USING 2D TENSOR VOTING IN TEXT DETECTION

Toan Nguyen, Jonghyun Park, and Gueesang Lee

Department of Computer Engineering, Chonnam National University, Korea

ABSTRACT

A novel text detection algorithm based on 2D tensor voting is proposed. Tensor voting is used to extract text line information by exploiting the curve saliency value and curve normal vector at each character. The text line information is useful to improve the results and reduce the effect of using heuristic rules of region-based methods. The experimental results attained from several natural scene images show that the proposed method successfully detects text with low false positive rate.

Index Terms— Tensor voting, text detection

1. INTRODUCTION

In recent years, a huge amount of still-images are created by digital cameras, often embedded in widespread mobile devices like cellular-phones. Text understanding in a natural scene image is very important for many purposes such as assistance system for impaired persons and text translation in foreign countries. Text detection is the first and critical step in text understanding. Since natural scene images can be captured under any conditions, they bring new challenges to correctly detect interested text regions. In natural scene images, the background may be very complex and the text may have different font styles, sizes, and orientations. Furthermore, to achieve good performance in text detection from these images, we need to overcome problems such as uneven illumination, reflection, shadows and highlights.

Text detection methods can be sorted into two main categories: texture-based and region-based. A complete survey of text detection applications and systems can be found in [1-2]. Texture-based approach including derivative and frequency methods usually requires highly computational complexity. Five localized features are proposed in [3] with a neural network classifier to locate regions of text in an image. The authors claim that their approach is invariant to scale and 3D orientation of the text and allows recovery of text in cluttered scenes. The frequency domain is also used such as Fourier transform [4], discrete cosine transform [5], and wavelet [6]. Since frequency domain methods are based on the fact that small texts produce strong texture response, they are only good for small characters.

Region-based approach is widely used because it is simple to implement and robust to illumination changes. Region-based approaches determine spatial cohesion based on edge features [7-8] or connected component features [9] of text strokes. In edge-based methods, the edges of the text boundary are identified and merged, and then several heuristics are used to filter out the non-text regions. The connected component based methods, on the other hand, use a bottom-up approach by grouping small components into larger components until all regions are identified in the image. In region-based approach, however, many non-text regions are misclassified as text regions because it does not consider the information in neighboring regions.

Previously proposed methods, especially in region-based approach, generate high false positive rate and require many well-defined heuristics depending on a specific application. In this paper, text line information is exploited by 2D tensor voting [10-11] to reduce false positive rate without using specific heuristic rules. The 2D tensor voting provides valuable information to locate characters and remove noises or to prevent the erroneous detection. Tensor voting generates curve saliency values and normal vectors at characters belonging to a text line. The text line is a virtual line that characters align on. The text line information can be used instead of specific heuristics to remove non-text regions in region-base methods. Thanks to tensor voting, our method can successfully detect text regions in many complex natural scene images.

The remainder of this paper is organized as follows. In section 2, we briefly introduce the 2D tensor voting framework. Our text detection method is represented in section 3. Experiment results are given in section 4. Finally, section 5 gives conclusions and future work.

2. DATA REPRESENTATION AND VOTING IN 2D

2.1. Data Representation

Tensor voting [10-11] is a unified computational framework for extracting perceptual structures from sparse, noisy data in 2D or 3D. The perceptual structures can be represented by their own properties. Points can simply be represented by their coordinates. Curve segments can be represented by their normal vectors. In tensor voting, all types of mentioned perceptual structures can be represented by tensors and inferred at the same time. In our application, each input token corresponds to a center point of a candidate character. This input token is represented in a form of the second order, symmetric, non-negative definite tensor. This kind of tensor can be stored in a 2 by 2 matrix and visualized by an
ellipse whose shape indicates the type of structure represented and its size the certainty of this information. Therefore, a ball tensor can represent for a pixel with no preference of orientation such as an isolate pixel or a junction of curves. A curve segment can be represented by a stick tensor with the biggest eigenvector is its normal vector.

Since we may not know types of input data, a token usually is represented by a general tensor which can be decomposed into stick tensor and ball tensor components corresponding to the first and second terms in the following equation.

$$T = \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T = (\lambda_1 - \lambda_2) \hat{e}_1 \hat{e}_1^T + \lambda_2 (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T),$$

where $\lambda_i$ are the eigenvalues in decreasing order and $\hat{e}_i$ are the corresponding eigenvectors.

### 2.2. Voting Process

Tensors communicate each other by voting process. In other words, tokens propagate their information such as their orientations to their neighbors. The shape and size of this neighborhood and the vote strength and orientation are encapsulated in predefined voting fields or kernels. Each type of feature requires a voting field. All voting fields can be generated from a fundamental stick voting field (Fig. 1). The voter O tries to cast its information to the recipient P. The normal vector at P is the normal vector of the most likely continuation curve from O to P. The strength of the vote at P is inversely proportional to the curve distance between O and P and calculated by the following decay function.

$$DK(s, k, \sigma) = e^{-(s^2 + ck^2)\sigma^2},$$

where $s=(\theta)/\sin(\theta)$ and $k=2 sin(\theta)/l$. The parameter $s$ is the arc length OP, $k$ is the curvature, $c$ is a constant, and $\sigma$ is the scale of voting field controlling the size of the voting neighborhood and the strength of votes. Each token in the voting domain receives several votes from its neighboring tokens. Vote accumulation is performed by tensor addition or equivalently by addition of 2 by 2 matrixes. Tokens that lie on the same smooth curve strongly support each other. After voting, saliency maps of data are computed. A saliency map shows the certainty of an interest kind of perceptual structure. In our application, curve saliency map and normal vectors at characters are computed to extract text regions.

![Fig. 1. Second order votes by 2D stick voting.](image)

### 3. TEXT DETECTION WITH 2D TENSOR VOTING

#### 3.1. System Overview

The flowchart of the proposed method is depicted in Fig. 2. The input and output of the system are a natural scene image and detected text region, respectively. If the input image is a color image, it is converted to grayscale. Candidate text regions are detected by a region-based method. Instead of heuristic rules, character center points extracted by tensor voting are used to remove non-text regions from these candidate text regions. By this approach, we not only avoid developing good heuristic rules but also exploit text line information to reduce false positive rate.

#### 3.2. Character Center Point Extraction

An edge image of the input image is created. The text line information including curve normal vectors and curve saliency values is then computed at center points of connected components. A connected component is a set of connected edge pixels in the edge image. Vertical edges of objects in the input image are generated by a Sobel vertical edge-emphasizing method which corresponds to the Sobel convolution matrix of

$$
\begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 
\end{bmatrix}
$$

Center points of all connected components are computed and used as input data for tensor voting. Tensor voting is applied to extract curve saliency values and curve normal vectors at tokens corresponding to these center points. The centre points of characters in a text region are usually close together and are mostly aligned on a line or a smooth curve. Therefore, the curve saliency of a token corresponding to a character centre point in a text region has higher curve saliency than that of a token in a non-text region. In
addition, the curve normal of a token in a text region indicates the normal vector of the text line. Since text is assumed to be horizontally aligned on a smooth curve or a line, candidates with low curve saliency values or nearly horizontal normal vectors are removed. The remaining character center points are used with a region-based method to detect text regions.

Fig. 3 shows an example of using tensor voting to extract the text regions from the input image in Fig. 3 (a). In Fig. 3 (b), connected components are represented by white pixels with center points in red color. After tensor voting, each token is assigned a normal vector (green line) that indicates the curve normal (vector direction) and curve saliency (vector magnitude) at this token. Remaining character center points are shown in Fig. 3 (c). Since with only this process, some non-text objects may be included and boundary boxes of characters are not well detected, region-based method is combined in next sections.

3.3. Candidate Text Region Detection

In this Section, a simplified region-based method is applied to detect candidate text regions. The Canny edge detection algorithm is used to generate an edge image from the input image. By Canny edge detection, we intend to extract good character boundary boxes for characters in text regions. Too small or too big connected components in the edge image are removed. To this stage, each remaining connected component represent for a character, a text region or a non-text region. Even with well-defined heuristic rules, many non-text regions cannot be removed because many objects in the image are similar to text when they are considered separately. For this reason, we use character center points to remove non-text regions.

3.4. Generating the Final Result

We use character center points to remove non-text regions in candidate text regions generated by the region-based method in Section 3.3. Fig. 3 (d) shows the candidate text regions achieved from the image in Fig. 3 (a). Many non-text regions still remain in the candidate text regions. The real text regions are connected components that are detected by region-based method and contain character center points. To evaluate this condition, a dilation image of the candidate text regions is created to connect nearby connected components together (Fig. 3 (e)). In the dilation image, connected components that do not overlap with character center points are classified as non-text regions. After removing non-text regions, final character bounding boxes are achieved (Fig. 3 (f)).

4. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed method, we use 30 home-made wine label images captured by cellular-phone cameras and 80 images taken from 2003 ICDAR Contest Test Images. All test images are rescaled into 640 by 480 images. Our method is compared with edge-based method [7-8] and connected component (CC) based method [9] in terms of Precision and Recall which are calculated by following equations.

\[
Precision = \frac{T}{T + F} \times 100\% \tag{3}
\]

\[
Recall = \frac{T}{S} \times 100\%, \tag{4}
\]

where S is the total number of characters in images; T is the number of characters that are correctly detected; F counts the number of regions incorrectly identified as texts. Table 1 summarizes the performance of the methods. The results show that our proposed method can successfully detect text in many natural scene images with low false positive rate compared to other methods. Some text detection results of our proposed methods are presented in Table 2.
Table 1. A comparison between the proposed method and other text detection methods (region of characters).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>2261</td>
<td>2261</td>
<td>2261</td>
</tr>
<tr>
<td>T</td>
<td>1860</td>
<td>1514</td>
<td>1713</td>
</tr>
<tr>
<td>F</td>
<td>231</td>
<td>651</td>
<td>875</td>
</tr>
<tr>
<td>Precision</td>
<td>88.95%</td>
<td>69.93%</td>
<td>66.19%</td>
</tr>
<tr>
<td>Recall</td>
<td>81.26%</td>
<td>69.96%</td>
<td>75.76%</td>
</tr>
</tbody>
</table>

With our method, text regions are not only detected but some information about corresponding text lines is achieved. By using normal vectors at characters we can estimate the curvature of the text line in which these characters belong to. Text alignment information is useful in text rectification applications.

5. CONCLUSIONS

In this paper, a new and efficient text detection method based on 2D tensor voting is proposed. In the proposed method, we combine the text line information and region-based method to detect text regions without using well-defined heuristic rules. In future work, we will exploit the text line information to straighten or rectify detected text for some specific applications such as wine label text recognition.

ACKNOWLEDGEMENT

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-313-D00999) and the MKE, Korea, under the ITRC support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2009-(C1090-0903-0008)).

Table 2. Some text detection results on natural scene images by the proposed method.

<table>
<thead>
<tr>
<th>Original images</th>
<th>Detected text regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image 1]</td>
<td>![Image 2]</td>
</tr>
<tr>
<td>![Image 3]</td>
<td>![Image 4]</td>
</tr>
</tbody>
</table>

REFERENCES