A New Technique for Multi-Oriented Scene Text Line Detection and Tracking in Video

Liang Wu, Palaiahnakote Shivakumara, Tong Lu, *Member, IEEE*, and Chew Lim Tan, *Senior Member, IEEE*

***Abstract*—Text detection and tracking in video is challenging due to contrast, resolution and background variations, and different orientations and text movements. In addition, the presence of both caption and scene texts in video aggravates the problem because these two text types differ in characteristics significantly. This paper proposes a new technique for detecting and tracking video texts of any orientation by using spatial and temporal information, respectively. The technique explores gradient directional symmetry at component level for smoothing edge components before text detection. Spatial information is preserved by forming Delaunay triangulation in a novel way at this level, which results in text candidates. Text characteristics are then proposed in a different way for eliminating false text candidates, which results in potential text candidates. Then grouping is proposed for combining potential text candidates regardless of orientation based on the nearest neighbor criterion. To tackle the problems of multi-font and multi-sized texts, we propose multi-scale integration by a pyramid structure, which helps in extracting full text lines. Then, the detected text lines are tracked in video by matching the subgraphs of triangulation. Experimental results for text detection and tracking on our video dataset, the benchmark video datasets, and the natural scene image benchmark datasets show that the proposed method is superior to the state-of-the-art methods in terms of recall, precision, and F-measure.**

***Index Terms*—Delaunay triangulation, multi-oriented video text detection, multi-sized text detection, text detection, text tracking.**

I. Introduction

RAPID proliferation of multimedia contents that are available for broadcast and Internet leads to an increasing need for their ubiquitous access at anytime and anywhere over a variety of receiving devices, especially due to the escalating popularity of high performance and low price digital camera devices or smart phones [1]–[4]. Taking video into daily life is becoming very common, which leads to an explosive growth of video and hence a dramatic increase in the size of heterogeneous video databases [5]. For example, the total number of YouTube video clips has amounted to over 120 million [5]. It is clear that when accessing lengthy and voluminous video programs, the ability to access highlights and to skip less interesting parts of video will save not only a viewer's time but also data downloading or airtime costs, especially when the viewer receives video wirelessly from remote servers. Moreover, it would be very attractive if users can access and view the content based on their preferences [1]. For instance, users prefer to retrieve exciting events or navigate events from personal collections for a particular person or vehicle. To realize the above needs, each source video has to be tagged with proper semantic labels. There are plenty of methods in the literature for annotating video based on their contents [6]. However, these methods may fail to annotate video with semantics due to the gap between low-level and high-level features [7]–[10]. This motivates researchers to use text information present in video to label the semantics of events because video texts provide a kind of high-level information closely related to video content with the help of an Optical Character Recognizer (OCR). Therefore, text detection and tracking can serve as a basis for numerous multimedia applications, such as landmark recognition, navigation guidance for the visually impaired, enhancing safe driving, traffic monitoring, license plate detection, tourists and transportation information systems. Furthermore, in augmented reality, signs and texts of foreign languages contained in video or natural scene images can be automatically translated for visitors by their smart phones. In summary, text detection helps in annotating video data, which in turn helps in video summarization, personalized video retrieval, etc. All these applications require robust and effective text detection and tracking in video [1]–[5].

Generally, video contains two types of texts: (1) caption text which is manually edited, and (2) scene text which exists in video naturally. Since caption text is edited, we can expect such texts to be of a good clarity or contrast, which are usually aligned in either horizontal or vertical direction. On the other hand, scene text is a part of a video frame/image and its nature is unpredictable. Thus it suffers from non-uniform illumination, perspective distortion, low resolution, low contrast, varying font types and font sizes, multiple colors and arbitrary orientations, etc. [11]–[19]. The presence of such types of texts in video adds more complexity to the text detection and tracking problem. Therefore, accurate text detection and tracking in unconstrained environments is still an elusive goal for researchers.

Manuscript received September 24, 2014; revised December 28, 2014; accepted May 28, 2015. Date of publication June 10, 2015; date of current version July 15, 2015. This work was supported in part by the Natural Science Foundation of China under Grant 61272218 and Grant 61321491, by the Program for New Century Excellent Talents under Grant NCET-11-0232, and by the University of Malaya HIR under Grant UM.C/625/1/HIR/MOHE/ENG/42. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jiebo Luo. *(Corresponding author: Tong Lu.)*

L. Wu and T. Lu are with the National Key Lab for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: wuliang0301@hotmail.com; lutong@nju.edu.cn).

P. Shivakumara is with the Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia (e-mail: hudempsk@yahoo.com).

C. Lim Tan is with the School of Computing, National University of Singapore, Singapore (e-mail: tancl@comp.nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2015.2443556

Text detection and recognition is actually not a new problem for the multimedia content analysis field as we can see several methods for text detection in natural scene images in the literature [8]–[10], [20]–[23]. Usually, text in natural scene images has a complex background. To solve this problem, many methods have been developed [20]–[23]. Since scene images are often captured by a high resolution camera, these methods take the advantage of the high contrast property of scene texts to propose features based on the shapes of characters. Therefore, these methods rely on connected component analysis or shape analysis of characters to achieve a good accuracy. However, these methods may not be used directly on video for text detection and tracking because video generally suffers from both complex background as well as low resolution. This results in a lot of disconnections and distorted shapes, and hence it is hard to preserve the shape of each character.

There are methods that have been proposed to detect texts in video by addressing the problems of low resolution and complex background in the literature [11]–[14]. These methods can be categorized as connected component based, texture based, and edge or gradient based methods. Since connected component based methods [24], [25] require the full shape of each character, these methods may not be suitable for text detection in video. To address the problem of complex background, texture based features are developed [17], [18], [26], [27]. These methods are computationally expensive and their performance lies in the training of a proper classifier and a large number of training samples. To achieve efficiency, the methods that are based on edge or gradient information have also been developed [6], [15], [16], [19], [28], [29]. These methods work well with less computation; however, they are sensitive to background, thus giving a rise to more false positives. Overall, most of the above methods focus on caption text detection in video but not for both caption and scene texts. Several methods have been developed to find a solution for the presence of both types of texts with different orientations in video [19], [26], [27], [29]. These methods basically explore the contrast information of text rather than the use of the characteristics of texts. Despite solving the problem of orientation, these methods do not utilize temporal information of video, but rather rely on individual frames. Therefore, these methods cannot be used for text tracking in video.

II. Related Work

This section provides a literature review of the existing methods [18], [30]–[40], which use individual frames and temporal frames for text detection and tracking in video, respectively.

Li *et al.* [18] proposed a method for video text tracking based on wavelet and moments features. This method uses the advantage of wavelet decomposition and spatial information provided by the moments with a neural network classifier for identifying text candidates. Text block shape features are used to track texts in temporal frames. Huang *et al.* [30] proposed a method for scrolling text detection in video using temporal frames. This method uses motion vector estimation for detecting texts. However, this method is limited to only scrolling texts but not arbitrarily oriented texts. Zhou *et al.* [31] exploited edge information and geometrical constraints to form a coarse-to-fine methodology to define text regions. Then candidate regions are labelled as connected components by morphological operations. Based on temporal redundancy, text authentication and enhancement are done. Mi *et al.* [32] proposed a text extraction approach based on multiple frames. Edge features are explored with similarity measures for identifying text candidates. Wang and Chen's method [33] used a spatio-temporal wavelet transform to extract text objects in video. Statistical features are used with a Bayesian classifier for final text region classification. Huang [34] detected video scene texts based on video temporal redundancy. The method performs motion detection in 30 consecutive frames to synthesize a motion image. Further, video scene text detection is implemented in single frames to retrieve candidate text regions. Finally, the synthesized motion image is used to filter out candidate text regions and only keep those candidate text regions that have motion occurrence as the final scene texts. Huang *et al.* [35] proposed a method for video text detection using temporal frames based on motion features by integrating multiple frames, which give text regions. Corner points are used to find candidate text pixels and then a growing process is proposed for connecting candidate text pixels. However, this method focusses on horizontal graphics texts but not arbitrary scene texts. Zhao *et al.* [36] proposed an approach for text detection using text corners in video. This method proposes to use dense corners for identifying text candidates. From the corners, the method forms text regions using morphological operations. Then the method extracts features such as area, aspect ratio and orientation from the regions to eliminate false ones. Finally, optical flow is used for the verification of text blocks. Liu *et al.* [6] proposed a method for video caption text detection using stroke like edges and spatio-temporal information. A color histogram is used for segmenting texts. Li *et al.* [37] proposed a method for video text detection using multiple frames integration. This method uses edge information to extract text candidates. Morphological operations and heuristic rules are proposed to extract the final texts from video. Mosleh *et al.* [38] proposed an automatic inpainting scheme for video text detection and removal based on stroke width transform to identify text objects. Then motion patterns of text objects of each frame are analyzed to localize video texts. The detected text regions are removed, and finally video is restored by an inpainting scheme. The objective of the method is to restore missing information due to overlapping caption texts. The method is proposed to detect horizontal texts but not for arbitrary text detection in video.

A few methods have been developed for text detection and tracking text in video [5], [18]. Yusufu *et al.* [5] proposed a video text detection and tracking system based on SURF features and a classifier. This method proposes many heuristics based on edge and morphological information for identifying text candidates. Moving text blocks are identified using motion estimation. The text blocks in the previous frame and the current frame are tracked when the features are found to be matching. We can observe from these methods that they are developed to track caption texts in the horizontal direction because detecting and tracking caption texts is much easier than detecting scene texts. Therefore, none of the methods addressed the problem of

scene text detection and tracking in video regardless of orientation, non-linear movement with multilingual ability.

Similar to this work, Wu *et al.* [19] recently proposed a method for detecting both caption and scene texts in video. The scope of this method is limited to text detection but not tracking. In addition, the method does not work well for texts of multiple fonts or sizes in video. Optical flow based properties have also been proposed by Shivakumara *et al.* [39] for dynamic curved text detection in video. The method is good when text is moving while the background remains static. It thus fails when the background moves. Moreover, the scope of the method is limited to text detection but not tracking. Arbitrary text detection is also proposed by Shivakumara *et al.* [29] by proposing gradient vector flows and grouping. Though the method finds a solution to the complex curved text detection problem, it does not explore temporal information for text detection in video. Therefore, the methods cannot be used for text tracking.

In summary, the above discussion shows that most of the current methods focus on caption texts in the horizontal direction for detection. The methods make use of temporal frames for enhancing text detection performance but not for tracking them in video. The use of spatial information for text detection and temporal information for tracking is ignored. Therefore, there is an immense scope for developing new methods that are capable of detecting and tracking texts in video accurately and efficiently irrespective of text types and orientations.

In this paper, we propose a new technique for text detection and tracking based on spatial and temporal information. Inspired by the work presented in [6] for caption text detection using temporal frames, we propose to use spatial information in a different way, namely, inter or intra character symmetry and dense corners by Delaunay Triangulation, for identifying text candidates. The proposed technique tracks every text by identifying its motion status based on transformations and templates. Moreover, the technique proposes multi-scale integration in a pyramid structure to tackle multi-font and multi-size problems. The main contributions are three folds: (1) Exploring invariant features, namely, symmetrical features that exist in inter and intra characters with the help of Delaunay Triangulation for identifying text candidates. These features work for any orientation and any type of texts and scripts; (2) The use of multi-scale integration for multi-sized and multi-oriented text lines detection in a new way; and (3) Tracking arbitrarily oriented texts through motion status identification in video by matching the sub-graphs given by Delaunay Triangulation with respect to different motion status.

III. Proposed Approach

The flow diagram of the proposed technique is shown in Fig. 1, where we can see that the proposed technique consists of two parts. The first part focusses on text detection, while the second part focusses on text tracking in video. As we are inspired by the work presented in [18], [36], [38] for text detection and tracking, where the methods used the first frame for text detection and then use temporal frames for text tracking and restoration, we follow the similar way to use the first frame in video for text detection and then temporal frames for text tracking. It is true that character components

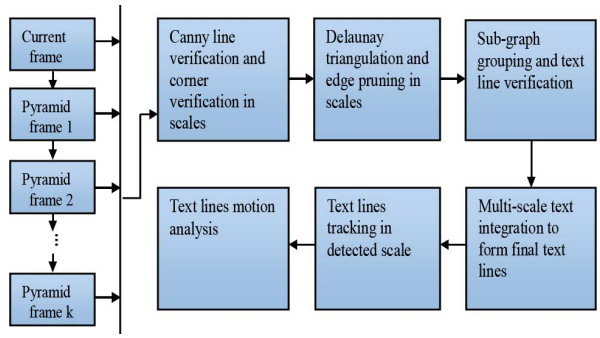


Fig. 1. Flow chart of our text detection, tracking and motion analysis.

have double strokes (parallel edges) with nearly constant stroke width [9]. This cue indicates the presence of texts. With this cue, we derive symmetry between strokes based on the gradient direction to smooth the edge map of an input frame, resulting in the elimination of unwanted components. In the same way, it is noted that the character components in the same text line exhibit symmetry between intra and inter character components due to regular spacing between character components and constant stroke width [10]. This observation motivates us to propose Delaunay Triangulation for the corners of the smoothed edge map, which extracts inter and intra symmetries by studying the spatial relationship between the corners. This results in text candidates for the input frame. The proposed method explores geometrical properties of sub-graphs formed by triangulation, such as edge strengths and density for text candidates to eliminate false ones, which results in potential text candidates. It is true that multi-level decomposition in a pyramid structure is an interesting idea for tackling multi-sized or multi-font texts [18]. Therefore, we propose multi-scale integration in a pyramid structure on different scales to extract the full text line. We can assert that the proposed features are independent of orientation, font, font size, orientation, text type and script. This is the main advantage of the proposed method, which is a departure from the existing methods. In the second part, the detected texts are tracked in video with respect to motion status such as zoom in, zoom out and rotation along with arbitrary movements (linear or non-linear) with 2D transformations and the KLT tracker. While tracking, the proposed technique tests the same invariant features mentioned above to find matches. The complete flow of the proposed technique can be seen in Fig. 1.

A. Gradient Directions Pair for Smoothing Edge Map

For each video frame, the proposed technique obtains its Canny edge image using Canny edge operator because Canny operation gives fine details for both low and high resolution texts in video compared to the other edge operators such as Sobel and Prewtt operators [10]. However, due to the complex background of video, Canny operation often gives erratic edges for the components in the background. To eliminate such erratic edges, we propose the idea of symmetry in gradient directions because of the fact that character components have parallel edges and exhibit symmetry in shape. We divide the directions of each pixel on an edge component in the edge map into eight bins as shown in Fig. 2(a) to study the distribution of the directions. The technique performs grouping by merging the

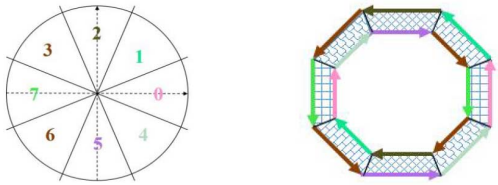


Fig. 2. Parallel double edges of character components. (a) Eight parts of directions. (b) Eight directions as four pairs.

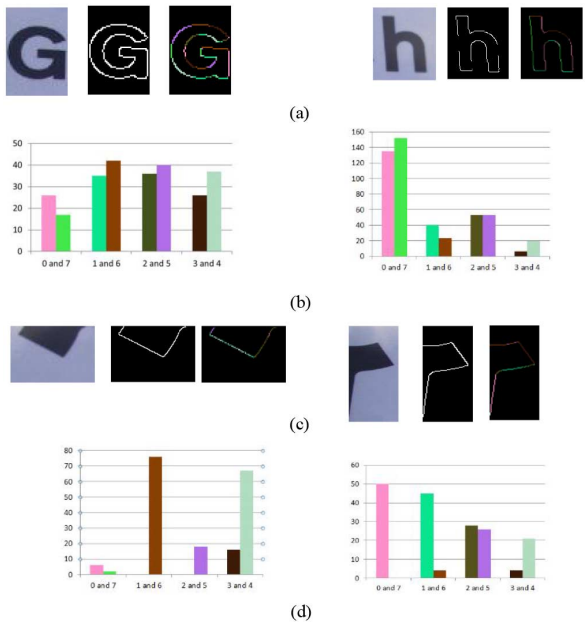


Fig. 3. Symmetry features of text and non-text components for directional pair. (a) Gray character image, Canny edge image, and gradient image. (b) Histograms based on directions for the gradient image in (a). (c) Gray non-text components, Canny edge image, and gradient image. (d) Histograms based on directions for the gradient image in (c).

gradient magnitude values that have opposite directions. Essentially, this leads to four paired groups as shown in Fig. 2(b), where we can see the parts that have different colors of both inner and outer contours share the same pattern. This is called symmetry. It is illustrated in Fig. 3, where (a) and (c) show sample text and non-text components with their respective Canny and gradient maps, respectively, while (b) and (d) show the histograms of the four directional pairs for the images in (a) and (c), respectively. It is observed from Fig. 3(b) and (d) that the bars in the histograms of the four paired groups look almost the same, while the bars of non-text components do not. This is valid because these pairs are formed based on the fact that character components have regular shapes and symmetric in nature, while non-text components do not exhibit regular shapes and hence they may not satisfy the directional symmetry condition. If any component in the edge map satisfies such symmetry, it is considered as a text component, or else it is considered as a non-text component. More specifically, the algorithmic steps are as follows.

First, we calculate the gradient map for the input gray frame. We segment the gradient direction from 0 to 360° into eight parts as shown in Fig. 2(a). Then we consider each part and its corresponding part whose direction is opposite to it as one

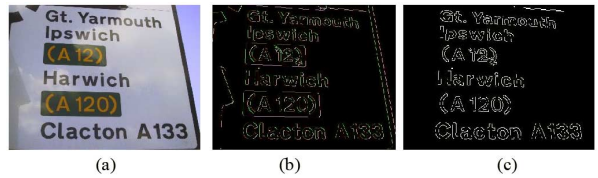


Fig. 4. Effect of smoothing. (a) Input frame. (b) Gradient map. (c) Smoothed map.

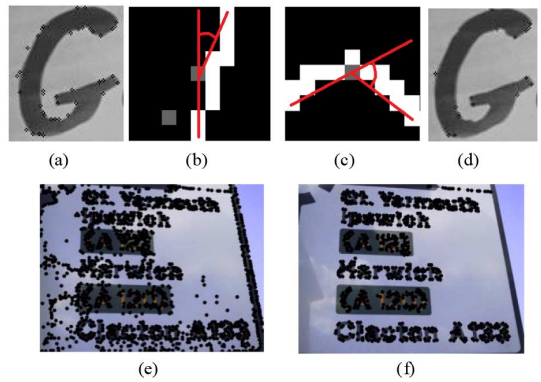


Fig. 5. Edge angle at corners for corner verification: (a) example of corners for text component; (b) false corner, which is to be removed; (c) example of a right corner, which is to be retained; (d) the effect of false corner removal; (e) original corners in the input image; and (f) verified corners in the input image.

pair, resulting in four pairs of parts as shown in Fig. 2(b). For example, $(337.5°, 360°) \cup (0, 22.5°)$ and $(157.5°, 202.5°)$ are said to be one pair as these have opposite directions to each other with the same gradient magnitude values. The effect of symmetry to eliminate non-text components which appear like straight lines on the full frame can be seen in Fig. 4, where (a) is the gray input frame, (b) is the gradient map with colors of different directions, and (c) is the smoothed edge map after filtering. However, we can notice from Fig. 4(c) that there are non-text components that satisfy the symmetry. Therefore, this step is considered as the preprocessing step as smoothing the edge map, which helps sub-sequent steps for obtaining better text candidates.

B. Spatial Study of Corners for Text Candidates Selection

It is noticed that in the previous step the symmetry misclassifies non-text components as text components due to the complex background of video. To filter out such non-text components, we propose corner and spatial study. Corner based features for text detection is a topic of interest due to its resilience to font, font size, orientation, text type, and to some extent distortion [36]. Initially, the technique obtains corners using Harris method as shown in one sample in Fig. 5(a), where one can notice that small curvatures are considered as corner points. Therefore, we propose modifications to the conventional Harris corner detection algorithm based on edge angle and its quantization as follows.

Let P be a corner point detected by the Harris method as shown in Fig. 5(a), we locate it at the center of a window of size $N \times N$. We determine empirically the size of N as 9 in this work. Along the edge points in the window, we find the farthest edge point from the center. Then the proposed technique finds

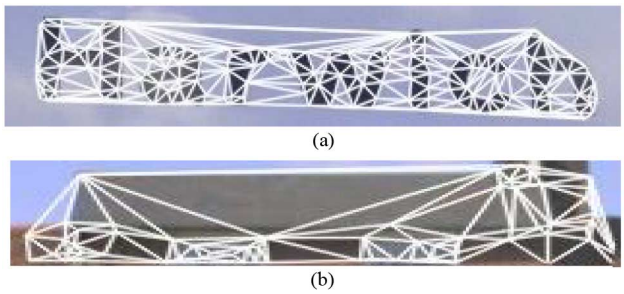


Fig. 6. Appearance pattern of Delaunay triangulation for text and non-text lines. (a) Delaunay triangulation for text line. (b) Delaunay triangulation for non-text line.

Fig. 7. (a) Triangulated mesh, (b) SWT_1, and (c) SWT_2.

an angle between the farthest point and the centroid as shown in Fig. 5(b). If this angle is too small then we ignore the corner which is shown in Fig. 5(b), where the corner points do not contribute much deviation. On the other hand, the corner points shown in Fig. 5(c) contribute enough deviation and hence these corners are considered as correct corner points. In this way, the proposed technique finds correct corners as shown in Fig. 5(d). The effect can be seen in Fig. 5(f), where the corner points that represent non-text components as shown in Fig. 5(e) are removed.

As stated in [40], Delaunay triangulation is good for preserving the spatial relationship between corners to study the characteristics of objects irrespective of orientation and to some extent distortion. Thus we propose to explore Delaunay triangulation for the detected stable corners to study their spatial features such as the proximity between the corners, and inter/intra character components through connected graph and sub-graph formation to identify text candidates. These properties motivated us to propose Delaunay triangulation for text detection and tracking, which requires robust features mentioned above to achieve good results. Further, these features help us to distinguish text and non-text components as shown in Fig. 6(a) and Fig. 6(b), where the triangular mesh pattern reflects text characteristics such as regular spacing between characters and constant stroke width distances, while the pattern of non-text does not. This is the advantage of Delaunay triangulation.

The same observation can be seen in Fig. 7(a), where the triangulated meshes for the whole image Fig. 5(f) using corners are shown. The proposed technique extracts the above features, which represent video texts through sub-graph properties as follows. For each triangle, the technique calculates the distances between nodes to eliminate the edges that connect from one text line to another as shown in Fig. 7(a), where those edges are marked by white color because they do not contribute to text components detection. Since the spacing between character components, words or successive lines is lesser than the

Fig. 8. (a) The results of spatial proximity between nodes. (b) Edges which represent non stroke width distance. (c) Result of stroke width property.

spacing between two random lines, the distance between the two nodes of the character components that belong to two successive text lines would be small compared to the distance between the nodes that represent two farthest text lines, as shown by the white color edges in Fig. 7(a). This results in the removal of unwanted edges as shown in Fig. 8(a), where the white color edges in Fig. 7(a) are removed. Mathematically, it can be defined as

$$Sp_{p,q} = len_{p,q} \tag{1}$$

where $Sp_{p,q}$ is the value of spatial proximity between nodes p and q, $\mathrm{len_{p,q}}$ is the length of the edge between nodes p and q. The threshold is determined automatically by calculating the average distance between characters, words and lines in the smoothed edge map. This is valid because the directional symmetry eliminates most of the non-text components from the Canny edge map. As a result, the image retains more text components.

Second, for each component in the smoothed map, we estimate stroke width using the gradient direction as described in [21]. Generally, stroke width distance is constant throughout a character [21], [40]. Therefore, we estimate stroke width consistency to identify text candidates. According to [21], the stroke width for a pixel p is defined as a ray $r = p + nd_p$, which moves in a direction perpendicular to the stroke direction from p until it reaches another edge pixel, say q. Here d_p denotes stroke orientation, and d_p is estimated for pixel q. The ray from p moves approximately opposite to $d_p (d_p = -d_q \pm \pi/6)$. The distance between p and q is considered as the stroke width. We also find the stroke width for the opposite gradient direction when the gradient directions point to the outside of a part, which means the background is darker than the character part as shown in the SWT images in Fig. 7, where (b) shows the stroke width of the gradient direction and (c) shows the stroke width of the opposite direction. With these stroke width distances, we calculate stroke width consistency by estimating the standard deviation of stroke widths along the graph edges. The formulation is as follows:

$$Cs_1^{p,q} = \sqrt{\frac{1}{len_{p,q}} \sum_{i=1}^{i=len_{p,q}} \left(\frac{SWT_1\left(p + \frac{q-p}{|q-p|} \cdot i\right)}{\overline{SWT_1^{p,q}}} - 1 \right)^2} \tag{2}$$

$$Cs_2^{p,q} = \sqrt{\frac{1}{len_{p,q}} \sum_{i=1}^{i=len_{p,q}} \left(\frac{SWT_2\left(p + \frac{q-p}{|q-p|} \cdot i\right)}{\overline{SWT_2^{p,q}}} - 1 \right)^2} \tag{3}$$

$$Cs^{p,q} = min(Cs_1^{p,q}, Cs_2^{p,q}) \tag{4}$$

(a) (b) (c)

Fig. 9. Candidate edges after considering the Canny rate: (a) dilation results, (b) triangulated mesh for image, and (c) is the result of Canny rate.

where SWT_1 and SWT_2 are two stroke width maps, respectively, generated from two opposite directions. $Cs_k^{p,q}$ is the value of the stroke width consistency of the edge between nodes p and q calculated from SWT_k, and $Cs^{p,q}$ is the smaller one between $Cs_1^{p,q}$ and $Cs_2^{p,q}$. $SWT_k(o)$ is the value of SWT_k at pixel o. $\overline{SWT_k^{p,q}}$ gives the average SWT_k value of the edge between nodes p and q, which is represented as $\overline{SWT_k^{p,q}} = \frac{1}{len_{p,q}} \sum_{i=1}^{i=len_{p,q}} SWT_k(p + \frac{q-p}{|q-p|} \cdot i)$. If the stroke width consistency of one edge $Cs^{p,q}$ is too large, then we prune this edge. As a result, the proposed technique retains the edges that satisfy stroke width consistency and eliminates the edges that do not satisfy as shown by the white color edges in Fig. 8(b). The final results can be seen in Fig. 8(c), where the white color edges that represent the spacing between text lines are successfully removed because these strokes do not meet stroke width consistency.

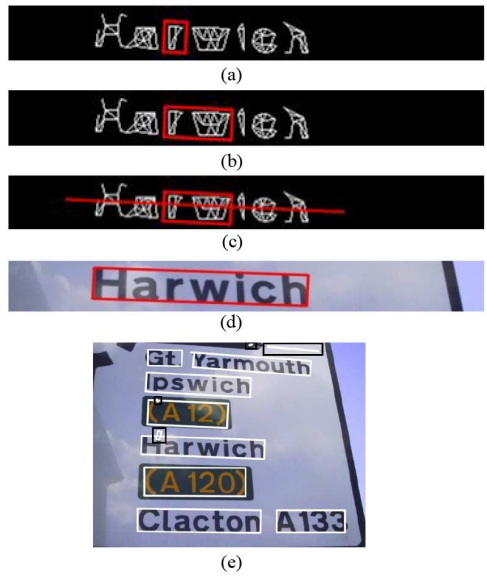
Fig. 8(c) still contains some unwanted edges that connect the components of one line to the components of another line. We propose one more feature called Canny rate which considers dense corners of text components. For each component in the smoothed edge map, we propose to use Canny rate through triangulated mesh as shown in Fig. 8(c) to identify candidate edges. Before testing the third property, we perform dilation over the smoothed edge map to connect small gaps between the edges as shown in Fig. 9(a) for the smoothed edge map in Fig. 4(c). As a result, we eliminate those edges in nontext regions, which are shown in Fig. 9(b) with white color. Formally, we define this feature as

$$Cr_{p,q} = \frac{\left|E_{p,q} \cap^{C_{p,q}}\right|}{|E_{p,q}|} \tag{5}$$

where $\mathrm{Cr_p, q}$ is the canny rate of the edge between nodes p and q, $\mathrm{E_{p,q}(t)}$ and $\mathrm{C_{p,q}(t)}$ respectively represent the set of the pixels belonging to the edge between nodes p and q, and the set of the pixels belonging to Canny edge components. We expect equation (5) to give high values (it is set to 1) for video texts and low values for non-texts in video (other than 1). The final result is shown in Fig. 9(c). In this way, the above three features help us to identify the candidate edges that represent text candidates.

C. Directional Region Grouping for Merging Text Candidates

Each text candidate given by the previous step is considered as a cluster. In order to extract the full text line, we need to group all the clusters that represent text candidates. For this purpose, we propose Directional Region Grouping (DRG) which first finds the direction of text clusters and then groups the nearest neighbor clusters in the text direction. Initially, the technique chooses a seed cluster as shown in Fig. 10(a) and then finds a direction using Principal Component Analysis (PCA) if the cluster has more than two character components. Here we consider the largest Eigen vector to find the angle for the text cluster. The proposed technique searches for the nearest neighbor cluster by finding the distance between the seed cluster and its neighbor clusters as shown in Fig. 10(b) along the text direction given by PCA as shown in Fig. 10(c). The final result of grouping for one line can be seen in Fig. 10(d). For the whole frame, DRG gives results as shown in Fig. 10(e), where it fixes a bounding box for the grouped regions. This grouping will terminate when the cluster set becomes empty. This idea works well because of the fact that the spacing between character components is usually lesser than the spacing between text lines. The advantage of DRG is that it works well for any orientation of text lines. The DRG is illustrated in Fig. 10, where bounding boxes are fixed for text regions.



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

Fig. 10. Illustration for DRG with some false positives in black bounding boxes. (a) Seed cluster. (b) Merging a nearest neighbor cluster of same size. (c) Direction of the text line. (d) Grouping all text clusters along text direction. (e) Final grouping result.

D. False Positives Removal

Due to the problem of low resolution and complex background, it is hard to remove false positives completely with the features. Therefore, we propose two rules of cluster groups, which represent text lines to reduce false positives.

Rule 1 is based on dense corners. It is true that due to more cursive nature of text, one can expect dense corners for text blocks and low corner density for non-text blocks. With this notion, we define rule-1 as the number of the corners of a text block should not be a too small value. Let Ngc be the number of the corners in a text block. If it is smaller than a threshold NT, we consider it as a false positive.

Rule 2 is based on edge strength. For an ideal text block, the number of the edges of the text block should be larger than the number of the corners in the text block. With this property, we define rule-2 as the ratio of the number of the edges in a text

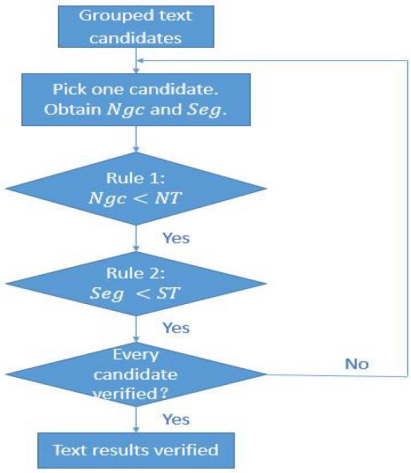


Fig. 11. Logical steps for removing false positives.

Fig. 12. False positive removal. (a), (c) False positives marked by green color and blue color. (b), (d) The effect of rule-1 and rule-2 where both the false positives are eliminated..

block divided by the number of the corner points in the same block. This results in edge strength of a text block as defined in (6)

$$Seg = \frac{Nge}{Ngc} \tag{6}$$

where Seg denotes edge strength and Ngc as earlier defined is the number of corners points in the text block. Thus, if Seg is smaller than a threshold ST, we discard it or else it is considered as a text block. The algorithmic steps of false positive verification are shown in Fig. 11. The false positive elimination is illustrated in Fig. 12, which shows two types of false positives that are denoted by green color and blue color. The output of Section C is considered as text blocks for text block verification. The effect of the two rules can be seen in Fig. 12, where false positives (green and blue color components) shown in Fig. 12(a) are removed as shown in Fig. 12(b). One more example is shown in Fig. 12(c) and (d), where the non-text components created by the background complexity are eliminated in Fig. 12(d) using (6).

The two threshold values for NT and ST are determined empirically by conducting experiments on the recent benchmark

Fig. 13. Effect of multi-scale integration. (a) Result of scale 0. (b) Result of scale 1. (c) Result of scale 2. (d) Final result of text detection.

ICDAR 2013 training dataset, which for the first time include video data for detecting texts. The details of the experiments can be found in the Experimental section.

In this work, we prefer to propose objective rules for removing false positives rather than using a classifier, which is popular nowadays for the purpose of false positive elimination [41]. The main reason is that the objective of the work is to propose a generalized algorithm for text detection and tracking, which should work well irrespective of scripts and orientations. This is also because text detection and tracking is actually a preprocessing step to text recognition in video. The proposed two rules work based on dense corners and edge strength of the text components. We believe these rules behave like objective heuristics but not hard and fast rules which may be sensitive to different situations. In the light of this, if we use a classifier for removing false positives, it may restrict the ability to detect and track the text of different script for different datasets. In addition, finding training samples for non-text components is not trivial because the boundary of a non-text is unknown and sometimes difficult to define for a classifier.

E. Multi-Scale Integration for Multi-Sized and Multi-Oriented Text Detection

Since the nature of scene text in video is unpredictable, it can have any orientation and font size. We propose multi-scale integration to handle this ill posed problem. We use a three-level pyramid structure approach to detect texts with a wide range of font sizes. For the input frame, we apply the steps from subsection A to subsection D on three levels to find text blocks. Then the bounding boxes at different levels are mapped to find actual text blocks. If a bounding box at one level does not overlap with the boxes at other levels, we consider it as a separate box that has a single text line. If the bounding box of a text line overlaps with the bounding boxes of the same text line at multiple levels, then we merge them by forming a new bounding box that can include all with union operation. The whole process is illustrated in Fig. 13, where (a) shows the results of the original input frame, (b) gives the results of scale 1 where we can see some of the characters are missed without bounding boxes, (c) illustrates the results of scale 2 where some of the characters are detected with bounding boxes and few characters are missed, and (d) shows the final results after integrating all the

text blocks at scale 0, scale 1 and scale 2, where we can see all the texts are covered by proper bounding boxes. This is the advantage of multi-scale integration.

We determine different scales automatically with the original input frame of scale 0 as follows. Each scale of an image is 0.8×0.8 of its next upper scale. This is required to keep the height of the top image in pyramid to be smaller than 100. Therefore, the input images of different sizes produce different numbers of layers or scales in the pyramid structure as defined in (7)

$$SN = \left\lceil \log_{0.8} \frac{100}{H} \right\rceil \tag{7}$$

where SN gives the number of scales and H is the height of the input image.

IV. Text Tracking Through Motion Status Analysis

The text detection step gives texts with bounding boxes for text lines in video. In this section, we further track text lines of any orientation or font size at different levels. We propose to use KLT tracker as it is simple and good for tracking texts in video. The tracking step involves motion status analysis and tracks text lines based on identifying motion status. In this work, we consider linear as well as non-linear motions. We propose to use 2D transform matrix for identifying motion status. Since the proposed technique identifies motion status, it works well for both linear and non-linear motions and hence improves text tracking results. This is the advantage of the technique compared to the existing tracking methods [18], [36], [38]. As detected text regions are represented by sub-graphs of corner points at different levels, the KLT tracker tracks them using corners of text regions in the same level where texts are detected and generates a set of trajectories within a specific time window of video sequence. The length of time window is named τ, which means the number of the frames in the video sequence.

A. Motion Status Analysis

Text motion in digital video consists of three types: static, simple linear and complex nonlinear such as zooming in and out or rotation. The goal of this section is to respectively derive the transformation matrices for zooming, rotation and translation, namely, M_z, M_r and M_t, which are all 3×3 matrices. Once we find these transformation matrices then it will be easy to identify the status of text motions. The theoretical analysis for deriving these transformation matrices is as follows.

The position of point $\mathrm{p}_0\ [xy]$ after zooming, rotation or translation can be obtained by

$$[x'\ y'\ 1] = [x\ y\ 1]\cdot M_T \tag{8}$$

where M_T represents one of 2D transformation matrices, namely, M_z, M_r and M_t. So $[xy1]\cdot M_z\cdot M_r\cdot M_t$ are the transforms required for point p_0 to identify its motion status such as zooming, rotation and translation, respectively, which is illustrated in Fig. 14. Given a 2D transform matrix $M = M_z\cdot M_r\cdot M_t$, then $[xy1]\cdot M_z\cdot M_r\cdot M_t = [xy1]\cdot M$, where

$$M = \begin{bmatrix} s_x cos\alpha & s_x sin\alpha & 0 \\ -s_y sin\alpha & s_y cos\alpha & 0 \\ dx & dy & 1 \end{bmatrix} \tag{9}$$

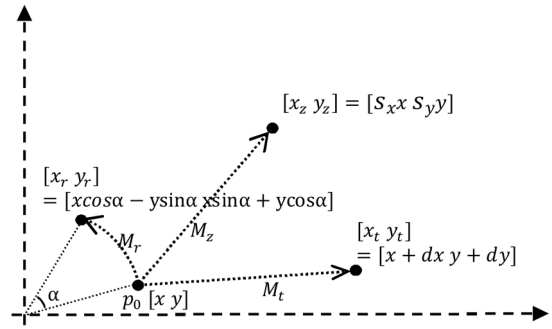


Fig. 14. Illustrating the transformation matrices, such as zooming matrix M_z, rotation matrix M_r, and translation matrix M_t for the point $p_0\ [xy]$.

in which α is the rotation angle, $[s_x\ s_y]$ is the zooming scale and $[dx\,dy]$ is the translation vector. According to the forms of M_z, M_r and M_t, we can easily obtain the exact values of them if we know M.

In our work, we apply these equations on the corners in successive temporal frames to identify text block motion status. Since the coordinates of the corners in two successive frames do not change much, we use the coordinates of the corners of every τ frame to identify motion status by deriving transform matrix M. For instance, let K be the number of the corners in a detected text and let the coordinates of the corners be $[x_k\ y_k](k\in\{1,2,\ldots,K\})$. After tracking them in 10 frames, their coordinates have become $[x'_k y'_k]\ \ (k\in\{1,2,\ldots,K\})$. Importantly, we need to move the origin to the center of the text because rotation does around the origin. After coordinate conversion, the previous coordinates have become $[x'_k y'_k]\ \ (k\in\{1,2,\ldots,K\})$ and the latter ones have become $[x''_k y''_k](k\in\{1,2,\ldots,K\})$. According to the theoretical analysis discussed above, we get the following equation:

$$\begin{bmatrix} x''_1 & y''_1 & 1 \\ y''_2 & y''_2 & 1 \\ \vdots & \vdots & \vdots \\ y''_k & y''_k & 1 \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 & 1 \\ y'_2 & y'_2 & 1 \\ \vdots & \vdots & \vdots \\ y'_k & y'_k & 1 \end{bmatrix} \cdot M \tag{10}$$

in which there are 6 unknown parameters in the transform matrix M. Since K is larger than 3, we propose to use the least square method for determining theses six parameters. With the transform matrix M, we can derive the zooming matrix M_z, rotation matrix M_r and translation matrix M_t easily. The same transform matrices are used for identifying text motion status as follows. The ratio of $\sqrt{{s_x}^2+{s_y}^2}$ is 1 or inverse when a text region is zoomed, or else we check another status, namely, if α in M_r is big enough, then the text is considered as having rotated with α degrees, and if $\sqrt{dx^2+dy^2}$ is large, the text is considered as having translated.

B. Text Tracking Through Motion Status Analysis

As discussed in the previous Section, the proposed technique determines motion status such as linear, rotation and zoom for each text line in a frame using KLT tracker and optical flow estimation. Here, the KLT tracker finds the correspondences in temporal frames. Then an optical flow is estimated between the correspondences. Based on the optical flow information, the

proposed method derives the motion status matrix as described in Section IV-A. As a result, the search domain is reduced in tracking text lines. For example, if the motion status is linear then the tracking algorithm searches in the linear direction rather than multiple directions in temporal frames. In this way, motion status determination helps in reducing the search time and the region for tracking text lines, which in turn helps in finding accurate locations of text lines. More specifically, the steps for tracking are as follows.

When we use the KLT tracker to calculate the corresponding point v_2 in the next frame f_2 for a point v_1 in the current frame f_1, we need to calculate an optical flow vector $\overline{v}$. Traditionally, $\overline{v}$ is initialized as zero: $\overline{v}^0 = [00]^T$. Next, an iterative scheme is applied to modify $\overline{v}^k$ to approximate the real optical flow by $\overline{v}^k = \overline{v}^{k-1} + \overline{n}^k$ until the computed pixel residual $\overline{n}^k$ is smaller than a certain threshold, or a maximum number of iterations is reached. The detail of how to calculate $\overline{n}^k$ can be found in [42]. Usually, around 5 iterations are enough to reach convergence for our experiments. Let $\overline{v}$ be equal to the convergence of $\overline{v}^k$. Thus, v_2 in frame f_2 is determined as $v_2 = v_1 + \overline{v}$. Once we have the motion status of a text line, the overall motion matrix M for all the corners in the text line is determined. Using this, we can predict the possible location of the tracked point with $\overline{v}_1 \cdot M$, in whch $\overline{v}_1$ is the extended vector of v_1. We use $\overline{v}_1 \cdot M$ to initialize $\overline{v}^0$ to reduce the number of iterative steps (about 2-3 steps) to reach convergence. In this way, motion status helps in reducing time for tracking text lines in video.

V. Experimental Results

The proposed technique consists of two major contributions, that is, text detection by spatial features and text tracking through motion status analysis on multi-oriented, multi-font, multi-sized texts in video. Further, the text detection part is divided into two sub-parts, namely, text detection in video and text detection in natural scene images. This is because when the proposed technique works for low resolution video, it should also work for high resolution scene images. We consider three video datasets for experimentation, namely, our own video dataset at the rate of 30 frames per second, which includes multi-oriented, multi-font, multi- sized graphics and scene texts, the video dataset from the recent benchmark database of ICDAR 2013 [43] which has mixed texts with less multi-orientations, and the video dataset from the standard database [7] YouTube (YVT) which is a collection of only scene texts along the horizontal direction but without graphics texts.

Similarly, we consider three benchmark scene image datasets from ICDAR family, namely, ICDAR 2013 scene text data, the Microsoft data [21] in which most of the texts are street view texts in the horizontal direction, and the MSRA-TD500 [44] data which contains horizontal and non-horizontal straight text lines. Since the resolutions vary greatly for both video and scene images, it adds more complexity to the problem. More details about the datasets are listed in Table I. In summary, 545 videos and 1269 scene text images are considered for evaluating the performance of the proposed technique. We believe that we have considered enough diversified data to test the generic property of the proposed technique. The second part, text tracking, is tested on all the three video datasets mentioned above for the purpose of evaluation.

TABLE I
Details of the Different Datasets Used for Experimentation

Dataset Name	Total number	Type	Resolution range
Our dataset	500	Video and Image	320 × 240 to 1280×720
ICDAR 2013 Video dataset	15	Video	720×480 to 1280×720
YVT	30	Video	Around 720×1280
ICDAR 2013 Scene dataset	462	Image	307×730 to 1296×960
Microsoft data	307	Image	1024×1360 to 1024×768
MSRA-TD500	500	Image	1296×864 to 1920×1280

We consider three standard measures, namely, recall, precision and F-measure along with the Average Processing Time (APT) for measuring the performance of the proposed technique. We follow the standard evaluation scheme as given in the ICDAR 2013 robust reading competition. Note that the measure proposed in [43] requires words segmentation because the ground truth is generated for words but not text lines. In our work, we detect text lines but not words because word segmentation for video is not easy for natural scene images due to low resolution and complex background. In order to use the standard measures, we combine the ground truths of words into text lines for calculating the measures. For all the experimentations in this work, we calculate the measures at text lines with the same measures. Specifically, the measures are defined formally as follows. A match m_p between two rectangles (detected and ground truth) is defined as the area of the intersection divided by the area of the minimum bounding box containing both rectangles. So the best match $m(r;R)$ for a rectangle r in a set of rectangles R is defined as

$$m\left(r;R\right) = max\{m_p\left(r;r'\right)|r' \in R\}. \tag{11}$$

This is defined to find the closest match in the set of ground truth targets for each rectangle in the set of estimates. Then Precision and Recall are defined as

$$Precision = \frac{\sum_{r_e \epsilon E_s} m\left(r_e;T_r\right)}{|E_s|}, Recall = \frac{\sum_{r_t \epsilon T_r} m\left(r_t;E_s\right)}{|T_r|} \tag{12}$$

where T_r and E_s are the sets of targets (ground truth) and estimated boxes, respectively. These two measures are combined to form a single measure f with a parameter α. We set it 0.5 to give precision and recall an equal weight

$$f = \frac{1}{\frac{\alpha}{Precision} + \frac{1-\alpha}{Recall}}. \tag{13}$$

In order to show the effectiveness of the proposed technique, we compare it with the state of the art methods, namely, Li *et al.*'s method [18] which uses moments and wavelet combination for both text detection and tracking in video, Zhao *et al.*'s method [36] which uses corner based features and optical flow for both text detection and caption text tracking in video, Mosleh

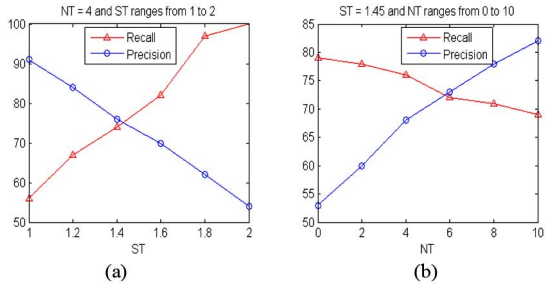


Fig. 15. Diagram of determining nt_1 and st_1. (a) Set nt_0 equals to 4 and range ST from 1 to 3, then we get $st_1 = 1.45$. (b) Set ST equals to st_1 and range NT from 0 to 10, then we get $nt_1 = 6$.

et al.'s method [38] which uses stroke width transform and inpainting for both text detection and text removal, Epshtein *et al.*'s method [21] which proposes stroke width transform for text detection from natural scene images, Yao *et al.*'s method [44] which uses an improved version of stroke width transform for non-horizontal text detection from natural scene images, and finally Yin *et al.*'s method [45] which uses Maximally Stable Extremal Regions (MSER) and single link clustering algorithms for non-horizontal text detection from natural scene images. Since the first three methods work for both text detection and text tracking, we compare these methods for all the experiments. On the other hand, since Epshtein *et al.*'s method [21] is for text detection from natural scene images but not video, we compare this method for text detection in video as well as natural scene images but not for text tracking experiments. The results of the last two methods are reported for MSRA-TD500 dataset at text line level, we compare these two methods on MSRA data but not all the other experiments since our technique requires text lines for experimentation.

The values for the thresholds NT and ST are determined empirically based on the experiments on the ICDAR 2013 training dataset as follows.

1. Initialize $nt_0 = 4$.
2. Fix NT as nt_{k-1} and range ST from 1 to 2, and then run our method on the ICDAR 2013 training dataset for each ST. Set ST equals to st_k where the Recall equals to Precision. Then fix ST as st_k and range NT from 0 to 10. Set NT equals to nt_k where Recall equals to Precision. Fig. 15 shows the illustration of finding st_1 and nt_1.
3. Repeat step 2 until $|nt_k - nt_{k-1}| < \Delta nt$ and $|st_k - st_{k-1}| < \Delta st$.

The illustration of determining nt_1 and st_1 is shown in Fig. 15, where the recall and precision curves intersect at some values. At last, the experimental study helps in setting the values as $NT = 6$ and $ST = 1.45$.

A. Evaluation of Text Detection in Video

As mentioned in the previous section, we consider three video datasets for experimentation, which are our own dataset, the ICDAR 2013 video dataset and the YVT video dataset. Since all the existing methods [18], [36], [38] detect texts in a key frame, the first frame, or individual scene images, we follow the same way for text detection in this work. For tracking, we use temporal frames after detecting texts in a key frame.

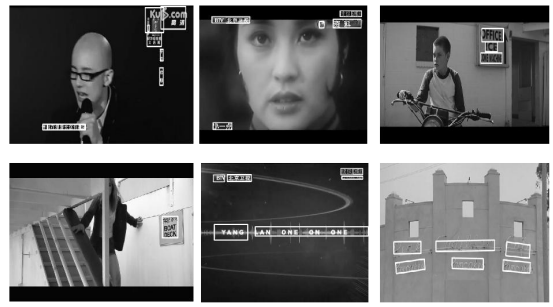


Fig. 16. Sample qualitative results of the proposed technique for our video data.

TABLE II
PERFORMANCE OF TEXT DETECTION ON OUR VIDEO DATA

Method	Precision	Recall	F-Measure	APT
Our Method	0.70	0.79	0.74	1.3
Mosleh et al. [38]	0.56	0.50	0.53	1.3
Epshtein et al. [21]	0.41	0.46	0.43	1.0
Li et al. [18]	0.10	0.83	0.18	0.8
Zhao et al. [36]	0.31	0.16	0.21	3.4

Experiments on Our Video Data: Since our dataset contains 500 videos which lasts for 1 or 2 seconds, we extract 500 key frames for text detection in this experiment. Our main objective is to achieve good results for both text detection and tracking, we consider the first frame as the key frame for text detection and the same text line would be traced in temporal frames. This helps us to speed up the text detection process by skipping duplicate frames. Therefore, 500 frames correspond to 500 videos for experimentation in this work. This dataset includes a wide range of texts that are multi-oriented, multi-sized, multi-fonts, and multi-scripts of both graphics and scene texts. Therefore, this dataset can be considered as a complex dataset for testing the capability of generalization of the proposed technique. Sample qualitative results are shown in Fig. 16, where one can see that the proposed technique detects almost all the texts with proper bounding boxes for different text lines including scene texts and graphics texts. Fig. 16 also shows that the proposed technique is good at detecting multi-script text lines. The quantitative results of the proposed and the existing methods are reported in Table II, where the proposed technique gives better recall and F-measure than the existing methods. Note that Li *et al.*'s method is good at recall and a short average processing time but is worse at precision compared to the proposed technique because their performance depends on the number of samples used for training and hence the method produces more false positives. It is faster than all the other existing methods including the proposed technique. Mosleh *et al.* and Zhao *et al.* are good for detecting caption texts but not for scene texts because their scope is caption text detection. Therefore, low precision, recall and F-measure are reported for both techniques compared to the proposed technique. On the other hand, the proposed technique reports good results in terms of precision, recall, F-measure and processing time because of its ability to handle both graphics and scene texts of different orientations.

Experiments on ICDAR 2013 Video Data: Since this dataset provides 15 videos lasting from around 10 seconds to 1 minute,

Fig. 17. Sample qualitative results of the proposed technique for ICDAR video data.

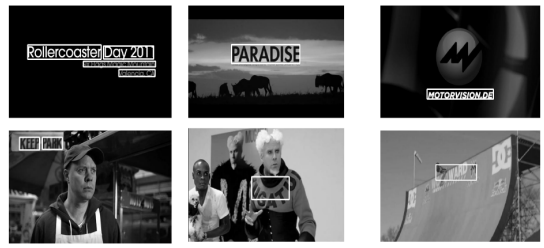


Fig. 18. Sample qualitative results of the proposed technique for YVT data.

TABLE III
PERFORMANCE OF TEXT DETECTION ON ICDAR 2013 VIDEO DATA

Method	Precision	Recall	F-Measure	APT
Our Method	0.63	0.68	0.65	1.3
Mosleh et al. [38]	0.50	0.49	0.49	1.2
Epshtein et al. [21]	0.48	0.47	0.47	0.9
Li et al. [18]	0.25	0.67	0.36	0.8
Zhao et al. [36]	0.23	0.32	0.27	3.4

TABLE IV
PERFORMANCE OF TEXT DETECTION ON YVT VIDEO DATA

Method	Precision	Recall	F-Measure	APT
Our Method	0.81	0.73	0.77	0.9
Mosleh et al. [38]	0.72	0.79	0.75	0.9
Epshtein et al. [21]	0.68	0.76	0.72	0.7
Li et al. [18]	0.32	0.57	0.41	0.7
Zhao et al. [36]	0.34	0.41	0.37	2.6

we extract 50 key frames from the video sequences for experimentation. This dataset represents a wide range of real life situations using different types of cameras. In addition, this dataset includes texts in various languages such as Spanish, French and English. Most of the video captures indoor and outdoor scenes to represent real life applications. Therefore, this dataset does not contain graphics texts. Since the ground truth is available at the word level for texts in frames, we use the same ground truth for calculating the measures by combining the ground truths of words into text lines. The same evaluation scheme as mentioned in [43] is used for evaluation. Sample qualitative results of the proposed technique are shown in Fig. 17, where we can see that the proposed technique detects well for complex background texts except for the last frame. This shows that the proposed technique is capable of handling texts embedded in indoor and outdoor frames. The quantitative results of the proposed and the existing methods are reported in Table III, where it is noticed that the proposed technique gives better results than all the existing methods in terms of recall, precision and F-measure. However, Li *et al.*'s method is the best in the average processing time among all the methods. Table III shows poor precision, recall and F-measure for the existing methods compared to the proposed technique because the existing methods are developed for graphics and caption text detection in video but not scene text in video. At the same time, this dataset contains a majority of scene texts with complex background and low resolution of different script types.

Experiments on YVT Scene Video Data: This is one more benchmark database of size 30 videos collected from YouTube available publicly. A special feature of this database is that this video dataset contains only scene texts with lots of greenery and buildings. Therefore, this is also a complex dataset for detecting texts. Each video has 15 seconds in duration. From this video dataset, we extract 35 key frames for text detection in this experiment. Sample qualitative results of the proposed technique are shown in Fig. 18, where the proposed technique detects well for almost all the frames except the last frame. Similarly, the quantitative results of the proposed and the existing methods are reported in Table IV, where the proposed technique gives better results than the existing methods in terms of recall, precision and F-measure but APT is the best for Li *et al.*'s technique. The main reason for poor precision, recall and F-measure for the existing methods is that this dataset does not contain any graphics and caption text. Though Li *et al.*'s and Epshtein *et al.*'s techniques work well for scene texts, they report poor precision, recall and F-measure compared to the proposed technique because Li *et al.*'s technique requires training samples and Epshtein *et al.*'s technique depends on Canny edge image of each input image. On the other hand, though the proposed technique uses Canny edge image, it selects potential text candidates successfully at different scales by eliminating unwanted edge information with the help of symmetry features. Therefore, the proposed technique outperforms the existing techniques.

In summary, when we look at the results of the proposed technique reported in Table II–Table IV for the our video, the ICDAR 2013 video and YVT datasets, the proposed technique gives better results for the YVT data but poorer results for ICDAR 2013 video data compared to that for our data. Since YVT does not contain mixed texts of caption and scene, and most of the texts are in the horizontal direction, the proposed technique gives better results. In case of the ICDAR 2013 video data, they are small texts with mixture of different resolutions captured by various devices such as mobile cameras, a Head mounted camera and a hand held camcorder, etc. In addition, this dataset contains texts of different languages while our data consists of multi-oriented captions and scene texts.

B. Evaluation of Text Detection in Natural Scene Images

The experimental results provided in Section V-A shows that the proposed technique works well for video of low resolution and complex background. The same technique is expected to work well for the texts in natural scene images captured by a high resolution camera.

Since the proposed technique and the existing techniques use single frame/image for text detection, we use the same way to detect texts in natural scene images. We use the ground truth at text line level for calculating recall, precision and F-measures

Fig. 19. Sample qualitative results of the proposed technique for ICDAR 2013 scene text data unsuccessful results.

TABLE V
PERFORMANCE OF TEXT DETECTION ON ICDAR 2013 SCENE TEXT DATA

Method	Precision	Recall	F-Measure	APT
Our Method	0.76	0.70	0.73	1.3
Mosleh et al. [38]	0.76	0.66	0.71	1.2
Epshtein et al. [21]	0.73	0.60	0.66	0.9
Li et al. [18]	0.21	0.61	0.31	0.8
Zhao et al. [36]	0.18	0.20	0.19	3.4

by combining the ground truths of words into text lines. There are existing techniques which report results at the word level using the ground truth. Since our method focuses on text line detection in video as well scene images, the implemented existing techniques are tested at text line level for a fair comparative study. The techniques which report results at the word level are not considered for comparative study in this work.

Experiments on ICDAR 2013 Scene Text Data: Sample qualitative results of the proposed technique are shown in Fig. 19, where it is found that the proposed technique works well for scene text detection. The results of the proposed and the existing techniques are reported in Table V, which shows that the proposed technique gives the best recall and F-measure among all the techniques under study. Since the technique proposed by Epshtein *et al.* is developed for scene text detection, it gives slightly lower precision, recall and F-measure compared to the proposed technique. In the same way, Zhao *et al.*'s technique gives the lowest precision, recall and F-measure as their technique is designed for caption texts in video but not scene texts in scene images. The same is true for Li *et al.*'s technique. However, Mosleh *et al.*'s technique gives the best at precision, recall and F-measure among the existing techniques (other than the proposed technique) because their technique considers both caption and scene texts in video.

Experiments on Microsoft Data: This dataset is much harder than the ICDAR 2013 scene dataset due to the presence of vegetation and repeating patterns such as windows, which are virtually undisguisable from texts without OCR. In addition, the text fonts in it are too small in most images. Achieving a good accuracy for this method is challenging. Sample results of the proposed technique are shown in Fig. 20, where the proposed technique detects texts correctly for a few images and incorrectly for some others. The quantitative results of the proposed and

Fig. 20. Sample qualitative results of the proposed technique for Microsoft data.

TABLE VI
PERFORMANCE OF TEXT DETECTION ON MICROSOFT SCENE TEXT DATA

Method	Precision	Recall	F-Measure	APT
Our Method	0.38	0.71	0.50	1.2
Mosleh et al. [38]	0.48	0.51	0.49	1.1
Epshtein et al. [21]	0.51	0.50	0.51	1.0
Li et al. [18]	0.23	0.35	0.28	0.7
Zhao et al. [36]	0.33	0.76	0.46	3.1

the existing techniques are reported in Table VI, which shows that the proposed technique is high at recall but low at precision compared to the existing techniques due to the presence of objects in the background which look like texts. As a result, false positives are high and hence precisions are low. However, overall, F-measure is high for the proposed technique compared to the existing techniques. Epshtein *et al.*'s method also reports low recall and F-measure due to the same problem. Note that the values of the measures of Epshtein *et al.* reported in their paper is a little different from the values of the measures reported in Table VI because in this experiment, we calculate the measures at text line level while the results reported in their paper is at the word level. The method proposed by Mosleh *et al.* is nearly close to Epshtein *et al.*'s technique because Mosleh *et al.*'s technique is an improved version of that proposed by Epshtein *et al.* In this study, Li *et al.*'s and Zhao *et al.*'s methods are much worse among the techniques.

Experiments on MSRA Data: Since our aim is to address multi-oriented text detection in video as well as scene images, we test our technique on another available MSRA dataset because it consists of multi-oriented text images. This dataset looks like the ICDAR 2013 dataset but not harder than the Microsoft dataset. Sample qualitative results of the proposed technique are shown in Fig. 21, where it is found that the proposed technique detects texts of different orientations well. The quantitative results of the proposed and the existing techniques are reported in Table VII, where the proposed technique gives a better at precision and F-measure than the existing techniques because all the existing techniques except Yin *et al.* and Yao *et al.* focus on horizontal text detection in scene images but not multi-oriented text detection. Note that we compare our technique with Yin *et al.* and Yao *et al.* only for this dataset because these techniques report results at text line level as our technique. Table VII shows though the precision of the proposed technique is almost the same as Yin *et al.* and Yao *et al.*, overall, the performance of the proposed technique is better than those two techniques especially in terms of recall.

Fig. 21. Sample qualitative results of the proposed technique for MSRA-TD500 multi-oriented scene text data.

TABLE VII
PERFORMANCE OF TEXT DETECTION ON MSRA-TD500 MULTI-ORIENTED SCENE TEXT DATA

Method	Precision	Recall	F-Measure	APT
Our Method	0.63	0.70	0.66	1.2
Mosleh et al. [38]	0.56	0.53	0.55	1.2
Epshtein et al. [21]	0.52	0.50	0.51	1.0
Li et al. [18]	0.26	0.65	0.37	0.8
Zhao et al. [36]	0.34	0.69	0.46	3.3
Yin et al. [45]	0.61	0.71	0.66	0.8
Yao et al. [44]	0.63	0.63	0.60	7.2

Fig. 22. Sample qualitative results of the proposed technique for Blur video and scene text data.

This is all because of the spatial and the symmetry features for identifying potential text candidates at multi-levels.

Experiments on Blurred Video Frames: It is known that while capturing video, there are chances of introducing blurring effects due to object movements or camera movements. Therefore, in this work, we test the proposed technique on 50 blurred frames chosen from ICDAR 2013 video and scene data [43] to study the effectiveness of the proposed technique. The sample results of the proposed technique are shown in Fig. 22, where the proposed technique detects text lines well for the first four images. However, for the last two images, the proposed technique does not detect text lines completely because of the blurring effects. This is because blurring affects Canny edge, gradient calculation and corner detection steps. The results of the proposed and the existing techniques are reported in Table VIII, where we can notice that the proposed technique is better than the existing techniques in terms of precision, F-measure and the average processing time. However, Mosleh *et al.*'s method is the best at recall compared to the others including the proposed technique. The poor result of the existing techniques is mainly because the techniques were developed for good images but not blurred images. When we compare the results of blurred data and other data, the proposed technique gives slightly lower results in terms of F-measure for the blurred data. Therefore, there is a scope for improvement to withstand blurring without sacrificing accuracy.

TABLE VIII
PERFORMANCE OF TEXT DETECTION ON BLURRED DATA

Method	Precision	Recall	F-Measure	APT
Our Method	0.63	0.28	0.39	1.3
Mosleh et al. [38]	0.54	0.30	0.39	1.3
Epshtein et al. [21]	0.53	0.29	0.37	1.0
Li et al. [18]	0.10	0.23	0.14	0.8
Zhao et al. [36]	0.13	0.22	0.16	3.4

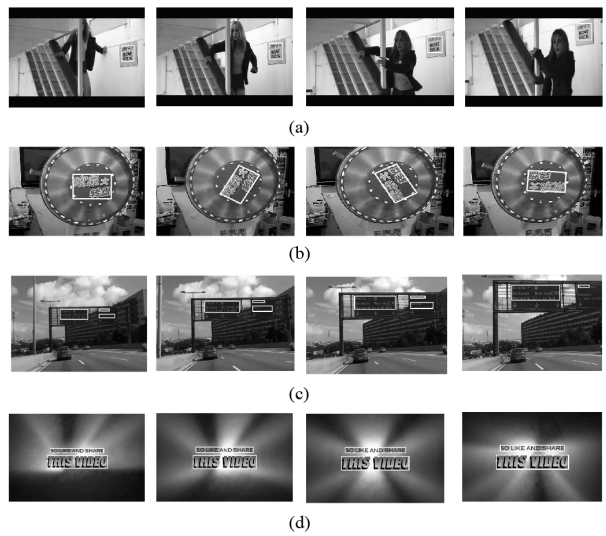


Fig. 23. Sample qualitative results of the proposed technique for txt tacking in video with different motion status. (a) Sample text tracking results of the proposed technique for our video data with linear translation motion status. (b) Sample text tracking results of the proposed technique for our video data with rotation motion status. (c) Sample text tacking results of the proposed technique for ICDAR video data with zooming out and linear translation motion status. (d) Sample text tacking results of the proposed technique for YVT data with zooming out motion status.

C. Evaluation of Text Tracking in Video

We consider three video datasets mentioned in Section V.A for evaluating the proposed tracking step. First, the proposed technique uses text detection step for detecting texts in the first frame and then tracks the same texts in temporal frames using the KLT tracker. Three existing techniques, namely, Mosleh *et al.*, Li *et al.* and Zhao *et al.* are used for comparative studies because these techniques involve text tracking in video. Besides, we also compare our tracking results with the tracking results reported in the ICDAR 2013 competition [43]. In this work, we consider tracking texts of different motion status, that is, video texts with linear motion, zoom and rotation (non-linear motion). This is different from the existing techniques since they generally focus on tracking texts with linear movements. Sample results for text tracking of the proposed technique are shown in Fig. 23, where we can see in (a)-(d), the proposed technique tracks video texts successfully for the different motion status chosen from the three video datasets.

TABLE IX
PERFORMANCE OF THE PROPOSED AND EXISTING TECHNIQUES FOR TRACKING TEXT IN VIDEO WITH RESPECT TO MOTION STATUS (P: PRECISION, R: RECALL, AND F: F-MEASURE)

Data Set	Our method			Mosleh et al.			Zhao et al.			Li et al.		
	P	R	F	P	R	F	P	R	F	P	R	F
Linear	98	96	97	78	92	84	95	97	96	90	91	90
Zooming	93	92	92	34	40	37	91	93	92	58	80	67
Rotation	90	91	90	13	23	17	91	87	89	82	78	80

TABLE X
PERFORMANCE OF THE PROPOSED AND EXISTING TECHNIQUES FOR TRACKING TEXT IN VIDEO

Data Set	Our method			Mosleh et al.			Zhao et al.			Li et al.		
	P	R	F	P	R	F	P	R	F	P	R	F
Our data set	98	91	94	51	90	65	93	91	92	84	83	83
ICDAR	89	91	90	27	36	31	81	89	85	65	78	71
YVT	96	96	96	45	96	61	92	95	93	81	90	85
Average	94	93	93	41	74	53	89	92	90	77	84	80
Average FPS	20			27			22			14		

Before successfully tracking texts, the proposed technique finds the motion status of each text, namely, static, linear, zoom and rotation using the derived transformation matrices as presented in Section IV. With the help of motion status identification, the technique achieves good results for text tracking because motion status identification helps in modifying the parameters for the text detection step and the tracker. The quantitative results of the proposed and the existing techniques are reported in Table IX, where one can see that the proposed technique outperforms the existing techniques. Table IX shows that the proposed and the existing techniques give good results in terms of precision, recall and F-measure for the texts of linear motion but lower precision, recall and F-measure for the texts with zoom in, out and rotation because tracking a text when it is moving with rotation, zoom in and out changes the shape of the text. This difficulty leads to poor performances for the existing techniques as they are not meant for handling such motions.

The final performance of the proposed and the existing techniques for tracking in the three video datasets is reported in Table X, where it can be seen that the proposed technique outperforms the existing methods for tracking. Zhao *et al.* use the KLT tracker as in our technique but their technique is more sensitive to corners due to background influence compared to our technique. The tracking performance of Li *et al.*'s technique depends on text detection results but does not consider text movements in complex background. Mosleh *et al.*'s technique uses Camshift for text tracking. This is good for tracking texts with linear motion but not texts with different rotations. In Table X, FPS denotes frames per second used for tracking texts. In summary, since the proposed technique tracks video texts according to motion status identification while the existing techniques use some tools for tracking which are sensitive to text movements, the proposed technique is better than the existing techniques and works for different situations in the sense that the proposed technique is independent of language, transformation, font and font size variations.

TABLE XI
PERFORMANCE OF THE PROPOSED AND EXISTING TECHNIQUES ON TRACKING DATA OF ICDAR 2013 VIDEO

Method	MOTP	MOTA	ATA
Our Method	0.61	0.46	0.29
Mosleh et al. [38]	0.45	0.13	0.03
Li et al. [18]	0.21	0.15	0. 07
Zhao et al. [36]	0.24	0.11	0.05
TextSpotter [43]	0.67	0.27	0.12
Baseline Algorithm [43]	0.63	-0.09	0

We also compare the text tracking results of the proposed technique with the tracking results given by the baseline algorithm (ABBAY OCR SDK) and TextSpotter as reported in [43] with the same measures to study the effectiveness of the proposed technique. In this experiment, we use the performance measures, such as MOTP which is the Mean Tracking Precision, MOTA which is the Mean Tracking Accuracy, and ATA which is the Average Tracking Accuracy to evaluate the tracking results given by the techniques. More details about the formulas and the definitions for these three performance measures can be found in [43]. Table XI shows that the proposed techniques is good at MOTA compared to TextSpotter, while TextSpotter is good at MOTP. However, the proposed technique is better than the baseline algorithms in terms of MOTA and ATA. The main cause for the poor MOTP for the proposed technique compared to the existing techniques is that the proposed technique considers the first frame as the key frame for text detection and the same text line is to be tracked in other frames. However, this experiment does not consider key frame selection for text detection, but rather it detects the lines as text appears in the whole video. Since our goal is to achieve good results for both text detection and tracking, we prefer to use key frame for text detection and then tracking the same text line, but TextSpotter is developed solely for text tracking. As a result, this experiment creates discrepancy between the text lines detected by the proposed technique and its ground truth. Therefore, there is a scope for testing our method on live video in real time environment without losing accuracy.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a novel technique for detecting and tracking texts of any orientation in video. The technique explores gradient directional information at component level and spatio-temporal information for text candidate selection. Delaunay triangulation is proposed to study the spatial relationship between corners in a different way. The characteristics of video texts are proposed to eliminate false candidates. Then grouping is proposed for extracting text regions of any orientation based on the nearest neighbor criterion and text direction. To tackle video texts of multi-fonts or multi-sizes, we further propose multi-scale integration for full text line detection. Then the detected texts are tracked in video by matching sub-graphs of triangulation. Experimental results on our video dataset, the benchmark video datasets and the natural scene image datasets show that the proposed technique is superior to the state of the art techniques in terms of recall, precision and F-measure. Since the reported accuracy is a bit low compared to the accuracy of document analysis, we are planning to further investigate

the proposed technique to improve the accuracy. In addition, our next work would be text recognition by considering text detection results as inputs.

REFERENCES

[1] C. Xu, Y. F. Zhang, G. Zhu, and Y. Rui, "Using webcast text for semantic event detection in broadcast sports video," *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1342–1355, Nov. 2008.

[2] J. Mao, H. Li, W. Zhou, S. Yan, and Q. Tian, "Scale based region growing for scene text detection," in *Proc. ACM MM*, 2013, pp. 1007–1016.

[3] X. C. Yin, K. Huang, and H. W. Hao, "Accurate and robust text detection: A step-in for text retrieval in natural scene images," in *Proc. SIGIR*, 2013, pp. 1091–1092.

[4] K. L. Bouman, G. Abdollahaian, M. Boutic, and E. J. Delp, "A low complexity sign detection and text localization method for mobile applications," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 922–934, Oct. 2011.

[5] T. Yusufu, Y. Wang, and Z. Fang, "A video text detection and tracking system," in *Proc. ISMM*, 2013, pp. 533–529.

[6] X. Liu and W. Wang, "Robustly extracting captions in videos based on stroke-line edges and spatio-temporal analysis," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 482–489, Apr. 2012.

[7] P. X. Nguyen, K. Wang, and S. Belongie, "Video text detection and recognition: Dataset and benchmark," in *Proc. WACV*, 2014, pp. 776–783.

[8] Y. Li, W. Jia, C. Shen, and A. V. D. Hengel, "Characterness: An indicator of text in the wild," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1666–1677, Apr. 2014.

[9] J. Zhang and R. Kasturi, "A novel text detection system based on character and link energies," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4187–4198, Sep. 2014.

[10] T. Q. Phan, P. Shivakumara, and C. L. Tan, "Detecting text in the real world," in *Proc. ACM MM*, 2012, pp. 765–768.

[11] D. Crandall, S. Antani, and R. Kasturi, "Extraction of special effects caption text events from digital video," *Int, H, Document Anal. Recognit.*, pp. 138–157, 2003.

[12] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," *Pattern Recognit.*, pp. 977–997, 2004.

[13] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," *Int, H, Document Anal. Recognit.*, pp. 84–104, 2005.

[14] J. Zang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in *Proc. DAS*, 2008, pp. 5–17.

[15] G. Miao, Q. Huang, S. Jiang, and W. Gao, "Coarse-to-fine video text detection," in *Proc. ICME*, 2008, pp. 569–572.

[16] P. Shivakumara, T. Q. Phan, and C. L. Tan, "Video text detection based on filtters and edge features," in *Proc. ICME*, 2009, pp. 514–517.

[17] Y. Liu, Y. Song, Y. Zhang, and Q. Meng, "A novel multi-oriented chinese text extraction approach from videos," in *Proc. ICDAR*, 2013, pp. 1387–1391.

[18] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 147–156, Jan. 2000.

[19] L. Wu, P. Shivakumara, T. Lu, and C. L. Tan, "Text detection using delaunay triangulation in video sequence," in *Proc. DAS*, 2014, pp. 41–45.

[20] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "Snooper-Text: A text detection system for automatic indexing of urban scenes," *Comput. Vis. Image Understanding*, vol. 122, pp. 92–104, 2013.

[21] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. CVPR*, 2010, pp. 2963–2970.

[22] Y. F. Pan, X. Hou, and C. L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 800–813, Mar. 2011.

[23] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," in *Proc. ICCV*, 2013, pp. 569–576.

[24] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video," *Pattern Recognit.*, pp. 977–997, 2004.

[25] D. Chen and J. M. Odobez, "Video text recognition using sequential monte carlo and error voting methods," *Pattern Recognit. Lett.*, pp. 1386–1403, 2005.

[26] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A Laplacian approach to multi-oriented text detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 412–419, Feb. 2011.

[27] P. Shivakumara, A. Dutta, C. L. Tan, and U. Pal, *Multi-Oriented Scene Text Detection in Video based on Wavelet and Angle Projection Boundary Growing*. Berlin, Germany: Springer-Verlag, 2013.

[28] M. R. Lyu, J. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization and extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 243–255, Feb. 2005.

[29] P. Shivakumara, T. Q. Phan, S. Lu, and C. L. Tan, "Gradient vector flow and grouping based method for arbitrarily-oriented scene text detection in video images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1729–1739, Oct. 2013.

[30] W. Huang, P. Shivakumara, and C. L. Tan, "Detecting moving text in video using temporal information," in *Proc. ICPR*, 2008.

[31] J. Zhou, "A robust system for text extraction in video," in *Proc. ICMV*, 2007, pp. 119–124.

[32] C. Mi, Y. Xu, H. Lu, and X. Xue, "A novel video text extraction approach based on multiple frames," in *Proc. ICICSP*, 2005, pp. 678–682.

[33] Y. K. Wang and J. M. Chen, "Detection video texts using spatial-temporal wavelet transform," in *Proc. ICPR*, 2006, pp. 754–757.

[34] X. Huang, "A novel approach to detecting scene text in video," in *Proc. CISP*, 2011, pp. 469–473.

[35] X. Huang, H. Ma, and H. Yuan, "A novel video text detection and localization Approach," in *Proc. PCM*, 2008, pp. 525–534.

[36] X. Zhao, K. H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: A novel approach to detect text and caption in videos," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 790–799, Mar. 2011.

[37] L. Li, J. Li, Y. Song, and L. Wang, "A multiple frame integration and mathematical morphology based technique for video text extraction," *Proc. ICCIA*, pp. 434–437, 2010.

[38] A. Mosleh, N. Bouguila, and A. B. Hamza, "Automatic inpainting scheme for video text detection and removal," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4460–4472, Nov. 2013.

[39] P. Shivakumara, M. Lubani, K. S. Wong, and T. Lu, "Optical flow based dynamic curved video text detection," in *Proc. ICIP*, 2014, pp. 1668–1672.

[40] D. Sugimura, K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait," in *Proc. ICCV*, 2009, pp. 1467–1474.

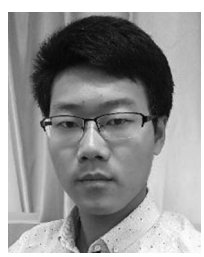
[41] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2014.

[42] J. Y. Bouguet, *Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the Algorithm*. Santa Clara, CA, USA: Intel Corp., 2001, vol. 1, pp. 1–9.

[43] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Boorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De las Heras, "ICDAR 2013robust reading competition," in *Proc. ICDAR*, 2013, pp. 1115–1124.

[44] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting text of arbitrary orientations in natural scene images," in *Proc. CVPR*, 2012, pp. 1083–1090.

[45] X. C. Yin, Z. Yin, K. Huang, and H. W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.



Liang Wu is currently working toward the M.S. degree in computer science and technology at Nanjing University, Nanjing, China.

His current research interests include media data analysis, computer vision, and pattern recognition algorithms.

Palaiahnakote Shivakumara received the B.Sc., M.Sc., M.Sc Technology, and Ph.D. degrees in computer science from the University of Mysore, Mysore, Karnataka, India, in 1995, 1999, 2001, and 2005, respectively.

He is currently a Visiting Senior Lecturer with the Department of Computer Systems and Information Technology, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia. From 1999 to 2005, he was a Project Associate with the Department of Studies in Computer Science, University of Mysore. He was a Research Fellow in the field of image processing and multimedia with the School of Computing, National University of Singapore, Singapore, from 2005 to 2007. He was also a Research Consultant with Nanyang Technological University, Singapore, for a period of six months in 2007. He was a Research Fellow with the National University of Singapore from 2008 to 2013. He has authored or coauthored more than 130 research papers in national and international conferences and journals. His research interests are in the area of image processing, pattern recognition, including text extraction from video and document image processing.

Dr. Shivakumara has been a reviewer for several conferences and journals.

Tong Lu (M'15) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Nanjing University, Nanjing, China, in 1993, 2002, and 2005, respectively.

He served as an Assistant Professor and an Associate Professor with the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2005 and 2007, respectively, where he is currently a Full Professor. He also has served as a Visiting Scholar with the National University of Singapore, Singapore, and the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. He is also a member of the National Key Laboratory of Novel Software Technology in China. He has authored or coauthored over 60 papers and two books in his area of interest, and holds more than 20 international or Chinese invention patents. His current research interests include multimedia, computer vision, and pattern recognition algorithms/systems.

Chew Lim Tan (M'03–SM'03) received the B.Sc. (Hons.) degree in physics from the University of Singapore, Singapore, in 1971, the M.Sc. degree in radiation studies from the University of Surrey, Surrey, U.K., in 1973, and the Ph.D. degree in computer science from the University of Virginia, Charlottesville, VA, USA, in 1986.

He is currently a Professor with the Department of Computer Science, School of Computing, National University of Singapore, Singapore. He has authored or coauthored more than 360 research publications. His current research interests include document image analysis, text and natural language processing, neural networks, and genetic programming.

Dr. Tan is an Associate Editor of *Pattern Recognition* and the *ACM Transactions on Asian Language Information Processing*, and is a member of the editorial board of the *International Journal on Document Analysis and Recognition*. He is a member of the Governing Board of the International Association of Pattern Recognition.