

## Research Article

# Natural Scene Text Detection and Segmentation Using Phase-Based Regions and Character Retrieval

Julia Diaz-Escobar  <sup>1</sup> and Vitaly Kober  <sup>1,2</sup>

<sup>1</sup>Department of Computer Science, CICESE, Ensenada, B.C. 22860, Mexico

<sup>2</sup>Department of Mathematics, Chelyabinsk State University, Chelyabinsk, Russia

Correspondence should be addressed to Julia Diaz-Escobar; jdiaz@cicese.edu.mx

Received 19 September 2019; Revised 15 May 2020; Accepted 2 June 2020; Published 19 June 2020

Academic Editor: Daniel Zaldivar

Copyright © 2020 Julia Diaz-Escobar and Vitaly Kober. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multioriented text detection and recognition in natural scene images are still challenges in the document analysis and computer vision communities. In particular, character segmentation plays an important role in the complete end-to-end recognition system performance. In this work, a robust multioriented text detection and segmentation method based on a biological visual system model is proposed. The proposed method exploits the local energy model instead of a common approach based on variations of local image pixel intensities. Features such as lines and edges are obtained by searching for the maximum local energy utilizing the scale-space monogenic signal framework. The candidate text components are extracted from maximally stable extremal regions of the local phase information of the image. The candidate regions are filtered by their phase congruency and classified as text and nontext components by the AdaBoost classifier. Finally, misclassified characters are restored, and all final characters are grouped into words. Experimental results show that the proposed text detection and segmentation method is invariant to scale and rotation changes and robust to perspective distortions, blurring, low resolution, and illumination variations (low contrast, high brightness, shadows, and nonuniform illumination). Besides, the proposed method achieves often a better performance compared with state-of-the-art methods on typical natural scene datasets.

## 1. Introduction

Nowadays, imagery has become an indispensable source of human communication and interaction. Millions of images are shared every day, and new content-based image applications have been developed. In particular, digital images with textual content provide useful information for tasks related to document classification, multimedia retrieval, language translator, text to voice converter, robotic navigation, and augmented reality, to name a few [1, 2]. The analysis of this textual information involves basically three stages: text detection, character segmentation, and word recognition. The fundamental goal of text detection is to determine whether there is text in a given image, while character segmentation considers the extraction and localization of characters from background pixels. Word recognition considers character

grouping and error correction in order to recognize the final words.

Since text localization, character segmentation, and word recognition stages are not necessarily applied in a specific order, the character segmentation as the first stage could provide a better performance for the following processes. However, text localization and character segmentation are still challenges in the document analysis and computer vision communities (<http://rrc.cvc.uab.es/?com=introduction>). Natural text scenes contain different types of fonts, symbols, colors, scales, and character orientations, which make text detection a complicated task. Moreover, natural scenes are commonly captured under uncontrolled conditions (illumination changes, partial occlusion, low resolution, sensor noise, blur, and alignment) and could contain complex backgrounds (people, buildings, fences, bricks, grass, trees, and cars) [1–3].

In the last decades, several techniques have been explored to solve the text detection and segmentation problem. These methods can be broadly divided into four categories: sliding window-based, connected component-based, deep learning-based, and hybrid methods [1]. Sliding window-based methods, also called texture-based methods, consider a sliding window across all over the images under different scales to identify text regions. Fourier-statistical features (FSF) [4], discrete cosine transform (DCT) [5], spatial filters [6], and wavelet coefficients [7] are commonly used as textural properties. Nevertheless, sliding window methods are sensitive to scale and rotation variations, besides they are computationally expensive. Connected component-based methods consider connected component properties such as color, stroke width, aspect ratio, and size to distinguish between character and noncharacter regions. Usually, connected components are obtained by color clustering [8, 9], image binarization [10, 11], edge detection [12], stroke width transform (SWT) computation [13], and maximally stable extremal region (MSER) extraction [14, 15].

In the last years, the MSER and SWT techniques have become the most used techniques for text detection process due to their invariance to scale and rotation transformations. Besides, not only the MSER but also all extremal regions (ERs) are used for text segmentation [16–20]. However, ER-based methods need to process multiple repeated regions to obtain correct character segmentation, generating classification errors and a high computational cost. Furthermore, SWT-based techniques are dependent on the accurate edge detector, which is not feasible in many cases.

Recently, deep learning-based techniques have become popular for pattern recognition. In particular, for the multioriented text detection task, different neural networks (NNs) and configurations have been proposed [21–24]. However, NNs need to be pretrained using thousands of images in order to achieve a good performance, and in many cases, a final fine-tune is realized with the training images of the dataset to be evaluated. Moreover, it has been shown that this kind of approach can be easily fooled by modifying some values of the image pixels [25].

Lastly, hybrid methods combine the sliding window techniques, connected components, and neural network-based methods [26–30]. Until now, most of the proposed methods related to natural scene text detection are based on the pixel intensity values. As a consequence, method performance is affected by the presence of nonuniform illumination, low contrast, blur, or noise degradations. In contrast, we propose a robust multioriented text detection and segmentation method based on the biological visual system model. Psychophysical evidence suggests that the human visual system decomposes the visual information into border and line components by using phase information. Furthermore, it is known that different groups of cells in V1 extract particular image features as frequency, orientation, and phase [31].

In this work, a new multioriented text detection and segmentation method based on the biological energy model is suggested. This paper is an extended version of the conference papers [32, 33]. Unlike the previous works, we

utilize the phase-based MSER approach and the AdaBoost classifier instead of applying only heuristic rules for the character filtering, retrieval, and grouping stages.

The main contribution of this work is as follows. First, the proposed character segmentation method is based on a biologically inspired model rather than being based on local intensities. Thus, the proposed text segmentation is robust to variations of the image pixel values (nonuniform illumination, low contrast, and shadows), and it is invariant to slight scale and rotation changes. Second, the phase congruency approach for character filtering and noise control is utilized, which significantly reduces the number of generated components, keeps a low number of regions, and preserves the most relevant regions. Third, AdaBoost classifiers are used rather than heuristic rules at character filtering, retrieval, and grouping stages. Finally, the computational complexity of the proposed system at the training stage is much lower compared with that of deep learning techniques, while the performance of the system with a small training set is competitive and, in some cases, better than that of the state-of-the-art algorithms.

The paper is organized as follows. In Section 2, a brief description of the related works is presented. In Section 3, the proposed text detection and segmentation method is described. In Section 4, experimental results are presented and discussed. Section 5 summarizes our conclusions.

## 2. Related Work

Until now, there are two representative connected component-based techniques used for text segmentation, that is, the SWT [13] and the MSER [14].

The local operator SWT computes the character stroke width for each edge map pixel. Therefore, strokes that have constant width values can be considered as characters, and those components which have similar stroke width values can be grouped into words. Since the original SWT is invariant to rotation and scale variations, several SWT-based methods have been developed. In [34, 35], a SWT-based method is proposed for multioriented text detection. The Canny edge detector is used to calculate the SWT map from the image. The image pixels are associated considering the SWT ratio and grouped into connected components. The obtained components are classified into character and noncharacter elements using a two-layer filtering scheme. A set of heuristic rules are considered, and a trained random forest (RF) classifier is applied. Finally, the character candidates are aggregated into text chains satisfying a certain set of rules. In [36], an extended version of the SWT, called stroke feature transform (SFT), is proposed. In addition to stroke width constraints, the SFT considers color uniformity and local relationships of edge pixels during ray tracking. Then, two text covariance descriptors are defined for component-level and text-line RF classifier training. In [37], an efficient stroke width value computation is proposed. The obtained stroke width value is used together with a perceptual diverge cue and an edge histogram of oriented gradient (HOG) descriptor to measure the properties of characters under a Bayesian framework.

On the contrary, the MSER method basically extracts image regions that remain stable under a certain number of thresholds, which are considered as potential character candidates. The MSER technique was first introduced by Matas and Zimmermann [15] for character detection and was recently extended for text detection and recognition [18]. In [16], an MSER-based text segmentation method is proposed. The character candidates are extracted using the MSER algorithm. The candidates are grouped using orientation, morphology, and protection clustering via adaptive hierarchical clustering. Then, the text candidates are classified into text and nontext components. In [17], a subpath division from the ER tree is done. Multiple subpaths are created according to the size and position similarities or ER regions. Then, an AdaBoost classifier is trained using mean local binary patterns (MLBP) for text and nontext classifications. Finally, heuristic rules are used for misclassified character filtering. In [20], the character candidates are extracted from low-variation ERs and classified using a support vector machine (SVM) and geometrical features. The obtained characters are grouped into text lines using heuristic rules, and a final restoration stage is considered if adjacent regions satisfy a set of predefined conditions. In [19], a similar ER-based method is proposed, but instead, geometrical features, the HOG, and local binary pattern (LBP) features are selected for character classification and recognition. Then, characters are grouped into text lines, and a CNN model is used to verify text lines, removing noncharacter components. In [28], a multichannel and multiresolution (MC-MR) strategy is proposed. The text candidates are extracted using MSER technique under RGB and YUV color spaces under different resolutions. Then, candidates are filtered by a coarse-fine strategy and classified as text and no-text components by a NN classifier.

### 3. Proposed Text Detection and Segmentation Method

In this section, the methodology for the proposed text detection and segmentation method is described. Connected components are obtained from the local image phase information. In order to extract the local phase-based image features, the scale-space monogenic signal framework [38, 39] is utilized. Basically, connected component regions are extracted from the local phase image using the MSER approach. Then, the obtained connected components are filtered considering geometrical properties, and the remaining components are considered as character candidates. Using an AdaBoost classifier, the character candidates are predicted as a character or noncharacter component. Finally, a second AdaBoost classifier is applied to restore misclassified characters. Figure 1 shows a block diagram of the proposed method.

**3.1. Image Preprocessing.** Morrone and Owens [40, 41] proposed a local energy model. This model argues that the biological visual system can locate features of interest by searching for maximum local energy and identifying the feature type (shadow, edge, or line) by evaluating the argument at that point. That is, edges, lines, and shadows, can be obtained at points where the Fourier components of the signal are maximum in the phase distribution, called phase congruency. Continuing with this approach, in [42], a dimensionless measure of phase congruency ( $PC(x)$ ) is proposed as follows:

$$PC(x) = \max_{\varphi \in [0, 2\pi]} \frac{\sum_n W(x)[A_n(x)[\cos(\varphi_n(x) - \bar{\varphi}(x))] - T]}{\sum_n A_n(x) + \varepsilon}, \quad (1)$$

where  $W(x)$  is a weight for the frequency spread;  $\varepsilon$  is a small constant to avoid division by zero; and  $T$  is a noise threshold parameter.  $PC(x)$  goes from 0 to 1. The  $PC(x)$  value indicates the significance of the current feature: unity means the most significant feature, and zero indicates the lowest significance. We refer to papers [42, 43] for more details.

In practice, local frequency information is obtained via banks of oriented 2D filters, which are computationally expensive. Instead, we used the scale-space monogenic signal framework to compute the local phase information of the image.

Let be  $f(x, y)$  an image and  $F(u, v) = \mathcal{F}\{f(x, y)\}$  be its Fourier transform. The scale-space monogenic signal ( $F_M$ ) representation is defined as [38]

$$F_M(u, v) = F_{bp}(u, v) + i\mathbf{R} \cdot F_{bp}(u, v), \quad (2)$$

where  $\mathbf{R} = (R_x, R_y)$  is the transfer function of the first-order Riesz transform in the frequency domain:

$$\begin{aligned} R_x(u, v) &= i \frac{u}{\sqrt{u^2 + v^2}} = \mathcal{F} \left\{ \frac{x}{2\pi(x^2 + y^2)^{3/2}} \right\}, \\ R_y(u, v) &= i \frac{v}{\sqrt{u^2 + v^2}} = \mathcal{F} \left\{ \frac{y}{2\pi(x^2 + y^2)^{3/2}} \right\}, \end{aligned} \quad (3)$$

and  $F_{bp}(u, v) = B_{s_0, \lambda, k}(u, v) \cdot F(u, v)$  represents the image  $F(u, v)$  filtered by the band-pass filter:

$$B_{s_0, \lambda, k}(u, v) = \left( e^{-2\pi s_0 \lambda^k \sqrt{u^2 + v^2}} - e^{-2\pi s_0 \lambda^{k-1} \sqrt{u^2 + v^2}} \right), \quad (4)$$

where  $\lambda \in (0, 1)$  indicates the relative bandwidth,  $s_0$  indicates the coarsest scale, and  $k \in \mathbb{N}$  indicates the band-pass number. Figure 2 shows a block diagram of the scale-space monogenic signal framework.

Then, the local amplitude  $A(x, y)$ , local orientation  $\theta(x, y)$ , and local phase  $\varphi(x, y)$  (note that the function  $a \tan 2(|y|/x) = \text{sign}(y) \cdot \tan^{-1}(|y|/x)$ , where the factor  $\text{sign}(y)$  indicates the direction of rotation) can be computed as follows:

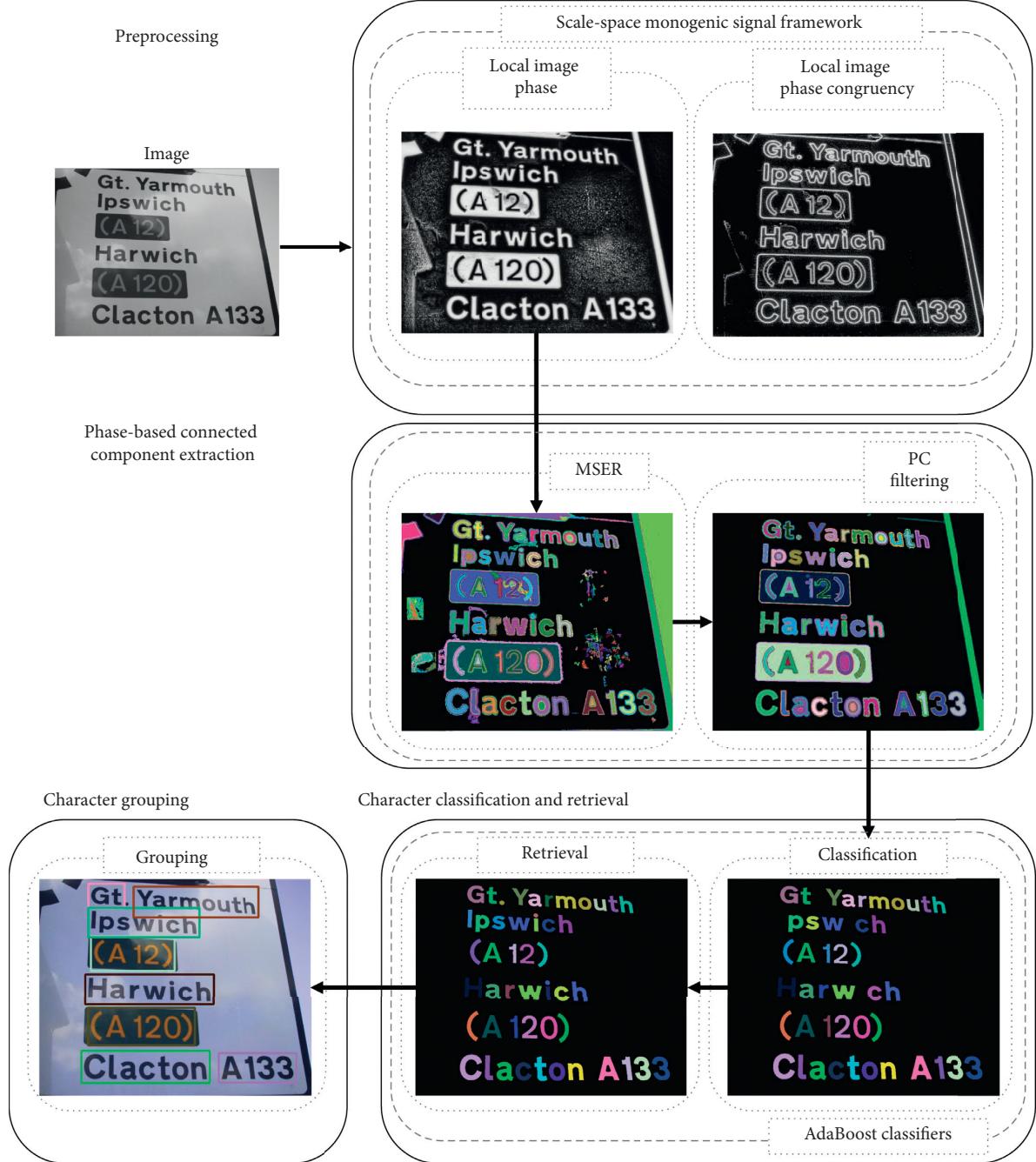


FIGURE 1: A block diagram of the proposed method.

$$A(x, y) = \sqrt{\left(\mathcal{F}^{-1}\{F_{bp}(u, v)\}\right)^2 + \left(\mathcal{F}^{-1}\{R_x(u, v) \cdot F_{bp}(u, v)\}\right)^2 + \left(\mathcal{F}^{-1}\{R_y(u, v) \cdot F_{bp}(u, v)\}\right)^2}, \quad (5)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{\mathcal{F}^{-1}\{R_y(u, v) \cdot F_{bp}(u, v)\}}{\mathcal{F}^{-1}\{R_x(u, v) \cdot F_{bp}(u, v)\}} \right), \quad (6)$$

$$\varphi(x, y) = a \tan 2 \left( \frac{\sqrt{\left(\mathcal{F}^{-1}\{R_x(u, v) \cdot F_{bp}(u, v)\}\right)^2 + \left(\mathcal{F}^{-1}\{R_y(u, v) \cdot F_{bp}(u, v)\}\right)^2}}{\mathcal{F}^{-1}\{F_{bp}(u, v)\}} \right). \quad (7)$$

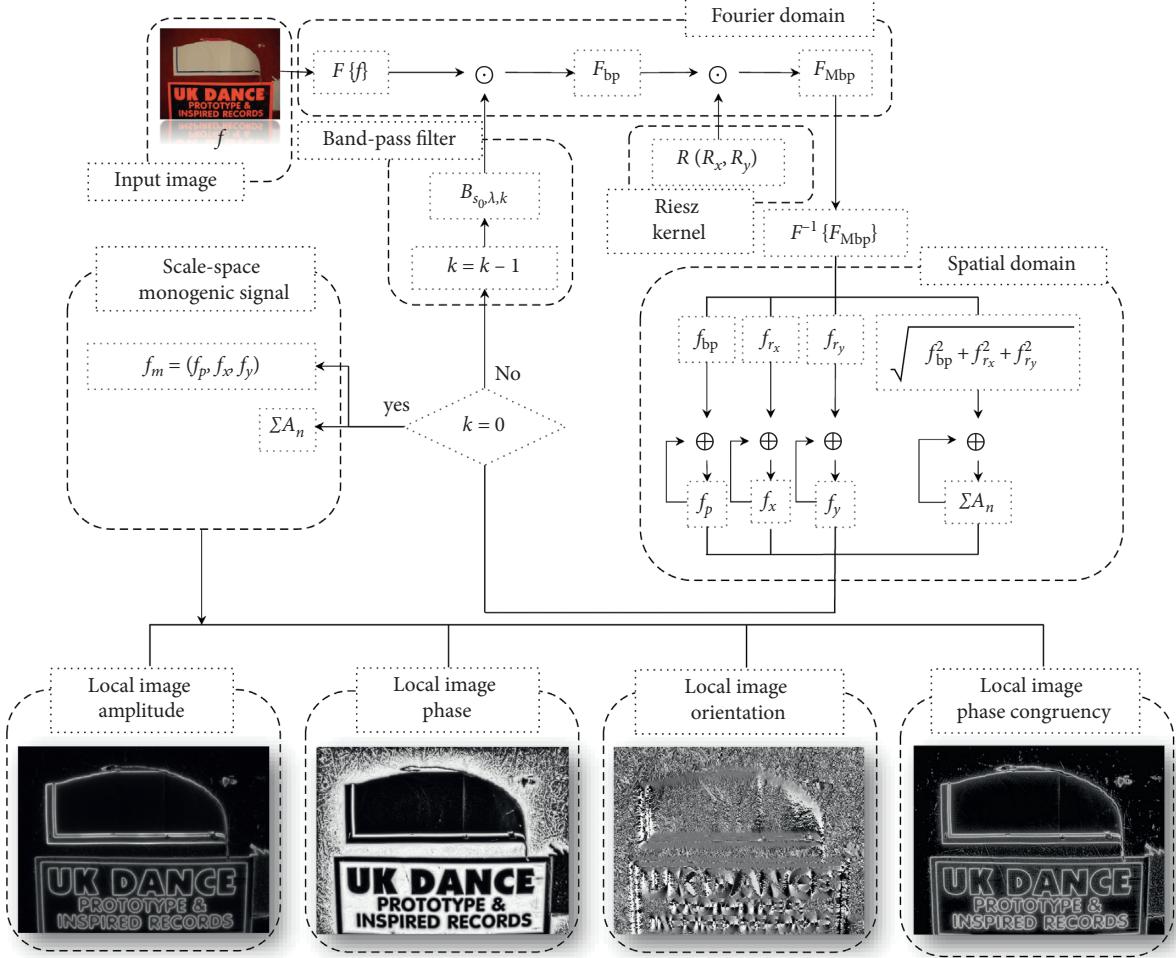


FIGURE 2: A block diagram of the monogenic signal framework.

**3.2. Phase-Based Character Candidate Generation.** As we mentioned earlier, the local image phase  $\varphi(x, y)$  describes the image structural information, while local amplitude gives us an intensity measure of the structure. Furthermore, the local phase allows us to distinguish between edge, edge-line, and line features. A phase value of 0 indicates an upward going step,  $\pi/2$  a bright line feature,  $\pi$  a downward going step, and  $3\pi/2$  a dark line feature [43]. However, we are not interested to make a distinction between dark or bright lines but in finding upward and downward going step features for region detection. For this reason, we consider the range from 0 to  $\pi$ , mapping the angles greater than  $\pi$  back into the range.

On the contrary, the MSER method [14] was first introduced for grayscale images, but it can be applied for any type of images as long as it maintains the two following conditions: totally ordered set and existence of adjacency relation. Thus, the proposed phase-MSER method is described as follows.

Let  $I$  be a grayscale image and  $\phi$  its local phase (equation (7)). The binary image  $I_{\text{bin}}^{(t)}$  is defined as

$$I_{\text{bin}}^{(t)}(x, y) = \begin{cases} 1, & \text{if } \phi(x, y) > t, \\ 0, & \text{if otherwise,} \end{cases} \quad (8)$$

where  $t$  denotes a threshold value. An extreme region  $R_t$  with threshold  $t$  is defined as

$$\forall p \in R_t, \quad q \in \partial R_t \implies I_{\text{bin}}^{(t)}(p) > I_{\text{bin}}^{(t)}(q) \text{ or } I_{\text{bin}}^{(t)}(p) < I_{\text{bin}}^{(t)}(q). \quad (9)$$

The extremal region  $R_{t^*}$  is maximally stable if and only if

$$q(t) = \frac{|R_{t+\Delta}| - |R_{t-\Delta}|}{|R_t|} \quad (10)$$

has a local minimum at  $t^*$ , with  $|\cdot|$  denoting cardinality, and  $\Delta$  is a parameter that considers the stability of the region under a certain number of thresholds. The obtained regions are called character candidates (CC). Figure 3 shows an example of the MSER technique and the proposed phase-MSER method.

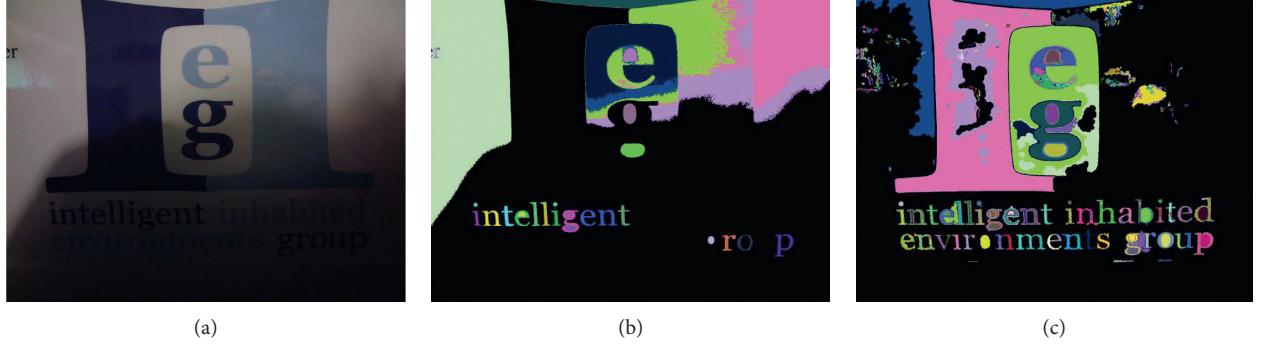


FIGURE 3: MSER vs. phase-MSER: (a) original image, (b) MSER, and (c) phase-MSER.

It is important to note that the local phase information is scale- and rotation-invariant. Moreover, due to the invariance-equivariance property, local phase information is independent of the local intensity; therefore, it is robust to contrast and illumination variations.

**3.3. Character Candidate Feature Computation.** Once the character candidate generation stage is done, a morphological closing operation is applied to each candidate in order to eliminate small holes. The size of the structural element was experimentally defined as  $\sqrt{[2]}\sqrt{[2]}CC_{area} \times \sqrt{[2]}\sqrt{[2]}CC_{area}$ . Next, for each candidate, geometrical connected component properties are computed.

Table 1 summarizes the computed properties.

Then, the obtained properties are used to compute the suggested candidate features:

- (1) The mean phase congruency value ( $PC_{mean}$ ) is computed to consider the phase congruency value of the candidate. As mentioned above, the  $PC(x)$  value indicates the significance of the current feature. Thus, one means the most significant edge component, and zero indicates the lowest significance.  $PC_{mean}$  is computed as follows:

$$PC_{mean} = \frac{1}{|CC_{contour}|} \sum_{i=1}^{|CC_{contour}|} PC(pt_i), \quad (11)$$

where  $\{pt_i \in CC_{contour}\}$  and  $|\cdot|$  denotes cardinality.

- (2) The phase congruency ratio ( $PC_{ratio}$ ) is computed to consider the contribution of the edge pixels of the candidate. One means a complete contribution from all the edge pixels, and zero indicates the lowest contribution.  $PC_{ratio}$  is obtained as

$$PC_{ratio} = \frac{1}{|CC_{contour}|} \sum_{i=1}^{|CC_{contour}|} D(PC(pt_i), PC_{thresh}), \quad (12)$$

where

$$D(PC(pt_i), PC_{thresh}) = \begin{cases} 1, & \text{if } PC(pt_i) > PC_{thresh}, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

and  $PC_{thresh}$  is a threshold from 0 to 1.

- (3) The filled convHull ratio is computed to consider the convexity of the candidate:

$$\frac{CC_{fill}}{\text{area}(CC_{convHull})}. \quad (14)$$

- (4) The approximated area ratio considers the stroke uniformity of the candidate. One means a complete uniformity of the candidate stroke, and zero indicates the lowest uniformity. The approximated area ratio is computed as

$$\frac{\text{abs}(CC_{area} - CC_{approx})}{\max(CC_{area}, CC_{approx})}, \quad (15)$$

where  $CC_{approx} = CC_{stroke} \cdot \text{length}(CC_{skel})$ .

- (5) The contour length ratio considers the difference between the external and internal candidate contours. This is to consider the complexity of the candidate edge. The contour length ratio is computed as

$$\frac{\text{abs}(\text{length}(CC_{contour}) - \text{length}(CC_{contourExt}))}{\text{length}(CC_{contourExt})}. \quad (16)$$

where  $CC_{contourExt}$  represents the external contour of the candidate.

In addition, the features used in [37, 44] are also considered:

- (1) The filled area ratio:

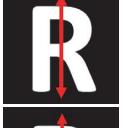
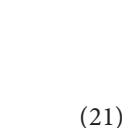
$$\frac{CC_{fill} - CC_{area}}{CC_{area}}. \quad (17)$$

- (2) The solidity:

$$\frac{CC_{area}}{\text{area}(CC_{hull})}. \quad (18)$$

- (3) The compactness:

TABLE 1: Connected component properties.

| Name                   | Property  |   |
|------------------------|---|---|
| CC                     | Connected component                                   |    |
| CC <sub>contour</sub>  | Contour pixels of the component                       |    |
| CC <sub>skel</sub>     | Morphological skeleton of the component               |    |
| CC <sub>convHull</sub> | Convex hull of the component                          |    |
| CC <sub>bb</sub>       | Bounding box of the component                         |    |
| CC <sub>rotRect</sub>  | Minimum rotated rectangle that encloses the component |   |
| CC <sub>stroke</sub>   | Mean stroke width of the component                    |  |
| CC <sub>minAxis</sub>  | Minimum axis of the component                         |  |
| CC <sub>maxAxis</sub>  | Maximum axis of the component                         |  |
| CC <sub>ratio</sub>    | Aspect ratio of the component                         |  |
| CC <sub>fill</sub>     | Filled component                                      |  |

(5) The eccentricity:

$$\frac{CC_{area}}{\text{length}(CC_{contour})^2}. \quad (19)$$

(4) The occupancy:

$$\frac{CC_{area}}{\text{area}(CC_{minRect})}. \quad (20)$$

$$\sqrt{1 - \left( \frac{\text{length}(CC_{minAxis})}{\text{length}(CC_{maxAxis})} \right)^2}. \quad (21)$$

(6) The aspect ratio:

$$\frac{\min(CC_{width}, CC_{height})}{\max(CC_{width}, CC_{height})}. \quad (22)$$

(7) The stroke width value:

$$\frac{\text{var}(CC_{stroke})}{E(CC_{stroke})^2}, \quad (23)$$

where  $E(\cdot)$  and  $\text{var}(\cdot)$  are mean and variance, respectively.

(8) The minimum stroke width ratio:

$$\frac{CC_{stroke}}{\min(CC_{width}, CC_{height})}. \quad (24)$$

(9) The maximum stroke width ratio:

$$\frac{CC_{stroke}}{\max(CC_{width}, CC_{height})}. \quad (25)$$

(10) The skeleton perimeter ratio:

$$\frac{\text{length}(CC_{skel})}{\text{length}(CC_{contour})}. \quad (26)$$

All the described features are used for AdaBoost classifier training to classify character candidates into text and nontext components. The text-component AdaBoost classifier was trained using the ICDAR2013 training dataset (299 images).

**3.4. Character Candidate Classification.** In this stage, the character candidate classification is performed. As a first step, coarse candidate filtering is applied taking into account the following noncharacter properties:

(1) The candidate area: to eliminate noncharacter candidates that are either larger or smaller than a predefined value, that is,

$$\max(50, 5 \times 10^{-4} \cdot I_{area}) < CC_{area} < \frac{1}{2} \cdot I_{area}, \quad (27)$$

where  $I_{area}$  is the image area.

(2) The aspect ratio: to eliminate noncharacter candidates that are too narrow or wide.  $CC_{ratio} < 0.10$  was considered.

(3) The phase congruency value: to eliminate low phase congruency value candidates. If  $PC_{mean}$  (equation (11)) is lower than a predefined threshold ( $PC_{thresh}$ ), then the candidate is discarded. Figure 4 shows an

example of the phase-based candidates under different  $PC_{thresh}$  values.

After the filtering stage, the remaining candidates are classified as text and nontext components using the already trained AdaBoost classifier. A candidate is considered as a text character (Char) if the sum of votes of the classifier is positive. The remaining candidates with the negative vote sum are considered as candidate neighbors (CN) and are used in the next stage of character retrieval.

**3.5. Character Retrieval.** During the classifier training stage, some characters were purposely mislabelled as noncharacters ("I," "i," "L," and "1") to reduce classification errors since these characters are usually similar to noncharacter structures in the image. The retrieval stage seeks to recover these characters and others that have been misclassified. The character retrieval method is described as follows.

For each Char, a neighborhood of radius  $R = 4 \cdot \max(Char_{height}, Char_{width})$  is defined. All the CNs inside the radius  $R$  are considered as character neighbors. If Char has no possible CNs, then the character is discarded from the retrieval stage but continues as a final character. It means that isolated characters are not discarded.

Next, each CN is evaluated to determine if it is a misclassified character. For this, a second AdaBoost classifier is applied. The classifier is trained using the following features between Char and its CN:

(1) The area difference:

$$\frac{\text{abs}(Char_{area} - CN_{area})}{\max(Char_{area}, CN_{area})}. \quad (28)$$

(2) The rotated rectangle area difference:

$$\frac{\text{abs}(\text{area}(Char_{rotRect}) - \text{area}(CN_{rotRect}))}{\max(\text{area}(Char_{rotRect}), \text{area}(CN_{rotRect}))}. \quad (29)$$

(3) The mean grayscale value difference:

$$\frac{\text{abs}(Char_{gray} - CN_{gray})}{255}. \quad (30)$$

(4) The height ratio:

$$\frac{\min(Char_{height}, CN_{height})}{\max(Char_{height}, CN_{height})}. \quad (31)$$

(5) The width ratio:

$$\frac{\min(Char_{width}, CN_{width})}{\max(Char_{width}, CN_{width})}. \quad (32)$$

(6) The mean stroke width difference:



FIGURE 4: Phase-based MSER regions filtered by different phase congruency thresholds: (a) phase-based MSER, (b)  $PC_{threshold} = 0.1$ , (c)  $PC_{threshold} = 0.2$ , (d)  $PC_{threshold} = 0.3$ , (e)  $PC_{threshold} = 0.4$ , (f)  $PC_{threshold} = 0.5$ , (g)  $PC_{threshold} = 0.6$ , (h)  $PC_{threshold} = 0.7$ , and (i)  $PC_{threshold} = 0.8$ .

$$\frac{\text{abs}(\text{Char}_{stroke} - \text{CN}_{stroke})}{\max(\text{Char}_{stroke}, \text{CN}_{stroke})}. \quad (33)$$

The character retrieval AdaBoost classifier was also trained using the ICDAR2013 training dataset.

Once the character retrieval AdaBoost classifier is trained, it is used to retrieve the CN as Char if the classifier vote sum is positive. Then, the retrieval neighbors are considered as characters, and they are also used for retrieval of their candidate neighbors recursively. The method stops when no new neighbor component is classified as a new character.

Note that no alignment feature is computed, as in many related works. Considering horizontal alignment helps to avoid character misclassification but restricts the method to horizontal text only. Thus, the proposed method can be applied for nonhorizontal text images.

**3.6. Character Grouping.** Since most of the state-of-the-art text detection methods evaluate word localization instead of character segmentation, a character grouping stage for text

detection is considered. Similar closest characters are grouped together and considered as candidate words. Then, the Hough transform is applied to obtain the final candidate word lines. The character grouping method is described as follows.

First, for each character, the distance between the character and all its neighbors within a radius  $R = 4 \cdot \max(\text{Char}_{height}, \text{Char}_{width})$  is computed. The distance is obtained as the minimum Euclidean distance between the convex hull of the character and its neighbors. All the characters are grouped into pairs, and a minimum region containing both components is created. The region is expanded to the minimum distance between characters.

All intersecting regions are considered as candidate words. Then, the Hough transform is applied to obtain the candidate word lines. Each of these lines is processed individually to verify if all the selected characters belong to a single word. This is done by applying the AdaBoost classifier used in the retrieval stage. All the characters from the candidate word are compared with each other. Those characters that are classified as nonword characters to all other characters form a new word, and so on. The method

stops when no new word is created. At the end, those final words that have only one element and its AdaBoost vote sum value is lower than zero, are eliminated. Figure 5 shows a character grouping example.

## 4. Experimental Results

**4.1. Evaluation Protocol.** The performance evaluation of the proposed method was realized using the following metrics. Two evaluation types are selected for text segmentation and text localization. For text segmentation, the character level recall-similarity rate [17] and the pixel-atom-based measures are utilized [45].

For character candidate generation evaluation, the recall-similarity rate is utilized. The recall-similarity is defined as the ratio between the total correctly detected candidate regions and the ground truth characters. A region is considered as a character candidate if the similarity value is up to 50%. The similarity value is defined as follows [17]:

$$\text{similarity}(D, \text{GT}) = \frac{\text{area}(D) \cap \text{area}(\text{GT})}{\text{area}(D) \cup \text{area}(\text{GT})}, \quad (34)$$

where  $D$  and GT represent the detected and ground-truth bounding box, respectively.

For pixel-level segmentation evaluation, the pixel- and atom-based measures are utilized. Pixel- and atom-based measures not only consider pixel-level accuracy but also take into account the morphological properties of characters. In [45], the minimal and maximal coverage criteria are introduced, which measure the degree of overlap between the ground truth area and the obtained segmented component. The minimal coverage criterion is fulfilled if the predefined threshold  $T_{\min} = 90\%$  of the ground-truth skeleton pixels is covered by the segmented component. Similarly, for the maximal criterion, the pixel distance to the ground-truth edge pixels should not exceed a maximum threshold  $T_{\max} = \min(5, 0.5 \cdot G)$ , where  $G$  is the maximum stroke width of the character.

On the contrary, although the proposed method is designed specifically for the text segmentation task, text localization evaluation is carried out to compare its performance with that of the state-of-the-art methods. The recall ( $R$ ), precision ( $P$ ), and F-measure ( $F$ ) are defined as follows [46]:

$$\text{precision}(G, D, t_r, t_p) = \frac{\sum_j \text{Match}_D(D_j, G, t_r, t_p)}{|D|},$$

$$\text{recall}(G, D, t_r, t_p) = \frac{\sum_i \text{Match}_D(G_i, D, t_r, t_p)}{|G|}, \quad (35)$$

$$F = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

$G$  and  $D$  represent the ground-truth rectangle set and detection rectangle set, respectively.  $t_r \in [0, 1]$  and

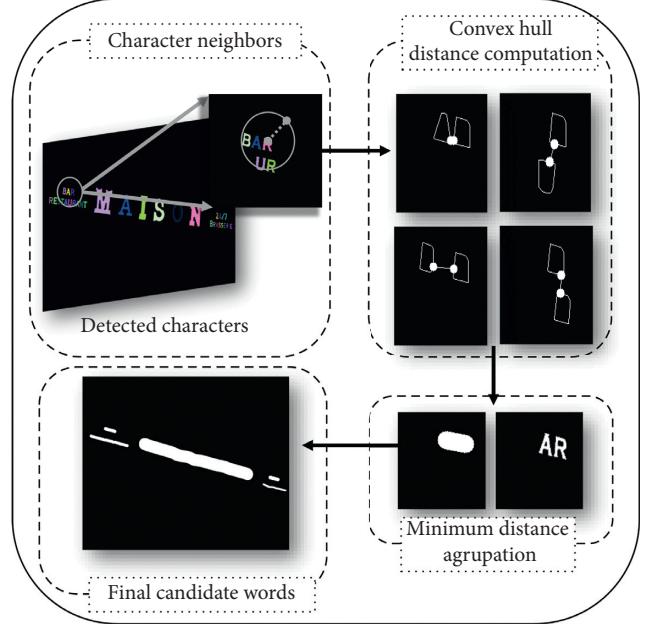


FIGURE 5: A block diagram of the proposed character grouping method.

$t_p \in [0, 1]$  are the recall and precision constraints, respectively. For more details, we refer to Wolf and Jolion [46].

For the MSER algorithm, the simulations were carried out using the reported MSER parameter [20], that is,  $\Delta = 4$ , maximum variation  $v = 0.5$ , and minimum diversity  $d = 0.1$ .

**4.2. Computer Simulations.** First, to analyze the tolerance of the proposed segmentation method to low contrast, high brightness, shadows, and nonuniform illumination degradations, computer simulations using synthetic images were performed. For the experiments, ten representative images from the ICDAR2013 dataset were selected. The selected images contain different symbols, font types, colors, sizes, and backgrounds. Each image was scaled, rotated, and synthetically degraded, obtaining 1000 synthetic images per degradation (see Figure 6). Table 2 shows the obtained results compared with the MSER method in terms of recall-similarity measure.

The proposed method shows a high candidate generation performance. The recall-similarity measure was up to 90% in most of the cases, excepting the brightness degradations. That is because brightness variations caused the loss of regions with low contrast (see Figure 6, second row, fifth column). Besides, the proposed segmentation method shows performance up to 30% for nonuniform illumination and shadow degradations and performance up to 10% for brightness and contrast variations compared with the MSER technique.

### 4.3. Typical Dataset Evaluation

**4.3.1. Datasets.** For performance evaluation of the proposed method, ICDAR2013 (<http://rrc.cvc.uab.es/>), USTB-SV1K [16], OSTD [47], and MSRA-TD500 [34] datasets are used.



FIGURE 6: Example of synthetic degraded images. From top to bottom: low contrast, high brightness, shadows, and nonuniform illumination.

TABLE 2: Character candidate generation results on the synthetic dataset (recall-similarity (%)).

| Method          | Contrast    | Brightness  | Illumination | Shadows     |
|-----------------|-------------|-------------|--------------|-------------|
| MSER (gray)     | 76.1        | 74.0        | 65.0         | 66.2        |
| Proposed method | <b>98.3</b> | <b>82.1</b> | <b>95.2</b>  | <b>94.2</b> |

The ICDAR2013 dataset consists of 462 complex scenes divided into training (299) and test (233) images. Note that the ICDAR2013 dataset contains images with horizontally aligned texts. Each image contains different complex backgrounds, font types, sizes, blurring, illumination, contrast, etc. The size of the images varies from  $480 \times 640$  to  $3888 \times 2592$ . USTB-SV1K dataset consists of 1000 Google Street View images ( $512 \times 512$ ) divided into training (500) and test (500) images. The images contain multiorientated and perspective-distorted text. OSTD dataset includes 89 multioriented text images. The images contain different font types, sizes, and orientations. The size of the images varies from  $640 \times 480$  to  $1024 \times 768$ . Finally, MSRA-TD500 contains 500 natural images divided into training (300) and test (200) images, which are taken from indoor and outdoor scenes. The resolution of the images varies from  $1296 \times 864$  to  $1920 \times 1280$ . The images contain English and Chinese texts, different fonts, sizes, colors, and orientations.

**4.3.2. Text Segmentation Evaluation.** Since text segmentation depends on the quality of connected component generation, the proposed phase-based character candidate

generation method is evaluated. Table 3 shows the obtained results in terms of recall-similarity measure and the obtained mean candidate regions. The obtained result shows that the proposed method obtains less character candidates with a high similarity rate than the other methods. Our method outperforms the results obtained in [8, 17], even when the methods utilize grayscale, RGB, Cb, and Cr channels. Although the recent methods [19, 28] report good similarity results for the given dataset, the mean number of candidates per image is too high, almost 30 and 15 times more than the proposed method. It is important to note that there exists a compromise between candidate region generation and computational complexity.

For the text segmentation evaluation, the precision and recall metrics were computed, as well as the F-measure. Table 4 shows the proposed method results on the ICDAR2013 dataset. The proposed method outperforms the methods [20, 48], which utilize grayscale images for character candidate extraction.

Both results, character candidate generation and text segmentation, show that the proposed method obtains fewer candidate regions with a more accurate pixel-level segmentation result.

Now, we provide the performance of the proposed method at different stages of its work. Table 5 presents character-level results in terms of recall, precision, and F-measure. We can observe that, after classification of candidates, the precision improves by 58%, while recall decreases by almost 24%. This is because at the classifier training stage, some characters were purposely mislabelled

TABLE 3: Character candidate generation results on the ICDAR2013 dataset.

| Method                            | Recall-similarity (%) | Candidate regions |
|-----------------------------------|-----------------------|-------------------|
| ER (G, H, S, and Cb) [19]         | 98.6                  | 6651              |
| MC-MR MSER [28]                   | 98.0                  | 2799              |
| MSER (gray)                       | 92.9                  | 754               |
| Sung et al. (gray + Cr + Cb) [17] | 87.7                  | 401               |
| Saric (gray) [20]                 | 89.9                  | 77                |
| Wu et al. (RGB) [8]               | 90.0                  | 1226              |
| Proposed method (gray)            | <b>91.0</b>           | <b>220</b>        |

TABLE 4: Character segmentation results on the ICDAR2013 dataset (%).

| Method           | Pixel-based (%) |             |             | Atom-based (%) |             |             |
|------------------|-----------------|-------------|-------------|----------------|-------------|-------------|
|                  | R               | P           | F           | R              | P           | F           |
| USTB_FuStar [48] | 69.5            | 74.4        | 71.9        | 68.0           | 72.4        | 70.1        |
| Saric [20]       | 65.9            | 77.3        | 70.8        | 67.7           | 80.2        | 72.8        |
| Proposed         | <b>69.9</b>     | <b>85.2</b> | <b>76.7</b> | <b>68.0</b>    | <b>80.1</b> | <b>73.5</b> |

TABLE 5: Stage evaluation: character-level results on the ICDAR2013 test dataset (%).

| Stage                    | Character-level rate (%) |      |      |
|--------------------------|--------------------------|------|------|
|                          | R                        | P    | F    |
| Candidate generation     | 91.0                     | 28.4 | 43.2 |
| Candidate classification | 67.3                     | 87.2 | 73.6 |
| Character retrieval      | 75.8                     | 83.2 | 79.3 |
| Character grouping       | 74.3                     | 85.9 | 79.6 |

as noncharacters. As expected, the retrieval stage recovers some characters that were misclassified; however, nontext components are also restored. Finally, the grouping stage discards noncharacters, which were recovered at the retrieved stage, as well as correct characters.

**4.3.3. Text Localization Evaluation.** Since most of the existing methods present text localization evaluation instead of character segmentation, we also carry out the same evaluation. Table 6 shows the text localization performance of the MSER-based techniques on the ICDAR2013 dataset. It can be seen that the proposed method shows better F-measure results than most other methods, except the techniques [17, 28] in which multiple image channels are used. However, the method [17] is designed for horizontal text only, decreasing its performance for multioriented text, while method [28] yields a lower F-measure than the proposed method with only grayscale images. Besides, the proposed method outperforms the latter one on the multioriented USTB-SV1K dataset (see Table 7).

Next, the performance of the proposed method and state-of-the-art algorithms [16, 20, 24, 28–30, 34, 37] on four datasets is evaluated using the protocol given in [34]. The results are shown in Table 7. One can observe that the proposed technique using only 299 training images outperforms the state-of-the-art methods on USTB and OSTD multioriented datasets. The performance of the methods [28, 29] drops by almost 30% compared with the

performance of these methods on the ICDAR2013 dataset containing horizontally aligned texts. Since the MSRA dataset has Chinese characters that we are not familiar with, we perform two evaluations of the proposed method: over the entire MSRA dataset and English text images of the dataset. Note that classifiers used in our method were only trained using Latin-based characters. For a fair comparison with other methods on this dataset, the proposed technique needs additional training with Chinese characters. It is of interest to note that the proposed method can detect parts of Chinese texts (see Figure 7). Although the deep learning-based method [30] outperforms the proposed method (for the complete test set), the authors report a decrease of 20% on F-measure using only the MSRA training set (300 images), thereby obtaining a lower F-measure than the proposed method.

Figures 8 and 9 show examples of correct text detection images and common errors of the proposed method in the USTB dataset, respectively. Three types of errors were found: the Google logo error (first row), where the proposed method recognized the Google watermark from the images; the unmarked text error (second row), where the proposed method recognized the text, but it was not considered as the text by the dataset ground truth; and the false positive and false negative errors (third row).

Finally, the average processing time of the proposed method was estimated using the ICDAR2013 dataset on a 2.8 GHz Intel Xeon E5-1603 PC with 16 GB of RAM. Table 8 summarizes the running time of all tested algorithms, as well

TABLE 6: Text localization evaluation on the ICDAR2013 test dataset (%).

| Method                                   | <i>R</i>    | <i>P</i>    | <i>F</i>    |
|--|-------------|-------------|-------------|
| Tian et al. (gray) [28]                  | 67.8        | 81.2        | 73.9        |
| Tian et al. (RGB + V) [28]               | 83.9        | 83.6        | 83.4        |
| Saric (gray) [20]                        | 67.7        | 80.2        | 72.8        |
| Wu et al. (RGB) [8]                      | 70.0        | 84.0        | 76.0        |
| Neumann and Matas (RGB + I + H + S) [18] | 71.3        | 82.1        | 76.3        |
| Yin et al. (gray) [16]                   | 65.1        | 83.9        | 73.3        |
| Sung et al. (gray + Cr + Cb) [17]        | 74.2        | 88.6        | 80.8        |
| Yin et al. (gray) [48]                   | 68.2        | 86.2        | 76.2        |
| Proposed method (gray)                   | <b>73.9</b> | <b>82.7</b> | <b>78.0</b> |

TABLE 7: Text detection comparison on ICDAR, MSRA, USTB, and OSTD datasets (%).

| Method                  | ICDAR    |          |             | MSRA     |          |             | USTB     |          |             | OSTD     |          |             |
|-------------------------|----------|----------|-------------|----------|----------|-------------|----------|----------|-------------|----------|----------|-------------|
|                         | <i>R</i> | <i>P</i> | <i>F</i>    |
| Ma et al. [30]          | 88.0     | 95.0     | <b>91.0</b> | 69.0     | 82.0     | 75.0        | —        | —        | —           | —        | —        | —           |
| Wei et al. [29]         | 81.1     | 87.3     | 84.3        | —        | —        | —           | 55.9     | 54.1     | 55.0        | 76.2     | 75.4     | 75.8        |
| He et al. [24]          | 81.0     | 92.0     | 86.0        | 70.0     | 77.0     | 74.0        | —        | —        | —           | —        | —        | —           |
| Tian et al. [28]        | 83.9     | 83.6     | 83.8        | —        | —        | —           | 48.7     | 53.8     | 51.1        | —        | —        | —           |
| Saric [20]              | 66.1     | 76.5     | 70.6        | —        | —        | —           | 31.8     | 44.6     | 37.1        | 45.4     | 49.8     | 47.5        |
| Yin et al. [16]         | 66.0     | 83.7     | 73.8        | 63.0     | 81.0     | 71.0        | 45.4     | 49.8     | 47.5        | —        | —        | —           |
| Li et al. [37]          | 62.0     | 80.0     | 70.0        | —        | —        | —           | —        | —        | —           | 60.0     | 72.0     | 61.0        |
| Yao et al. [34]         | 66.0     | 69.0     | 67.0        | 63.0     | 63.0     | 60.0        | —        | —        | —           | 73.0     | 77.0     | 74.0        |
| Proposed                | 73.9     | 82.7     | 78.0        | 63.9     | 74.3     | 65.6        | 58.8     | 68.8     | <b>63.1</b> | 89.0     | 90.1     | <b>88.0</b> |
| Proposed (English only) | 73.9     | 82.7     | 78.0        | 73.9     | 81.7     | <b>75.7</b> | 58.8     | 68.8     | 63.1        | 89.0     | 90.1     | 88.0        |



FIGURE 7: Examples of text detection on the MSRA-TD500 dataset. Green rectangle: ground truth; red rectangle: proposed text detection method.

as hardware features reported by the authors. Note that the processing time of the algorithms at each stage depends on various factors, such as hardware features, specific implementation of algorithms, size, and contextual complexity of processed images, which make a fair comparison difficult.

One can observe that methods [28, 30] achieved the best runtimes of recognition since GPU was utilized for implementation. Methods [18, 48] work only for the horizontal text, which reduces the computational complexity (runtime) of these methods. Note that all deep learning

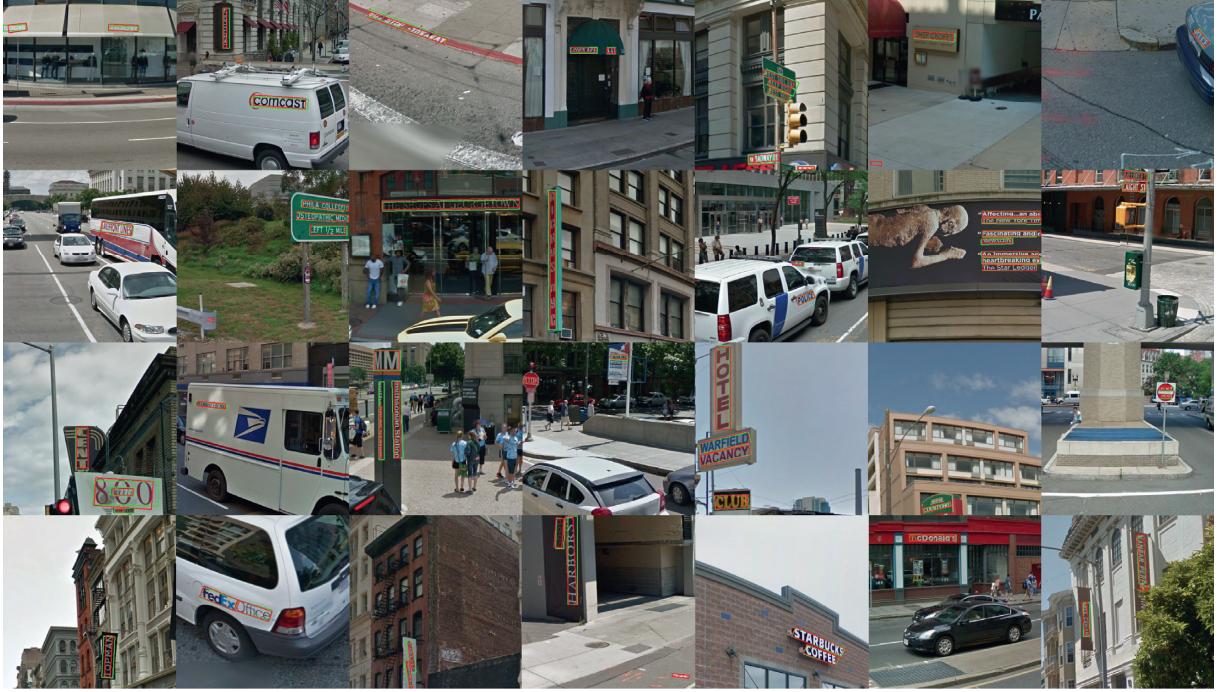


FIGURE 8: Correct detected text on the USTB-SV1K dataset. Green rectangle: ground truth; red rectangle: proposed text detection method.

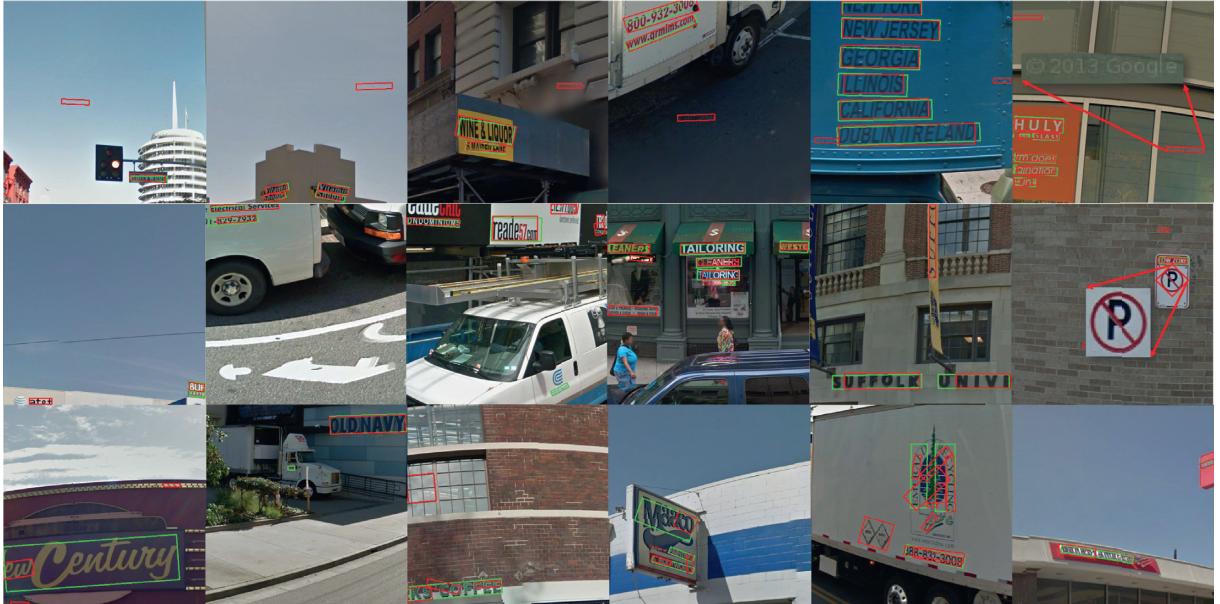


FIGURE 9: Text detection errors of the proposed method on the USTB-SV1K dataset. Green rectangle: ground truth; red rectangle: proposed text detection method.

TABLE 8: Running time on ICDAR2013.

| Method          | Features   | Time (s) |
|-----------------|--|----------|
| Ma et al. [30]  | NVIDIA TITAN X GPU                                     | 0.2      |
| Wei et al. [29] | 2.4 GHz Intel(R) Core (TM)<br>i7 4-core CPU, 16 GB RAM | 2.1      |
| He et al. [24]  | 2.9 GHz 12-core CPU<br>256 G RAM, GTX Titan X          | 0.9      |

TABLE 8: Continued.

| Method                 | Features  | Time (s) |
|------------------------|---|----------|
| Tian et al. [28]       | 3.40 GHz Inter(R) Core(TM)<br>i7-3770 CPU, 16 GB RAM<br>NVIDIA GeForce GT 630 GPU | 0.8      |
| Saric [20]             | 3 GHz Intel Core 2 Duo  | —        |
| Wu et al. [8]          | 4.0 GHz 4-core CPU, 32 GB RAM   | 8.0      |
| Neumann and Matas [18] | —   | 1.6      |
| Yin et al. [16]        | 2.20 GHz Linux laptop   | 1.4      |
| Sung et al. [17]       | —   | —        |
| Li et al. [37]         | —   | —        |
| Yin et al. [48]        | 2.20 GHz Linux laptop   | 0.8      |
| Yao et al. [34]        | 2.53 GHz CPU  | 7.2      |
| Proposed method        | 2.8 GHz Intel Xeon<br>E5-1603, 16 GB RAM  | 3.2      |

algorithms require significantly longer training time compared with the proposed method, which is reasonably fast for detection and segmentation even using a conventional computer without a graphics processor. Further optimization of the method implementation, as well as the use of GPU technology, can definitely reduce the overall processing time of our method.

## 5. Conclusion

In this paper, a novel multioriented text detection and segmentation method inspired by the human vision system was proposed. The method is based on the local energy model and the scale-space monogenic signal framework to extract essential local phase information. The proposed method consists of phase-based text segmentation, character retrieval, and character grouping stages. The phase-based candidate regions are extracted by applying the MSER algorithm to the local phase image; meanwhile, character retrieval and grouping are done by applying AdaBoost classifiers to avoid the use of heuristic rules.

The proposed method proved to be robust to geometric distortions, font variations, complex backgrounds, low contrast, high brightness, shadows, and illumination changes. The method achieves a high character segmentation performance possessing low computational complexity (number of extracted components). The method outperforms the state-of-the-art algorithms on typical databases in terms of character segmentation, text localization, and the number of candidate regions. Besides, our method is not restricted to only horizontal texts like most of the existing methods but also to multioriented texts.

Finally, the proposed method can be used for text detection in different languages or handwritten texts.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the RFBR (grant 18-08-00782).

## References

- [1] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.
- [2] Q. Ye and D. Doermann, "Text detection and recognition in imagery: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, 2015.
- [3] H. Zhang, K. Zhao, Y.-Z. Song, and J. Guo, "Text extraction from natural scene image: a survey," *Neurocomputing*, vol. 122, pp. 310–323, 2013.
- [4] P. Shivakumara, T. Q. Phan, and C. L. Tan, "New fourier-statistical features in RGB space for video text detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1520–1532, 2010.
- [5] S. Angadi and M. Kodabagi, "A texture based methodology for text region extraction from low resolution natural scene images," *International Journal of Image Processing (IJIP)*, vol. 3, no. 5, p. 229, 2009.
- [6] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1631–1639, 2003.
- [7] T. Saoi, H. Goto, and H. Kobayashi, "Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features," in *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pp. 690–694, IEEE, Seoul, South Korea, August 2005.
- [8] H. Wu, B. Zou, Y.-q. Zhao, Z. Chen, C. Zhu, and J. Guo, "Natural scene text detection by multi-scale adaptive color clustering and non-text filtering," *Neurocomputing*, vol. 214, pp. 1011–1025, 2016.
- [9] P. Tang, Y. Yuan, J. Fang, and Y. Zhao, "A novel similar background components connection algorithm for colorful text detection in natural images," in *Proceedings of the 2015 IEEE International Conference on Signal Processing*.

- Communications and Computing (ICSPCC)*, pp. 1–5, IEEE, Ningbo, China, September 2015.
- [10] Z. Liu and S. Sarkar, “Robust outdoor text detection using text intensity and shape features,” in *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1–4, IEEE, Tampa, FL, USA, December 2008.
- [11] S. Karaoglu, B. Fernando, and A. Tréneau, “A novel algorithm for text detection and localization in natural scene images,” in *Proceedings of the 2010 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 635–642, IEEE, Sydney, NSW, Australia, December 2010.
- [12] C. Yu, Y. Song, and Y. Zhang, “Scene text localization using edge analysis and feature pool,” *Neurocomputing*, vol. 175, pp. 652–661, 2016.
- [13] B. Epshtain, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2963–2970, IEEE, San Francisco, CA, USA, June 2010.
- [14] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.
- [15] J. Matas and K. Zimmermann, “A new class of learnable detectors for categorisation,” *Image Analysis*, pp. 541–550, 2005.
- [16] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, “Multi-orientation scene text detection with adaptive clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1930–1937, 2015.
- [17] M.-C. Sung, B. Jun, H. Cho, and D. Kim, “Scene text detection with robust character candidate extraction method,” in *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 426–430, IEEE, Tunis, Tunisia, August 2015.
- [18] L. Neumann and J. Matas, “Real-time lexicon-free scene text localization and recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1872–1885, 2016.
- [19] Y. Zheng, Q. Li, J. Liu, H. Liu, G. Li, and S. Zhang, “A cascaded method for text detection in natural scene images,” *Neurocomputing*, vol. 238, pp. 307–315, 2017.
- [20] M. Saric, “Scene text segmentation using low variation extremal regions and sorting based character grouping,” *Neurocomputing*.
- [21] X. Zhou, C. Yao, H. Wen et al., “East: an efficient and accurate scene text detector,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2642–2651, IEEE, Honolulu, HI, USA, July 2017.
- [22] X. Liu, D. Liang, S. Yan et al., “Fast oriented text spotting with a unified network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5676–5685, Salt Lake City, UT, USA, June 2018.
- [23] C.-Y. Lee and S. Osindero, “Recursive recurrent nets with attention modeling for ocr in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2231–2239, Las Vegas, NV, USA, June 2016.
- [24] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, “Deep direct regression for multi-oriented scene text detection,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.
- [25] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: high confidence predictions for unrecognizable images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, Boston, MA, USA, June 2015.
- [26] Y. F. Pan, X. Hou, C. L. Liu et al., “A hybrid approach to detect and localize texts in natural scene images,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 800–813, 2011.
- [27] Z. Zhao, C. Fang, Z. Lin, and Y. Wu, “A robust hybrid method for text detection in natural scenes by learning-based partial differential equations,” *Neurocomputing*, vol. 168, pp. 23–34, 2015.
- [28] C. Tian, Y. Xia, X. Zhang, and X. Gao, “Natural scene text detection with MC-MR candidate extraction and coarse-to-fine filtering,” *Neurocomputing*, vol. 260, pp. 112–122, 2017.
- [29] Y. Wei, W. Shen, D. Zeng, L. Ye, and Z. Zhang, “Multi-oriented text detection from natural scene images based on a cnn and pruning non-adjacent graph edges,” *Signal Processing: Image Communication*, vol. 64, pp. 89–98, 2018.
- [30] J. Ma, W. Shao, H. Ye et al., “Arbitrary-oriented scene text detection via rotation proposals,” *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.
- [31] E. Gladilin and R. Eils, “On the role of spatial phase and phase correlation in vision, illusion, and cognition,” *Frontiers in Computational Neuroscience*, vol. 9, p. 45, 2015.
- [32] J. Diaz-Escobar and V. Kober, “Scene text segmentation based on local image phase information and mser method,” in *Proceedings of the Mexican Conference on Pattern Recognition*, Springer, Querétaro, Mexico, pp. 211–220, June 2018.
- [33] J. Diaz-Escobar and V. Kober, “Text detection in natural scenes with phase congruency approach,” in *Applications of Digital Image Processing XL*, vol. 10396, International Society for Optics and Photonics, 2017.
- [34] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, “Detecting texts of arbitrary orientations in natural images,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1083–1090, IEEE, Providence, RI, USA, June 2012.
- [35] C. Yao, X. Bai, and W. Liu, “A unified framework for multi-oriented text detection and recognition,” *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.
- [36] W. Huang, Z. Lin, J. Yang, and J. Wang, “Text localization in natural images using stroke feature transform and text covariance descriptors,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1241–1248, Sydney, NSW, Australia, December 2013.
- [37] Y. Li, W. Jia, C. Shen, and A. van den Hengel, “Characterness: an indicator of text in the wild,” *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1666–1677, 2014.
- [38] M. Felsberg and G. Sommer, “The monogenic scale-space: a unifying approach to phase-based image processing in scale-space,” *Journal of Mathematical Imaging and Vision*, vol. 21, no. 1, pp. 5–26, 2004.
- [39] J. Diaz-Escobar, V. Kober, and J. A. Gonzalez-Fraga, “Luift: luminance invariant feature transform,” *Mathematical Problems in Engineering*, vol. 2018, pp. 1–17, 2018.
- [40] M. C. Morrone and R. A. Owens, “Feature detection from local energy,” *Pattern Recognition Letters*, vol. 6, no. 5, pp. 303–313, 1987.
- [41] M. C. Morrone and D. Burr, “Feature detection in human vision: a phase-dependent energy model,” *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 235, no. 1280, pp. 221–245, 1988.
- [42] P. Kovesi, “Image features from phase congruency,” *Videre: Journal of Computer Vision Research*, vol. 1, no. 3, pp. 1–26, 1999.

- [43] P. Kovesi, "Edges are not just steps," in *Proceedings of the Fifth Asian Conference on Computer Vision*, pp. 22–28, Melbourne, Australia, 2002.
- [44] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3538–3545, IEEE, Providence, RI, USA, June 2012.
- [45] A. Clavelli, D. Karatzas, and J. Lladós, "A framework for the assessment of text extraction algorithms on complex colour images," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 19–26, ACM, Boston MA, USA, 2010.
- [46] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 4, pp. 280–296, 2006.
- [47] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 20, no. 9, pp. 2594–2605, 2011.
- [48] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, 2014.