J Inf Process Syst, Vol.11, No.4, pp.528~537, December 2015
http://dx.doi.org/10.3745/JIPS.02.0026

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Text Detection in Scene Images Based on Interest Points

Minh Hieu Nguyen* and Gueesang Lee**

Abstract

Text in images is one of the most important cues for understanding a scene. In this paper, we propose a novel approach based on interest points to localize text in natural scene images. The main ideas of this approach are as follows: first we used interest point detection techniques, which extract the corner points of characters and center points of edge connected components, to select candidate regions. Second, these candidate regions were verified by using tensor voting, which is capable of extracting perceptual structures from noisy data. Finally, area, orientation, and aspect ratio were used to filter out non-text regions. The proposed method was tested on the ICDAR 2003 dataset and images of wine labels. The experiment results show the validity of this approach.

Keywords

Connected Component, Interest Point, Tensor Voting, Text Detection

1. Introduction

As the number of images increase exponentially, the needs for extracting meaningful information from these images also escalates. Although numerous approaches on text detection have been published, it is still challenging to localize text in natural scene images. This has caused researchers to be interested in many aspects of the problem.

Text detection methods can be divided into the two main categories of texture-based and region-based. A complete survey of text detection with related applications can be found in [1]. Texture-based approaches usually require a high computation time. Five localized features are proposed in [2] with a neural network classifier to locate the regions of text in an image, in which the result is invariant to scale and the 3D orientation of the text. It also allows for the recovery of text in cluttered scenes. The frequency domain was also considered using the Fourier transform [3], discrete cosine transform (DCT) [4], and wavelets [5]. Frequency domain methods are based on the fact that small texts produce strong texture responses, and therefore, are only effective for scenes with small character strokes.

The region-based approach is widely used because it is simple in its algorithm and robust against illumination changes. Region-based approaches determine spatial correlation based on edge features [6] or connected component features [7] 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

Manuscript received May 13, 2014; first revision March 11, 2015; accepted April 14, 2015; onlinefirst August 10, 2015.
Corresponding Author: Gueesang Lee (gslee@jnu.ac.kr)
* FTP University, Hanoi City, Vietnam (hieunp131@gmail.com)
** Dept. of Electrical and Computer Engineering, Chonnam National University, Gwangju, Korea (gslee@jnu.ac.kr)

regions or outliers. On the other hand, the connected component based methods use a bottom-up approach by grouping small components into larger components until all of the regions are identified. In region-based methods, however, many non-text regions are misclassified as text regions because the information in neighboring regions is not considered.

These approaches, especially in the region-based method, generate a high false positive rate and require many parameter settings in heuristics depending on the specific application.

In contrast, Nguyen et al. [8] first used tensor voting for detecting text from images. By adopting tensor voting, the use of heuristic rules was minimized. Nevertheless, since tensor voting deals with points, or more specifically the pixel level, this method did not take into account a region's properties (such as area, orientation, aspect ratio, etc.) when candidate text regions were selected. As a result, a large amount of false positives, or commonly known as a 'false alarm,' were extracted. This leads to low precision in a lot of cases, which are shown in Fig. 1. Moreover, as the only input for the tensor voting process, the centroids obtained from connected components of an edge map could not cover all of the sections that are text regions in the image. Consequently, several texts were left behind undetected, as shown in Fig. 2. Since the publication of this original paper, there has been limited progress in this direction.

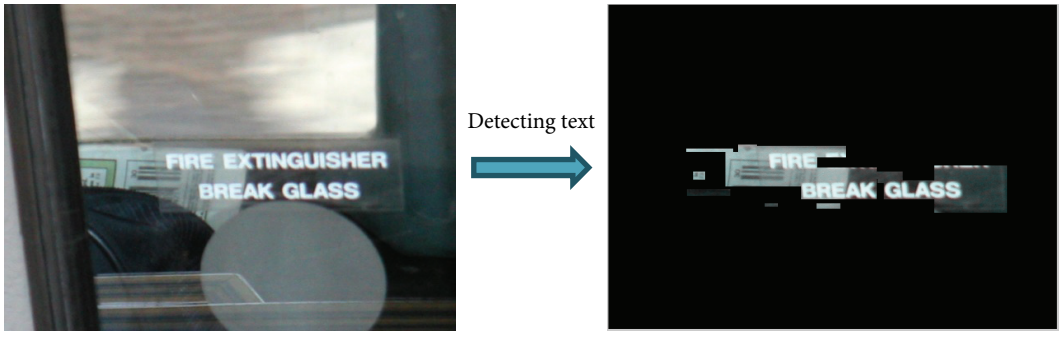


Fig. 1. Example with many false results from early implementation of tensor voting.

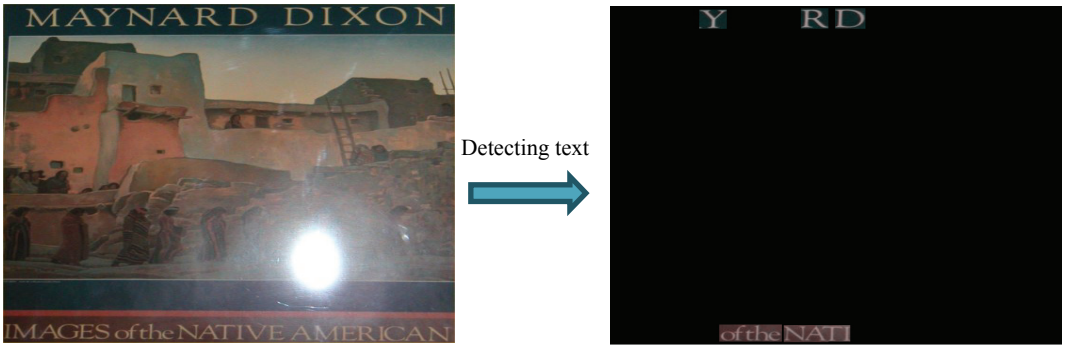


Fig. 2. Undetected text in early implementation of tensor voting.

Inspired by the observation that there is usually a large amount of corner points and center points of edge components close together in text characters, we are proposing a novel combination of corner point detection, edge detection, tensor voting, and a few rules to locate text in natural images. The experiments show that these changes did improve previous development in various cases.

The remainder of this paper is organized as follows: in Section 2, we present the proposed method. The experiment results are given in Section 3. Finally, Section 4 provides our conclusions and future work.

2. Proposed Approach

The proposed system has four stages: pre-processing, interest points detection, tensor voting to extract candidate text regions, and post-processing (Fig. 3). The input is a natural scene image and the output is the same image with red bounding boxes around the detected text regions. The input will be resized if it's a small sized image. Then, it is converted to grayscale, filtered by a bilateral filtering technique [9], and duplicated into two images with different resolutions for the purpose of locating text in different sizes. In our implementation, both corner points detected by the Förstner-Köthe corner detector [10] and center points of connected components, which are obtained from a Sobel edge map, are aggregated together. These points are then encoded with 2D tensors and they are called tokens. Later, a tensor voting framework is applied to extract curve saliency values and normal vectors of these tokens. Finally, some rules are used to eliminate the rest of the non-text regions that are still remaining after previous phases.

2.1 System Flowchart for Text Detection

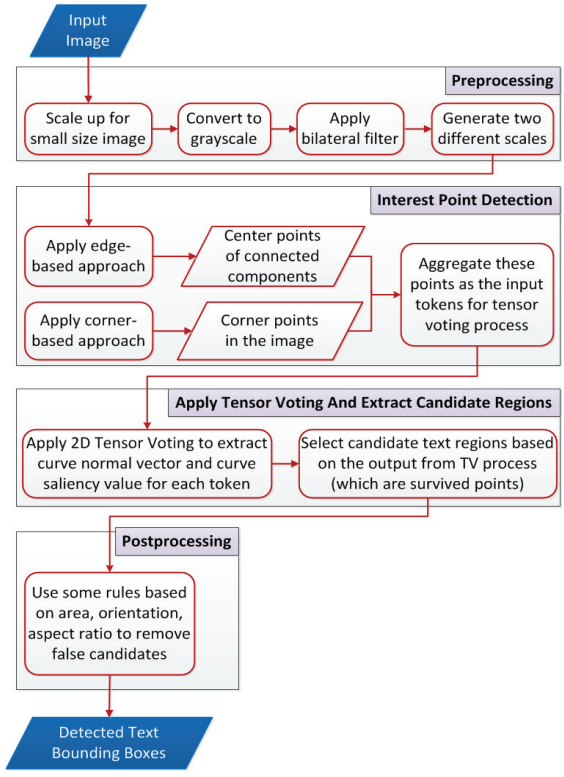


Fig. 3. Flowchart of proposed system for text detection.

2.2 Interest Point Detection

By 'interest point' we simply mean any point in the image for which the signal changes two-dimensionally. Conventional 'corners' such as L-corners, T-junctions, and Y-junctions satisfy this, but so do black dots on white backgrounds, the center of connected components, the endings of branches, and any location with a significant 2D texture [11]. In this paper we use the general term 'interest point' to indicate corner points and center points of edge connected components. Fig. 4 shows an example of general interest points detected on a sample image.

Many different interest point detectors have been proposed with a wide range of definitions for what points in an image are interesting. Some detectors find points of high local symmetry; others find areas of highly varying texture, while others locate corner points. Corner points are interesting as they are formed from two or more edges and edges usually define the boundary between two different objects or parts of the same object.

Förstner and Gülch [12] propose a two-step procedure for localizing interest points. The first points are detected by searching for optimal windows using the auto-correlation matrix A. This detection yields systematic localization errors (i.e., in the case of L-corners). A second step based on a differential edge intersection approach improves localization accuracy.

Köthe [12] improved the structure tensor computation using an increased resolution and non-linear averaging to optimize the localization accuracy. In our implementation, we applied a modified version of a Förstner corner detector that embraces some extensions proposed by Köthe.

These days, the Harris corner detector is frequently used [13], as shown in Fig. 4.

Fig. 4. Interest points detected by Harris detector.

To generate the binary edge map, the Sobel vertical edge-emphasizing method is applied to the grayscale image. The Sobel vertical edge magnitude is computed through the following convolution operation:

$$E(x,y) = I(x,y) * S, \qquad S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

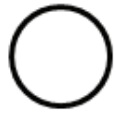
where, I is the grayscale image, E is the edge map, and S is the vertical Sobel kernel.

Edge connected components are defined as sets of connected edge pixels in the edge map. Center points of these connected components are then extracted.

2.3 Tensor Voting for Text Detection

In 2D tensor voting [14,15], each input token corresponds to a point or curve segment. Table 1 shows how input tokens are encoded as second order tensors. If the input token is a point, which does not have a preferential orientation, it is encoded with a ball tensor. Otherwise, it is encoded as a stick tensor with normal vector information.

Table 1. Encoding input tokens as 2D second order symmetric tensors

Input token	Tensor	Matrix form	Eigenvalues
Point (un-oriented)	Ball tensor 	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\lambda_1 = \lambda_2 = 1$
Curve segment (oriented) 	Stick tensor 	$\begin{bmatrix} n_1^2 & n_1 n_2 \\ n_1 n_2 & n_2^2 \end{bmatrix}$	$\lambda_1 = 1, \lambda_2 = 0$

Tensors support information including proximity and the smoothness of continuity by voting process. Tensors that lie on smooth salient features (i.e., curves) strongly support each other. Each tensor votes for its neighboring tensors with its information and also receives votes from them. The shape and size of this neighborhood and the voting strength and orientation are encapsulated in predefined voting fields or kernels. Each type of feature requires a voting filed. All voting fields can be generated from a fundamental stick voting field. For details of tensor voting, refer to [14,15].

The positions of detected interest points are encoded with tensors and after that they are input to the 2D tensor voting framework. Tensors then propagate their information to their neighbors by voting algorithm. Since the distances between interest points are rather big, we set a large value for the scale of the voting field $\sigma = 100$. After tensor voting, the curve saliency value and normal vector of the token at each point are extracted. Fig. 5(c) visualizes these values with extracted interest points, in which the points, saliency values, and normal vectors of tokens are represented by the yellow dots, the lengths of the white lines, and the orientations of the while lines, respectively. It's noticeable that in the text region, the tokens have high curve saliency values and their normal directions are likely the same as the normal vectors of text lines that they belong to. Therefore, we can use this information to remove more non-text regions.

The interest points corresponding to tokens that have small curve saliency values and nearly horizontal normal vectors, as described in Fig. 6, are removed. This step is presented in Fig. 5(d), which shows the remaining interest points after we remove others by using token information. The remaining interest points will be used to select the text regions.

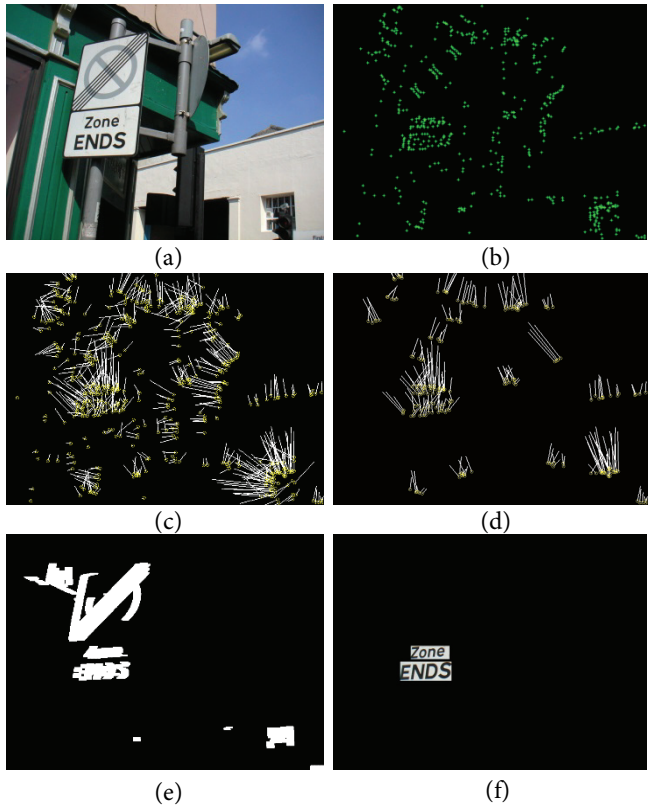


Fig. 5. An example of the proposed method. (a) Original image, (b) detected interest points, (c) results from applying tensor voting on interest points: saliency values, and normal vectors of tokens, (d) remaining points after removing tokens with small curve saliency values and nearly horizontal normal vectors, (e) dilated area of remaining interest points, and (f) final detected text bounding box after post-processing.

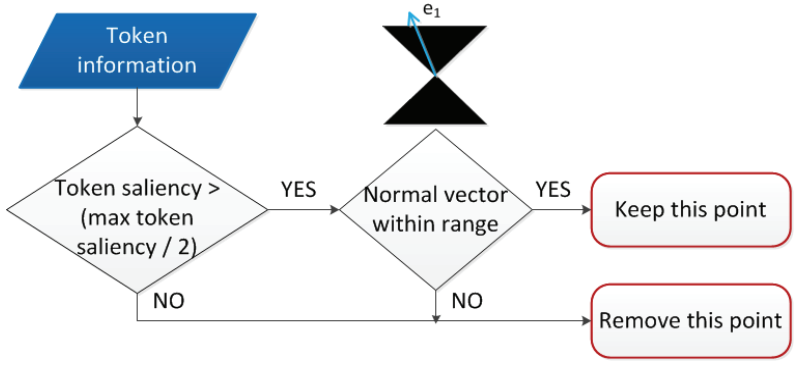


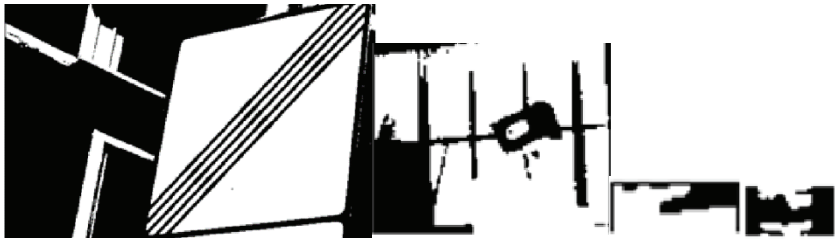
Fig. 6. Using token information to remove points of non-text regions.

2.4 Post-Processing

Rules used in the post-processing phase are based on area, orientation, and the aspect ratio of the input regions. These rules help to eliminate the rest of non-texts in various cases. They are constructed empirically by experiments and are described below.

2.4.1 Area

The area of a region is computed as the number of the foreground (or white) pixels in that region. We remove the regions, as shown in Fig. 7(d), with an area less than a threshold, which corresponds to noises in the test data



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

Fig. 7. Regions removed in post-processing phase. (a–c) By checking aspect ratio and (d) by checking area.

2.4.2 Aspect ratio

The aspect ratio of a bounding box is computed as its width over its height. Since texts are often positioned horizontally, a large enough aspect ratio value would be the sign of text than a small one would. We kept the regions with aspect ratios > 2, which comes from the observation that most of the bounding boxes for the text line have several characters aligned along the text line. Examples of removed regions are shown in Fig. 7(a) and (b).

2.4.3 Orientation

Orientation is defined as the angle (in degrees ranging from -90° to 90°) between the x-axis and the major axis of the ellipse that has the same second-moments as the region. Fig. 8 illustrates the axes and orientation of an input region. If the orientation is > 10° or < -10°, then it violates the assumption that the text lines are mostly horizontally aligned and the region will be removed.

Fig. 8. Orientation feature of a region.

3. Experimental Result

To evaluate the performance of the proposed method, we adopted the process in [8]. The input image set included 30 homemade wine label images captured by cellular-phone cameras and 80 images taken from the ICDAR 2003 Trial Test dataset [16]. The proposed method was implemented on a Windows XP system with Intel i5-2400 3.10 GHz and compared with the first implementation of a tensor voting-

based method [8], edge-based method [6], and connected component (CC)-based method [7]. The comparison is carried out in terms of Precision and Recall, which are calculated by the following equations:

$$Precision = \frac{T}{T+F} \times 100\%$$

$$Recall = \frac{T}{S} \times 100\%$$

where, S is the total number of characters in images, T is the number of characters that are correctly detected, and F counts the number of regions incorrectly identified as texts. Table 2 shows some results from the proposed method. Table 3 summarizes the performance of the methods. The results show that the proposed method has better precision and recall than others. The tensor voting based method by [8] exhibits better performance than conventional edge-based [6] or CC-based [7] methods, while our method outperforms [8] because of the introduction of feature selection.

Table 2. Sample results of proposed method

Table 3. Comparison of methods

	Proposed method	**TV-based [8]**	**Edge-based [6]**	**CC-based [7]**
S	2261	2261	2261	2261
T	1919	1860	1514	1713
F	46	231	651	875
Precision (%)	**97.65**	88.95	69.93	66.19
Recall (%)	**84.87**	81.26	69.96	75.76

TV=tensor voting, CC=connected component.

4. Conclusions

Since the first implementation of tensor voting in text detection, there have not been many improvements. It is true that by applying 2D tensor voting, the use of heuristic rules is minimized. Nevertheless, the tensor voting-based method failed to locate text in many cases and often generated various non-text regions in the result. To overcome these limitations, we proposed a novel method based on interest point detection. This method utilizes the fact that there is often a large amount of corner points of characters, and center points of connected components in text regions. The experimental results show that our method is effective in the detection of text in complex natural scene images.

For future work, we will consider using the curve saliency value and normal vectors to estimate the curvature of the text lines so that we utilize this information for text rectification.

Acknowledgement

This research was supported partly by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by MEST (2015-024474) and by the technological innovation R&D Program of SMBA (S2173771).

References

[1] K. Jung, K. I. Kim, and A. K. Jain, "Text Information extraction in images and video: a survey," *Pattern Recognition,* vol. 35, no. 5, pp. 977-997, 2004.

[2] P. Clark and M. Mirmehdi, "Recognising text in real scenes," *International Journal of Document Analysis and Recognition*, vol. 4, no. 4, pp. 243-257, 2002.

[3] B. K. Sin, S. K. Kim, and B. J. Cho, "Locating characters in scene images using frequency features," in *Proceedings of 16th International Conference on Pattern Recognition,* Quebec, Canada, 2002, pp. 489-492.

[4] D. Crandall, S. Antani, and R. Kasturi, "Extraction of special effects caption text events from digital video," *International Journal of Document Analysis and Recognition,* vol. 5, no. 2-3, pp. 138-157, 2003.

[5] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and robust text detection in images and video frames," *Image and Vision Computing,* vol. 23, no. 6, pp. 565-576, 2005.

[6] J. Samarabandu and X. Liu, "An edge-based text region extraction algorithm for indoor mobile robot navigation," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 1, no. 7, pp. 2008-2015, 2007.

[7] J. Gllavata, R. Ewerth, and B. Freisleben, "A robust algorithm for text detection in images," in *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis (ISPA2003)*, Rome, Italy, 2003, pp. 611-616.

[8] T. Nguyen, J. Park, and G. Lee, "Using 2D tensor voting in text detection," in *Proceedings of 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, TX, 2010, pp. 818-821.

[9] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of 6th International Conference on Computer Vision (ICCV)*, Bombay, India, 1998, pp. 839-846.

[10] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct point, corners and centres of circular features," in *Proceedings of Intercommission Conference on Fast Processing of Photogrammetric Data,* Interlaken, Switzerland, 1987, pp. 281-305.

[11] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision,* vol. 37, no. 2, pp. 151-172, 2000.

[12] U. Köthe, "Integrated edge and junction detection with the boundary tensor," in *Proceedings of 9th International Conference on Computer Vision*, Nice, France, 2003, pp. 424-431.

[13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of Alvey Vision Conference*, Manchester, UK, 1988, pp. 147-151.

[14] G. Medioni, M. S. Lee, and C. K. Tang, *A Computational Framework for Segmentation and Grouping*. Amsterdam: Elsevier, 2000.

[15] W. S. Tong, C. K. Tang, and G. Medioni, "First order tensor voting, and application to 3-D scale analysis," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2001)*, Kauai, HI, 2001, pp. 175-182.

[16] ICDAR 2003 datasets, http://algoval.essex.ac.uk/icdar/Datasets.html.

Minh Hieu Nguyen

He received the B.S. degree in Software Engineering from FPT University, Vietnam in 2011. He received the M.S. degree in the department of Electronics and Computer Engineering in Chonnam National University, South Korea. His research interests are image processing, computer vision and text extraction.

Gueesang Lee http://orcid.org/0000-0002-8756-1382

He received the B.S. degree in Electrical Engineering from Seoul National University in 1980. In 1982, he received the M.S. degree in Computer Engineering from Seoul National University. In 1991, he received Ph.D. degree in Computer Science from Pennsylvania State University. He is currently a professor of the department of Electronics and Computer Engineering in Chonnam National University, Korea. Research Interests: Image processing, computer vision and video coding.