

A New Method for Word Segmentation from Arbitrarily-Oriented Video Text Lines

Author

Sharma, Nabin, Shivakumara, Palaiahnakote, Pal, Umapada, Blumenstein, Michael, Tan, Chew Lim

Published

2012

Conference Title

Proceedings of the International Conference on Digital Image Computing Techniques and Applications (DICTA), 2012

DOI

<https://doi.org/10.1109/DICTA.2012.6411703>

Copyright Statement

© 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from

<http://hdl.handle.net/10072/51287>

Link to published version

<http://dicta2012.csse.uwa.edu.au/>

Griffith Research Online

<https://research-repository.griffith.edu.au>

A New Method for Word Segmentation from Arbitrarily-Oriented Video Text Lines

Nabin Sharma*, Palaiahnakote Shivakumara[†], Umapada Pal[‡], Michael Blumenstein* and Chew Lim Tan[†]

*Griffith University, Queensland, Australia. Email: {m.blumenstein, nabin.sharma}@griffith.edu.au

[†]School of Computing, National University of Singapore, Singapore. Email: {shiva, tanc1}@comp.nus.edu.sg

[‡]Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, India. Email: umapada@isical.ac.in

Abstract—Word segmentation has become a research topic to improve OCR accuracy for video text recognition, because a video text line suffers from arbitrary orientation, complex background and low resolution. Therefore, for word segmentation from arbitrarily-oriented video text lines, in this paper, we extract four new gradient directional features for each Canny edge pixel of the input text line image to produce four respective pixel candidate images. The union of four pixel candidate images is performed to obtain a text candidate image. The sequence of the components in the text candidate image according to the text line is determined using nearest neighbor criteria. Then we propose a two-stage method for segmenting words. In the first stage, for the distances between the components, we apply K-means clustering with $K=2$ to get probable word and non-word spacing clusters. The words are segmented based on probable word spacing and all other components are passed to the second stage for segmenting correct words. For each segmented and un-segmented words passed to the second stage, the method repeats all the steps until the K-means clustering step to find probable word and non-word spacing clusters. Then the method considers cluster nature, height and width of the components to identify the correct word spacing. The method is tested extensively on video curved text lines, non-horizontal straight lines, horizontal straight lines and text lines from the ICDAR-2003 competition data. Experimental results and a comparative study shows the results are encouraging and promising.

Keywords: Video text line, Video text candidates, Video word segmentation.

I. INTRODUCTION

The improvement of recognition system performance through text region segmentation from video frames is a challenging and an interesting problem currently in the field of image processing, pattern recognition and document image analysis [1-4]. This is because video text recognition helps in bridging the gap between low level features and high level features for efficient, accurate indexing and retrieval of events from sports video databases [1-4]. In addition, if we apply the present OCR systems, which are developed for recognizing simple and high resolution document images, directly on video frames which have low resolution, complex background, arbitrary oriented text etc, gives very poor recognition rates [5-6]. To overcome this problem, rather than sending the whole video text frame to OCR, methods have been developed to detect/segment text in the video frame to pass segmented regions/lines to OCR. As a result, the OCR receives segmented

text with less complex background information. There are plenty of methods [7-8] for text detection and segmentation from video frames available in literature, which detect horizontal straight lines, non-horizontal straight lines and even curved lines also (arbitrarily-oriented text). For instance, the method [9] based on a Laplacian approach and the skeleton was proposed recently for multi-oriented text detection in video. This method works well only for horizontal and non-horizontal straight lines but not for arbitrarily-oriented text lines because the skeleton concept for classification of simple and complex components works well for multi-oriented text lines. Recently, the method [10] based on gradient direction and the perimeter of component growing is proposed to detect arbitrarily-oriented text from video frames, which works well for different kinds of text in video frames. Therefore, we considered the output of this method as input for the word segmentation in this work. Despite the text lines being given as input to the OCR, recognition rates for the video text does not improve much because of the unpredictable characteristics of video. There are methods [11-12] to improve the binarization to achieve better result for video text recognition, by considering text lines as input. But these methods do not give consistent results for video data due to variations in background and contrast. Additionally, obtaining feasible thresholds to binarize the video image is hard compared to a document image. Hence, we cannot rely on binarization methods to improve the video text recognition rate.

Consequently, the methods [13-14] for word and character segmentation from text lines were proposed to improve the recognition rate by reducing the complex background complexity. In this regard, a Fourier-moments based method [15] was proposed for word and character segmentation from text lines, which uses clustering and run-length criteria to segment the words, and text height difference at the character boundary was used for segmenting characters. A gradient vector flow based method [16] for video character segmentation was developed. Gradient flow is used for identifying seed points and a least cost path is used for segmenting the characters. The main drawback of these methods is that the methods consider horizontal and non-horizontal text lines but not arbitrarily-oriented text lines because segmenting words and characters from arbitrary text lines is much more complex than horizontal and non-horizontal text lines. Hence, we propose a new method for segmenting words from arbitrarily-oriented text

lines. To the best of our knowledge, this is the first attempt towards word segmentation from arbitrarily-oriented video text lines.

II. PROPOSED METHOD

The proposed method takes arbitrarily-oriented text lines detected by the method presented in [10] as input for word segmentation since text detection is beyond the scope of this work. The reason for choosing this method is that it works well despite the above-mentioned problems in video. Since video text suffers from low resolution, complex background and arbitrary-orientation, it is hard to extract the complete text information. Therefore, we propose four gradient directional features to extract respective pixel candidate images of the text in video. We are inspired by the work presented in [17-18] for text pixel selection using gradient directional features, where it is clearly shown that gradient directional features are useful in separating text and non-text pixels. Therefore, in this work, we combine four pixel candidate images to restore the possible text information, which we call the text candidate image. In this way, the proposed method is different and effective to obtain text information, as compared to other research presented in the literature. The sequence of the components in the text candidate image is determined based on the nearest neighbor criteria. Then we introduce a two stage method for segmenting words from a sequence of components and it works based on the fact that the space between the words is greater than the space between the characters. The flow diagram of the method is depicted in Figure 1.

A. Text Candidate-based Gradient Directional Features

The output of the text detection method as shown in Figure 2(a) is considered as input for our work. The input image is converted into grey scale to make implementation simple as

shown in Figure 2(b). For the image shown in Figure 2(b), the method obtains the Canny edge map as shown in Figure 2(c) to compute gradient information for each edge pixel. We use the default parameter values of the MATLAB function to get the Canny edge map. For each pixel in the Canny edge map, the method computes four gradient directional features by referring to the gradient image of the input text line, to extract pixel candidates. The basis for computing four directional features is that there exists a parallel edge for every edge in a character and it is also observed that the edge of a stroke typically appears as a pair [17-18]. We identify the two edge pixels of parallel edges in the following way: for an edge pixel $P_s(x, y)$ consider the gradient direction $\theta(P_s)$ and magnitude $f(P_s)$. Traverse towards the direction roughly perpendicular to $\theta(P_s)$ until an edge pixel $P_e(x, y)$ is found, as shown in Figure 2(d). Considering P_s and P_e , the four directional features are computed as given below.

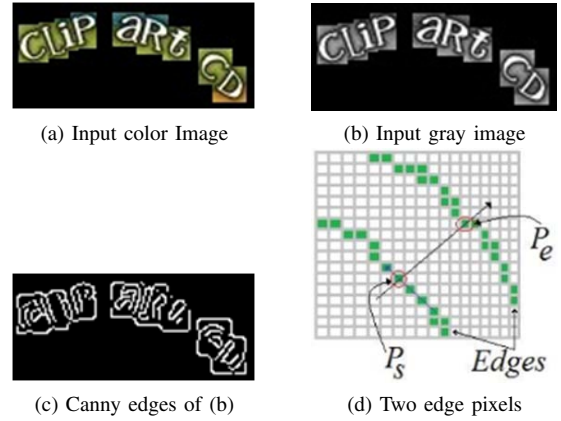


Fig. 2: Two edge pixels found based on gradient direction and traversing towards perpendicular to gradient direction.

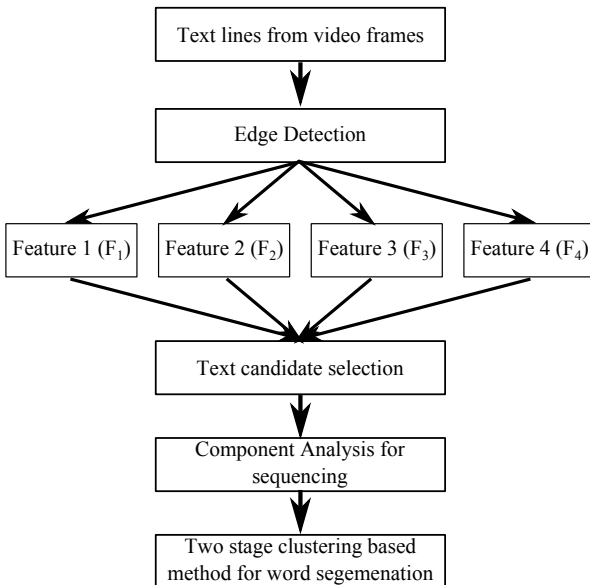


Fig. 1: Flow diagram for word segmentation

1) *Magnitude Symmetry (F_1)*: If the magnitude of the two pixels (P_s and P_e) shown in Figure 2(d) is almost the same then we consider the two pixels as candidates. i.e if $f(P_s) - f(P_e) < T$, then $\{P_s, P_e\}$ are considered as pixel candidates. The threshold T is a positive value, and is considered as nearer to zero as possible. In this way, the method obtains a Pixel Candidate Image (PCI) for all pairs of edge pixels as shown in Figure 3(a), where almost all text pixels are obtained by checking magnitude symmetry for each pair of edge pixels.

2) *Reverse Directional feature (F_2)*: Consider the gradient direction $\theta(P_s)$ at the starting pixel(P_s) as shown in Figure 2(d). Now from P_s traverse towards the direction perpendicular to $\theta(P_s)$, and check whether a pixel(P_e) is reached. If P_s can be traversed back from P_e along the direction perpendicular to $\theta(P_e)$, then $\{P_s, P_e\}$ are considered as pixel candidates. This results in the pixel candidate image (PCI) by F_2 , as shown in Figure 3(b), where almost all text pixels are obtained.

3) *Opposite Gradient Directional Pixel Identification*(F_3): For the two pixels on parallel edges shown in Figure 2(d), the method finds opposite gradient direction of those two pixels (towards background). If the directions of both pixels is the same, then we consider those pixels as pixel candidates. This results in the pixel candidate image by F_3 . The effect of F_3 is shown in Figure 3(c) where text pixels are retained.

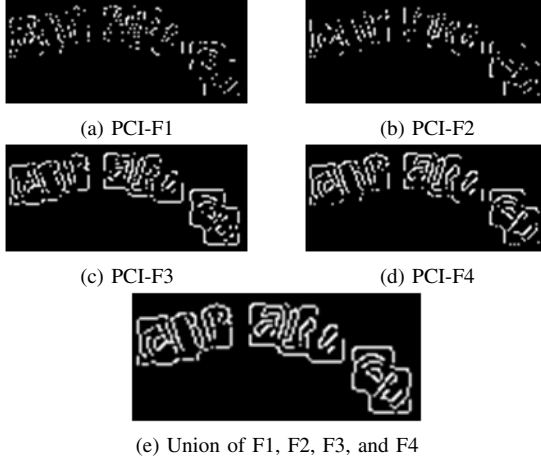


Fig. 3: Steps for computing text candidate images.

4) *Gradient Directional Pixel Identification* (F_4): For the two pixels P_s and P_e as shown in Figure 2(d), the method matches the gradient direction of those two pixels. If the gradient directions of the pixels are the same then both the pixels are considered pixel candidates. The effect of this feature can be seen in Figure 3(d), where it retains almost all text pixels as present in the Canny edge map.

To restore the possible text information, we combine all four pixel candidate images given by F_1, F_2, F_3 and F_4 , respectively by performing the Union operation. The result is shown in Figure 3(e), where clear text pixels as in the Canny edge map can be seen. The output of the Union operation is considered as the text candidate image.

B. Extracting Consecutive Components

To find space between the words from an arbitrarily oriented text line, it is essential to remember the sequence of the text components in the text candidate image. Therefore, we propose nearest neighbor criteria to find the sequence of the text components by estimating distance between them. The distance is computed as follows. Let CF_1 and CF_2 be the two bounding boxes of filled components as shown in the Figure 4. Let $\alpha_1(x_1, y_1)$ and $\alpha_2(x_2, y_2)$ be the top and bottom corners of the right face of the component CF_1 respectively, and $\beta_1(x_3, y_3)$ and $\beta_2(x_4, y_4)$ be top and bottom corners of the left face of the component CF_2 respectively. We calculate the distance D_1 (between α_1 and β_1) and D_2 (between α_2 and β_2) as given below. Finally, average of D_1 and D_2 is considered as actual distance (d) between components as defined in equation (3). It is found that the distance values

consist of too high and too low values. Therefore, we use \log_e to normalize the distance values as defined in equation (3). Let $D = \{d_1, d_2, d_3, \dots, d_n\}$ be the set of distances computed between the consecutive components.

$$D_1 = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} \quad (1)$$

$$D_2 = \sqrt{(x_2 - x_4)^2 + (y_2 - y_4)^2} \quad (2)$$

$$d = \log\left(\frac{D_1 + D_2}{2}\right) \quad (3)$$

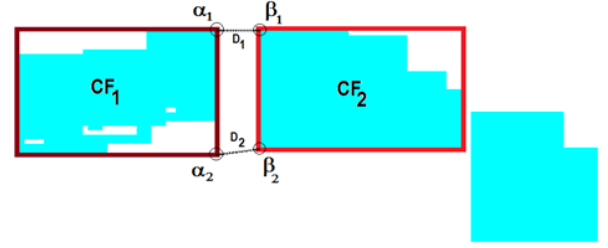


Fig. 4: Illustration for computing distance between text components to find consecutive components

C. Two Stage Method for Word Segmentation

Generally the space between the words is greater than the space between the characters in a text line. In this section, we exploit this feature to segment the words using the distance vector (D). Since the input is arbitrary-oriented text in video, there are chances of getting over segmentation and under segmentation. We believe solving under segmentation is easier than over segmentation because finding space between the components is easy if space exists, than merging the components. Therefore, we propose a two-stage method for correct word segmentation.

1) *Stage-1 for Probable Word Segmentation*: It is noted that the distance vector contains high values, which relate to the space between the words and low values relating to space between the characters, because a text line usually has only word and characters spaces. In order to separate high and low values, we propose K-means clustering with K=2. However, K-means clustering is sensitive to initial guesses. Therefore, we use the Min-Max clustering method to select an initial guess from the distance vector. The mathematical steps for Min-Max clustering are as follows. The minimum and maximum values are computed as

$$MinD = \text{Min}\{d_1, d_2, d_3, \dots, d_n\} \quad (4)$$

$$MaxD = \text{Max}\{d_1, d_2, d_3, \dots, d_n\} \quad (5)$$

Then Min and Max clusters are obtained as follows, if $(d_i > 0 \text{ and } (d_i - \text{MinD}) \leq (\text{MaxD} - d_i))$, then $\text{MinCluster}_x = d_i$ else

$$MaxCluster_y = d_i$$

Where, x and y are the number of elements in the Min and Max clusters, respectively. Once all the distances are clustered into either Min or Max, the average of each cluster is computed,

$$MinAvg = \frac{MinCluster_x}{x} \quad (6)$$

$$MaxAvg = \frac{MaxCluster_y}{y} \quad (7)$$

The MinAvg and MaxAvg are considered as an initial guess for the K-means clustering algorithm to segment the words. The K-means clustering produces two clusters for the distance vector with the initial guesses, where cluster-1 having high distance values (probable word spaces) and cluster-2 having low distance values (probable non-word spaces). Since distances in cluster-2 do not represent word spaces, we simply pass those components to stage-2 for further verification. The distances in cluster-1 are representing word spaces. Hence, we segment the text components and pass the segmented results to stage-2 for verification. This is true because of the fact that the space between the words is greater than the space between the characters. The flow diagram of stage-1 is shown in Figure 5. However, segmentation based on cluster-1 distances may allow under-segmentation but not over-segmentation. Therefore, we consider the output of stage-1 as probable words. One such example is shown in Figure 6, where for the input image shown in Figure 6(a), stage-1 allows under-segmentation but not over-segmentation as shown in Figure 6(b), while for the image shown in Figure 6(c), stage-1 segments all words

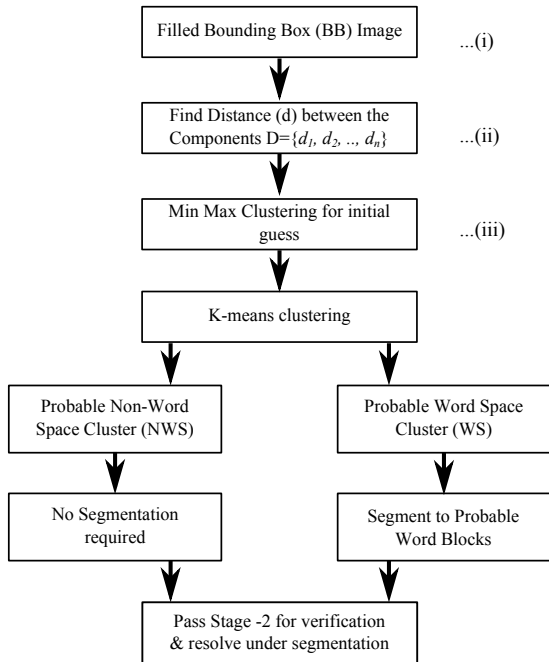


Fig. 5: Flow chart for stage-1 of word segmentation

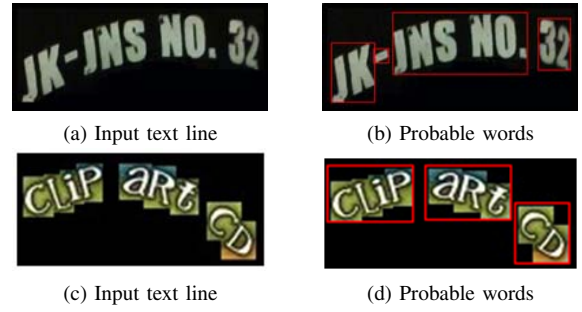


Fig. 6: Result of probable word segmentation

correctly as shown in Figure 6(d). This is the advantage of the stage-1 method.

2) *Stage-2 of Correct Word Segmentation*: Upon the output of stage-1, the method applies steps (i), (ii) and (iii) shown in Figure 5 until K-means clustering produces two clusters. In this stage, we introduce key features such as Height (H) and Width (W) of components to decide whether the component is a word or character. This hypothesis is based on the fact that generally the width is greater than height for the words and vice versa for characters. Since the probable segmentation is done in stage-1, K-means clustering may produce one cluster containing one or many equal distance values, or two clusters having both high and low values, or no clusters. Hence, there is a need to introduce other features such as H and W to decide the correct words apart from the two clusters. The various situations are discussed below,

Case 1: Number of Clusters = 0: This signifies that the distance vector is empty i.e. $D=\{\phi\}$. Hence, as no clusters exist, no more segmentation is required, and it is considered as the final result.

Case 2: Number of Clusters = 1: Then the following conditions are considered, if 'D' contains only one value then the value is compared with the MinAvg and MaxAvg of cluster 1 and cluster 2, respectively, formed in stage-1. If the distance is close to MinAvg then the distance is considered as a character space, otherwise it is considered as a correct word space. If 'D' contains more than one value and all in one cluster then it signifies that all the distances are equal. In this situation, Height and Width are considered to decide whether the distances belong to word or character space. If any one of the components in the word image satisfies $H > W$ then all components in the probable word image are considered as part of the same word, else if all components have $W > H$ then each component is considered as one word.

Case 3: Number of Clusters = 2: If distance (d) belongs to cluster-1 (representing word space-WS) and $W > H$, then 'd' is considered as a word space.

If $d \in NWS$ (non-word space) and $W > H$ then 'd' is considered as a character space. Here $W > H$ signifies that the component contains more than a single character but may not be a complete word. As a result, the component is not considered as a separate word.

If $d \in NWS$ and $W \leq H$ then 'd' is considered as a character

space as it belongs to the NWS cluster. Here $W \leq H$ signifies that the component is a single character. Hence ‘d’ is a character space.

The sample results for each case are shown in Figure 7, 8 and 9 respectively, where height(H), width(W) and the number of clusters plays a vital role in segmenting words correctly. Figure 7 shows an example of case 1 where the results of both stage-1 and stage-2 are the same. The result of stage-1 shown in Figure 7(a), segmented the words correctly, and there is no space between the characters, which can be seen in the figure 7(b), where three word patches are completely filled. Hence, the stage-2 method doesn’t produce any clusters for this case, and the final word segmentation results are shown in figure 7(c). It can be observed from Figure 8 that the word “PUBLICO” was segmented correctly by stage-1 as shown in Figure 8(a). Stage-2 produces clusters of equal distances as shown in Figure 8(b), where five distance values between the patches appears to be equal. This results in one cluster having five equal distance values. In this situation, our method checks whether any one of the patches has $H > W$, then the whole image is considered as a correct word, as shown in Figure 8(c). It can be noted from Figure 8(b) that the patch for “CO” does not satisfy the condition ($H > W$) but other patches satisfy the condition. Figure 9 shows an example of case-3, where the result shown in Figure 9(a) was obtained by stage-1, and “JNS NO” was considered as one word. Stage-2 produces two clusters with high and low values for “JNS NO” as shown in Figure 9(b). The distance between the patch for “JNS” and “NO” is greater the distance between ‘N’ and ‘O’. In this scenario, our method segments the “JNS” as a word since the distance between “JNS” and “NO” is referring to a word space. “NO” is considered as another word, because the distance between ‘N’ and ‘O’ refers to a character space. The final results can be seen in Figure 9(c). The complete flowchart of stage-2 including case-1, case-2 and case-3 for word segmentation is shown Figure 10(a)-(c).

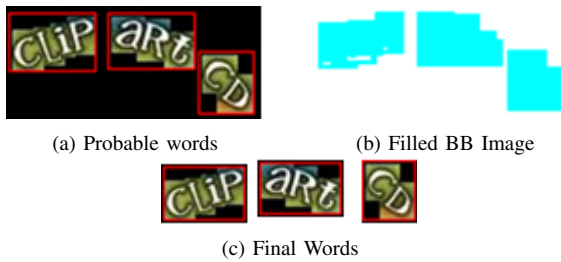


Fig. 7: Example of Case 1. There is no space between characters, hence $D = \{\phi\}$ and number of clusters = 0

III. EXPERIMENTAL RESULTS

To evaluate the proposed method, we create our own dataset as there is no benchmark dataset available in the literature for word segmentation from arbitrarily-oriented text lines. We extract 70 arbitrarily oriented text lines from video frames (curved text which excludes non-horizontal and

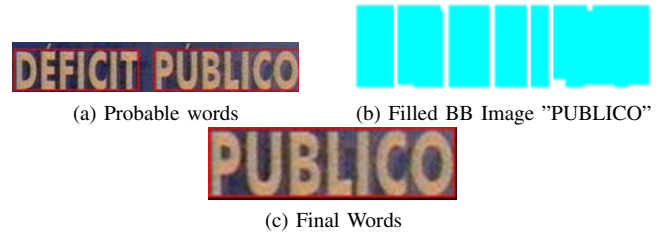


Fig. 8: Example of Case 2. There is an equal space between characters, hence number of clusters = 1 and at least one component has $H > W$

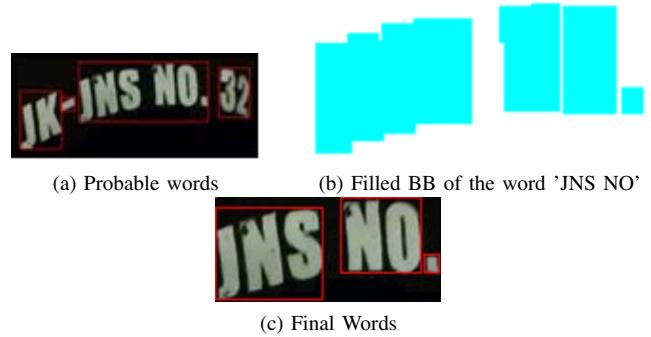


Fig. 9: Example of Case 3. There are both word and character spaces, hence number of clusters = 2

horizontal straight lines), 352 non-horizontal text lines, 1047 horizontal text lines, and 93 text lines from Hua’s publicly available data [19]. It is to be noted 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. To test the robustness and effectiveness of the proposed method, we also tested on 125 text lines extracted from the ICDAR-2003 competition dataset [20] of camera images. In summary, 1535 (70+325+1047+93) video text lines and 125 text lines from camera images are used for the purpose of experimentation. The following definitions are used for measuring the method.

Truly Detected Word (TDW): A segmented block that contains correctly segmented words. *Under Segmented Blocks (USB)*: A segmented block which contains less number of words than expected. *Over Segmented Blocks (OSB)*: A segmented block that contains more words than expected. Since there is no ground truth available, we manually count the *Actual Number of Words (ANWs)* in the text lines and it is considered as ground truth for evaluation. We use the standard *Recall (R)*, *Precision (P)*, and *F-measure (F)* as performance measures. The performance measures are defined as follows. $Recall (R) = TDW / ANW$, $Precision (P) = TDW / (TDW + USB + OSB)$, $F-measure (F) = (2 \times P \times R) / (P + R)$.

A. Experiments on Curved Data

Sample results for word segmentation on curved data are shown in Figure 11, where (a) and (b) are the input images,

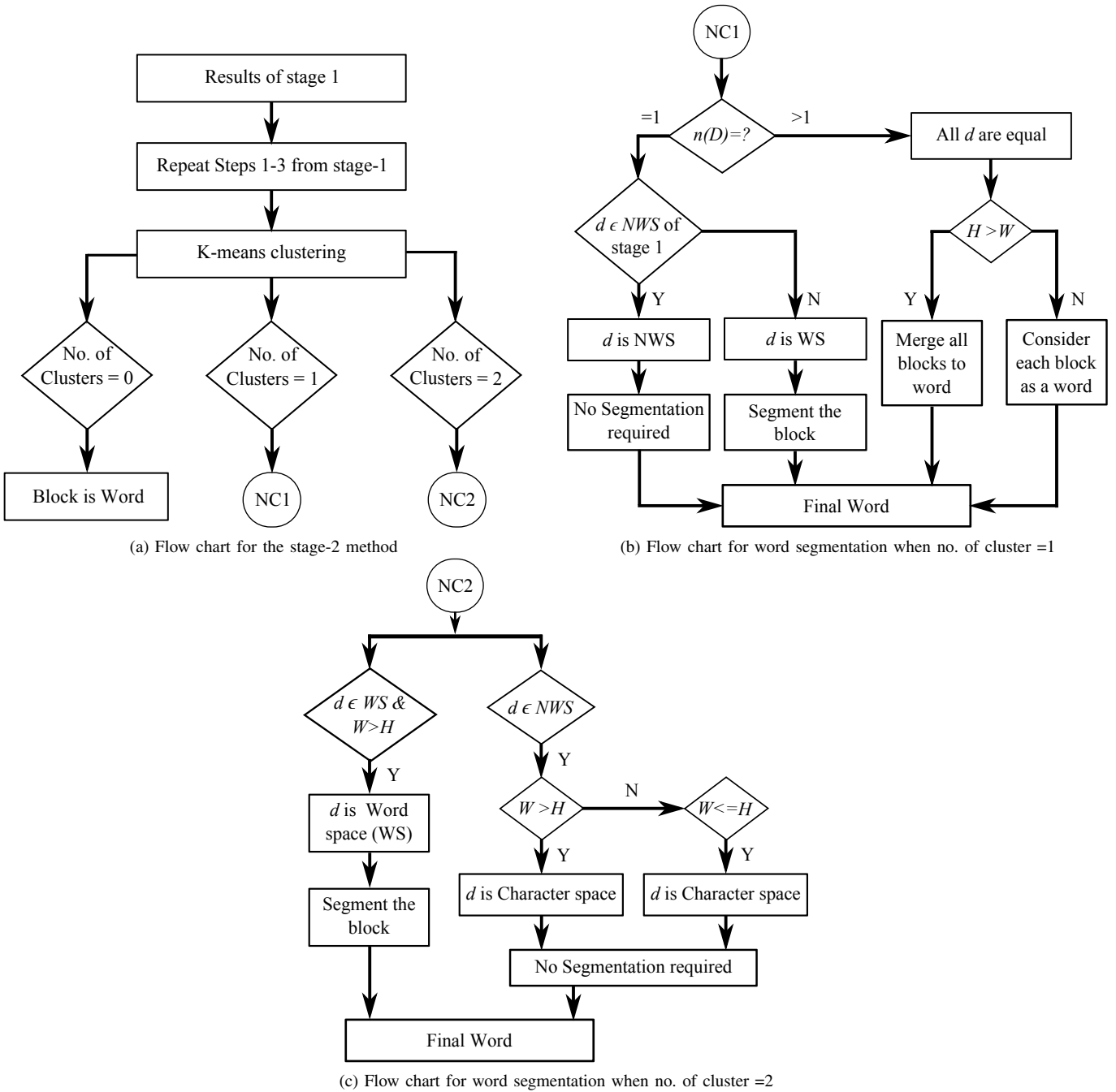


Fig. 10: Flow chart for stage-2 of word segmentation

(c) and (d) are the results of stage-1, and (e) and (f) are the result of stage-2. Figure 11 shows that stage-1 allows under segmentation (Figure 11(c)) for the image shown in Figure 11(a) and stage-2 segments words properly while for other images shown in Figure 11(b), stage-1 and stage-2 gives the same results. The quantitative results for curved data are reported in Table-I, where the F-measure is low and precision is a bit higher than recall. This shows that there is scope for further improvement. The main reason for low results is too low contrast and too small font text in video frames. In these

cases, we lose edge information since the gradient is not good for too low contrast images.

B. Experiments on Non-Horizontal Straight Line Data

For non-horizontal straight lines, sample results are shown in Figure 12, where stage-1 doesn't produce any results for the image shown in Figure 12(a) because the distance values between the components are classified into Probable-Non-Word-Space cluster. Therefore, stage-2 gives correct results for the image where height and width plays a role to decide the word spacing. Whereas for other images, which have

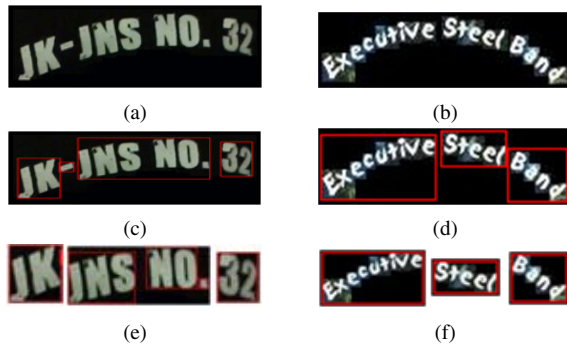


Fig. 11: (a),(b)Samples of input arbitrarily-oriented text lines; (c),(d)Word segmentation results obtained in stage-1; and (e),(f)Final word segmentation results obtained in stage-2.

TABLE I: Experimental results on various datasets

Dataset	R	P	F
Curve dataset	0.60	0.70	0.65
Non-Horizontal Straight Line	0.84	0.85	0.82
Horizontal Straight Line	0.85	0.91	0.88
Hua dataset	0.86	0.79	0.83
ICDAR 2003	0.81	0.83	0.82

low resolution, as shown in Figure 12(b), the same results are produced for both stage-1 and stage-2. The experimental results reported in Table-I reveal that the proposed method gives promising results and are higher than the curve data.

C. Experiments on Horizontal Straight Line Data

Figure 13 shows sample results for horizontal straight line data. We consider different scripts (Bangla, Urdu, and English) to show that the proposed method works irrespective of scripts. It is observed from Figure 13 that stage-1 allows under segmentation for the image shown in Figure 13(a) and stage-2 gives correct results. For the other image shown in Figure

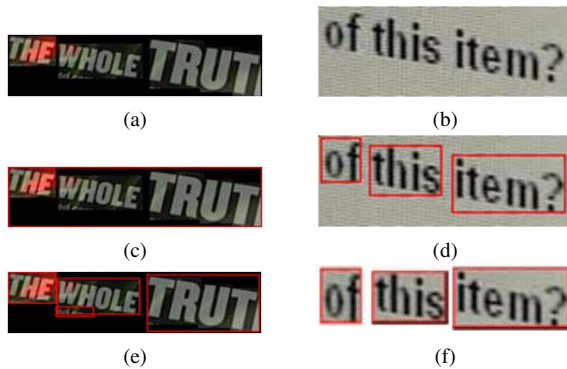


Fig. 12: (a),(b)Samples of Non-horizontal text lines; (c),(d)Word segmentation results obtained in stage-1; and (e),(f)Final word segmentation results obtained in stage-2.

13(b), stage-1 and stage-2 give the same results. Table-I shows that the F-measure of the proposed method is slightly higher than non-horizontal straight lines and better than curved data. As a result, we can infer that as the complexity increases, the performance of the method decreases.



Fig. 13: (a),(b)Samples of horizontal text lines; (c),(d)Word segmentation results obtained in stage-1; and (e),(f)Final word segmentation results obtained in stage-2.

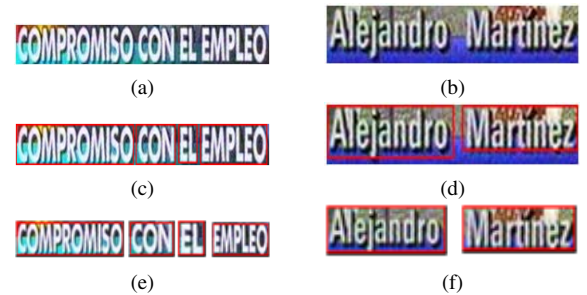


Fig. 14: (a),(b)Samples of input text lines from Hua dataset; (c),(d)Word segmentation results obtained in stage-1; and (e),(f)Final word segmentation results obtained in stage-2.

D. Experiments on Hua dataset

Even though this dataset [19] is small, it provides an element of objective testing for the proposed method as it is publicly available and is considered as a standard dataset. For the images shown in Figure 14, the proposed method segments words properly, even though the images having complex backgrounds and font difference. The results reported in Table-I also reveal that the proposed method results are comparable and encouraging.

E. Experiment on ICDAR-2003 Data

We also considered Camera based scene images to evaluate the proposed method's performance because the ICDAR-2003 competition data is considered as a very complex dataset for text detection and word segmentation according to the accuracy reported in literature [20]. The sample results shown in Figure 15 shows that the proposed method works well for the image having blur (Figure 15(a)) and image having complex backgrounds and multiple colours (Figure 15(b)). Table-I shows that the recall, precision and F-measure are promising and are comparable to results of video data. Therefore, we can

conclude that the proposed method is good for camera-based scene images and video images as well.



Fig. 15: (a),(b)Samples of input text lines from ICDAR 2003; (c),(d)Word segmentation results obtained in stage-1; and (e),(f)Final word segmentation results obtained in stage-2.

F. Comparative Study

Since there is no method available for word segmentation from arbitrarily-oriented video text in the literature, we cannot compare our results. There is a recent method [15] for word segmentation that works only for horizontal and non-horizontal straight lines and we give a comparative study using this work. We tested our method on the same dataset used by the method [15] for word segmentation and found almost the same accuracy as that of the method [15]. The comparative study is presented in Table-II. The scope of the method [15] is limited to Horizontal (H) and Non-Horizontal (NH) text lines only. But the strong contribution of our method is that it works for complex arbitrarily oriented text as well as curved lines, irrespective of font, orientation, script, contrast and background variations, without sacrificing the accuracy for the horizontal and non-horizontal data.

TABLE II: Experimental results obtained for the comparative study

Data	Proposed Method			Rajendran et. al [15]		
	R	P	F	R	P	F
NH and H	0.85	0.86	0.85	0.85	0.87	0.85

IV. CONCLUSION

In this paper, we have explored four gradient directional features to obtain text candidates from arbitrarily-oriented text line images, located by the text detection method based on information from parallel edges in characters. We have introduced a two stage method for word segmentation based on the fact that the space between the words is greater than the space between characters. Stage-1 allows under segmentation and stage-2 segments and verifies the segmented words using cluster nature and aspect ratio of the text components in the image. The method is evaluated on different datasets such as curved data, non-horizontal, horizontal, Hua's data and ICDAR-2003 competition data. The experimental

results show that the proposed method works well irrespective of font, orientation, script, contrast and background variations.

V. ACKNOWLEDGMENT

This work is done jointly by NUS, Griffith University, Australia and Indian Statistical Institute, Kolkata, India. This research is partly supported by the A*STAR grant 092 101 0051 (WBS no. R252-000-402-305).

REFERENCES

- [1] D. Doermann, J. Liang and H. Li, "Progress in Camera-Based Document Image Analysis", In Proc. ICDAR, 2003, pp.606-616.
- [2] J. Zang and R. Kasturi, "Extraction of Text Objects in Video Documents: Recent Progress", In Proc. DAS, 2008, pp.5-17.
- [3] K. Jung, K.I. Kim and A.K. Jain, 2004. "Text information extraction in images and video: a survey", Pattern Recognition, 2004, pp.977-997.
- [4] C. Wolf and J. M Jolion, "Extraction and Recognition of artificial text in multimedia documents", Pattern Analysis and Applications, 2003, pp.309-326
- [5] S. H. Lee and J. H. Kim, "Complementary combination of holistic and component analysis for recognition of low resolution video character images", Patten Recognition Letters, 2008, pp.383-391.
- [6] D. Chen and J. M. Odobez, "Video text recognition using sequential Monte Carlo and error voting methods". Pattern Recognition Letters, 2005, pp.1386-1403.
- [7] N. Sharma, U. Pal, M. Blumenstein, "Recent Advances in Video Based Document Processing: A Review", In Proc. DAS, 2012. pp.63-68.
- [8] X. Huang, H. Ma and H. Zhang, "A New Video Text Extraction Approach", In Proc. ICME, 2009, pp.650-653.
- [9] P. Shivakumara, T. Q. Phan and C. L. Tan, "A Laplacian Approach to Multi-Oriented Text Detection in Video", IEEE Transactions on PAMI, 2011, pp.412-419.
- [10] N. Sharma, P. Shivakumara, U. Pal, M. Blumenstein, Chew Lim Tan, "A New Method for Arbitrarily-Oriented Text Detection in Video", In Proc. DAS, 2012, pp.74-78.
- [11] Z. Zhou, L. Li and C. L. Tan, "Edge based Binarization for Video Text Images", In Proc. ICPR, 2010, pp.133-136.
- [12] S. Bolan, L. Shijian and Chew Lim Tan, "Binarization of Historical Document Images Using the Local Maximum and Minimum", In Proc. DAS, 2010, pp.159-165.
- [13] S. Kopf, T. Haenselmann, W. Effelsberg, "Robust character recognition in low resolution images and videos", Technical report, Department for mathematics and computer science, University of Mannheim, 2005.
- [14] P. Shivakumara, S. Bhowick, S. Bolan, C. L. Tan and U. Pal, "A New Gradient based Character Segmentation Method for Video Text Recognition", In Proc. ICDAR, 2011, pp.126-130.
- [15] D. Rajendran, P. Shivakumara, S. Bolan, S. Lu and C. L. Tan, "A New Fourier-Moments based Video Word and Character Extraction for Recognition", In Proc. ICDAR, 2011, pp.1165-1169.
- [16] T. Q. Phan, P. Shivakumara, S. Lu and C. L. Tan, "A Gradient Vector Flow-Based Method for Video Character Segmentation", In Proc. ICDAR, 2011, pp.1024-1028.
- [17] J. Zhang and R. Kasturi, "Text Detection using Edge Gradient and Graph Spectrum", In Proc. of ICPR, 2010, pp.3979-3982.
- [18] B. Epshtein, E. Ofex and Y. Wexler, "Detecting text in natural scenes with Stoke Width Transform", In Proc. CVPR, 2010, pp.2963-2970.
- [19] X. S. Hua, L. Wenyin and H. J. Zhang, "An automatic performance evaluation protocol for video text detection algorithms", IEEE Trans. CSVT, 2004, pp.498-507.
- [20] S. M. Lucas, "ICDAR 2005 text locating competition results", In Proc. ICDAR, 2005, pp.80-84.