Multimed Tools Appl
DOI 10.1007/s11042-013-1385-0

Multi-oriented scene text detection in video based on wavelet and angle projection boundary growing

Palaiahnakote Shivakumara · Anjan Dutta · Chew Lim Tan · Umapada Pal

Abstract In this paper, we address two complex issues: 1) Text frame classification and 2) Multi-oriented text detection in video text frame. We first divide a video frame into 16 blocks and propose a combination of wavelet and median-moments with k-means clustering at the block level to identify probable text blocks. For each probable text block, the method applies the same combination of feature with k-means clustering over a sliding window running through the blocks to identify potential text candidates. We introduce a new idea of symmetry on text candidates in each block based on the observation that pixel distribution in text exhibits a symmetric pattern. The method integrates all blocks containing text candidates in the frame and then all text candidates are mapped on to a Sobel edge map of the original frame to obtain text representatives. To tackle the multi-orientation problem, we present a new method called Angle Projection Boundary Growing (APBG) which is an iterative algorithm and works based on a nearest neighbor concept. APBG is then applied on the text representatives to fix the bounding box for multi-oriented text lines in the video frame. Directional information is used to eliminate false positives. Experimental results on a variety of datasets such as non-horizontal, horizontal, publicly available data (Hua's data) and ICDAR-03 competition data (camera images) show that the proposed method outperforms existing methods proposed for video and the state of the art methods for scene text as well.

Keywords Wavelet-median-moments · Text symmetry · Text frame classification · Text representatives · Angle projection boundary growing · Multi-oriented video text detection

P. Shivakumara (✉)
Multimedia Unit, Department of Computer Systems and Information Technology, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
e-mail: hudempsk@yahoo.com

P. Shivakumara
e-mail: hudempsk@gmail.com

A. Dutta · U. Pal
Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, India

U. Pal
e-mail: umapada@isical.ac.in

C. L. Tan
School of Computing, National University of Singapore, Singapore, Singapore
e-mail: tancl@comp.nus.edu.sg

Published online: 17 February 2013

1 Introduction

Advances in video technology together with social media such as YouTube have led to a tremendous growth in video databases. Therefore, there is great demand to develop an efficient and accurate method for indexing in the field of image processing, computer vision and multimedia. Text detection and recognition have been introduced to fill the semantic gap in indexing where necessary as it is discussed in [3, 8, 17, 29]. Sharma et al. [17] state that the presence of both graphics and scene text in individual video frames helps in retrieving semantic events. Video text recognition and temporal information can be used for tracking events [29]. Text on book images and video could be useful for information retrieval [8]. Special events can be retrieved with the help of text detection and tracking [3]. Furthermore, [5, 25, 26, 30] present examples of text detection and recognition for sport events and other applications such as driving assistance for visually impaired people. In summary, text detection and recognition help in improving performance of the information retrieval system.

There are two types of text in video, namely, caption and scene text. Caption text is manually edited text which includes subtitles and superimposed text in video while scene text is the text generally embedded in scene. Scene text includes text such as those on trucks and t-shirts, street names, building names, billboards etc. Since caption text is edited, it is easy to process and detect while scene text is unpredictable due to variable background, font types and sizes, orientation and perspective distortion that can cause disconnections, loss of information, impaired shapes, blurring in text images. Hence scene text detection is much more complex and challenging.

2 Related work

We first review methods for text detection in camera images of natural scenes and point out their inadequacies for video images of natural scenes. We then survey the methods generally used for video text detection to show their deficiency in dealing with natural scene text. This leads us to the research gap that we will address by means of our proposed method.

First on scene text detection in camera images, Epshtein et al. [4] proposed the use of a stroke width transform to detect scene text using Canny edge map. This method can be used for multi-oriented text detection since stroke width is invariant to rotation. However, extraction of stroke width requires prior knowledge of the shapes of the characters in the scene. Therefore, the stroke width concept may not work for video text detection due to frequent disconnections and impaired shapes in video. Minetto et al. [13] proposed a multi-resolution system for text detection in complex visual scene images. Pan et al. [15] designed a text region detector to estimate the text detection confidence and scale information using an image pyramid to obtain candidate text components. To filter out non-text components, conditional random field is used in the method. Yi and Tian [28] then explored the use of local gradient and color information to find character candidates and structure based grouping to achieve text detection. Neumann and Matas [14] introduced the idea of external regions (EF) to overcome blurring effects so as to find and group probable characters to lead to text detection. Yao et al. [27] in turn proposed the use of intrinsic characteristics of text to deal with multi-oriented text lines in natural scenes. This method considers many features based on connected component analysis and shapes of characters using several classifiers. Finally, Phan et al. [16] proposed the use of Gradient Vector Flow (GVF) information to detect scene text by exploiting character gap features and finding common symmetry pixels in both Sobel and Canny edge maps of the input image. Due to small fonts, low contrast and

complex background in video, the method sometimes fails to find the required common symmetry points. Overall, most of the scene text detection methods are classifier dependent and data dependent subject to the constraints of the database in use. While, these constraints may be true for camera based images, they are not necessarily applicable to video based images due to the unpredictable nature of scene text in video.

Next we survey the methods for video text detection. Generally, they can be classified into three categories, namely, connected component-based methods [7, 12], edge and gradient based methods [1, 2, 24] and texture-based methods [9, 10, 19]. Among the connected component-based methods, Jain and Yu's method [7] is good for big font and high contrast images while Mariano and Kasturi's work [12] requires uniform color for the text lines. Thus connected component based methods are generally good for caption text which usually exhibits uniform color and regular spacing. Among the edge and gradient based methods, Chen et al. [2] proposed a machine learning based approach using contrast independent features based on gradient to do localization and verification of video text. On the other hand, Wong and Chen [24] used the combination of edge and gradient features for efficient text detection with low false positives while Cai et al. [1] used edge and color features to detect low contrast text. These methods tend to be sensitive to constant thresholds required for classifying text and non-text pixels. The last category, i.e. texture-based methods aims to address the problem of complex background and false positives. Among these methods, Doermann and Kia [28] proposed the use of moments based features in the wavelet domain using a neural network classifier. Liu et al. [10] and Shivakumara et al. [19], on the other hand, used texture features without a classifier. Generally, texture-based methods use expensive texture features and they are sensitive to font types, font sizes and text like features in the background. They also tend to require more time to process due to large number of features.

All the video text detection methods surveyed above except for Doermann and Kia's method [9] assume that the frame in question for text detection already contains text. This means that these methods will not work properly if the input frame does not contain text at all. This calls for the need to preprocess the input video stream to classify text frames from non-frames before sending the text frames for text detection. Though Doermann and Kia's method [9] reported an experimental study on text frame classification, it achieves only 62 % precision as it is hard to differentiate text and non-text. Recently, we have proposed a mutual nearest neighbor based symmetry method [20] for text frame classification. There is however much room for improvement for text frame classification.

We also note from the above survey that most of the existing methods aim at horizontal text detection rather than multi-oriented text detection. We have found two methods, namely one by Crandell and Kasturi [3] and the other by Zhou et al. [32] that address multi-oriented text lines. However, these two methods are limited only to caption text appearing in a few orientations. We have partially addressed the issue of multi-oriented text lines in our recent works [18, 21, 22]. However, our work in [21] and [22] assumes text frame as the input, while the method in [18] requires classification of horizontal and non-horizontal text images before extracting text lines.

The above observations lead to the following research gap in two aspects which we would like to address in this paper: (1) Lack of an accurate text frame classification as a preprocessing step prior to text detection, and (2) Lack of a multi-oriented text detection capability that does not require the input to be a text frame.

Hence in this paper, we propose the use of wavelet-median-moments with k-means clustering method to identify text frame and to detect the text lines. The new method called angle projection boundary growing (APBG) is able to detect straight text lines at multiple orientations. Wavelet and moments combination has been used for horizontal text detection

by Doermann and Kia [9]. However, this method uses the mean to compute central moments and proposes a huge feature vector with expensive classifiers such as neural network for horizontal text detection. Instead, we propose the combination of wavelet and median-moment without a classifier and training. The main contributions of this work in contrast to [18, 21, 22] are as follows. i) Introducing median-moments with high frequency sub-bands of wavelet at block level to find probable text blocks for text frame classification. ii) The same features extracted in a different way to identify potential text pixel for text detection. iii) APBG for tackling the problem of multi-oriented text detection and iv) An attempt to increase the precision by introducing novel feature for false positive elimination on different dataset without classifier and training samples help.

3 Proposed methodology

As it is noted from [9, 21] that text detection methods erroneously produce many false positives when we present non-text frames as input. Thus, there is a need to select text frames accurately before text detection to minimize false detection and reduce computations. The proposed median-moments-wavelet features work based on the fact that text pixel have high contrast compared to its background information. Therefore, we expect these features to give high values for text and low values for non-text pixels in the frame. The proposed method is therefore divided into two stages. In the first stage, we propose text frame classification based on a symmetry property of the text candidates. The second stage describes multi-oriented text detection based on text representatives and angle projection boundary growing concept. The scope of this work is limited to straight line text detection at multiple orientations in one video frame. Based on our experience in video text detection, most of the text lines in a typical frame generally appear as straight lines in one or multiple orientations. Curved text line detection is thus outside the scope of this paper.

3.1 Text frame classification based on text candidates

Text frame classification method presented in [20] shows that the mutual nearest neighbor criteria is too strict measure for text frame classification. This results in a good non-text frame classification rate but a poor text frame classification rate. We will now propose a new symmetry measure to achieve a better classification rate for both text and non-text frames. The proposed method resizes the input frame to 256×256 regardless of its original size to facilitate implementation. Though this results in some distortions due to resizing, it will not affect the text location as shapes of objects in video frames are not important for text detection. Besides, it is possible to restore the original frame size after processing and re-compute the actual location of the text in the original frame. Wavelet (Haar) decomposition provides high contrast values such as diagonal, horizontal and vertical for text pixels in high frequency sub-bands HH, HL and LH, respectively as it is noted in [9]. Therefore, we take the average of the three high frequency sub-bands to assemble vital information of the text pixels to extract features. The proposed method divides the whole average frame into 16 blocks of size 64×64 without overlapping to study the local text information which is usually scattered over the entire frame as small clusters, instead of performing feature extraction on the whole frame in one operation. This division makes classification process faster. The reason behind the choice of 64×64 sized blocks is that the block should contain at least two characters for sufficient identity as a text block or not. Besides, our intention is to identify at least one candidate text block out of 16 blocks to classify a given frame as a text

frame but not to extract complete text information in the divided blocks. A sample text frame is shown in Fig. 1a and its 16 blocks are shown in Fig. 1b. We consider block 5 and 7 shown in Fig. 1b as a non-text and a text block, respectively for the purpose of illustration. After wavelet decomposition, for each block we compute median moments on it to increase the gap between text and non-text pixels. It is noted from [9] that those moments with respect to the mean for wavelet decomposition help in discriminating text and non-text pixels because wavelet decomposition provides successive approximations to the image by down sampling and has the ability to detect edges during the high pass filtering. The low pass filter creates successive approximation to the image while the detailed signal provides features rich representation of the textual content. In this work, we use the first level decomposition to obtain the three high frequency sub-bands. However, we propose moments with respect to the median rather than mean because median considers neither high nor low values while mean considers all values. Since low values do not contribute to text detection, median moments are more suitable.

For each block B shown in Fig. 1b, we compute level-one Haar wavelet transform to obtain four sub-bands, namely, HH, HL, LH and LL as defined in Eqs. 1–4, respectively and it determines average of sub-bands (HH, HL and LH) as defined in Eq. 5 and 6.

$$HH = \left\{ \downarrow_{2,2} \left(\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} * B \right) \right\} \otimes \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{1}$$

$$HL = \left\{ \downarrow_{2,2} \left(\begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} * B \right) \right\} \otimes \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2}$$

$$LH = \left\{ \downarrow_{2,2} \left(\begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} * B \right) \right\} \otimes \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{3}$$

$$LL = \left\{ \downarrow_{2,2} \left(\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} * B \right) \right\} \otimes \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{4}$$

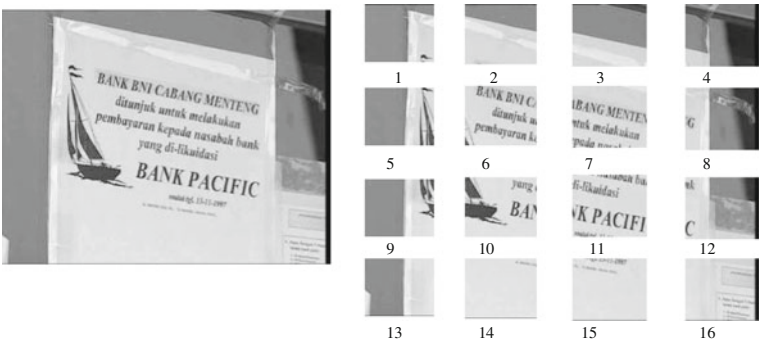


Fig. 1 (**a**) Input frame, (**b**) The 16 blocks of the input frame

Here, $\downarrow_{2,2}$ denotes downsampling by a factor of 2 in each dimension, * denotes the convolution, and $\otimes$ denotes the Kronecker product. Then, the average of the three high-frequency sub-bands is computed:

$$B' = \frac{HH + HL + LH}{3} \tag{5}$$

Such an image can be understood as the following convolution product:

$$B' = \frac{B - LL}{3} \tag{6}$$

Thus, B′ is obtained by removing a low-pass filtered component of B, which has larger responses around the edges.

For each average sub-band block (B′), we calculate the median (Me), 2nd order moments (μ_2) from median and 3rd order moments(μ_3) from median as follows:

$$Me(B') = \frac{SI\left(\frac{N^2-1}{2}\right) + SI\left(\frac{N^2+1}{2}\right)}{2} \tag{7}$$

$$\mu_2(B') = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(B'(i,j) - Me(B'))^2 \tag{8}$$

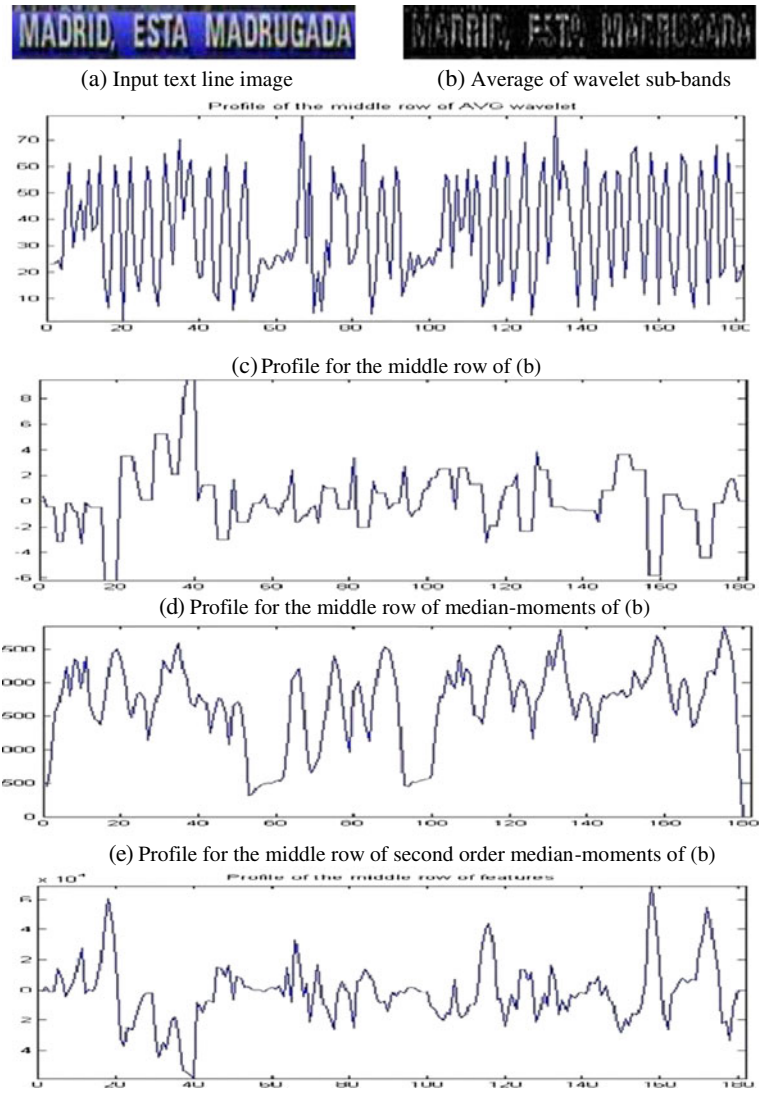
$$\mu_3(B') = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(B'(i,j) - Me(B'))^3 \tag{9}$$

Here, SI is the sorted list of pixel values of the 64×64 sized block, N=64.

The usefulness of the combination of wavelet and median-moments is shown in Fig. 2 where for the input image shown in (a), the average of wavelet sub-bands images shown in (b) sharpen edge pixels of the texts by suppressing non-text pixels. It is noticed from the profiles shown in Fig. 2c–f that the effect of the background reduces gradually from Fig. 2c–f because the high positive and negative peaks in the profiles are disappearing where non-text regions are present while for text regions, the high positive and negative peaks are retained. This shows that the combination of wavelet and median-moments are more effective than only wavelet sub-bands for text and non-text discrimination.

3.1.1 Probable text block classification

For each of the 16 blocks, we get the 3 feature values as mentioned in Eqs. 1–3 to construct a vector of dimension 3. Thus, for the entire frame, we get a feature matrix of dimension 16×3 from 16 such 3 dimensional vectors. Each element of the matrix is then normalized within [0, 1] by making the minimum and maximum values in the feature matrix as 0 and 1, respectively. We then apply k-means algorithm with k=2 to cluster the 16 given feature vectors into 2 classes: text and non-text. Cluster which gives high values is considered as a text cluster. This is valid because features representing text pixels in the cluster have higher values than features representing non-text pixels in the cluster. The output of k-means



(a) Input text line image

(b) Average of wavelet sub-bands

(c) Profile for the middle row of (b)

(d) Profile for the middle row of median-moments of (b)

(e) Profile for the middle row of second order median-moments of (b)

(f) Profile for the middle row of third order median-moments of (b)

Fig. 2 Profiles analysis for text and non-text regions using wavelet and median-moments

algorithm gives a set of text blocks and a set of non-text blocks. For the blocks in Fig. 1b the method classifies blocks 5, 6, 7, 11 and 12 as text blocks and the rest as non-text blocks. It is noted that non-text blocks 5 and 12 have been misclassified as text blocks and block 10 which is a text block has been misclassified as a non-text block. In this way the Probable Text Block Classification (PTC) gives at least one text block for every frame and hence each non-text frame contains at least one misclassified blocks. Despite its misclassification, it helps in reducing the computational burden of the next step (Candidate text block classification) to avoid processing all the 16 blocks unnecessarily to identify them as non-text blocks. For instance, in Fig. 1b, this step has reduced the 16 blocks to 5 probable text blocks for further feature extraction.

3.1.2 Candidate text block classification

At first, for each of the pixel in a block (B) selected by PTC, we consider a window W of size M×M (M=4) pixels over the average sub-band (B′). For each window, its median, 2nd order and 3rd order median moments are calculated according to Eqs. 7–9, respectively. We then normalize the features matrix of size 64×3 within [0, 1] by making the minimum and maximum feature values of the block as 0 and 1, respectively. The normalized feature matrix of the block is fed to k-means clustering with k=2 to classify text cluster as shown in Fig. 3a for blocks 5 and 7 of Fig. 1b. It is noticed from Fig. 3a that for block 5, some of the non-text pixels are classified as text pixels and for block 7, almost all text pixels are classified correctly and we call the classified text pixel as potential text pixels.

In the next step, we eliminate small blobs from the potential text pixels because we expect the cluster of pixels to be all from the text regions. We also eliminate straight components from the blocks as we assume that isolated straight components do not contribute to text region detection as shown in Fig. 3b which gives the text candidates for checking the symmetry property to identify the candidate text block.

We introduce the concept of Percentage of Pixel Based Symmetry Property (PPSP) for identifying candidate text block. This symmetry property is derived based on the observation that the pixel distribution of a text region over four quadrants [Q1-Q4] exhibits a pattern of symmetry. To check this property, first, the block with text candidates shown in Fig. 3b is divided into four quadrants with respect to x and y axes whose origin sits at the centroid of the image region as shown in Fig. 3c. The method computes the percentage of pixels in each of the four quadrants Q1-Q4 by counting the number of pixels in each quadrant which we call the four "quadrant values" in Table 1 for blocks 5 and 7. Note that if any one of the quadrants contains zero pixels then the method considers the block as a non-text block without testing the symmetry property. To measure this symmetry property, we introduce a Max-Min Nearest Neighbor (MMNN) concept. In each block, we group the four quadrant values into two clusters as follows:

$$C_1 = \left\{ Q_i | Q_i \leq \frac{\max Q_i + \min Q_i}{2} \right\} \tag{10}$$

$$C_2 = \left\{ Q_i | Q_i > \frac{\max Q_i + \min Q_i}{2} \right\} \tag{11}$$

Then, we classify the block B according to the sizes of the clusters:

$$B = \begin{cases} \text{text} & \text{if } |C_1| = |C_2| = 2 \\ \text{non} - \text{text} & \text{otherwise} \end{cases} \tag{12}$$

As per our definition to classify a frame as a text frame, if at least one block satisfies such a symmetry property then the entire frame is considered as a text frame. Otherwise it is a non-text frame.

(a) Block 5 Block 7 (b) Block 5 Block 7 (c) Block 5 Block 7

Fig. 3 (**a**) Potential text pixels, (**b**) Result of filtering (Text candidates) and (**c**) Quadrant formation of the blocks 5 and 7 of Fig. 1b

Table 1 Percentage of pixels computed from each quadrant and the clusters in each of the quadrants of blocks 5 and 7 of Fig. 1b

	Quadrant values				Clusters	
Block	Q_1	Q_2	Q_3	Q_4	C_1	C_2
5	20.53	21.67	22.43	35.37	[Q_1, Q_2, Q_3]	[Q_4]
7	29.31	21.65	27.15	21.89	[Q_1, Q_3]	[Q_2, Q_4]

For the considered frame shown in Fig. 1a, blocks 7 and 11 shown in Fig. 1b satisfy the symmetry property by the PPSP algorithm and hence the entire frame is classified as a text frame. In summary, for the frame shown in Fig. 1a, the classifications by PTC and CTC are shown in Table 2.

This shows that CTC is good since it extracts the symmetry property of text pattern while PTC does not extract any such symmetry property but it serves to first identify probable text blocks. Hence, these two complement each other to achieve a better accuracy.

3.2 Multi-oriented text detection

We now integrate the text candidates of each block obtained from the method described in Section 3.1.2. For example, the results shown in Fig. 3b of block 7 are considered as text candidates. These text candidates represent text in the video frame but they may not be the complete information of text in the video frame as shown in Fig. 4a and b. Therefore, we introduce a new method based on text representatives to restore the lost information with the help of Sobel edge map (Fig. 4c) of the input frame shown in Fig. 1a. The Sobel edge map is obtained by performing Sobel edge operation on the input frame. This step will be discussed below.

3.2.1 Text representatives

For each of the text candidates in Fig. 4b, we get the corresponding edge components in the Sobel edge map (Fig. 4c) for which the bounding boxes are drawn and are called text representatives. We eliminate false representatives if the centroid of the representative does not fall on its major axis. This results in candidate text representatives as shown in Fig. 4d where it is noticed that most of the false representatives are removed. Then these candidate text representatives are used for fixing bounding boxes for multi-oriented text lines using angle projection boundary growing method given below.

3.2.2 Angle projection boundary growing (APBG)

We propose APBG based on the nearest neighbor concept to determine the angle of orientation of a text line from the candidate text representatives shown in Fig. 4d. We

Table 2 Summary of classification of PTC and CTC for the blocks shown in Fig. 1b

Type	PTC		CTC	
	Text	Non-text	Text	Non-text
True text	6, 7, 11	10	7, 11	6
True non-text	5, 12	1–4, 8, 9, 13–16		5, 12

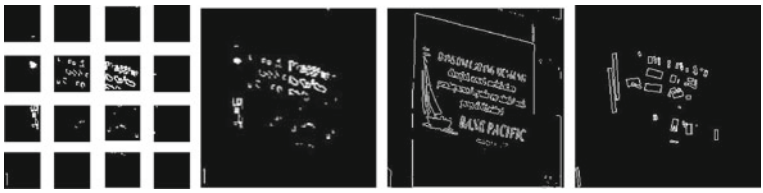


Fig. 4 (**a**) Text candidates in the blocks, (**b**) Integrating blocks containing text candidates, (**c**) Sobel edge map and (**d**) Bounding boxes for candidate text representatives after eliminating some false representatives

use a candidate text representative belonging to that line as the seed to grow the boundary. Here, the algorithm considers every candidate text representatives in the integrated image as a seed point for growing. For instance, the candidate text representative in Fig. 4d corresponding to the fifth text line shown in Fig. 4c is used as a seed point to begin the boundary growing process. The method starts growing boundary of the seed point by expanding the boundary, pixel by pixel. For each expanded boundary, the method checks whether the expanded boundary contains a white pixel or not. If the expanded boundary does not contain any white pixel then growing continues until it gets a white pixel of the nearest neighbor component in the text line. This process continues along the text edge information in the Sobel edge map from the seed representative till it satisfies some convergent criteria. Unlike the methods in [18, 22] which use some threshold to stop the boundary growing process, the proposed method uses the converging criteria based on angle information, which allows the proposed method to grow along the direction of the text line without covering the background information and touching adjacent text lines. Therefore, the proposed APBG is more effective than the boundary growing in [18, 22] to fix the bounding box for multi-oriented text lines in video frame. Besides, it restores the missing text information which does not have candidate text representative while growing in Sobel edge map. First APBG fixes the boundary for the edge component corresponding to the candidate text representative in the Sobel edge map and then we compute the angle for that component using PCA as shown in Fig. 5a. Here, we prefer PCA to determine the angle for the text component because we believe PCA is

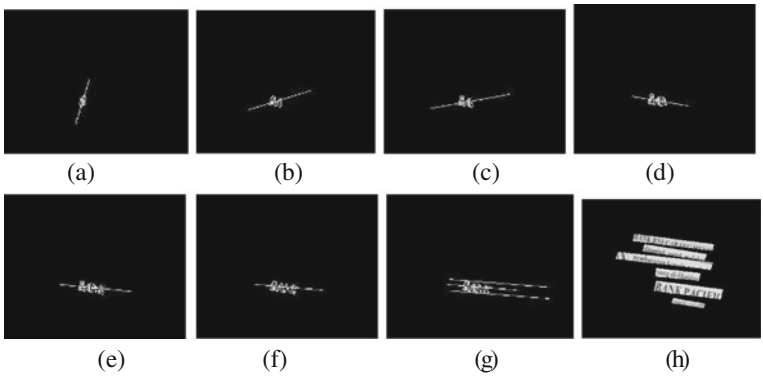


Fig. 5 Angle projection boundary for text lines. (**a**) angle 86.32, (**b**) angle 36.86, (**c**) angle 25.98, (**d**) angle −10.25, (**e**) angle −6.67, (**f**) angle −6.54, (**g**) Projected angle and (**h**) Text lines extraction results

better than other methods in distinguishing from objects that appear as disconnected text components. The computed angle for the first edge component is considered as the first iteration angle. The APBG allows the boundary of the first edge component to grow incrementally pixel by pixel. Then iteratively it expands the boundary for identifying neighboring text pixels until it reaches the nearest edge component and then it merges the boundary of the present component with the previous component to make one boundary. We compute the angle for the merged edge components using PCA. This step is considered as the second iteration angle as shown in Fig. 5b. If the absolute difference between two successive iterations is less than 1° (refer to angles in Fig. 5e and f), the angle calculation stops and the last iteration angle is used as the projection angle to draw the top and bottom boundaries for the text line as shown in Fig. 5g. If not, then the iteration continues. This angle computation is valid because we assume multi-oriented text lines to be straight as it is stated in the proposed methodology section. Though the angle computation stops upon convergence, the boundary growing process continues till the end of the text line within the top and bottom boundaries with no regards to any noise in between. This is the advantage of the proposed method and it fixes exact bounding box even if any noisy edges are present in the space between the two boundaries due to the high contrast edges in the background. This is the novelty of the work which differs from the existing methods [18, 22]. The end of the text line is determined based on an experimental study on the space between characters, words and text lines. However, we ignore the initial three iterated angles (Fig. 5a–c) before checking the convergence criteria because the computed angles for initial iterations may not be the actual text direction due to insufficient edge information. This is illustrated in Fig. 5 where we check the convergent criteria for iterated angles from (d) to (f). It is noticed from Fig. 5e–f that the absolute difference between the iteration angles (e) and (f) is 0.11 which is less than 1°. Therefore −6.54° angle is considered as the angle projection boundary as shown in Fig. 5g. The APBG repeats on other text lines, giving rise to the bounding boxes for all the text lines in the frame as shown in Fig. 5h where the results are shown after false positive elimination by the steps to be explained in Section 3.2.3.

3.2.3 False positive elimination

The preceding APBG process essentially detects all the multi-oriented text lines with their respective bounding boxes and orientation angles. However, some of these detected text lines could well be false positives because as it is reported in [23, 31] that in many cases, some text like background blocks may be marked as text blocks incorrectly, due to their high contrast, sharp edges and corresponding high text energies. Therefore, it is critical to eliminate false positives to improve the accuracy of the method. To remove such false positives and based on the assumption that the multi-oriented text lines are straight, two objective heuristics are proposed in this work.

The first heuristic is based on the distances between centroids of edge components in a text line. If the standard deviation of these distance values is close to zero, then the text line is considered truly a text line. Otherwise, it is a false positive and is eliminated. The second heuristic is based on the angle of the text line. The whole text line is divided into two equal parts. The two divided parts should give the same angle if it is a text line. Otherwise, it is eliminated as a false positive. These heuristics are said to be objective heuristics. They are new and they can work for different images unlike the existing methods [1, 10, 18–22, 24, 32] which use ad hoc heuristics based on density of Sobel edges and shape of the

components with constant thresholds. Hence, the proposed method gives low false positive rate.

4 Experimental results

We evaluate the proposed method on our own dataset as well as some standard datasets such as 45 video images of Hua's data [6] and 251 cameras images of ICDAR 2003 data [11] to study performance of the method. Our own dataset includes a variety of video frames such as, frames from movies, news clips containing scene texts, sports and web video. Though there is work on performance evaluation in [31], the problem lies in the non-availability of bench mark data and the fact that existing evaluation methods are based on the assumption of graphics text and well structured text in video frames. Nevertheless, we still use a small dataset of 45 video images of Hua's data used in [6] as benchmark dataset to evaluate the performance of the method. We present experimental results on video frames that are not necessarily constrained by these assumptions.

We present our results in two parts, namely, the first part for text frame classification and the second part deals with text detection from frame. We conducted all the experiments on a PC with P4 2.6 GHz processor with 1 GB RAM running Windows XP operating system and we use C++ to implement the algorithms.

4.1 Experimental results on text frame classification

We consider 1220 text frames and 800 non-text frames to create a general dataset which includes a variety of frames from different sources. We evaluate the text frame classification method at the block level in terms of recall and precision and the frame level in terms of classification rate. The evaluation includes experiments based on PTC alone, CTC alone, and combination of PTC and CTC. They are presented in the subsequent sections. We noted that Average Processing Time (APT) required for text frame classification is about 1.97 s.

4.1.1 Experiments on text frame classification using PTC

For illustrating the performance of the proposed text frame classification method using PTC, we test a text frame (Fig. 6a) and a non-text frame (Fig. 7a). For the text frame shown in Fig. 6a the PTC classifies blocks 6, 7, 10 and 11 as text blocks and the rest as non-text blocks. For the non-text frame shown in Fig. 7a, PTC classifies blocks 9–11 and 14–16 as text blocks though they are non-text blocks. This is mainly because of the unsupervised k-means clustering algorithm used for text blocks classification. Therefore, PTC alone may not be sufficient to classify text blocks, accurately. The experimental result of PTC is reported in the second row of Table 3. (In Table 3, R denotes Recall, P denotes Precision). From Table 3 it can be seen that for text blocks recall is quite high but precision is low since PTC classifies non-text blocks as text blocks. On the other hand recall is low but precision is high for non-text blocks classification. This shows that PTC definitely helps in classification of text blocks. It is also observed that the classification rate (CR) for text frame is 100 % and 0 % for non-text frames because PTC identifies at least one candidate text block for both text frame and for non-text frames.

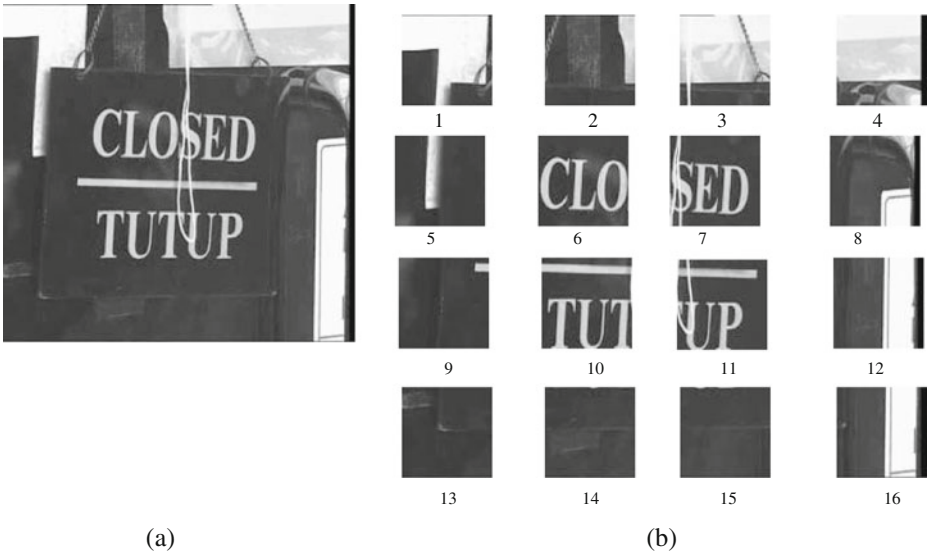


Fig. 6 (**a**) Example of an input text frame (**b**) 16 blocks of the input frame

4.1.2 Experiments on text frame classification using CTC

To study the effectiveness of CTC without PTC, we conduct experiments on the same dataset (1220 text and 800 non-text frames), separately. For the frame shown in Fig. 6a, CTC classifies its blocks 6, 7 and 9 as text and the other blocks as non-text. Blocks 10 and 11 are misclassified as non-text blocks by CTC. Similarly, block 5 and block 9 look similar as shown in Fig. 6b. CTC classifies block 5 as non-text correctly but it classifies block 9 as text

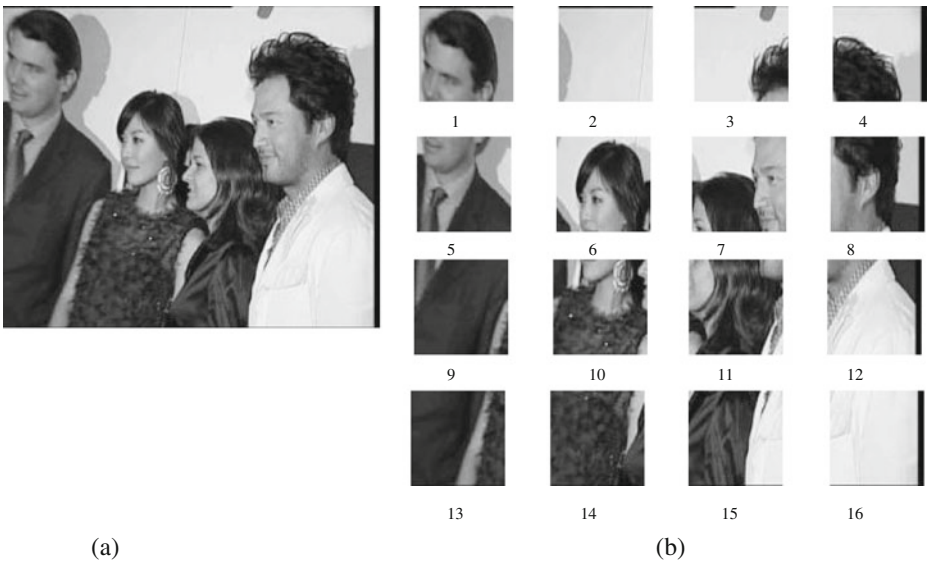


Fig. 7 (**a**) Input non-text frame (**b**) 16 blocks of the input frame

Table 3 Performance (in %) of Probable text block classification method (PTC), Candidate text block classification method (CTC) and Combined (PTC+CTC)

Methods	Block Level				Frame level	
	Text		Non-Text		Text	Non-text
	R	P	R	P	CR	CR
PTC	85.0	18.2	73.4	98.6	100	0
CTC	63.8	41.5	93.7	97.3	92.5	93.5
PTC+CTC	58.4	88.0	99.4	97.1	94.0	98.8

wrongly. However, the same block 9 was classified as non-text block by PTC, correctly. This is due to presence of text like objects in block 9 which satisfy the defined symmetry property. This also shows that CTC alone is not sufficient to identify the real candidate text block. For the non-text frame shown in Fig. 7a, none of the blocks are classified as text by CTC. This shows that CTC is good for classification of both text blocks and frames compared to PTC. The experimental results of CTC are reported in the third row of Table 3 where precision is low and recall is high for text block classification while for non-text block classification, precision is high and recall is low compared to PTC. Besides, classification rate for non-text frames is better than PTC as CTC works based on the symmetry concept to identify text blocks. Hence, both PTC and CTC are necessary to classify text frames, accurately.

4.1.3 Experiments on text frame classification by combining PTC and CTC

Based on the above experiments, this experiment combines both PTC and CTC to get good result for classification of text frames. For the text frame shown in Fig. 6a, the combined method classifies blocks 6 and 7 as text blocks and the others as non-text blocks. We take a closer look at these blocks 6 and 7 in Fig. 8a and b. Here, we get two distinct pairs of clusters namely, C1={Q1, Q2}, C2={Q3, Q4} for block 6 and C1={Q1, Q3}, C2={Q2, Q4} for block 7. Hence, blocks 6 and 7 are classified as candidate text blocks. We observe another two blocks 10 and 11 shown in Fig. 8c and d which appear somewhat similar to blocks 6 and 7. However, we see that there is no need to test symmetry property since the second quadrant of each of the two blocks has zero text candidates which render them as non-text according to our symmetry property test. Thus blocks 10 and 11 are classified as non-text blocks though they are earlier classified as text blocks by PTC. Since at least one candidate text block exists for the frame, this frame is considered as a text frame. Similarly for the non-text frame shown in Fig. 7a, none of the blocks are classified as text blocks by the combined

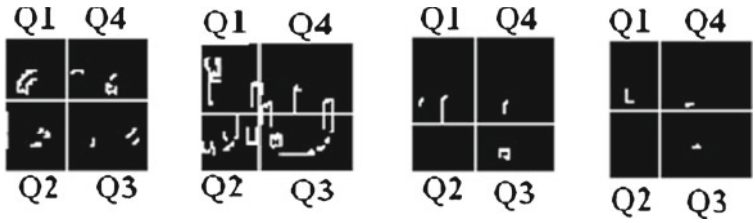


Fig. 8 Quadrant information for the blocks 6, 7, and 10, 11 shown in Fig. 6b. Here Q1-Q4 are the four quadrants

method. Thus it is considered as non-text frame correctly. The experimental results of the combined PTC and CTC method are reported in the fourth row of Table 3 where precision for text blocks improves compared to the results of PTC and CTC alone. In addition, the classification rate for text and non-text frame is better than CTC. Hence the combined method is useful for classification of text frames from the large dataset.

To analyze the symmetry property further, we conduct experiments on some objects which appear to satisfy the symmetry property as shown in Fig. 9 where (a) and (c) are two non-text objects in the video frame. Since the proposed wavelet-moments feature and k-means clustering presented in Section 3.1.2 filter out pixels belonging to non-text objects before getting text candidates, the objects that look like text get eliminated before testing the symmetry property as shown in Fig. 9b and d. This is because the contrast of non-text pixels is lower than that of text pixels. Therefore, the method eliminates those pixels before testing the symmetry property. Fig. 9b and d show text candidates for the blocks shown in Fig. 9a and c, respectively. For Fig. 9b, the symmetry test gives quadrant values {Q1=46.15, Q2=17.95, Q3=17.95, Q4=17.95}, and clusters C1={Q1} and C2={Q2, Q3, Q4}. Since we do not find two distinct pairs of clusters, the block is considered as a non-text block. For Fig. 9d, the symmetry test gives nothing since there are no text candidates in the image. Hence the block is classified as a non-text block.

4.2 Experimental results on text detection

We used 220 frames (which include 176 scene text frames and 44 graphics text frames) that contain non-horizontal text lines, 800 frames (which include 160 Chinese text, 155 scene text and 485 English graphics text frames) that contain horizontal text, and publicly available Hua's data of 45 frames (which include 12 scene text and 33 graphics text frames) for our text detection algorithm. We also experiment on the publicly available ICDAR-03 competition dataset of 251 images (all images are scene text) to check the effectiveness of our method on camera based images. In summary, 1065 (220+800+45) video frames and 251 camera images are used for this experimentation.

We consider six existing methods for comparison of results to show the effectiveness of our method. Out of these, "Gradient" [24], "Edge-Color" [1] and "Edge-Texture" [10] are considered for comparing results of horizontal text detection because these methods are designed to handle horizontal video text detection only and the fourth method "Edge-Caption"[32], which detects both horizontal and vertical text lines. In addition, we also include our recently published method "Laplacian" [21] and "Bayesian" [22] which detects multi-oriented text in video frame. Since the methods [21, 22, 32] work for multi-oriented text, we use these three methods for comparative study on all datasets with the proposed method. The main reason to consider these existing methods is that these methods work with fewer constraints for complex background without a classifier and training as in our

Fig. 9 Testing the combined method on non-text objects. (**a**) and (**c**) show non-text

proposed method. Here we evaluate the performance at the text line level, which is a common procedure in the literature [1, 2, 9, 10, 18–22, 24, 32], rather than the word or character level because we have not considered any recognition in this work. The following categories are defined for each detected block by a text detection method.

Truly Detected Block (*TDB*): A detected block that contains at least one true character. Thus, a TDB may or may not fully enclose a text line. *Falsely Detected Block* (*FDB*): A block which is erroneously detected as a text block that does not contain text. *Text Block with Missing Data* (*MDB*): A detected block that misses more than 20 % of the characters of a text line (MDB is a subset of TDB). The percentage is chosen according to [2], in which a text block is considered correctly detected if it overlaps at least 80 % of the ground-truth block. We count manually Actual Number of Text Blocks (ATB) in the images and it is considered as ground truth for evaluation. The performance measures are defined as follows. *Recall (R)*=TDB / ATB, *Precision (P)*=TDB (TDB+FDB), *F-measure (F)*=(2 × P × R) / (P+R), *Misdetection Rate* (*MDR*)=MDB / TDB.

4.2.1 Experiments on non-horizontal dataset

To get the idea of qualitative results, sample input frames for non-horizontal text are shown in the first row of Fig. 10 and we have shown results of different methods (Proposed, Laplacian, Bayesian and Edge-Caption) in the other rows of the corresponding input frames. It is observed from the results that for varieties of text (Scene text with perspective deformation, scene text with small font, different orientations and multi-oriented text lines as shown in the first row in Fig. 10), the proposed method detects almost all text lines correctly. On the other hand, the Bayesian [22] and the Laplacian methods [21] produce false positives for the input images (Fig. 10a and b) as shown in the third and the fourth row of Fig. 10. For the images shown in Fig. 10c–d, the Bayesian and the Laplacian methods fail to detect text line properly. In case of the Bayesian method, the boundary growing causes a problem while in case of the Laplacian classification of text and non-text components causes a problem and generates poor results due to low contrast in the images. For the images shown in Fig. 10a and c, the Edge-Caption method [32] detects text with improper bounding boxes and for Fig. 10b and d, the method fails to detect text as shown in fifth row in Fig. 10. The main reasons for this are heuristics and different limitations such as the method detects only caption text, horizontal and vertical texts, etc.

The detailed quantitative experimental results for the proposed, the Bayesian, Laplacian Edge-Caption methods are reported in Table 4 for non-horizontal data. Table 4 shows that F-measure is higher with lower misdetection rate and computational time for the proposed method than the Bayesian, the Laplacian and the Edge-Caption methods. It is also noted from Table 4 that the Bayesian method and the Laplacian method requires about 9 times more computational time for text detection compared to the proposed method as reported in Average Processing Time (APT) (See the last column of Table 4). Therefore, it can be concluded that the proposed method is good for multi-oriented text detection.

4.2.2 Experimental results on horizontal dataset

To have the idea of qualitative results, here Fig. 11a shows the original frame and (b) shows the text extraction results of the proposed method, when horizontal data are considered. Figure 11c–h show the results of the existing methods mentioned above. It is observed from Fig. 11 that the proposed, the Bayesian, the Laplacian and the Edge-Texture methods detect text correctly while the other methods fail to detect text properly and draw improper

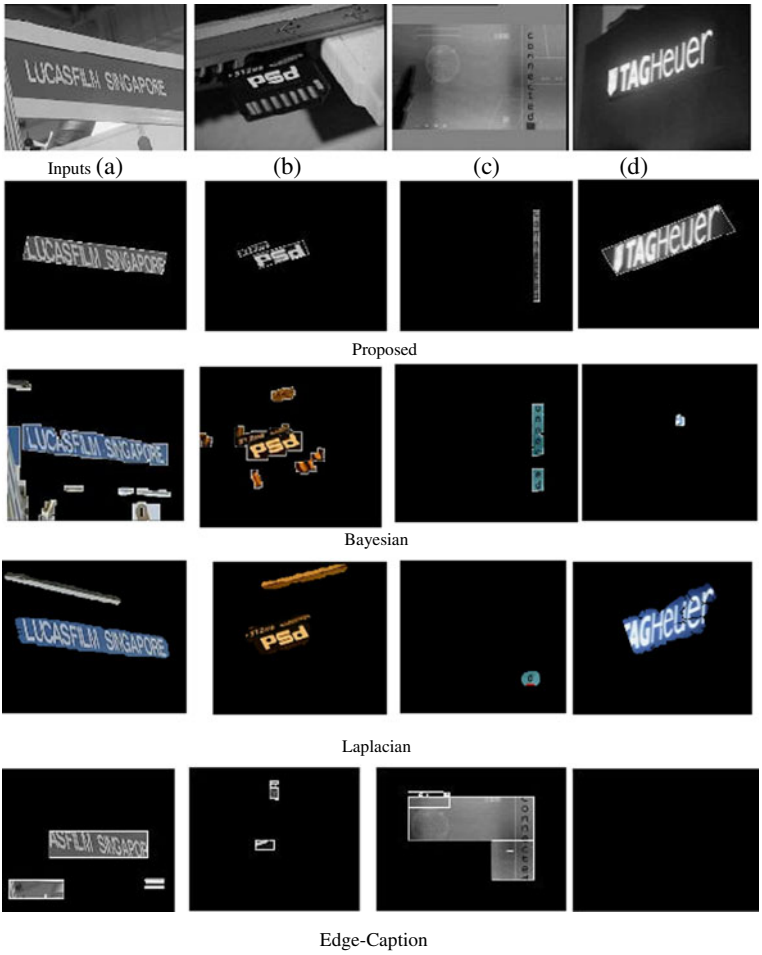
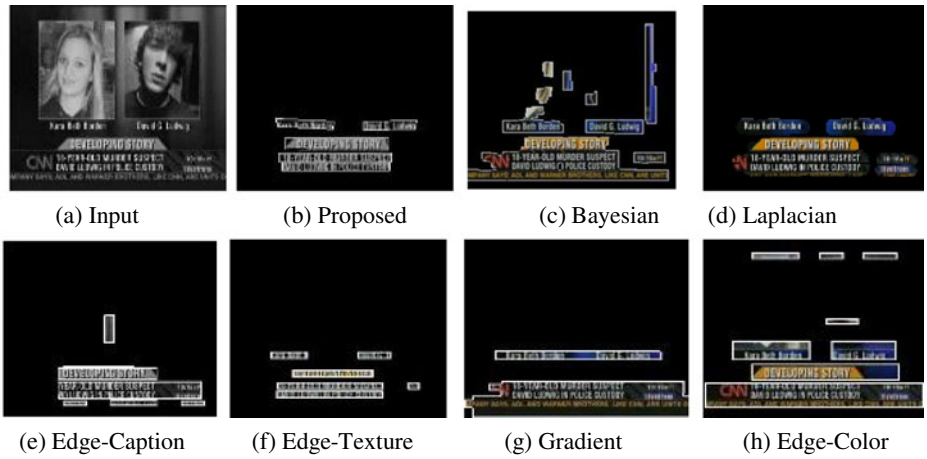


Fig. 10 Experiment on different non-horizontal text. Here first row shows four input images and the other rows show results on different methods

bounding boxes for the text lines. Quantitative results on horizontal data are shown in Table 5. Table 5 shows that the proposed method outperforms the existing methods in terms of Precision, F-measure and APT. However, the Gradient method gives low misdetection rate compared to all the methods. On the other hand, our proposed method is the second best in MDR with a high F-measure and is the best in computational time. The main reason for

Table 4 Performance on non-horizontal dataset

Methods	R	P	F	MDR	APT (s)
Proposed Method	0.85	**0.85**	**0.85**	**0.12**	**1.12**
Bayesian [22]	**0.86**	0.76	0.80	0.14	9.5
Laplacian [21]	0.83	0.75	0.79	0.24	10.3
Edge-Caption [32]	0.39	0.67	0.49	0.40	1.34



(a) Input (b) Proposed (c) Bayesian (d) Laplacian

(e) Edge-Caption (f) Edge-Texture (g) Gradient (h) Edge-Color

Fig. 11 Experimental results of different methods on horizontal text

poor performance of the existing methods is that the methods use several constant thresholds and make assumptions as stated in the previous sections for text detection.

4.2.3 Experiments on Hua's dataset

We will now test the proposed method on an independent publicly available (http://www.cs.cityu.edu.hk/~liuwy/PE_VTDetect/) Hua's dataset comprising of 45 different frames obtained from [6]. While the dataset is small, it provides an objective test of the proposed method in comparison with the six existing methods. Experimental results are shown in Fig. 12 where (a) is the input frame, (b) is text extraction by the proposed method and (c)-(h) are the results of the existing methods. Figure 12 show that the proposed method detects all text lines in the frame while the other methods fail to detect text and fix bounding boxes wrongly. Quantitative results are shown in Table 6. It is seen from the results reported in Table 6 that the proposed method outperforms the existing methods in terms of precision, MDR and APT. Although the Bayesian and the Laplacian method give slightly better recall and F-measure, they require more computational time than our method. The boundary growing methods used in Bayesian method requires more time because of Bayesian classifier and the boundary growing process that covers background while the Laplacian method involves connected component labeling to classify simple and complex components. On the

Table 5 Performance on horizontal dataset

Methods	R	P	F	MDR	APT (s)
Proposed Method	0.86	**0.88**	**0.86**	**0.11**	**0.87**
Bayesian [22]	**0.88**	0.73	0.79	0.12	8.9
Laplacian [21]	0.82	0.78	0.80	0.18	7.8
Edge-Caption [32]	0.61	0.85	0.71	0.25	1.19
Edge-Texture [10]	0.68	0.65	0.66	0.32	22.1
Gradient [24]	0.67	0.79	0.73	0.06	1.10
Edge-Color [1]	0.59	0.37	0.46	0.15	6.1

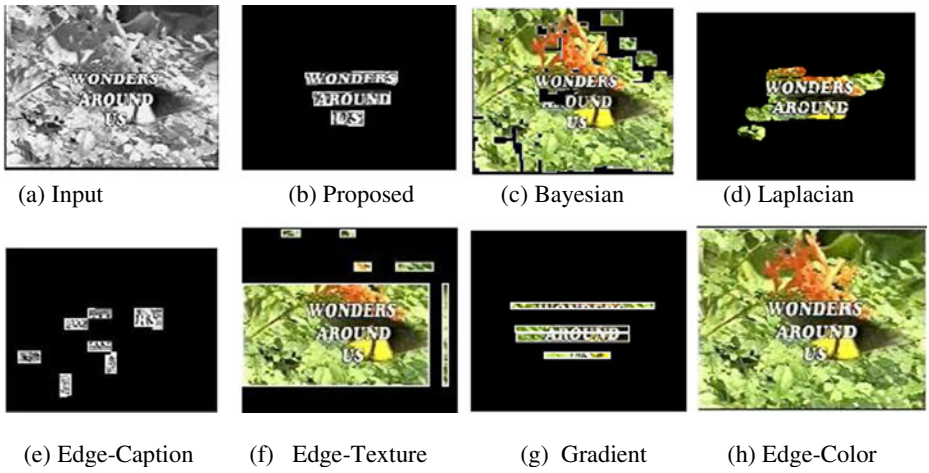


Fig. 12 Experimental results of different methods on Hua's Data

other hand, the proposed method does not require much time because APBG terminates quickly with the convergent criteria.

4.2.4 Experiments on ICDAR-2003 dataset

Earlier experiments were done on video based images. Here the present experiment is done to show that the capability of the proposed method for camera based images. To verify its capability, we conduct two experiments on the benchmark database of ICDAR-03 competition data [11].

The first experiment is based on the same evaluation metrics that we use in the preceding experiments on video images for consistent comparison. A sample result on ICDAR-03 data is shown in Fig. 13 where (a) shows the input image, (b) shows the results of the proposed method and (c)-(h) show the results of the existing methods. The Bayesian, the Laplacian and the other existing methods fail to detect text line properly for the image shown in Fig. 13a while the proposed method detects text line correctly. The quantitative line-level text detection results are reported in Table 7 where it is observed that the proposed method outperforms the existing methods in terms of precision and F-measure.

It may be noted that our evaluation metrics are defined to cater to low quality video text. The evaluation thus allows lesser defined bounding boxes and hence more tolerance to

Table 6 Performance on Hua's dataset

Methods	R	P	F	MDR	APT(s)
Proposed Method	0.81	**0.86**	0.83	**0.02**	**0.67**
Bayesian [22]	0.87	0.85	0.85	0.18	5.6
Laplacian [21]	**0.93**	0.81	**0.87**	0.07	11.7
Edge-Caption [32]	0.72	0.82	0.77	0.44	1.13
Edge-Texture [10]	0.75	0.54	0.63	0.16	24.9
Gradient [24]	0.51	0.75	0.61	0.13	1.6
Edge-Color [1]	0.69	0.43	0.53	0.13	9.2

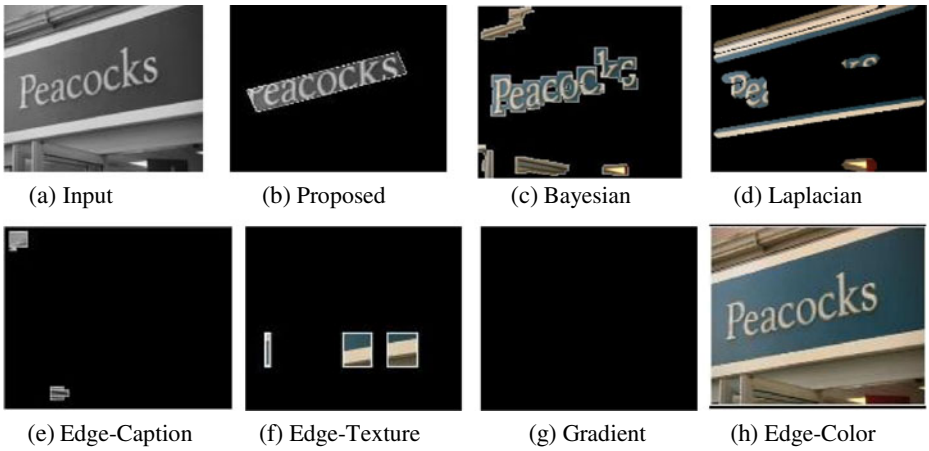


Fig. 13 Experimental results of different methods on an image of ICDAR-03 competition dataset

missing characters in text detection. This is the same practice in other video text evaluation [1, 9, 10, 18–22, 24, 32]. On the other hand, we note that the ICDAR-03 competition actually has its own set of evaluation measures [11]. Thus, to have the comparative idea of same evaluation measure, our second experiment follows the ICDAR-03 competition measures to evaluate the proposed method on the ICDAR-03 dataset. The ICDAR-03 measures are computed at the word level (fixing bounding box for the words using space information between the components and nearest neighbor concept) by matching the detected word boundary with the ground truth. The performance evaluation using ICDAR-03 measures can be found in the following link http://www.comp.nus.edu.sg/~tancl/Shiva/. In addition to the proposed method, we also test the ICDAR-03 data on some recent methods which also follow the ICDAR-03 competition measures. The seven methods compared here are the method proposed by Phan et al. [16], the best method in terms of recall, the method by Yao et al. [27], the best method in terms of F-measure, the method by Neumann and Matas [14], the second best method in terms of precision, the method by Epshtein et al. [4], the best method in terms of precision, the method by Yi et al. [28], the method by Minetto et al. [13] and the method which obtain the best performance of ICDAR-05 competition [11]. The quantitative results on word-level are reported in Table 8 which shows that the proposed method achieves the best F-measure and precision compared to these state of the art methods. Thus, we can infer that the proposed method is good both for text detection from natural scene images and video frames.

Table 7 Line-level Performance on ICDAR-03 dataset

Methods	R	P	F	MDR
Proposed Method	0.85	**0.87**	**0.85**	0.14
Bayesian [22]	**0.87**	0.72	0.78	0.14
Laplacian [21]	0.86	0.76	0.81	0.13
Edge-Caption [32]	0.66	0.83	0.73	0.26
Edge-Texture [10]	0.53	0.61	0.57	0.24
Gradient [24]	0.52	0.83	0.64	**0.08**
Edge-Color [1]	0.67	0.33	0.44	0.43

Table 8 Word-level Performance on ICDAR-03 dataset

Methods	R	P	F
Proposed Method	0.62	**0.74**	**0.68**
Phan et al. [16]	**0.69**	0.63	0.66
Yao et al. [27]	0.66	0.69	0.67
Neumann and Matas [14]	0.62	0.72	0.67
Yi et al. [28]	0.62	0.71	0.62
Epshtein et al. [4]	0.60	0.73	0.66
Minetto et al. [13]	0.61	0.63	0.61
ICDAR-05 [23]	0.62	0.67	0.62

It is also observed that the results reported in Tables 4 to 7 of the proposed method are higher than the results reported in Table 8. This is because the measures used in Tables 4 to 7 are in the line of text detection and they allow tolerance as in [1, 9, 10, 18–22, 24, 32] for computing measures while the measures used in Table 8 are computed based on the measures suggested in the ICDAR-03 competition using the available ground truth. Another reason for getting higher results in Tables 4 to 7 is because of line level text detection but Table 8 considers word level text detection. The conclusion drawn from the above experiments is that in both evaluation measures, the proposed method is found to be superior to existing methods for both video and camera based images.

The main cause of poor accuracy in text detection in the existing methods is due to their lack of good features to eliminate false positives. This is because separation of text and non-text is a difficult task which leads to false positives. As such, in this work, we have proposed a method that can give good precision by eliminating false positives correctly. This is accomplished by the use of Angle Projection Boundary Growing (APBG) and objective heuristics based on the direction of text lines to eliminate false positives. The advantage of APBG is that once it converges, it just grows within the top and bottom boundaries of the text line regardless of noise and character touching between the lines. As a result, it does not include background information in the bounding boxes of text lines thereby preserving the text line properties.

4.2.5 Discussion on text detection experiments

The above experiments show that the proposed method generally excels the other methods in terms of precision and F-measure, though there are a couple of instances where it suffers somewhat in recall. However, we first note that the experiments shown in Tables 5 to 8 have not tested the multi-oriented text detection capability of the proposed method in full as we allowed comparison with baselines that are unable to handle multi-oriented text lines. These experiments serve to demonstrate the overall better performance than the other existing methods even for ICDAR'03 dataset which is not really video dataset. The real test of the full capability of multi-oriented text is in Table 4 which is a main concern of the paper. Here it can be seen that the proposed method is a strong contender over others with a much higher F-measure. It only marginally loses to the Bayesian method in terms of recall, but the Bayesian method is notably worse in precision and F-measure, not to mention its much higher computational cost.

5 Conclusion and future work

In this paper, we have proposed wavelet-moments based method for text detection from video frames. Firstly, the method introduces a new concept called symmetry that is computed based on wavelet and median moments for text frame classification. Secondly, the method presents a new idea of text representatives from text candidates identified that uses angle projection boundary growing for multi-oriented text line detection from video frames. Experimental results on text detection in video and camera images show that the proposed method gives better results compared to the results of existing methods. The proposed method achieves the best accuracy for ICDAR-03 data according to ICDAR-03 competition measures compared to state of the art methods. However, the proposed method fails to detect complete text when the text is distorted due to perspective distortion. In future, in addition to addressing the perspective distortion problem by exploring temporal information, we plan to deal with curved text lines in video frames using background and foreground information.

Acknowledgments This work is done jointly by National University of Singapore and Indian Statistical Institute, Kolkata, India. This research is supported in part by the A*STAR grant 092 101 0051 (WBS no. R252-000-402-305). We thank the anonymous reviewers for their valuable comments and suggestions that improve the quality of the work. Our special thanks to Prof. Andy Ming-Ham Yip, Department of Mathematics, National University of Singapore for his helpful discussion and comments on wavelet operations and other mathematical details.

References

1. Cai M, Song J and Lyu MR (2002) “A new approach for video text detection”. In Proc ICIP 117–120
2. Chen D, Odobez JM and Thiran JP (2004) “A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning”. Signal Process Image Commun 205–217
3. Crandall D and Kasturi R (2001) “Robust detection of stylized text events in digital video”. In Proc ICDAR 865–869
4. Epshtein B, Ofek E and Wexler Y (2010) “Detecting text in natural scenes with stroke width transform”. CVPR 2963–2970
5. Guo J, Gurrin C, Lao S, Foley C and Smeaton AF (2011) “Localization and recognition of the scoreboard in sports video on sift point matching”. In Proc MMM 337–347
6. Hua XS, Wenyin L and Zhang HJ (2004) “An automatic performance evaluation protocol for video text detection algorithms”. IEEE Trans CSVT 498–507
7. Jain AK and Yu B (1998) “Automatic text location in images and video frames”. Pattern Recogn 2055–2076
8. Jung K, Kim KI and Jain AK (2004) “Text information extraction in images and video: a survey”. Pattern Recogn 977–997
9. Li H, Doermann D and Kia O (2000) “Automatic text detection and tracking in digital video”. IEEE Trans IP 147–156
10. Liu C, Wang C and Dai R (2005) “Text detection in images based on unsupervised classification of edge-based features”. In Proc ICDAR 610–614
11. Lucas SM (2005) “ICDAR 2005 text locating competition results”. In Proc ICDAR 80–84
12. Mariano VY and Kasturi R (2000) “Locating uniform-colored text in video frames”. In Proc ICPR 539–542
13. Minetto R, Thome N, Cord M, Fabrizio J and Marcotegui B (2010) “SNOOPERTEXT: a multiresolution system for text detection in complex visual scenes”. In Proc ICIP 3861–3864
14. Neumann L and Matas J (2012) “Real-time scene text localization and recognition”. In Proc CVPR 3538–3545
15. Pan YF, Hou X and Liu CL (2011) “A hybrid approach to detect and localize texts in natural scene images”. IEEE Trans on IP 800–813
16. Phan TQ, Shivakumara P and Tan CL (2012) “Detecting text in the real world”. In Proc ACM MM 765–768
17. Sharma N, Pal U and Blumenstein M (2012) “Recent advances in video based document processing: a review”. In Proc DAS 63–68
18. Sharma N, Shivakumara P, Pal U, Blumenstein M, Chew Lim Tan (2012) “A new method for arbitrarily-oriented text detection in video.” In Proc DAS 74–78

19. Shivakumara P, Trung Quy Phan and Chew Lim Tan (2010) "New fourier-statistical features in RGB space for video text detection". IEEE Trans CSVT 1520–1532
20. Shivakumara P, Dutta A, Phan TQ, Tan CL and Pal U (2011) "A novel mutual nearest neighbor based symmetry for text frame classification in video". Pattern Recogn 1671–1683
21. Shivakumara P, Phan TQ and Tan CL (2011) "A laplacian approach to multi-oriented text detection in video". IEEE Trans PAMI 412–419
22. Shivakumara P, Sreedhar RP, Phan TQ, Lu S and Tan CL (2012) "Multi-oriented video scene text detection through bayesian classification and boundary growing". IEEE Trans CSVT 1227–1235
23. Wang X, Huang L and Liu C (2009) "A new block partitioned features for text verification". In Proc ICDAR 366–370
24. Wong EK and Chen M (2003) "A new robust algorithm for video text extraction". Pattern Recogn 1397–1406
25. Wu W, Chen X and Yang J (2004) "Incremental detection of text on road signs from video with applications to a driving assistant systems". In Proc ACM MM 852–859
26. Xu C, Wang J, Wan K, Li Y and Duan L (2006) "Live sports event detection based on broadcast video and web-casting text". In Proc ACM MM 221–230
27. Yao C, Bai X, Liu W, Ma Y and Tu Z (2012) "Detecting texts of arbitrary orientations in natural images". In Proc CVPR 1083–1090
28. Yi C and Tian Y (2011) "Text string detection from natural scenes by structure-based partition and grouping". IEEE Trans Image Process 2594–2605
29. Zang J and Kasturi R (2008) "Extraction of text objects in video documents: recent progress". In Proc DAS 5–17
30. Zhang D and Chang SF (2002) "Event detection in baseball video using superimposed caption recognition". In Proc ACM MM 315–318
31. Zhang J, Goldgof D and Kasturi R (2008) "A new edge-based text verification approach for video". In Proc ICPR
32. Zhou J, Xu L, Xiao B and Dai R (2007) "A robust system for text extraction in video". In Proc ICMV 119–124

Originality and contribution

There are three main contributions in this work that are (1) text frame classification by proposing new symmetry features on text candidates, (2) multi-oriented text detection in video with good accuracy, where we have proposed new angle projection boundary growing method to tackle the multi-orientation problem and (3) achieving the best accuracy for ICDAR-03 data according to ICDAR-03 measures compared to the state of the art methods. Originality: (1) The way we combine wavelet-median moments, (2) Defining symmetry based on text pattern appearance, (3) use of directional features for false positive elimination and (4) angle projection boundary growing method for traversing multi-oriented texts.

Palaiahnakote Shivakumara is a Visiting Senior Lecturer in the Department of Computer Systems and Information Technology, Faculty of Computer Science and Information Technology, University of Malaya. He received B.Sc., M.Sc., M.Sc Technology by research and Ph.D degrees in computer science respectively in 1995, 1999, 2001 and 2005 from University of Mysore, Mysore, Karnataka, India.

From 1999 to 2005, he was Project Associate in the Department of Studies in Computer Science, University of Mysore, where he conducted research on document image analysis, including document image

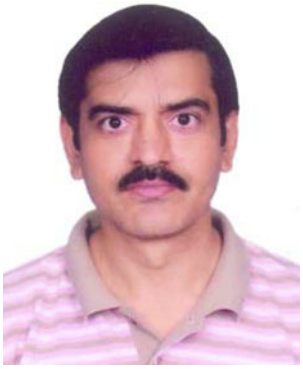
mosaicing, character recognition, skew detection, face detection and face recognition. He worked as a Research Fellow in the field of image processing and multimedia in the Department of Computer Science, School of Computing, National University of Singapore, from 2005-2007. He also worked as a Research Consultant in Nanyang Technological University, Singapore for a period of 6 months on image classification in 2007. He worked as a Research Fellow (RF) in National University of Singapore (NUS) from 2008 to 2013 on video text extraction and recognition. He has published more than 100 research papers in national, international conferences and journals. He has been reviewer for several conferences and journals.

His research interests are in the area of image processing, pattern recognition, including text extraction from video and document image processing.

Anjan Dutta received the B.Sc. degree in Mathematics from the University of Calcutta, Kolkata, India in the year 2006 and MCA degree in Computer Applications from the West Bengal University of Technology, Kolkata, India in the year 2009. Currently he is doing his Master degree in Computer Vision and Artificial Intelligence from the Universitat Autònoma de Barcelona, Barcelona, Spain and also at the same time he is working as a PhD student in the Computer Vision Centre, Barcelona, Spain under the supervision of Dr. Josep Lladós and Dr. Umapada Pal. His main research interests include Graphics Recognition, Structural Pattern Recognition using graph matching technique.

Chew Lim Tan is a Professor in the Department of Computer Science, School of Computing, National University of Singapore. He received his B.Sc. (Hons) degree in physics in 1971 from University of Singapore, his M.Sc. degree in radiation studies in 1973 from University of Surrey, UK, and his Ph.D. degree in computer science in 1986 from University of Virginia, U.S.A. His research interests include document image analysis, text and natural language processing, neural networks and genetic programming. He has

published more than 400 research publications in these areas. He is an associate editor of Pattern Recognition, associate editor of Pattern Recognition Letters, an editorial member of the International Journal on Document Analysis and Recognition. He is a fellow and a member of the Governing Board of the International Association for Pattern Recognition (IAPR). He is also a senior member of IEEE.



Umapada Pal received his Ph.D. from Indian Statistical Institute and his Ph.D. work was on the development of Printed Bangla OCR system. He did his Post Doctoral research on the segmentation of touching English numerals at INRIA (Institut National de Recherche en Informatique et en Automatique), France. From January 1997, he is a Faculty member of the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata and at present he is a Professor. His primary research is Digital Document Processing. He has published 214 research papers in various international journals, conference proceedings and edited volumes. In 1995, he received student best paper award from Chennai Chapter of Computer Society of India. He received a merit certificate from Indian Science Congress Association in 1996. Because of his significant impact in the Document Analysis research domain of Indian language, TC-10 and TC-11 committees of IAPR (International Association for Pattern Recognition) presented 'ICDAR Outstanding Young Researcher Award' to Dr. Pal in 2003. In 2005-2006 Dr. Pal has received JSPS fellowship from Japan government. In 2008 Dr. Pal received Visiting fellowship from Spanish Government. Dr. Pal has been serving as a program committee member of many conferences including International Conference on Document Analysis and Recognition (ICDAR), International Conference on Frontiers of Handwritten Recognition (ICFHR), International Conference on Pattern Recognition (ICPR), Internal Workshop on Document Analysis and Systems (DAS) etc. He is the Editorial board member of International Journal of Computer, Mathematical Sciences and Applications; Electronic Letters on Computer Vision and Image Analysis; and ACM Transactions on Asian Language Information Processing. Also he has served as the guest editor of special issue of VIVEK, Electronic Letters on Computer Vision and Image Analysis, and International Journal on Document Analysis and Recognition. He has served as Program-Chair for ICDAR-2009 and DAS-2012 and Organizing Chair for ICFHR-2010. He is a life member of IUPRAI (Indian unit of IAPR) and senior life member of Computer Society of India.