A New Wavelet-Laplacian Method for Arbitrarily-Oriented Character Segmentation in Video Text Lines

Guozhu Liang¹, Palaiahnakote Shivakumara², Tong Lu¹, and Chew Lim Tan³
¹National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China
²Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
³School of Computing, National University of Singapore, Singapore
597087208@qq.com; hudempsk@yahoo.com, lutong@nju.edu.cn; tancl@comp.nus.edu.sg

Abstract—Character segmentation is an important topic to improve the overall performance of text recognition methods especially for recognizing arbitrarily-oriented text lines, which is considered a difficult problem due to low resolution, complex background and lots of visual variations in video. This paper presents a novel idea for segmenting characters from arbitrarily-oriented text lines based on wavelet and Laplacian combination. Firstly, we explore wavelet which decomposes a given input image into sub-levels like a pyramid structure for segmenting words based on the fact that as decomposition level increases, the gap between the characters decreases due to the reduction in the size of the input image, which results in a single component for each word. Secondly, for each segmented word, we propose the Laplacian wavelet combination in a new way to extract text candidates. Thirdly, we propose horizontal and vertical sampling for character segmentation from the words. The proposed method is tested on curbed, non-horizontal and horizontal text lines of video and the ICDAR 2005 natural scene dataset to evaluate the performance. A comparative study with an existing method shows that the proposed method outperforms it in terms of precision and f-measure.

Keywords: Wavelet image decomposition, Wavelet image pyramid, Word segmentation, Laplacian-wavelet, Horizontal and vertical sampling, Character segmentation.

I. INTRODUCTION

It is noted from the literature [1] that character segmentation or detection in text lines of video or natural scene images plays a vital role in enhancing the overall performance of text detection and recognition methods. This is because character segmentation is an integral part of text detection and recognition, which increases the discriminative power of feature extraction by reducing the influence of complex background [2]. However, segmenting characters from text lines in video or natural scene images is not as easy as segmenting characters from scanned or camera based document images because generally video and natural scene images suffer from low resolution and complex background, font or font size variations, and motion blur, in contrast to document images where we can see plain background without many variations in fonts [3]. In addition, arbitrary orientations of text lines make the problem more challenging and interesting [3]. For example, it is stated in [1] that character segmentation requires much attention of researchers in order to achieve good recognition rates at character and word levels because matching the extracted features depends on character shapes. If a method is not able to segment characters with clear shapes, the extracted features probably produce more false positives during recognizing. The same conclusions can be drawn from the Yao et al. method [2], where a unified framework for multi-oriented text detection and recognition is proposed using component linking and word partition process. The same thing is true for text detection. Zhang et al. [4] proposed a novel text detection system based on character and link energies. This method gives good results if it is able to separate characters in text lines because its feature extraction depends on character shapes. Kang et al. [5] proposed an orientation robust text line detection method in natural scene images based on maximally stable extremal regions and clustering. It is known that maximally stable extremal region expects good shapes of characters. Otherwise, it produces multiple sub-components for one character component [6, 7].

To solve the character segmentation problem, there are methods available in the literature. For instance, Rajendran et al. [8] proposed a Fourier-moment based method for word and character segmentation from text lines, which uses clustering and run-length criteria to segment words, and text height difference at character boundary for segmenting characters. The main scope of this method is to segment words and characters from horizontal text lines but not arbitrarily oriented text lines. Phan et al. [9] proposed a gradient vector flow based method for video character segmentation. Gradient vector flow is used for identifying seed points and the least cost path is estimated for segmenting the characters. Though the method works well for complex background text lines without segmenting words, it is sensitive to seed point selection. Sharma et al. [10] proposed the combination of profile based features for seed point guessing and cost path for character segmentation from multi-oriented text lines. Since the method uses profiles based features, it is good for multi-oriented text lines but not those curve shaped texts.

In this paper, we propose a novel method that explores wavelet decomposition for word segmentation by shrinking the characters of words into a single component. This is valid because as wavelet decomposition level increases when the size of the image decreases. This results in shrinking characters of words into one single component. For the segmented words, we propose a new combination of Laplacian-wavelet as inspired by the work proposed in [11] for text detection in video, where the Laplacian-Fourier combination has been used for text candidates detection. Motivated from the word segmentation proposed in [12] for video text lines of any orientation, we...
explore the same idea of horizontal sampling and vertical sampling for character segmentation from the words in Laplacian-Wavelet domain. In this way, the proposed method is a departure from the existing methods.

II. PROPOSED METHOD

We use the text detection method in [13], which works well for arbitrarily-oriented texts in video for text line extraction from input video images. Therefore, for this work, text lines of any orientation are inputted for character segmentation. The proposed method consists of three stages: (1) Word segmentation from the input text lines by exploring wavelet decomposition. Based on our experiments, it is observed that as the decomposition level goes up, the gap between characters reduces because of the reduction in the size of the image. It is true that the spacing between words is greater than that between characters in text lines. It is also true that character boundaries have a high contrast compared to the spacing between words. When the decomposition level goes up, the high contrast of boundaries spreads over the space between characters or words. Since the gap between characters is small compared to the gap between words, the influence of character boundaries over the space results in connecting neighboring characters, which in turn results in the whole word as a single component. Due to the fact that the space between words is larger compared to that between characters, words would not merge with neighboring ones.

This observation motivated us to explore wavelet decomposition for word segmentation. Then with the help of k-means clustering, the proposed method segments the words. For the segmented word, the proposed method further introduces the Laplacian-Wavelet combination for text candidate detection. It is noted from the work [11] that the combination of Laplacian-Fourier has been used successfully for detecting text candidates from multi-oriented text lines in video, we thus propose the combination Laplacian with Wavelet instead of Fourier for detecting text candidates. Additionally, as motivated by the work presented in [12] for word segmentation from arbitrarily-oriented text lines by introducing horizontal and vertical sampling, we explore the same idea for segmenting characters by giving words as the input in Laplacian-Wavelet domain. In this way, the proposed method contributes to achieving good results for character segmentation on arbitrarily-oriented text lines in video and natural scene images as well.

A. Wavelet Decomposition Levels for Word Segmentation

For each text line detected by the text detection method as shown in Fig 1(a), the proposed method deploys wavelet decomposition (Haar wavelet) to obtain its sub-bands, such as LL, LH, HL, HH for the first level as shown in Fig. 1(b), where we can see LL provides the text line with reduced size, LH, HL, and HH highlights horizontal, vertical and diagonal information of the input text line, respectively. Fig. 1 shows that when we apply wavelet decomposition, LL provides the image with a reduced size compared to the original size of the input image. As a result, the neighboring pixels get merged. This cue motivated us to use decomposition levels for merging character components of words into a single component. Since the space between characters is smaller than that between words, when we increase the decomposition levels, the character components are merged into a single component at a certain level of decomposition. This is illustrated in Fig. 2, where (a) is the input curved text line, (b)-(d) are the results of LL for the decomposition level-1, level-2 and level-3, respectively, and (e) is the magnified result of level-3 in (d). It is clearly observed from Fig. 2(e) that the characters of the words are merged into a single component, which means we get three word components that look like three patches with clear spacing between the words for the input image in Fig. 2(a). This is the main advantage of the proposed method for segmenting words, which is independent of scripts, orientation etc. and does not require any features or rules to identify the space between words as the existing methods proposed in the literature [8, 12, 14].

It is true that as the height of decomposition level increases, the adjacent pixels get merged into a single component, but the question is how many levels are to be used when the size of the input image changes. For the above case shown in Fig. 2, three levels are sufficient to segment the words. To overcome this problem, we apply k-means with k=2 clustering on LH and HL to obtain the cluster which represents high frequency values called candidate pixels. Then we study the distribution of the candidate pixels in LH and HL as decomposition level goes up. The distribution of candidate pixels looks sparse as the number of decomposition levels increases because of the reduction in the number of pixels. With this observation, we define the ratio of the number of the
candidate pixels in LH and HL of the previous level with the number of the candidate pixels in LH and HL of the current level should satisfy a certain threshold. The value of the threshold is determined based on experimental study. It is illustrated in Fig. 3, where (a)-(c) show candidate pixels of LH and HL at level-1, level-2 and level-3, respectively. It is observed from Fig. 3 that the number of candidate pixels decreases as the number of levels increases.

![Candidate pixels of LH and HL at Level-1](image1)
![Candidate pixels of LH and HL at Level-2](image2)
![Candidate pixels of LH and HL at Level-3](image3)

**Fig. 3: Candidate pixel distribution as decomposition level increases**

Once we determine the number of levels for the input text line image, we consider the results of LL of the top level for word segmentation. Before word segmentation, the proposed method performs smoothing operation over the LL image to equalize the values of pixels, which results in filling small gaps based on neighbor information. Then the proposed method applies k-means clustering with k=2 on the smoothed LL image to extract word components as shown in Fig. 4, where (a) is the result of k-means clustering on the smoothed LL image, and (b) and (c) respectively show the results of word segmentation for the curved and horizontal text lines.

![Sample word segmentation](image4)

**Fig. 4: Sample word segmentation: (a) Word patches after smoothing by k-means clustering, (b) Word segmentation results for curved text line and (c) Word segmentation results for horizontal text line**

### B. Text Candidates Detection for the Segmented Words

As motivated by the work presented in [11] which successfully combines Laplacian with Fourier for text candidate detection in video, we propose to combine Laplacian with wavelet high frequency sub-bands, namely, LH, HL and HH. The proposed method convolves the Laplacian masks of different directions, such as horizontal, vertical and diagonal with sub-band images in frequency domain, resulting in three enhanced images. For instance, the horizontal Laplacian mask with its convolution with LH sub-band is shown below:

\[
\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)
\]

We define the mask as \(W^1\text{Lap}\). In the same way, the proposed approach obtains vertical and diagonal enhanced images.

The main reason to choose different directions of Laplacian masks to convolve with sub-bands is to handle arbitrary orientation of words. The inverse transform is used to convert frequency images to spatial ones. It is true that the obtained three enhanced images have high values for text pixels and low values for non-text pixels because of the advantage of the combination on Laplacian and wavelet sub-bands. To separate text pixels from non-text ones, we propose to use k-means clustering with k=2. The result of k-means clustering on three enhanced images can be seen in Fig. 5, where we can see that non-text pixels are eliminated while text pixels are retained for all the three enhanced images. Further, the results of LH, HL and HH are fused through an “AND” operation to obtain text candidates for the word image shown in Fig. 5, where (a)-(c) denote the results of k-means clustering on LH, HL and HH, respectively, and (d) denotes the final fused image where we can see most of text pixels are retained while non-text pixels are discarded.

![Text candidates for the word](image5)

**Fig. 5: Text candidates for the word**

### C. Character Segmentation

We noted from the work proposed in [12], where zero crossing points are explored for word segmentation from video text lines. Inspired by the work, we propose to explore the same zero crossing points for character segmentation in this work from the words. For each word, we set a rectangle sliding window with its height equals to the height of the word block, and its width equals to the half of the height. The sliding window follows the direction of word block, and goes ahead 1/4*width pixels in each movement. For each sliding window, we calculate the percentage of zero crossing points. It is true that when there is a text, we get a high percentage of zero crossing points, while when there is no text, we get a low percentage of zero crossing points because of more transitions from the background to the foreground of text. In order to select the window which represents the probable spacing between characters, we again apply k-means clustering on the percentages of zero crossing points of sliding windows. The cluster which represents a low percentage of zero crossing points is considered as the window which represents probable spacing between characters as shown in Fig. 6, where (a) is the input word and (b) shows the results of the text candidates.
given by the previous section, and (c) shows probable spacing given by k-means clustering. It is observed from Fig. 6(c) that the probable spacing looks like actual spacing between characters. However, this may not be true for all the cases due to low resolution and complex background. To tackle such a problem, we choose the seed window which gives the lowest percentage of zero crossing points from the probable spacing windows given by k-means clustering as shown in Fig. 7(a).

For the seed window, we propose horizontal and vertical sampling as proposed in [12] for character segmentation in this work. The process of horizontal sampling is to reduce the window size pixel by pixel horizontally from the left to the right side of the seed window until it reaches no more reductions. For every reduced size window, the proposed method calculates the percentage of zero crossing points and compares the percentage of zero crossing points of the previous window before allowing the reduction process. It is expected that the percentage of zero crossing points of the current window must be higher than the percentage of the zero crossing points of the previous window as the window covers more space when it reduces horizontally. As a result, this process gives us the width of character spacing as shown in Fig. 7(b). The same width is verified with the other window, which represents the probable spacing between characters to identify the actual spacing between characters as shown in Fig. 7(c), where we can see for this example, the method identifies spacing between characters correctly without over-segmentation of false segmentations. However, since video and natural scene images usually have complex background, horizontal sampling may still give false segmentations.

To solve this problem, we further propose vertical sampling in the same way as horizontal sampling for finding the height of the seed window. Instead of reducing the size of the seed window horizontally, the proposed method reduces the size of the window vertically. The final results of vertical sampling are shown in Fig. 7(d), where we can see all the characters are segmented properly.

### III. EXPERIMENTAL RESULTS

We consider the dataset that was used for evaluating word segmentation in [12] for evaluating character segmentation in this work. In addition, we also consider the ICDAR 2005 dataset [15] as the benchmark data, as well as to represent natural scene data. We use the same measures as the method used [12], such as recall, precision and f-measure for measuring the performances of the proposed method. Our dataset include 70 arbitrarily oriented text lines from video frames (including curved texts but excluding non-horizontal and horizontal straight lines), 325 non-horizontal text lines, 1047 horizontal text lines, and 93 text lines from publicly available Hua’s data [16] and 125 text lines from the ICDAR-2003 competition dataset. In summary, 1535 (70+325+1047+93) video text lines and 125 text lines from camera images are used for the purpose of experimentation.

To show the superiority to the existing method, we implement the method [9], which is developed for character segmentation from text lines in video without word segmentation by using Gradient vector flow and cost estimation for comparative studies. We use the following definitions which are used in measuring the method:

- **Truly Detected Character (TDC):** A segmented block that contains correctly segmented characters.
- **Under Segmented Blocks (USB):** A segmented block which contains more than one character.
- **Over Segmented Blocks (OSB):** A segmented block which contains no complete character.

Since there is no ground truth available, we manually count the Actual Number of Character (ANC) in text lines, which are considered as ground truth for evaluation. We compute Recall(R), Precision(P), and F-measure(F) for evaluating the performances of the proposed method. These performance measures are defined as follows:

- **Recall (R) = TDC / ANC,**
- **Precision (P) = TDC / (TDC + USB+OSB),**
- **F-measure (F) = (2 x P x R) / (P + R).**

The qualitative results of the proposed method for character segmentation on curved, non-horizontal, horizontal data are shown respectively from (a)-(c) in Fig. 8, where one can notice that the proposed method segments characters correctly for different oriented text lines with different contrasts and background. The quantitative results of the proposed and the existing method [9] for all the datasets mentioned above are reported in Table 1, which shows that the proposed method is better than the existing method for all the

![Fig. 6: The probable window which represent spacing between characters](image)

![Fig. 7: Character segmentation for the curved word](image)
datasets in terms of precision and f-measure. This is mainly because the process of the minimum cost path finding in GVF does not work when it comes to segment the characters like “WXYVYH”. For these characters, the existing method suffers from under segmentation. In addition, the existing method is sensitive to seed points which are required for cost path estimation. On the other hand, the proposed method takes the advantage of wavelet decomposition for word segmentation, horizontal and vertical sampling for correct gap identification. It can be confirmed from Table 1 that the precision of the proposed method is higher than that of the existing method.

![Character segmentation for curved text lines](image1)

(a). Character segmentation for curved text lines

![Character segmentation for non-horizontal text lines](image2)

(b). Character segmentation for non-horizontal text lines

![Character segmentation for horizontal text lines](image3)

(c). Character segmentation for horizontal text lines

Fig. 8: Sample character segmentation of the proposed method for the different orientated text lines

Table 1: Performance of the proposed and existing methods on different datasets

<table>
<thead>
<tr>
<th>Data</th>
<th>Proposed Method</th>
<th>Existing Method [9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curved dataset</td>
<td>0.58 0.67 0.62</td>
<td>0.58 0.65 0.61</td>
</tr>
<tr>
<td>Non-Horizontal Straight Line</td>
<td>0.79 0.84 0.81</td>
<td>0.76 0.82 0.79</td>
</tr>
<tr>
<td>Horizontal Straight Line</td>
<td>0.86 0.90 0.88</td>
<td>0.81 0.87 0.84</td>
</tr>
<tr>
<td>Hua data</td>
<td>0.86 0.87 0.86</td>
<td>0.79 0.76 0.77</td>
</tr>
<tr>
<td>ICDAR 2003</td>
<td>0.86 0.89 0.87</td>
<td>0.78 0.84 0.81</td>
</tr>
</tbody>
</table>

**IV. CONCLUSION AND FUTUER WORK**

We have proposed a novel idea for segmenting characters from text lines of any orientation in video as well as natural scene images. The proposed method explores wavelet decomposition in a new way for segmenting words from text lines. The combination of Laplacian-Wavelet has been introduced for text candidate detection for word images. Furthermore, horizontal and vertical sampling concept is proposed for segmenting characters without over segmentation and under segmentation. Experimental results on different datasets show that the proposed method is superior to the existing method in terms of f-measure. We are planning to extend this idea for improving the accuracies for curved and multi-script data in future.

**ACKNOWLEDGEMENT**

The work described in this paper was supported by the Natural Science Foundation of China under Grant No. 61272218 and No. 61321491, and the Program for New Century Excellent Talents under NCET-11-0232.

**REFERENCES**


