A New Method for Arbitrarily-Oriented Text Detection in Video

[1]Nabin Sharma, [2]Palaiahnakote Shivakumara, [3]Umapada Pal, [1]Michael Blumenstein and [2]Chew Lim Tan
[1]Griffith University, Queensland, Australia, Email: {m.blumenstein, nabin.sharma}@griffith.edu.au
[2]School of Computing, National University of Singapore, Singapore, Email: {shiva, tancl}@comp.nus.edu.sg
[3]Computer Vision and Pattern Recognition Unit, Indian Statistical Unit, Kolkata, India. Email: umapada@isical.ac.in

***Abstract*—Text detection in video frames plays a vital role in enhancing the performance of information extraction systems because the text in video frames helps in indexing and retrieving video efficiently and accurately. This paper presents a new method for arbitrarily-oriented text detection in video, based on dominant text pixel selection, text representatives and region growing. The method uses gradient pixel direction and magnitude corresponding to Sobel edge pixels of the input frame to obtain dominant text pixels. Edge components in the Sobel edge map corresponding to dominant text pixels are then extracted and we call them text representatives. We eliminate broken segments of each text representatives to get candidate text representatives. Then the perimeter of candidate text representatives grows along the text direction in the Sobel edge map to group the neighboring text components which we call word patches. The word patches are used for finding the direction of text lines and then the word patches are expanded in the same direction in the Sobel edge map to group the neighboring word patches and to restore missing text information. This results in extraction of arbitrarily-oriented text from the video frame. To evaluate the method, we considered arbitrarily-oriented data, non-horizontal data, horizontal data, Hua's data and ICDAR-2003 competition data (Camera images). The experimental results show that the proposed method outperforms the existing method in terms of recall and f-measure.**

Keywords- Video text frame, Gradient direction, Dominant text pixels, Video text representative, Angular region growing, Arbitrarily-oriented text detection.

I. INTRODUCTION

Text detection and extraction from video is an emerging area for research in the field of image processing and multimedia as it is useful in bridging a gap between low level feature and high level features to retrieve video events based on semantic with the help of Optical Character Recognition (OCR). Besides, scene text detection in video is challenging because of low resolution, complex background, different fonts, font size, orientation and color bleeding [1, 2]. Arbitrary orientation of text in video makes the problem even more complex and challenging.

There are several methods for natural scene text detection in camera based images in document analysis. It is seen that [3, 4] these methods required high resolution and clear shape of the character to identify the regular pattern of text for text detection in natural scenes. For instance, Epshtein et al. [3] have proposed text detection in natural scenes based on stroke width transform. The stroke width transform works well if there are no disconnections in the character components. Pan et al. [4] also proposed a hybrid approach for text detection in natural scene in images based on conditional random field. The conditional random field involves connected component analysis to label the text candidates. We can also see method on multi-oriented text extraction from camera images in [5] but this method works well if text with clear character shape is present in the images. These constraints are true for high resolution like scanned and camera images but not necessarily true for video based images due to undesirable properties of video. Thus, document analysis based methods used for text extraction from camera images and natural scene images may not be suitable without modifications for scene text detection or extraction from video frames.

Generally, video contains two types of text that are scene text and graphics text. Scene text is captured by the camera. Examples of scene text include street signs, billboards, and text on trucks and writing on shirts. Graphics text is manually added to video frames to supplement the visual and audio content. Since it is manually added, detection of such text is easier than scene text. In case of sports domains, scene text helps in retrieving sports events and it is useful in many applications such as navigation, surveillance, video classification, or analysis of sports events [1].

The major categories of text detection method are (a) connected component-based [6, 7] (b), texture-based methods [8, 9], and edge and gradient based methods [10, 11]. Since connected component based methods expect character shape, the methods may not be suitable for scene text detection in video with complex background. While texture based method are better than connected component as they work well for complex background of video. However, there is a problem in defining texture property for scene text detection as background may give defined texture property and they are sensitive to fonts and font size. On the other hand, the combination of edge and gradient feature based method are good for text detection in terms of efficiency and some extent to complex background. However, these methods suffer from setting threshold values at several stages of the algorithms.

Based on the above discussion, we can conclude that arbitrarily-oriented text detection in video frames is not addressed fully. Multi-oriented text has only been partially addressed in [12, 13] where the algorithm is limited to caption text and a few selected directions. Recently, Shivakumara et al. [14] have addressed this multi-oriented issue which is based on Laplacian and skeletonization methods. However, the goal of this method is limited to multi-oriented text detection but not arbitrarily-oriented text detection video. That method works well for a video frame with different oriented text but not for a frame with curve text.

Therefore, in this paper, we propose the method for dominant text pixel selection based on gradient pixel direction and magnitude checking. For each dominant text pixels, we obtain text representatives from the Sobel edge map of the input video frame. To tackle the problem of arbitrary orientation of the text, we introduce region growing to find text direction and then grouping to extract text lines.

II. PROPOSED METHOD

The proposed method consists of five subsections. In subsection A, we introduce the combination of gradient pixel direction and pixel magnitude features to identify the dominant text pixels from the Sobel edge map of the input video frame. By mapping dominant text pixel to the Sobel edge map of the input frame, text representatives are obtained. We eliminate the broken segments of text representatives based on connected component analysis to obtain candidate text representatives in subsection B. To find

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.6

direction of arbitrary text, we propose two kinds of region growing, region growing-1 to get word patches (discussed in subsection C), and region growing-2 for grouping word patches based on direction of the word patches to extract the whole text line from the video frame (discussed in subsection D). Sometimes the region growing of arbitrary direction combines two text lines as one line due to less space between text lines. To overcome this problem, we introduce a new idea of classification of frames containing horizontal text and frame containing arbitrarily-oriented text in subsection E. However the classification algorithm is good for the frame having horizontal and non-horizontal text separately. If the frame contains both horizontal and non-horizontal text then the classification method fails sometimes to classify correctly.

A. Gradient Pixel Direction and Magnitude for Dominant Text Pixels

For each edge pixel in the Sobel edge map of the input frame, the Gradient directional and Gradient magnitude features are computed as follows.

Let G_x and G_y be the Sobel masks in 'x' and 'y' directions respectively, given by equation 1 and 2. The strength of gradient $f(x,y)$ and direction of gradient $\theta(x,y)$ are defined as,

$$G_x = (p_7 + 2p_8 + p_9) - (p_1 + 2p_2 + p_3) \quad (1)$$

$$G_y = (p_3 + 2p_6 + p_9) - (p_1 + 2p_4 + p_7) \quad (2)$$

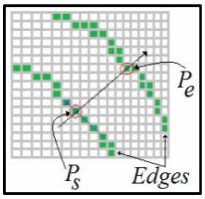
Where $p_1, p_2, \ldots, p_9$ are pixels in the 8-connected neighborhood of p_5 , as shown in Figure 1(a).

$$f(x,y) = [\, G_x^2 + G_y^2 \,]^{1/2} \quad (3)$$

$$\theta(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (4)$$

p_1	p_2	p_3
p_4	p_5	p_6
p_7	p_8	p_9

(a)



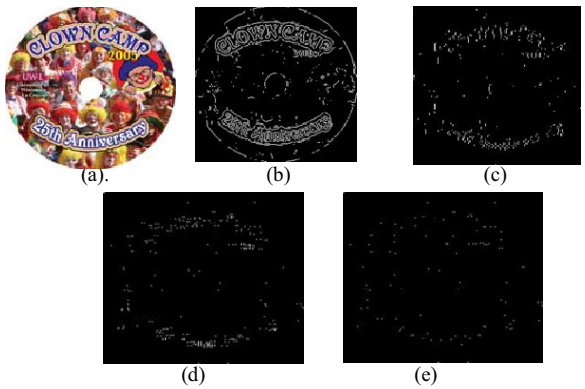
(b)

Figure 1: (a) 8-connected pixel neighborhood; (b) Example of candidate dominant computation. Traversing in the direction perpendicular to gradient direction of P_s leads to P_e

From the gradient directional features, the method identifies the dominant text pixels. Our approach is motivated from the fact that there exists a parallel edge for every edge in a character [3, 15]. The following technique is used to identify the dominant text edge pixels or 'Dominant points' (DP).

1. For an edge pixel $P_s(x,y)$ consider the gradient direction $\theta(P_s)$. Traverse towards the direction roughly perpendicular to $\theta(P_s)$ until an edge pixel $P_e(x,y)$ is found.
2. Consider the gradient direction $\theta(P_e)$ at P_e . Now from P_e traverse towards the direction perpendicular to $\theta(P_e)$, and check whether P_s is reached. If P_s can be traversed back from P_e, then $\{P_s , P_e\}$ are considered as DPs. An example of DP selection is shown in Figure 1(b).
3. If $f(P_s) - f(P_e) = 0 \; or < T$, then $\{P_s , P_e\}$ are considered as candidate DPs. The threshold T is considered as nearer to zero as possible.

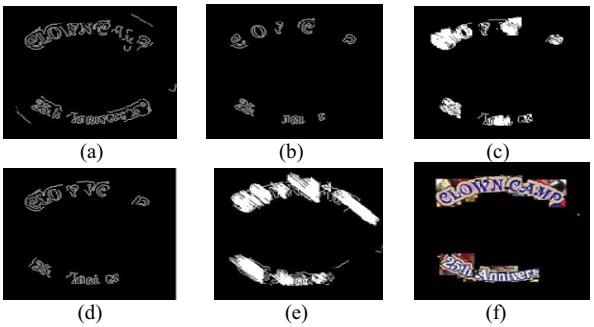
We remove redundant candidate DPs and retain only one candidate DP per edge component. To remove the redundant DPs, one candidate DP is chosen randomly from a list of candidate DPs belonging to the same edge component. The dominant point selection is illustrated in Figure 2 where (a) shows input frame, (b) shows Sobel edges of frame in (a), (c) shows the result of dominant point selection by checking condition in step 2, (d) shows the result of candidate dominant point selection by checking the condition in step 3 and (e) shows final DPs after removing redundant candidate DPs.



(a). (b) (c)

(d) (e)

Figure 2: (a) Input frame, (b) Sobel edge map for (a), (c) dominant points, (d) candidate dominant point, and (e) final dominant point per edge component.



(a) (b) (c)

(d) (e) (f)

Figure 3: (a) Text representatives; (b) Candidate text representatives after false text representative elimination; (c) Region growing of candidate text representatives; (d) Text edge found after region growing-1; (e) Region growing -2 for grouping; (f) Extracted edge components

B. Candidate Text Representatives

The method extracts edge components corresponding to final dominant text pixels in the Sobel edge map of the input frame, which we call as text representatives (TB_p) as shown in Figure 3(a) for the frame shown in Figure 2(a). This operation not only extracts true text representatives but also false-text representatives as shown in Figure 3(a). Therefore, we propose new idea of removing false representatives from the results of text representative selection based on connected component analysis. Since text representatives refer text, we expect at least one fully connected component from each text representatives which we call as candidate text representative. Due to low resolution of video, it is hard to get fully connected components. Therefore, we define the following when there are no fully connected components present in text representatives. We extract candidate text representatives based on the end point distance of disjoint components. There can be multiple end points in a disjoint component. Hence, we consider the longest traversal path to find the end points. Let $E_1(x,y)$ and $E_2(x,y)$ be the two ends of a

disjoint component. Then, if $Dist(E_1, E_2)$ =0, or < d_1, consider the component as candidate text representative, else eliminate it. This step helps in eliminating false text representatives. As a result, we get candidate text representatives for true text representatives as shown Figure 3(b). The distance function $Dist(E_1, E_2)$ is defined as,

$$Dist(E_1, E_2) = \sqrt{(y_{E_1} - y_{E_2})^2 + (x_{E_1} - x_{E_2})^2} \quad (5)$$

C. Region Growing-1for Angle Computation

To extract text edges in the Sobel edge map from the candidate text representative, we grow candidate text representatives based on nearest neighbor concept along text direction in the Sobel edge map till the condition meet. As a result, we get word patches by grouping broken segments and character components as shown in Figure 3(b). These word patches are used in growing-2 for grouping adjacent word patches based on direction of the word patches. The conditions for stopping the growing are as follows. Each candidate text representatives are grown to the length of its radius (r) and new components found are grouped. Before merging a new text representative (TB_N) to the current text representative (TB_C), it is verified whether TB_N is a true representative, based on statistical analysis of height and width of text blocks. Once the TB_N is grouped with TB_c, the region growing operation on TB_C is stopped and is applied to TB_N. This region growing step yields a list of potential word patches (WP_n) as shown in Figure 3(d), and is used for further region growing based on the orientation of each word patches.

D. Region Growing-2 for Grouping

The potential word patches (WP_n) formed in the first stage of region growing are used to find there angle/orientation. The region growing is performed in the angular direction, from both ends of a word patch, which helps in restoring the missing text blocks. The following steps are performed on each of the word patches present in WP_n.

If $n(WP_i) > 2,\ where\ i = 1, \dots, (Total\ Word\ Patches)$, we compute the orientation of the word patch if there are more than two text representatives in the word patch 'i'.

The orientation of the WP_i is calculated in the following way,

1. Compute the orientation α_1 of the first two text representatives in WP_i.
2. Compute the orientation α_2 of the last two text representatives in WP_i.

Let the set of first two and last two text representatives in WP_i be WP_{i1} and WP_{i2}, respectively. The WP_i is now grown in both α_1 and α_2 directions from the respective ends.

When a TB_N is found, following situations are considered,

1. $TB_N \in TB_p$, then add TB_N to the respective ends WP_{i1} or WP_{i2}. Where, TB_p are potential text representatives.
2. $TB_N \in WP_j$, where $1 \le j \le n$ and $j \ne i$, then add TB_N to the respective ends WP_{i1} or WP_{i2}.
3. $TB_N \notin TB_p$, validate TB_N based on the height and width ratio of the WP_j.

When $TB_N's$ are added to both or either of the ends WP_{i1} or WP_{i2}, α_1 and α_2 are re-calculated using the updated WP_i, according to the procedure mentioned above, and angular region growing continues till no more components can be added to either ends WP_{i1} or WP_{i2}. The above mentioned steps are repeated for each of the word patches in WP_n. After region growing-2, the WP_n is populated with roughly all the expected text blocks with some false positives as shown in Figure 3(e) and the final result can be seen in Figure 3(f) where the region growing-2 extracts almost all text in Figure 2(a).

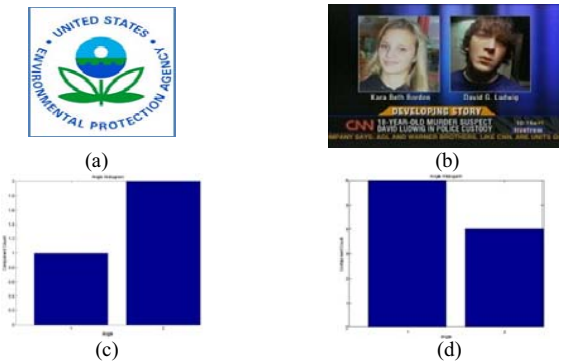


Figure 4: (a) Curve text frame; (b) Frame with horizontal text; (c) Angle histogram of word patches in (a); (d) Angle histogram of word patches in (b);

E. Classification of Horizontal and Non-horizontal Frames

It is observed that the region growing-2 method described in Section II-D works well when there is enough space between the text lines. If the space between text lines is less than two pixels then the Region Growing-1 method often detects two text lines as a single text line because the Region Growing-1 method grows in all directions of the candidate text representatives to extract arbitrarily-oriented text. To overcome this problem, we propose a classification algorithm based on angle histogram for classifying horizontal and non-horizontal text frames. Here non-horizontal text frames include curve text and non-horizontal straight lines. With this classification, we can allow region growing to grow in horizontal direction for horizontal frames and multi-direction for non-horizontal frames. The word patches WP_n formed after region growing-2 are considered for angle/orientation calculation. We compute angle 'β_j' for each $n(WP_i) > 2$. If $0° \le \beta_j \le 15°$, the WP_i is Horizontal (H), If $15° < \beta_j \le 90°$, the WP_i is Non-horizontal (NH). A histogram of the total number of H and NH word patches found in the frame are plotted as shown in Figure 4(c) and (d) for the curve frame in Figure 4(a) and horizontal frame in Figure 4(b). If $H > NH$ then the text orientation in the frame is horizontal, else Non-horizontal. For horizontal text frames, the classification technique not only segregates the text lines but also helps in restoring the missing text components after region growing 2.

III. EXPERIMENTAL RESULTS

To evaluate the proposed method, we created our own dataset as there is no benchmarked dataset on arbitrarily-oriented data, available in the literature. We collected video frames from different sources such as movies, news clips containing scene texts, sports and web video to make sure that dataset includes all kind of frames. We also found a small dataset of 45 video frames, which is available publicly [16] to evaluate the performance of the method. In order to show that the proposed method is effective, we select 142 arbitrary video text frames (curve text which excludes non-horizontal straight lines), 220 non-horizontal text frames, 800 horizontal text frames, and publicly available Hua's data of 45 frames. Note that arbitrary data may contain multi-oriented characters, words and lines while non-horizontal data may contain

only multi-oriented text lines but not characters and words. The method is also tested on ICDAR-2003 competition dataset [17] of 251 images (camera images) to check the effectiveness of our method. In summary, 1207 (142+800+45) video frames and 251 camera images are used for evaluation.

We choose a recently developed method Zhou et al. [13] which detects both horizontal and vertical text in video images, for the comparative study. Since the existing method detects only horizontal and vertical text, we use this existing method [13] for comparative study on all datasets with the proposed methods except arbitrarily-oriented text detection. We define the following categories for each detected text block (extracted text patches as shown in Figure 5(c) and (d) for the frames shown in Figure 5(a) and (b)) to evaluate the performance of the proposed method, as it is widely used in literature [10, 14].

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 detected 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 [10], in which a text block is considered correctly detected if it overlaps at least 80% of the ground-truth block. Since there is no ground truth, 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. There are two other performance measures commonly used in the literature, *Detection Rate* and *False Positive Rate*; however, they can also be converted to Recall and Precision: Recall = Detection Rate and Precision = 1 – False Positive Rate. We also consider Computational Time (CT) per frame as a measure to evaluate the proposed method. The values for the threshold T and d_1 are determined empirically as 0.01 and 4, respectively.

A. Performance of H and NH Classification Algorithm

The classification algorithm is tested on the whole dataset containing 1458 frames which include 362 non-horizontal and 1051 horizontal frames to evaluate the performance of the classification algorithm. In this experiment, we consider 142 curve text as non-horizontal data. The confusion matrix for horizontal and non-horizontal frame is shown in Table 1 where one can see classification rate for horizontal and non-horizontal is good. However, there are still misclassifications due to missing text information and multi-oriented characters in a word.

Table 1. Confusion matrix for classification of H and NH dataset (in %)

Operation	Horizontal	Non-Horizontal
Horizontal	80.7	19.1
Non-Horizontal	17.1	82.8

B. Performance on Aribitrarily-orientd Data

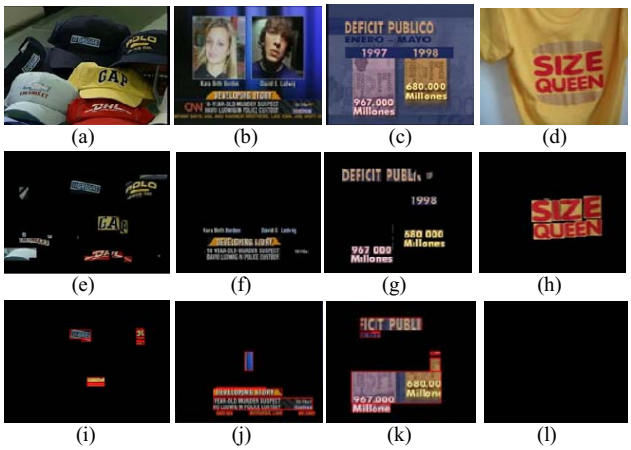
Sample results of the proposed method are shown in Figure 5 where (a), (b) are input frames with different orientation and background and (c), (d) are the text detection results for the frames shown in (a) and (b). It is noticed from the Figure 5 that the proposed method works well for arbitrarily-oriented text in video with few false positives and misdetections. The experimental results are reported in Table 2 where recall is lower than precision and misdetection rate is slightly high because the Sobel edge map which we use for dominant text point selection and lose text information when text has too low contrast. As a result, we lose some text components in a text line and sometimes we lose whole text line due to loss of dominant point. In addition, the computational time is also quite high as the proposed method involves connected component analysis while grouping.

Table 2. Experimental results for arbitrarily-oriented data

Method	R	P	F	MDR	CT
Proposed method	0.73	0.88	0.79	0.28	10.32

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

Figure 5: (a), (b) are sample of arbitrary text frames and, (c) and (d) are text detection results of the proposed method for (a) and (b), respectively.



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

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

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

Figure 6: (a)-(d) are sample frames selected from non-horizontal, horizontal, Hua's and ICDAR-2003 data, respectively. (e)-(h) are the text detection results corresponding to (a)-(d) of the proposed method and (i)-(l) are the text detection results corresponding to (a)-(d) of Zhou et al.

Table 3. Experimental results of the proposed and existing method on non-horizontal (NH), horizontal (H), Hua's and ICDAR-2003 data (ICDAR)

Data	Proposed method					Zhou et al. method [13]				
	R	P	F	MDR	CT	R	P	F	MDR	CT
NH	0.76	0.90	0.82	0.24	15.48	0.39	0.67	0.49	0.40	1.3
H	0.86	0.81	0.83	0.22	19.56	0.61	0.85	0.71	0.25	1.1
Hua's	0.88	0.77	0.82	0.32	9.06	0.72	0.82	0.77	0.44	1.1
ICDAR	0.81	0.78	0.79	0.23	5.36	0.66	0.83	0.73	0.26	3.1

C. Performance on Non-horizontal data

Figure 6(a) shows sample input frame for non-horizontal text detection and text detection result by the proposed method for the frame in (a) is shown in Figure 6(e). We can see that the proposed method detects almost all text with few false positives and missing while Zhou et al. method fails to detect text for the non-horizontal text frame in Figure 6(a) as shown in Figure 6(i). The experimental results for the proposed method and existing method on non-horizontal data is reported in Table 3 where the proposed method gives better results than the existing method. The reason for poor results by the existing method is the number of heuristics and the feature based on connected component analysis.

D. Performance on Horizontal Data

Figure 6(b) shows sample input frame and the text detection results of the proposed method for this frame is shown in Figure 6(f). The proposed method detected almost all text except the last line having a small font. The last line is missing because of missing dominant text pixel in the Sobel edge map of the input frame. Figure 6(b) and (j) shows that the existing method also gives good results for horizontal text frames with some missing texts and false positives. According to experimental results on horizontal data in Table 3, the F-measure for the proposed method is higher than existing method. However, precision of the proposed method is lower than the existing method. This shows the existing method gives less false positives compared to the proposed method. The classification of the horizontal text frames also contributed in getting better results than the non-horizontal data.

E. Performance on Hua's Data

A sample frame from Hua's data is shown in Figure 6(c) and the text detection results of the proposed method for the frame in Figure 6(c) is shown in Figure 6(g). The proposed method detected almost all text properly. The existing method for the frame shown in Figure 6(c) fixes improper bounding boxes for the text lines as shown in Figure 6(k) because of heuristics and constant thresholds. The dataset is available publicly and can be found at (http://www.cs.cityu.edu.hk/~liuwy/PE_VTDetect/). The experimental results of the proposed method and existing method are reported in Table 3 where the recall and F-measure are higher than the existing methods but precision is lower than the existing method due to more false positives.

F. Performance on ICDAR-2003 Data

This experiment is to show that the proposed method works well for high resolution camera based images, since it works well for video images with low resolution. To verify it, we conducted experiments on the benchmark database (ICDAR-2003 competition data). The sample result on ICDAR-2003 data is shown in Figure 6(d) and (h) where (d) shows the input image, (h) shows the results of the proposed method and (l) shows the result of the existing method for the frame in Figure 6(d). It is observed from Figure 6(d), (h) and (l) that the proposed method works well for camera scene text images while the existing method [13] fails to detect text in the frame. The reported quantative results in Table 3 for ICDAR-2003 data reveal that recall and F-measure is higher than the existing method but precision is lower than the existing method. This is because of more false positives and missing text information. The recall is low for the existing method because it is designed for caption text detection but not scene text detection. We can conclude from the experimental study that the proposed method is good for different kinds of frames.

Table 3 shows that the existing method requires less computational time (CT) for all data compared to the proposed method because the existing method focuses on caption text and big font with high contrast while the proposed method focuses on both scene text and graphics text where we can see more disconnections and broken segments. As a result, grouping criterion based on connected component analysis consumes more time. Thus, we can infer that the proposed method works well for complex arbitrarily oriented text detection at the cost of computations.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a gradient direction and magnitude feature based method for arbitrarily-oriented text detection in video frames. In this work, we have introduced a new method for obtaining candidate text representatives for text lines in the video frames with the help of Sobel edge map of input frame. To handle the problem of arbitrarily-oriented text, we propose two stage region growing methods. In first stage, the method merges the nearest neighbor components in Sobel edge map to estimate the direction of the word patches. In the second stage, the method joins word patches based on direction information and restore the missing word patches during first stage region growing. Experimental results and comparative study with existing method shows that the proposed method outperforms the existing method for arbitrarily-oriented, non-horizontal and horizontal text frames.

ACKNOWLEDGMENT

This work is done jointly by NUS, Griffith University, Australia 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).

REFERENCES

[1] J. Zang and R. Kasturi, "Extraction of Text Objects in Video Documents: Recent Progress", DAS, 2008, pp. 5-17.

[2] K. Jung, K.I. Kim and A.K. Jain, "Text Information Extraction in Images and Video: a Survey", Pattern Recognition, 2004, pp. 977-997.

[3] B. Epshtein, E ofek and Y. Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform", CVPR, 2010, pp. 2963-2970.

[4] Y. F. Pan, X. Hou and C. L. Liu, "A Hybrid Approach to Detect and Localize Texts in Natural Scene Images", IEEE Trans. on IP, 2011, pp. 800-813.

[5] P. P .Roy, U. Pal, J. Llados and F. Kimura, "Multi-Oriented English Text Line Extraction using Background and Foreground Information", DAS, 2008, pp. 315-322.

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

[7] K. Jung and J. H. Han, "Hybrid Approach to Efficient Text Extraction in Complex Color Images", Pattern Recognition Letters, 2004 pp. 679–699.

[8] K. L Kim, K. Jung and J. H. Kim. "Texture-Based Approach for Text Detection in Images using Support Vector Machines and Continuous Adaptive Mean Shift Algorithm" IEEE Trans. on PAMI, 2003, pp. 1631-1639.

[9] M. Anthimopoulos, B. Gatos and I. Pratikakis, "A Two-Stage Scheme for Text Detection in Video Images", Image and Vision Computing, 2010, pp. 1413-1426.

[10] D. Chen, J.M. Odobez and J.P. Thiran, "A Localization/Verification Scheme for Finding Text in Images and Video Frames based on Contrast Independent Features and Machine Learning", Signal Processing: Image Communication, 2004, pp. 205-217.

[11] A. Jamil, I. Siddiqi, F. Arif and A. Raza, "Edge-based Features for Localization of Artificial Urdu Text in Video Images", ICDAR, 2011, pp. 1120-1124.

[12] H. Tran, A. Lux, H. L. T. Nguyen and A. Boucher, "A Novel Approach for Text Detection in Images using Structural Features", ICAPR, 2005, pp. 627-635.

[13] J. Zhou, L. Xu, B. Xiao and R. Dai, "A Robust System for Text Extraction in Video", ICMV, 2007, pp. 119-124.

[14] P. Shivakumara, T, Q. Phan and C. L. Tan, "A Laplacian Approach to Multi-Oriented Text Detection in Video", IEEE Trans. on PAMI, 2011, pp. 412-419.

[15] A. Mishra, K. Alahari and C. V. Jawahar, "An MRF Model for Binarization of Natural Scene Text", ICDAR, 2011, pp 11-16.

[16] X.S Hua, L. Wenyin and H.J. Zhang, "An Automatic Performance Evaluation Protocol for Video Text Detection Algorithms", IEEE Trans. on CSVT, 2004, pp 498-507.

[17] S. M. Lucas, "ICDAR 2005 Text Locating Competition Results", ICDAR, 2005, pp. 80-84.