A New Method for Character Segmentation from Multi-Oriented Video Words

Nabin Sharma*, Palaiahnakote Shivakumara†, Umapada Pal‡, Michael Blumenstein* and Chew Lim Tan§
*Griffith University, Queensland, Australia 4222. Email: {nabin.sharma, m.blumenstein}@griffith.edu.au
†University of Malaya (UM), Kuala Lumpur-50603, Malaysia. Email: shiva@um.edu.my
‡CVPR Unit, Indian Statistical Institute, Kolkata, India 700108. Email: umapada@isical.ac.in
§School of Computing, National University of Singapore, Singapore. Email: tancl@comp.nus.edu.sg

***Abstract*—This paper presents a two-stage method for multi-oriented video character segmentation. Words segmented from video text lines are considered for character segmentation in the present work. Words can contain isolated or non-touching characters, as well as touching characters. Therefore, the character segmentation problem can be viewed as a two stage problem. In the first stage, text cluster is identified and isolated (non-touching) characters are segmented. The orientation of each word is computed and the segmentation paths are found in the direction perpendicular to the orientation. Candidate segmentation points computed using the top distance profile are used to find the segmentation path between the characters considering the background cluster. In the second stage, the segmentation results are verified and a check is performed to ascertain whether the word component contains touching characters or not. The average width of the components is used to find the touching character components. For segmentation of the touching characters, segmentation points are then found using average stroke width information, along with the top and bottom distance profiles. The proposed method was tested on a large dataset and was evaluated in terms of precision, recall and f- measure. A comparative study with existing methods reveals the superiority of the proposed method.**

Keywords: Video Document Analysis, Video Character Segmentation, Multi-oriented Document Processing, Video Character Recognition, Piece-wise Linear Segmentation Line (PLSL).

I. Introduction

Content-based indexing and retrieval of videos is becoming more essential due to the increasing size of multimedia databases. Text present in videos plays an important role in video indexing and retrieval and video text has been classified into two groups [2], [3] namely *'Scene text'* (e.g. text on vehicles, commodities, buildings, sign boards on roads, etc.) and *'Graphic text'* or *'Caption text'* (news video, sports video, etc). Hence both types of text can be extremely useful in the effective indexing and retrieval of the videos. The major challenges in text information extraction from video are low resolution, complex/non uniform backgrounds and blur, to mention a few. Though the text extraction step segments the text region, but still it may contain considerable amounts of non-text portions, which consequently hampers correct Optical Character Recognition (OCR). Minimizing the effect of non-text background has a high potential in improving the OCR results. Hence, word and character segmentation aims to refine the actual text area by reducing the non-text background. Word segmentation actually divides the text line/region into smaller regions, which in turn reduces the interaction with the background noise. Whereas, character segmentation reduces the words into further smaller regions comprising of single characters, thereby further minimizing the non-text portion. The segmented characters are then sent to the OCR engine for recognition. The problem becomes more challenging when the words are multi-oriented in video. Hence in this paper, a new method for character segmentation from multi-oriented video words is proposed. Words extracted from multi-oriented and horizontal straight lines from video were considered for experiments.

A comprehensive survey of character segmentation techniques was presented by Casey and Lecolinet [1]. Three main approaches where mentioned in the survey, namely, dissection-based, recognition-based and holistic segmentation. Recently a few techniques for character segmentation from video frames have been reported in the literature [7], [9], [8]. Phan et al. [7] proposed a Gradient Vector Flow (GVF) based technique for video character segmentation. The authors used GVF for the identification of cut candidates and framed character segmentation as a minimum cost path finding problem. The input image is considered as a graph where pixels are considered vertices connected to their neighboring pixels. Rajendran et al. [9] used Fourier-Moments features to extract the words and characters from video text lines. The authors relied on the feature which specified that the text height difference at the character boundary column is smaller than the other columns, but the same may not be true in cases where the text is slanted or is in italics. The same idea was also used by Shivakumara [8] to segment characters, but the authors used gradient-based features with Max-Min clustering to obtain the text clusters.

The presence of touching characters due to poor resolution, blur and background noise is a major bottle neck in character segmentation and OCR of video text. Hence, a two-stage approach is proposed, wherein for the stage-I, the isolated or non-touching characters are segmented and subsequently stage-II concentrates on touching character segmentation. Inspired by the work reported in [7], where the character segmentation was formulated as a minimum cost path finding problem, we propose the use of Piece-wise Linear Segmentation Lines (PLSL) for segmenting characters in both stages. Distance profile features are used in stage-I to estimate the candidate segmentation points. Distance profile features along with stroke width and average width of the characters are used to determine the candidate touching segmentation points. The main advantages of the proposed method is that it not only allows a curved segmentation path when required (in case of multi-oriented and touching character) but also does not require any thresholds in order to classify the character gaps. Unlike [7] where a cost function is used to guide the segmentation path, the text and non-text clusters are used as guides for the PLSLs in

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.90

the proposed method, which is computationally less expensive than the use of a cost function. In the present work, candidate segmentation points are found rather than trying to find the segmentation path at a fixed interval, as proposed in [7]. The use of fixed interval to find the segmentation path between characters in [7] resulted in more false positives, which in turn creates an overhead for a dedicated false positive elimination step. Whereas, in the proposed method false positives seldom occurred in stage-I and considerably less occurred during touching character segmentation.

The rest of the paper is organized as follows. The proposed character segmentation method is detailed in Section II. Section III presents the experimental results, comparative study with the existing methods and discussion on the results obtained as well as failure cases. Section IV concludes the paper providing the scope for future work.

II. Proposed character segmentation methodology

The word segmentation method proposed in [4] was used to segment the words from a video text line detected using [5]. The resulting word image is the input for the proposed character segmentation method. The proposed method which consists of two stages is discussed in the sub-sections given below. The method for candidate text cluster selection is described in Section IIA. The candidate segmentation point selection and character segmentation technique are described in the sub-section IIB. A detailed description of the touching character segmentation and false positive elimination techniques are given in sub-section IIC. A high level overview of the proposed method is presented in the flow chart shown in Figure 1.

A. Candidate text cluster selection

The input word image is a color image having a non-uniform background with noise. Hence, it is essential to identify the text and non-text pixels. In order to over come the problem with non uniform backgrounds a Difference of Gaussian filter is applied to remove the low frequency background blobs. The text/non-text clustering is performed using min-max clustering [8]. The minimum and maximum gray values are calculated for min-max clustering. Once the minimum (C_{min}) and maximum (C_{max}) clusters are found, a text cluster is then identified. It is generally observed that the pixels near the image border or outside the minimum bounding box of the word belong to the background cluster. Hence, the number of pixels min_{border} belonging to the C_{min} and max_{border} to C_{max} cluster are calculated considering the border rows and columns of the word image.

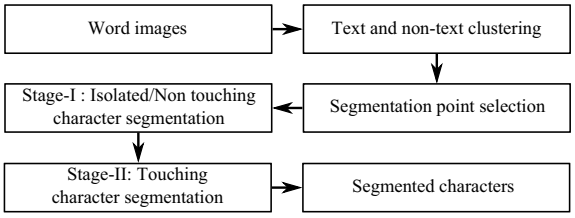


Fig. 1: Flow diagram of the proposed character segmentation method

If $min_{border} > max_{border}$, then C_{min} is considered as the background cluster else C_{max} is consider as the background cluster. Once the text cluster is found Connected Component Analysis (CCA) is applied, and the very small components are removed as they usually represent the background noise. The sample word images and the identified text clusters are shown in Figure 2. Both clusters are used to segment characters in the later steps of the method.

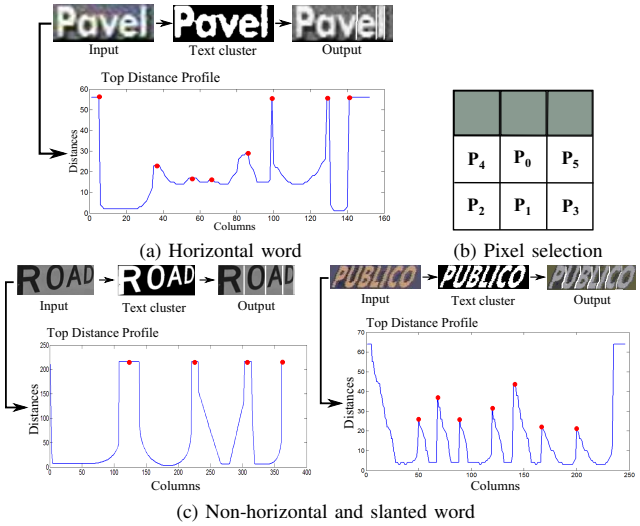
B. Stage-I: Non-touching character segmentation

Stage-I of the proposed method concentrates on segmenting non-touching characters. Text and non-text clusters identified are used as guides to find the piece-wise linear segmentation line between characters. The following are the steps to segment the isolated characters.

1) Candidate segmentation point selection: In-order to find the candidate segmentation points, orientation of the text cluster is first estimated followed by the estimation of the top distance profile. The orientation of the word is calculated using a PCA-based approach [6]. For computational simplicity, words having orientation less than a threshold T_{deg} (derived empirically and considered as 15°) are considered as horizontal otherwise multi-oriented. As multi-oriented words are considered for the experiments, the candidate segmentation points are found in the direction perpendicular to the orientation of the word, instead of rotating the word image to make it horizontal. To avoid an extra operation for rotation and its drawbacks, the multi-oriented words were processed in their original form. In case of multi-oriented words, the minimum bounding box is considered and a distance profile calculated in the direction perpendicular to its orientation.

Considering horizontal word as an example, for each column in the word image, the top distance is the distance between the topmost pixel and the first text pixel in the column. A sample top distance profile graph is shown in Figure 2(a). It can be seen that the graph contains many local peaks, some of them represent clear gaps between characters as the distance is equal to the height of the word. In the case of slanted or italic words it is not possible to find a distance peak which is equal to the height of the word. In such cases the distance is less than the height but there are high peaks between the characters. The top distance profiles shown in Figure 2(c) explains the scenario of non-horizontal and slanted words. Once the top distance profiles are calculated, the local peaks (marked in red) as shown in Figure 2, are found and considered as candidate segmentation points.

2) Character segmentation: The candidate segmentation points found are used to segment characters using the piece-wise linear segmentation lines. The proposed method is inspired by [7], where the character segmentation problem is formulated as a minimum cost path finding problem. Using a cost function and calculating the cost at each instance is computationally expensive, hence non-text clusters are considered in the proposed method. Starting from a candidate point, the method tries to move towards the other boundary of the word image using the non-text (C_{NT}) cluster. If the orientation of the word is horizontal, the bottom five neighboring pixels as shown in Figure 2(b) i.e. pixels $p_1(x+1, y)$, $p_2(x+1, y-1)$, $p_3(x+1, y+1)$, $p_4(x, y-1)$ and $p_5(x, y+1)$ are considered to find piece-wise linear lines. In the case that all five pixels



(c) Non-horizontal and slanted word

Fig. 2: Example of character segmentation in Stage-I

belongs to the text cluster, the path is marked as 'blocked'. If a neighbor pixel is found, which belongs to the non-text cluster, the pixel is marked as a starting pixel and the update operation is repeated. For example, let $P_s(x, y)$ be the starting pixel belonging to the non-text (C_{NT}) cluster and $P_n(x', y')$ be the neighboring pixel belonging to the set $NP = \{p_1, p_2, p_3, p_4, p_5, p_5\}$, which also belongs to the non-text cluster. Let $CPLS(i, j)$ be the candidate segmentation line vector,

If $P_n \epsilon NP$ and $P_n \epsilon C_{NT}$, then $P_s = P_n$, i.e. P_s is updated with P_n and is added to $CPLS(i, j+1) = P_n$, where, 'j' is a pixel in the i^{th} PLSL. The same rules are applied to the new pixel (P_s) and the process is repeated until the opposite boundary of the word is reached (marked as 'success') or the path is marked as 'blocked' as the path fails to reach the opposite boundary. The neighboring pixels corresponding to the direction perpendicular to the orientation of the word are considered in the case of multi-oriented words.

An important characteristic of using piece-wise linear lines is it produces straight segmentation lines instead of curved paths as proposed in [7]. In the case of horizontal words which are not slanted, PLSLs are nearly vertical, whereas in the case of slanted or multi-oriented words, the PLSLs are curved when required. The above process is applied to all the candidate segmentation points and the corresponding successful segmentation paths as well as the blocked paths are noted.

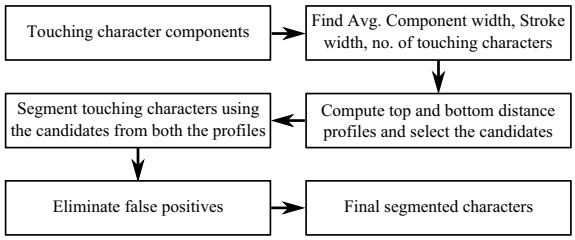


Fig. 3: Flow diagram for touching character segmentation

C. Stage-II: Touching Character segmentation

The presence of touching characters is a more challenging problem to handle character segmentation from video words. Hence, a new technique is proposed here to segment the touching characters in a video frame. A flow diagram of the same is given in Figure 3.

First, the touching component is identified using the average width (Wd_{av}) of the characters. The average width is found by dividing the width of the word by the total number of text components. The text components having width more than Wd_{av} are marked as touching components and considered for touching character segmentation. The approximate number of touching characters (Tc_{av}) is also estimated by dividing the width of touching component by Wd_{av}. Hence, the total number of expected segmentation lines should be $(Tc_{av}) - 1$. Upon careful examination of the touching components, it was found that the touching parts between the characters could exist at any location of the character boundary. But the length of the touching portion is usually less than the stroke width (Sw) of the characters. The stroke width (Sw) was calculated using the method proposed in [10]. Hence, the stroke width along with the top and bottom distance profiles are used to segment the touching characters.

The top profile of the touching component is calculated in the same way as discussed in Stage-I. The bottom profile is also calculated in a similar way. The top and bottom profiles of a sample touching character are shown in Figure 4. Here the bottom profile is also considered because touching parts can exist at any portion of the character boundary. The highest (Tc_{av}) number of peaks is selected from both top and bottom profiles, which are considered as the candidates for touching character segmentation. Starting from a candidate point which is placed just outside the text cluster, the possible segmentation paths are found as follows:

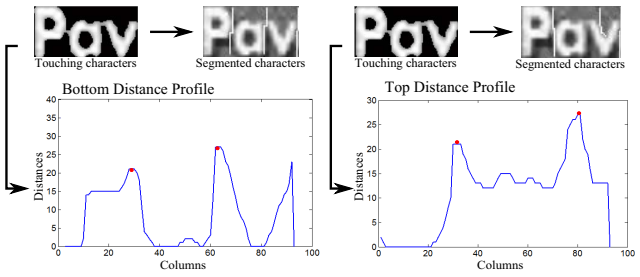
1. Move in three direction i.e. $p_1(x+1, y)$, $p_2(x+1, y-1)$ and $p_3(x+1, y+1)$ through the text cluster individually till a background cluster pixel is reached. Note the corresponding distances to reach the background pixel, dp_1, dp_2, and dp_3, respectively.
2. If any of the distances in the order dp_1, dp_2, and dp_3 are less than Sw the path through the text cluster is considered as the touching component segmentation path. An example of the same is shown in Figure 4.
3. Repeat the operation for both top and bottom candidate touching segmentation points.
4. As both the top and bottom profiles are considered, false positives exist and should be eliminated. The following criteria are used for false positive elimination.

(i) Consider the first segmentation point which is away from the left boundary by a distance approximately equal to Wd_{av}. In the case of more than one touching characters the next touching point should also exist at a distance nearly equal to Wd_{av} from the first segmentation point and so on.

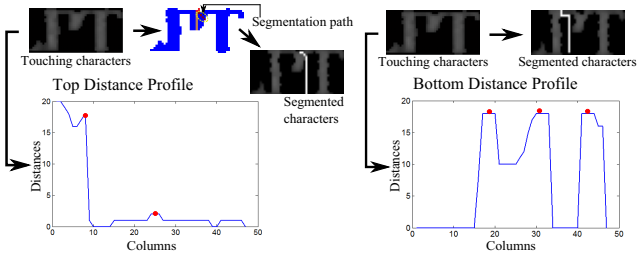
(ii) Apply the above rule to both the top and bottom profile's successful candidate touching points. If both the top and bottom profile candidates satisfy the criteria, consider the top profile candidate segmentation line as final.

(iii) In the case of multiple touching characters, if the top and bottom candidate points partially satisfy first criteria, then consider a combination of both the profile candidate points. For example, if there are three characters in a touching component,

the first candidate from the top profile follows the first criteria and the other candidates do not. Whereas, the second candidate from bottom profile satisfies the rule with respect to the first candidate point from the top profile, then it is considered as the second touching segmentation point. Illustration of the touching character segmentation technique is shown in Figure 4. From Figure 4(a) it can be clearly seen that 'PAV' is a touching string, hence each distance profiles are computed as shown in the figure. As there are three characters in the string, two PLSLs are required to segment them. In the bottom distance profile the two highest local peaks are marked in red. Both the peaks in the bottom profile are at a distance nearly equal to Wd_{av}. Whereas, the peaks in the corresponding top profile have a distance greater than Wd_{av} between them. The first peak in both the profiles is nearly in the same column and correctly segments 'P' and 'a', but the second peak in the top profile doesn't satisfy both Wd_{av} and the Sw criteria. Hence it is discarded. Whereas the second peak in the bottom profile satisfies both the criteria and successfully segments 'a' and 'v'. The other example in Figure 4(b) is an example of touching characters where a slanting segmentation path is found in the text cluster. The vertical segmentation path through the text cluster marked in dark yellow of the top profile, doesn't satisfy the stroke width rule. Hence the slanting paths in the direction of P_3 and P_1 were explored. The path in the direction of P_3 follows the stroke width rule and was considered as the touching segmentation path from the top distance profile peak which successfully segmented 'P' and 'T'. None of the peaks in the bottom profile of Figure 4(b) satisfied the Wd_{av}, and were thus discarded.



(a) Example 1: Segmentation using vertical path



(b) Example 2: Segmentation using slanted path

Fig. 4: Example of touching character segmentation in Stage-II

III. Experimental Results

The same dataset employed by the authors in [7] was used for our experiments to have the comparative idea. The dataset comprised of 700 words (3527 characters) which was divided into four subsets: English horizontal (200 images), English non-horizontal (100 images), Chinese horizontal (200 images) and Chinese non-horizontal (100 images). The dataset includes various types of multi-oriented images, with various levels of background noise. The detailed results are discussed below.

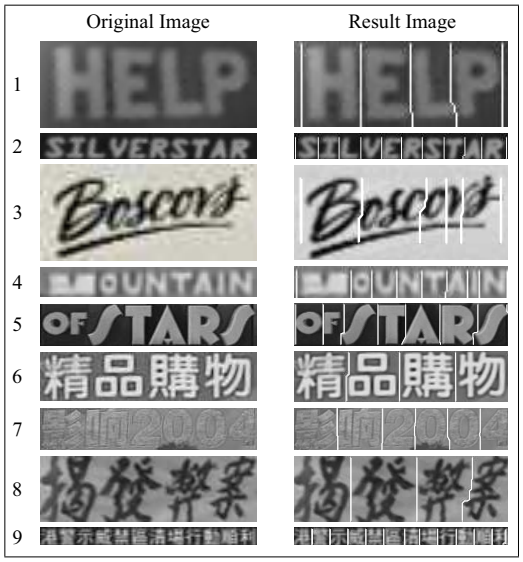


Fig. 5: Example of some segmentation results

A. Sample segmentation results

Some samples of segmentation results are shown in Figure 5. The first column shows some of the original images of both English and Chinese words, and the second column shows the results of character segmentation obtained by the proposed method. It can be seen that even though the first and fourth images suffer from blur and low resolution, the proposed method segmented all the characters correctly. Whereas in the third image it misses the segmentation of 'OSC', as the word is written in italics (hand written fashion) where all the three characters are connected and the touching portion has nearly the same stroke width as that of the characters. The seventh image is a combination of both Chinese and English characters, with low resolution and strong interaction with the background. The proposed method worked well and segmented all the characters. It can be seen in the sample images that PLSLs tries to segment characters using vertical straight lines and also it uses curved lines when required.

B. Segmentation accuracy and comparative study

Recall(R), Precision(P) and f-measure(F) were used as performance measures based on *Actual Cuts(AC), True Cuts(TC)* and *False Cuts(FC)* detailed in [7]. The performance measures [7] are defined as follows,

1) $R = TC/AC$
2) $P = TC/(TC + FC)$
3) $F = (2XPXR)/(P + R)$

We compared the proposed work with the method proposed by Trung et al. [7] and Shivakumara et al. [8]. The first two rows of Tables I and II show the performance of the proposed method on English horizontal and non-horizontal

and Chinese horizontal and non-horizontal words, respectively. Whereas the last two rows shows the performance reported by [7] and [8], respectively on the same dataset. The proposed method achieved a higher recall rate and f-measure value than both the results shown in [7], [8] for the English dataset in Stage-II. This signifies that the proposed method segments the relevant characters quite well. The precision achieved for English Horizontal dataset was a little less than that of [7], but is still comparable because the precision achieved for all other datasets was higher than the precision achieved by [7] and [8]. Due to low resolution, blur and noise there was confusion in the estimation of touching portion for some images (refer Figure 6) in stage-II. This resulted in the decrease of recall rate during the stage II for English Horizontal and Non-Horizontal dataset.

The performance on the Chinese horizontal dataset reported in Table II shows that higher precision and f-measure was achieved. Whereas, the recall rate was the same as that achieved by [7]. Higher precision achieved for both Chinese Horizontal and Non-Horizontal datasets shows that the characters where segmented with higher confidence. In Chinese Non-Horizontal dataset the spacing between characters may not be uniform as in English and single component sometimes has many sub-components. Hence, the algorithm gets confused and resulted in a lower recall rate than [7]. Overall the accuracy achieved was promising and comparable with the existing methods which have used the same dataset for the experimentation.

TABLE I: Performance on English dataset

Method	English Horizontal			English Non-horizontal		
	R	P	F	R	P	F
Proposed (Stage-I)	0.97	0.84	0.90	0.95	0.82	0.88
Proposed (Stage-II)	**0.96**	**0.87**	**0.91**	**0.94**	**0.88**	**0.91**
Trung et al.[7]	0.89	0.91	0.90	0.91	0.85	0.88
Shivakumara et al. [8]	0.93	0.73	0.82	0.82	0.77	0.79

TABLE II: Performance on Chinese dataset

Method	Chinese Horizontal			Chinese Non-horizontal		
	R	P	F	R	P	F
Proposed (Stage-I)	0.95	0.84	0.90	0.87	0.76	0.81
Proposed (Stage-II)	**0.95**	**0.86**	**0.91**	**0.88**	**0.79**	**0.83**
Trung et al.[7]	0.95	0.81	0.87	0.96	0.74	0.84
Shivakumara et al. [8]	0.93	0.69	0.79	0.83	0.78	0.80

C. Failure cases

Failure cases were examined thoroughly and it was found that the major reason for incorrect or under-segmentation was the poor quality of the images. Low resolution, blur, and particularly the interaction of various noisy background with text, contributed substantially to the failure. A few samples of failure cases obtained from the experiment are shown in Figure 6. The first image in the Figure 6 suffers from strong background interaction even though the text is reasonably clear. Hence, in that case under-segmentation occurs and the characters '*getp*' were not segmented. The second and third images suffer from both a complex background with non-uniform illumination and blur. The result of the second image shows both incorrect and under-segmentation. Whereas, for the third image no segmentation line was found as all the characters seem to be highly connected due to background noise and blur.

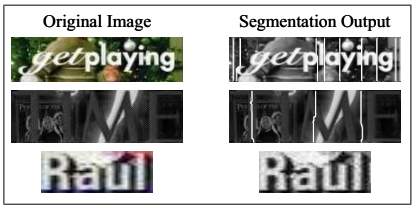


Fig. 6: Example of some failure cases

IV. Conclusions and Future Work

A new method for multi-oriented video character segmentation has been proposed in this paper. The proposed method is a two-stage approach, where in the first stage the isolated (non-touching) characters are segmented, and in the second stage the touching characters are segmented. Piecewise Linear Segmentation Lines (PLSL) are used for character segmentation, which allows both vertical and partially curved segmentation paths based on the orientation and slant of the word. A new technique based on the stroke width, distance profiles and average width of the characters was used for touching character segmentation, which is novel for character segmentation in video. Experiments were performed on a large dataset comprising both English and Chinese horizontal and non-horizontal words. The proposed method performed well and promising results were obtained which are competitive with the state of the art methods. The future plan includes improving the text clustering technique in order to get a better basis for the latter steps of the approach and to effectively consider more complex shaped curve word segmentation.

References

[1] R.G. Casey and E. Lecolinet, *A survey of methods and strategies in character segmentation*, IEEE PAMI, Vol 18(7), pp.690-706, 1996.

[2] N. Sharma, U. Pal, and M. Blumenstein, *Recent Advances in Video Based Document Processing: A Review.*, DAS, pp.63-68, 2012.

[3] K. Jung, K.I. Kim and A.K. Jain, *Text information extraction in images and video: a survey*, Pattern Recognition, pp.977-997, 2004.

[4] N Sharma, P Shivakumara, U Pal, M Blumenstein and C L Tan, *A New Method for Word Segmentation from Arbitrarily-Oriented Video Text Lines*, DICTA, pp 1-8, 2012.

[5] N. Sharma, P. Shivakumara, U. Pal, M. Blumenstein, C. L. Tan, *A New Method for Arbitrarily-Oriented Text Detection in Video*, DAS, pp.74-78, 2012.

[6] Y.S. Lee, H.S. Koo and C.S. Jeong, *A straight-line detection using principal component analysis*, Pattern Recognition Letters, 27, pp. 1744-1754, 2006.

[7] T. Q. Phan, P. Shivakumara, B. Su, C. L. Tan, *A Gradient Vector Flow-Based Method for Video Character Segmentation*, ICDAR, pp. 1024-1028, 2011.

[8] P. Shivakumara, S. Bhowmick, B. Su, C. L. Tan, U. Pal, *A New Gradient based Character Segmentation Method for Video Text Recognition*, ICDAR, pp.126-130, 2011.

[9] D. Rajendran, P. Shivakumara, B.Su, S. Lu, C. L. Tan, *A New Fourier-Moments based Video Word and Character Extraction method for Recognition*, ICDAR, pp.1165-1168, 2011.

[10] B. Epshtien, E. Ofek, Y. Wexler, *Detecting text in natural scenes with Stroke Width Transform*, CVPR, pp. 2963-2970, 2010.