We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Image Segmentation Based on Mathematical Morphological Operator

Jianjun Chen, Haijian Shao and Chunlong Hu

Additional information is available at the end of the chapter

Abstract

Image segmentation is the process of partitioning a digital image into multiple regions (sets of pixels); the pixels in each region have similar attributes. It is often used to separate an image into regions in terms of surfaces, objects, and scenes, especially for object location and boundary extraction. Until now, many general-purpose algorithms and techniques have been proposed for image segmentation. Typical and traditional methods are: (1) threshold-based method; (2) edge-based method; and (3) region-based method. In this chapter, we propose an approach of image segmentation based on mathematical morphology operator: toggle operator. The experimental result shows that the proposed method can segment natural scene images into homogeneous regions effectively.

Keywords: homogeneous region, image segmentation, mathematical morphology, toggle operator

1. Introduction

Image segmentation is typically used to partition an image into meaningful parts. Thus, it has a significant application in image analysis and understanding. The result of image segmentation is a set of regions (each region is a set of pixels) that collectively cover the entire image, or a set of contours (i.e., boundaries, consisting of lines, curves, etc.) extracted from the image. The pixels in one region have similar characteristics in terms of color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s) [1, 2].

Until now, a great variety of algorithms have been proposed for image segmentation [3, 4]. These methods are generally classified into three categories: thresholding segmentation, edge-based segmentation, and region-based segmentation. Each method has its own advantages and disadvantages; there is no single method that can be applied effectively for segmenting all kinds of images. This is because different images have different features and properties.

Therefore, for different images, the segmentation requires different techniques. Images can be divided into the following categories according to their attributes and characteristics. From the color point of view, images include grayscale images, binary images, and color images; from the texture point of view, images include texture images and nontexture images. Based on the image features, the following subsections will consider some proposed classical approaches in more detail.

1.1. Threshold-based segmentation

Thresholding segmentation is a pixel-based method for image segmentation. It is the simplest method based on the variation of intensity between the object pixels and background pixels. Therefore, it is often used to separate out regions of an image corresponding to objects that we are interested in.

In order to differentiate the pixels that are located in the region of interest from the rest, a comparison is performed for each pixel intensity value with respect to a threshold. In this method, pixels are divided into two classes that are typically named "foreground" and "background." Pixels with values less than threshold are placed in one class, and the rest are placed in the other class. Therefore, this method is often used to convert a grayscale image into a binary image.

$$I_{bw}(x,y) = \begin{cases} 1 & I_{gray}(x,y) \geq T \\ 0 & I_{gray}(x,y) < T \end{cases} \tag{1}$$

where I_{gray} is the grayscale image, I_{bw} is the binary image, (x,y) is the coordinate of target pixel, and T is the threshold value. This method is most effective for images with high levels of contrast. However, the key of this method is to select the well-suited threshold value.

Many researchers contribute to the work for automated selecting the threshold T based on the computer. Their thresholding methods can be categorized into the following groups based on the information the algorithm manipulates [5–9]:

(1) Histogram shape-based methods: the peaks, valleys, and curvatures of the smoothed histogram are analyzed.

    a. Double-peak threshold method: suppose the histogram of the image is a bimodal distribution (regions with uniform intensity give rise to strong peaks in the histogram), then the value of valley point can be chosen as the threshold.

    b. Minimum variance method: suppose that a region has relatively homogeneous gray values, it will make sense to select a threshold that minimizes the variance of the gray values within regions.

    c. Maximum variance method (usually called the Otsu method [10]): chooses a good threshold value by maximizing the variance between objects and background. The histogram can be divided into two classes, while the interclasses' variance is maximized.

(2) Clustering method: representing the image as a set of clusters, an ideal threshold value is determined by iteratively reducing the clusters until there are two clusters left. The two remaining clusters are the background and the foreground (object).

(3) Maximum-entropy-based method: determines an ideal threshold value by maximizing entropy for the probability distribution of the foreground and background regions, which are represented in the histogram.

(4) Local thresholding method: chooses the threshold value for each pixel according to the local image characteristics. In this method, a different threshold value is selected for each pixel in the image.

(5) Watershed method: considers the gradient magnitude of an image as a topographic surface where high gradient denotes peaks, while low gradient denotes valleys. Start by filling every isolated valley with different colored water. As the water rises, water from different valleys will start to merge. To avoid that, barriers are built in the locations where water merges. Continue the work of filling water and building barriers until all the peaks are under water. Then the created barriers give the segmentation result.

Thresholding methods are applied to segment, not only grayscale images but also color images. For color images, one approach is to determine a separate threshold for each color channel of the image and then combine them with an AND operation. Segmentation based on color information (e.g., RGB, HSL, and HSV color models) may be more accurate than gray-scale images [11, 12].

1.2. Edge-based segmentation

An edge is the boundary between two regions with different properties; it represents the change from one object or surface to another. Edges are used to characterize the physical extent of objects, since there is often a sharp adjustment in intensity at the region boundaries. Thus, detection of edges is a very important step toward image feature understanding; it is often used to divide images into areas corresponding to different objects.

The main idea of most edge-detection techniques is the computation of a local derivative of an image, including first- and second-order derivatives. The first-order derivative of choice in image processing is the gradient; it can be used to detect the presence of an edge in an image. Second-order derivatives in image processing generally are computed using the Laplacian. The sign of the second derivative can be used to determine whether an edge pixel is on the dark or light side of an edge [13–17]. Gradient operator and Laplacian operator are defined as follows:

(1) Gradient operators: the gradient of an image, $f(x, y)$, at location (x, y) is defined as the vector.

$$\nabla f(x, y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \qquad (2)$$

The magnitude of this vector is computed from.

$$|\nabla f(x,y)| = \mathrm{mag}(\nabla f(x,y)) = \left(g_x^2 + g_y^2\right)^{\frac{1}{2}} = \left[(\partial f/\partial x)^2 + (\partial f/\partial y)^2\right]^{\frac{1}{2}} \tag{3}$$

To simplify this computation, this quantity is approximated by using the following expression:

$$\nabla f \approx |g_x| + \left|g_y\right| \tag{4}$$

The direction angle of gradient vector is the maximum rate of change of f at coordinates (x,y):

$$\theta(x,y) = \arctan\left(\frac{g_y}{g_y}\right) \tag{5}$$

For a discrete image, the gradient can be calculated by the following expressions:

$$|\nabla f(x,y)| = \left\{[f(x,y) - f(x+1,y)]^2 + [f(x,y) - f(x,y+1)]^2\right\}^{\frac{1}{2}} \tag{6}$$
$$\approx |f(x,y) - f(x+1,y)| + |f(x,y) - f(x,y+1)|$$

or

$$|\nabla f(x,y)| = \left\{[f(x,y) - f(x+1,y+1)]^2 + [f(x+1,y) - f(x,y+1)]^2\right\}^{\frac{1}{2}} \tag{7}$$
$$\approx |f(x,y) - f(x+1,y+1)| + |f(x+1,y) - f(x,y+1)|$$

Edges can be detected with the help of first-order derivative type operators, as follows:

a. Sobel edge detector: using convolutions with row and column edge gradient masks, it is suitable to detect edges along the horizontal and vertical axis.

b. Robert detection: calculates the square root of the magnitude squared of the convolution with the row and column edge detectors; it is able to detect edges that run along the vertical axes of 45° and 135°.

c. Prewitt operator: detects edges by calculating the Prewitt compass gradient filters that return the result for the largest filter response.

d. Kirsch edge detector: performs convolution using eight filters that are applied to calculate gradient.

e. Frei-Chen edge operator: uses only the row and column filters.

The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. However, these gradient operators tend to be sensitive to noise.

(2) Laplacian operator: the Laplacian of an image $f(x,y)$ is defined as

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \tag{8}$$

where $\frac{\partial^2 f(x,y)}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$, $\frac{\partial^2 f(x,y)}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$, so that

$$\nabla^2 f(x,y) = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y) \tag{9}$$

The Laplacian is seldom used directly for edge detection because, as a second-order derivative, it is unacceptably sensitive to noise, its magnitude produces double edges, and it is unable to detect edge direction. In order to improve the effectiveness of edge detection, the following algorithms are proposed:

a. Laplacian of Gaussian (LoG): Gaussian filtering is combined with Laplacian to break down the image where the intensity varies, to detect the edges effectively [16]. However, this operator cannot find the orientation of edge because of using the Laplacian filter [17].

b. Canny edge detector (colored edge detectors): uses a multistage algorithm to detect a wide range of edges in images. First, Gaussian convolution filter is applied to smooth the images in order to reduce the noise. Second, a first-derivative operator (e.g., Sobel, Robert algorithm) is applied to the smoothed image in order to output a gradient magnitude image. Third, a process of nonmaximum suppression is applied to get rid of spurious response to edge detection in order to give a thin line in the output. Fourth, a double threshold is applied to determine potential edges. Last, tracking edges by hysteresis helps to finalize the detection by suppressing all the other edges. Canny edge detector performance is good; the only drawback is that it takes more time to compute and it is more complex [18, 19].

1.3. Region-based segmentation

An edge-based technique may attempt to find the object boundaries and then locate the object itself by filling them in; a region-based technique takes the opposite approach.

Region-based segmentation algorithms operate iteratively by grouping together neighboring pixels that have similar properties (such as gray level, texture, color, shape) and splitting groups of pixels that are dissimilar in value [20, 21]. There are a variety of approaches of region-based segmentation. These methods can be classified into two categories:

(1) Region growing method: This is the simplest region-based segmentation method. It is also classified as a pixel-based segmentation method since it involves the selection of initial seed points.

This approach to segmentation groups of pixels or subregions into larger regions is based on predefined criteria. First, a set of seed points is selected based on some user criterion (e.g., pixels in a certain grayscale range). Second, the regions are grown from these seed pixels to neighboring pixels, which are examined to ascertain if they should be added to the region,

according to a region membership criterion (e.g., pixel intensity, grayscale texture, or color). Third, the second process is iterated on, in the same manner as general data clustering algorithms.

Region-growing-based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect. However, it is computationally expensive.

(2) Split-and-merge method: This method consists of two steps: region splitting and region merging.

Region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.

Region merging is the opposite of region splitting and works as a way of avoiding over-segmentation. It starts with small regions and merges the regions that have similar characteristics (such as gray level, variance).

There are a great number of various approaches [22–26] excepting for the methods described above, also including the improvements of above methods [27–37]; for example, matching-based segmentation, clustering-based segmentation, fuzzy-inference-based segmentation, generalized PCA (principal component analysis)-based segmentation. Each segmentation method has its advantages and disadvantages. A universal algorithm of segmentation does not exist, as each type of image corresponds to a specific approach. Therefore, choice of technique depends on peculiar characteristic of individual problems. The emphasis of this paper lies on an improved method of scene image segmentation based on mathematical morphological operator-toggle operator.

This chapter is organized as follows: Section 1 presents an overview of methodologies and algorithms for image segmentation. A new proposed image segmentation method is then introduced in Section 2. In Section 3, the experimental results are analyzed to prove the validity of the proposed method. Finally, the chapter concludes in Section 4.

2. Scene image segmentation based on mathematical morphology

Signs and public notices are ubiquitous indoors and outdoors, and they are often used for route finding, finding public places and other locations. The texts in natural scene images contain important information. Therefore, text detection has attracted wide interest due to its usefulness in a variety of real-world applications, such as robots navigation, assisting visually impaired people, tourist navigation, enhancing safe vehicle driving, and so on [38, 39]. To date, a great number of algorithms have been proposed for detecting text on scene images or video [40–49]. However, most approaches proposed in the past research contribute to detect the text regions by analyzing the entire image. The image is segmented into text regions and non-text regions according to their features, respectively. The performance of these methods relies on the text detection algorithm and image complexity. Actually, scene text is usually presented on signboards. Because of the uniform color for the background of signboard, the ideal way for

extracting text from scene images is to cut out the signboard regions first, and then detect text from the signboard regions. Thus, this chapter aims to propose an algorithm for segmenting a natural scene image into homogeneous regions. In our method, we first perform the image segmentation in order to detect homogeneous regions. Signboard regions are then detected with a simple criterion in order to remove the noise, such as trees and other non-signboard areas. In the following subsections, the proposed method is described and discussed in detail.

2.1. Image smoothing preprocessing

A natural scene image, I_{rgb}, is supposed to be a bitmap image based on the RGB (red-green-blue) color model. A smoothing process, edge preserving smoothing filter (EPSF) [20], is first applied to I_{rgb}.

The EPSF is applied independently to every pixel using different coefficients as shown in the following convolution mask:

$$\frac{1}{\sum\limits_{i=1}^{8} c_i} \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & 0 & c_5 \\ c_6 & c_7 & c_8 \end{bmatrix} \tag{10}$$

where c_i $(i = 1, 2, 3, \cdots, 8)$ are calculated using the following equation:

$$c_i = (1 - d_i)^p, \text{where } p \geq 1 \tag{11}$$

where d_i $(i = 1, 2, 3, \cdots, 8)$ are the Manhattan color distances, which are extracted between the target pixel and the 8 neighboring pixels in a 3×3 window. That is,

$$d_i = \frac{\left| I_R^0 - I_R^i \right| + \left| I_G^0 - I_G^i \right| + \left| I_B^0 - I_B^i \right|}{3 \times 255} \tag{12}$$

where I_R^0, I_G^0, I_B^0 are the RGB color values of the target pixel, and I_R^i, I_G^i, I_B^i are the RGB color values of the ith neighboring pixel.

The filtering of the image is achieved by applying the convolution mask, Eq. (10), on each of the three color channels. Factor p in Eq. (11) scales exponentially the color differences; it controls the amount of blurring performed on the image. A fixed value $p = 13$ is used for all of our experiments because this results in very good performance. The target pixel of the convolution mask is set to zero to remove impulsive noise.

Finally, a smoothed image, I_{EPSF}, is obtained. I_{EPSF} is then converted to a grayscale image I_{gray} through the following equation:

$$I_{gray}(x, y) = 0.2126 \cdot I_{EPSF}^R(x, y) + 0.7152 \cdot I_{EPSF}^G(x, y) + 0.0722 \cdot I_{EPSF}^B(x, y) \tag{13}$$

where (x, y) is the coordinate of target pixel. $I_{EPSF}^{R}(x, y)$, $I_{EPSF}^{G}(x, y)$, and $I_{EPSF}^{B}(x, y)$ are the intensities for red, green, and blue, respectively, of the smoothed image I_{EPSF}.

2.2. Homogeneous region segmentation

A measure of region homogeneity is variance (i.e., regions with high homogeneity will have low variance). In this section, a mathematical morphological operator, Toggle Mapping (TM) [34], is introduced to segment a grayscale image into homogenous regions according to the pixel intensity. This is a simple way to segment a grayscale image into homogeneous regions based on a toggle operator. Such operator is defined as follows:

$$H(x, y) = \begin{cases} 1 & if \ D(x, y) - E(x, y) \leq th \\ 0 & if \ D(x, y) - E(x, y) > th \end{cases} \tag{14}$$

where H is a binary image taking two values, th is a threshold value, and D and E are the dilation image and erosion image of input image, respectively.

In order to meet the needs of the application, Dorini [35] and Fabrizio [36] have modified and improved this operator by adding new factors or weight coefficients. In their algorithms, the toggle operator is used one time for segmenting an image, so the values of thresholds and coefficients are fixed in their algorithms. However, for different images, the optimal values for thresholds and coefficients should be different. In order to overcome over-segmentation or under-segmentation, we propose a new algorithm for grayscale image segmentation. In our method, the toggle operator is applied iteratively on input image, and the value of threshold is changed in each iteration step. This is because, while applying Eq. (14) on a grayscale image, in the output image, the area of connected component will increase with the increase of threshold value. **Figure 1** shows an example for the increment of threshold value. Based on such feature, we propose an approach that tries to search for homogeneous regions by calculating the standard deviation of intensity for connected components. The detail procedure of our proposed algorithm is described in the following steps.

Step 1: Initializing $\leftarrow \phi$, $THSD = 20$, $ThRatio = 0.99$, $i \leftarrow 1$, $th \leftarrow i$, and applying Eq. (14) on I_{gray} then gets a set of connected components $CC^{(1)} = \left\{ C_1^{(1)}, C_2^{(1)}, \cdots, C_{n(1)}^{(1)} \right\}$, where $n(1)$ is the number of elements of $CC^{(1)}$.
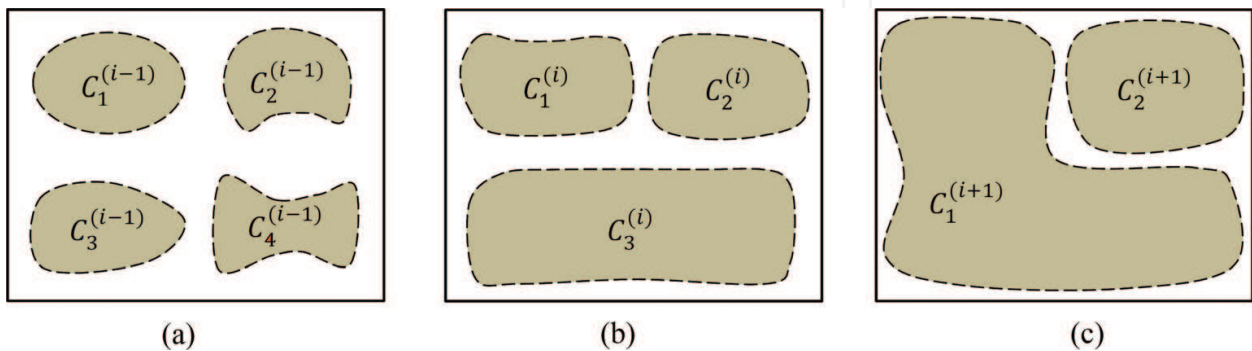

Figure 1. An example for result of Toggle Mapping (a) $th = i-1$, (b) $th = i$, (c) $th = i + 1$.

Figure 2. An example for natural scene image segmentation (a) Original Image, (b) Grayscale Image, (c) Labeling Image for Homogeneous Regions.

Step 2: Updating $i \leftarrow i + 1$, $th \leftarrow i$, and performing Eq. (14) on I_{gray} then gets a set of connected components $CC^{(i)} = \left\{ C_1^{(i)}, \ C_2^{(i)}, \ \cdots, \ C_{n(i)}^{(i)} \right\}$, where $n(i)$ is the number of elements of $CC^{(i)}$. Setting $j \leftarrow 1$.

Step 3: For connected component $C_j^{(i)}$, where $C_j^{(i)} \in CC^{(i)}$, calculating its standard deviation of intensity $SD\_C_j^{(i)}$. If $SD\_C_j^{(i)} > THSD$, go to Step 4; else go to Step 5.

Step 4: Supposing $C_j^{(i)} \supseteq \left\{ C_{j(1)}^{(i-1)}, C_{j(2)}^{(i-1)}, \ \cdots, C_{j(m)}^{(i-1)} \right\}$, it means that $C_{j(1)}^{(i-1)}, \ C_{j(2)}^{(i-1)}, \ \cdots, \ C_{j(m)}^{(i-1)}$ are covered by $C_j^{(i)}$, where $\left\{ C_{j(1)}^{(i-1)}, \ C_{j(2)}^{(i-1)}, \ \cdots, \ C_{j(m)}^{(i-1)} \right\}$ is a subset of $CC^{(i-1)}$. Updating $U \leftarrow U \cup \left\{ C_{j(1)}^{(i-1)}, C_{j(2)}^{(i-1)}, \ \cdots, C_{j(m)}^{(i-1)} \right\}$ and $CC^{(i)} \leftarrow CC^{(i)} - \left\{ C_j^{(i)} \right\}$.

Step 5: Updating $j \leftarrow j + 1$. If $j \leq n(i)$, go to Step 3; else go to Step 6.

Step 6: Calculating the number of pixels $NP\_CC^{(i-1)}$ for $CC^{(i-1)}$ and $NP\_CC^{(i)}$ for $P\_CC^{(i)}$, respectively. If $\frac{\min\left(NP\_CC^{(i-1)}, \ NP\_CC^{(i)} \right)}{\max\left(NP\_CC^{(i-1)}, \ NP\_CC^{(i)} \right)} > ThRatio$, go to Step 7; else go to Step 2.

Step 7: Terminating. Finally, U is a set of homogeneous regions.

As shown in **Figure 2**, the natural scene image can be segmented into homogeneous regions. The result showed that our proposed method can work effectively with high accuracy.

3. Experiment and results

3.1. Experimental images

In our experiment, 500 natural scene images are captured with various signboards, shop names, traffic signs, and more. All the original images are saved in RGB24 bitmap format with

a size of 1000×1500 pixels. In order to provide a wide range of real-life scenarios, images are captured with different compact digital cameras at different angles, positions, and under variable lighting and weather conditions. **Figure 3** shows some examples used in this experiment. **Table 1** shows the experimental environment.

3.2. Evaluation of image segmentation

In this subsection, our proposed method is compared to the methods of watershed segmentation using gradient [1], Canny edge detection [18], and region growing [1]. In order to evaluate the accuracy of our proposed method. There are many parameters included in not only our method but also the other three methods. Therefore, 100 images, selected from the total 500 images, are used for training and deciding the value of parameters based on the grid search method. The remaining 400 images that differ from the 100 training images are used to do the experiment in order to evaluate the accurancy of segmentation.

(a)   (b)   (c)

(d)   (e)   (f)

Figure 3. Examples of scene images.

OS	Microsoft Windows 10 Enterprise
CPU	Intel(R) Xeon(R) E5–2620 v4 2.10GHz (dual processor)
Memory	128GB
Programming Language	MATLAB

Table 1. Experimental environment.

The purpose of our research is to support visually impaired people to access the scene text. This paper aims to segment the natural scene images into homogenous regions. This is because, after the segmentation, specified criteria can be applied to select the signboard regions and the text can then be extracted from these regions. Therefore, in the experiment, we only focus on the accuracy of signboard segmentation.

The result of segmentation relies on not only segmentation algorithms but also the quality of the images. From the observation, the signboard regions can be segmented into four categories: PERFECT, FRAGMENT, EXCALATION, and FRAGMENT and EXCALATION.

PERFECT: a signboard has been segmented correctly, as shown in **Figure 4(a)**.

FRAGMENT: a signboard is segmented out in fragments, as shown in **Figure 4(b)**, where the extracted results are part of one signboard.

EXCALATION: one part of the signboard is extracted, but the others are lost, as shown in **Figure 4(c)**, where the extracted region is part of one signboard.

EXCALATION and FRAGMENT: one part of a signboard is segmented into fragments, but the other part is lost, as shown in **Figure 4(d)**.

In the experiment, PERFECT and FRAGMENT are evaluated as correct results, EXCALATION and EXCALATION and FRAGMENT are evaluated as incorrect results.

There are 482 target signboards in 400 experimental images. After the experiment, the accuracy of signboard segmentation and the average image processing time are calculated, respectively. The results are shown in **Table 2**.

As shown in **Table 2**, the average processing time of our proposed method is not so short. This is because our algorithm iteratively applies the toggle operator to segment image and find homogeneous regions. So, it is time-consuming. For the region-growing method, it first searches the seeds in an image and then performs the growing processing. This is also time-consuming.
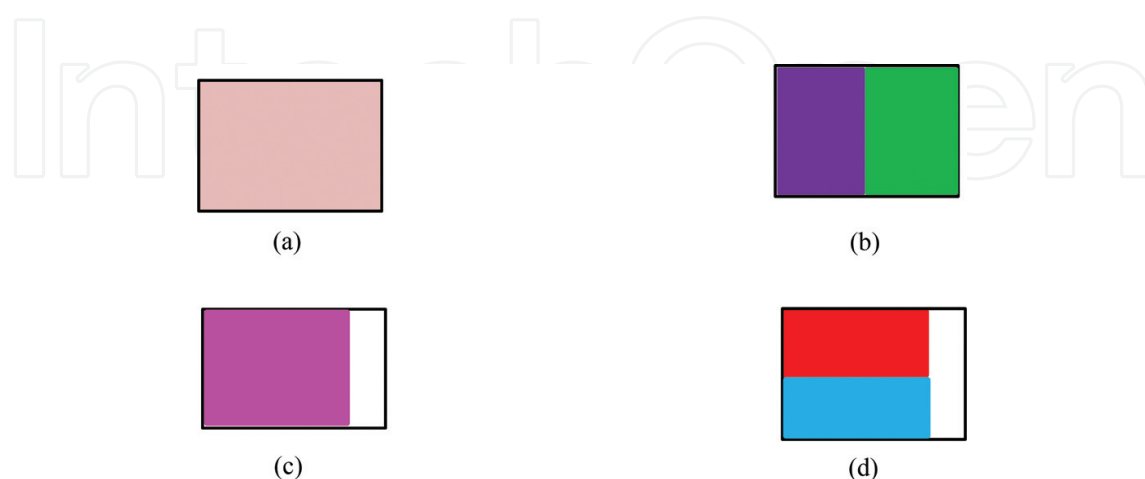

(a)   (b)

(c)   (d)

Figure 4. Cases of signboard segmentation (a) Complete Signboard, (b) Fragmentary Signboard, (c) Partial Signboard, (d) Partially & Fragmentary.

Methods	Accuracy (%)	Average execution time (s)
Our method	94.5	3.21
Watershed segmentation	83.6	2.82
Canny edge detector	91.4	0.73
Region growing	92.2	4.08

Table 2. Segmentation accuracy and average processing time.

Figures 5–8 are the segmentation results of **Figure 3**, by applying our menthod, watershed segmentation, Canny edge detection, and region-growing method, respectively. From the observation, our proposed algorithm can segment an image into homogeneous regions effectively, and some results are better than those applying the Canny edge detector and the region-growing method, because of the Canny edge operator not always detecting the closed boundary of object and the result of region-growing method deeply depending on the initial seeds selection.

Each method can achieve a high accuracy value if the quality of images is very good. But if the images include much noise, the accuracy of segmentation is very low any method. The signboard regions cannot be segmented completely due to the following reasons: (1) the surface of signboard is corroded, for example, **Figure 9(a)**; (2) shadow exists on the signboard, for example, **Figure 9(b)**; and (3) reflective effect, for example, **Figure 9(c)**.

(a)   (b)   (c)
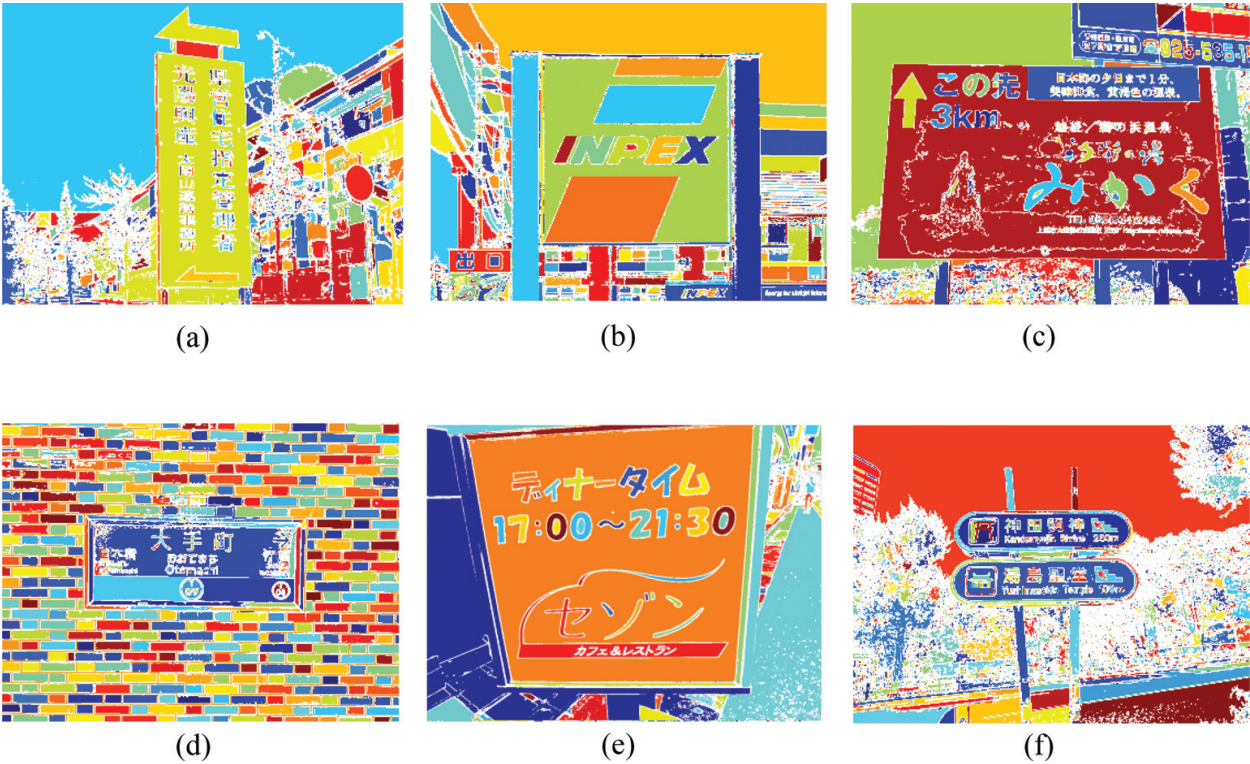
(d)   (e)   (f)

Figure 5. Result of our method for segmenting **Figure 3** (a) Homogeneous Regions, (b) Homogeneous Regions, (c) Homogeneous Regions, (d) Homogeneous Regions, (e) Homogeneous Regions, (f) Homogeneous Regions.
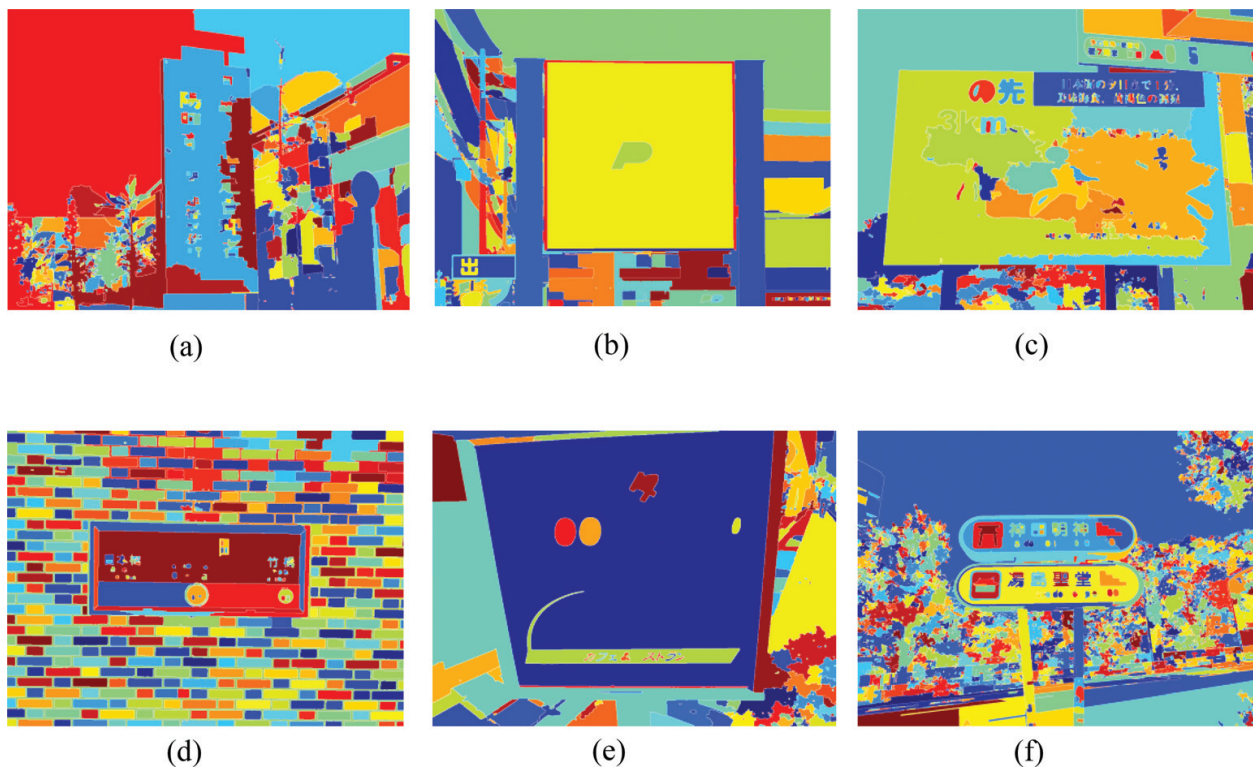
(a)

(b)

(c)

(d)

(e)

(f)

Figure 6. Watershed segmentation result of **Figure 3** (a) Homogeneous Regions, (b) Homogeneous Regions, (c) Homogeneous Regions, (d) Homogeneous Regions, (e) Homogeneous Regions, (f) Homogeneous Regions.
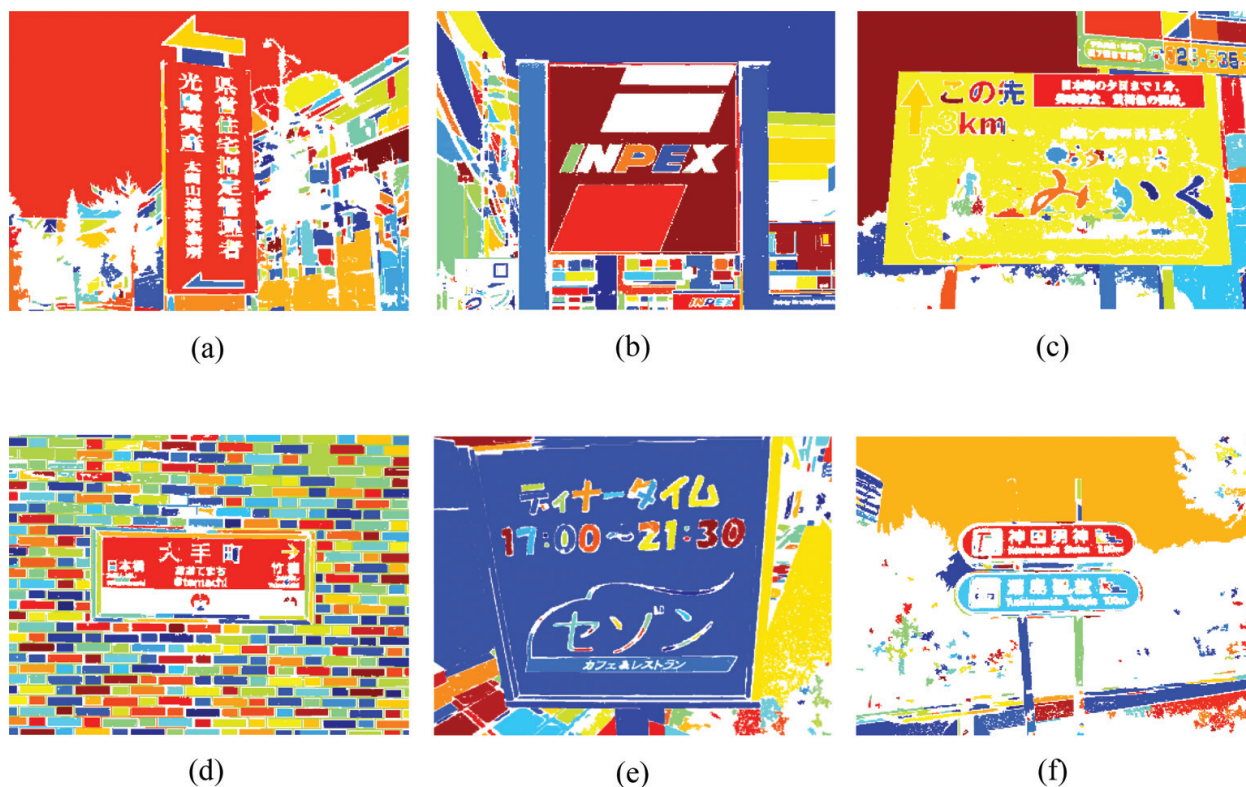

(a)

(b)

(c)

(d)

(e)

(f)

Figure 7. Canny edge detector for segmentation of **Figure 3** (a) Homogeneous Regions, (b) Homogeneous Regions, (c) Homogeneous Regions, (d) Homogeneous Regions, (e) Homogeneous Regions, (f) Homogeneous Regions.

Figure 8. Region growing for segmentation of **Figure 3** (a) Homogeneous Regions, (b) Homogeneous Regions, (c) Homogeneous Regions, (d) Homogeneous Regions, (e) Homogeneous Regions, (f) Homogeneous Regions.
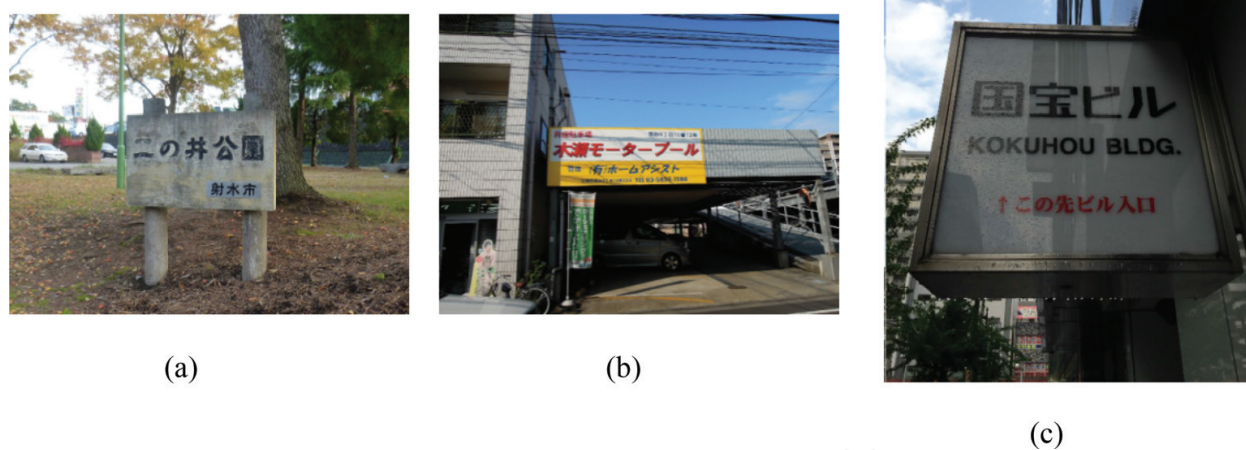

Figure 9. Examples of scene images.

4. Summary

This chapter proposes a method of mathematical morphology-based natural scene image segmentation. First, a number of typical segmentation algorithms are reviewed and discussed, and the objective of this chapter was introduced. Second, our proposed method was described. Third, the experiment was done and discussed.

The proposed method was tested on different images, and the results showed that our method can be an effective way for scene images segmentation. However, the results indicated that the signboard regions were extracted with low accuracy due to the presence of shadows or even corroded signboards.

In order to improve the accuracy of segmentation results, in the near future, we will introduce techniques for removing shadows and reflections in images.

Acknowledgements

Author details

Jianjun Chen[1,2]*, Haijian Shao[1] and Chunlong Hu[1]

*Address all correspondence to: kengunchin@foxmail.com

1 Jiangsu University of Science and Technology, Zhenjiang, China

2 Toyama Prefectural University, Imizu, Japan

References

[1] Gonzalez RC, Woods RE. Digital Image Processing. (3rd edition), Pearson International 3rd ed. New Jersey, NJ: Pearson Prentice Hall; 2008

[2] Shapiro LG, Stockman GC. Computer Vision. New Jersey: Prentice-Hall; 2001. pp. 279-325

[3] Zhang YJ, Gerbrands JJ. Objective and quantitative segmentation evaluation and comparison. Signal Processing. 1994;**39**(1–2):43-54

[4] Zhang YJ. A survey on evaluation methods for image segmentation. Pattern Recognition. 1996;**29**(8):1335-1346

[5] Sezgin M, Sankur B. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging. 2004;**13**(1):146-165

[6] Tobias OJ, Seara R. Image segmentation by histogram thresholding using fuzzy sets. IEEE Transactions on Image Processing. 2002;**11**(12):1457-1465

[7] Luessi M, Eichmann M, Schuster GM, Katsaggelos AK. Framework for efficient optimal multilevel image thresholding. Journal of Electronic Imaging. 2009;**18**(1):013004-1-013004-10

[8] Lai YK, Rosin PL. Efficient circular thresholding. IEEE Transaction on Image Processing. 2014;**23**(3):992-1001

[9] Saini R, Dutta M, Kumar R. A comparative study of several image segmentation techniques. Journal of Information and Operations Management. 2012;**3**(1):21-24

[10] Otsu N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics. 1979;**9**(1):62-66

[11] Pham N, Morrison A, Schwock J, et al. Quantitative image analysis of Immunohistochemical stains using a CMYK color model. Diagnostic Pathology. 2007:2-8

[12] Plataniotis KN, Venetsanopoulos AN. Color Image Processing and Applications. Berlin, Heidelberg: Springer; 2000

[13] Senthilkumaran N, Rajesh R. Edge detection techniques for image segmentation—A survey of soft computing approaches. International Journal of Recent Trends in Engineering. 2009;**1**(2):250-254

[14] Senthilkumaran N Rajesh R Edge detection techniques for image segmentation—A surveyIn: Proceedings of the International Conference on Managing Next Generation Software Applications. 2008. pp. 749–760

[15] Torre V, Poggio T. On edge detection. IEEE Transaction Pattern Analysis Machine Intelligence. 1986;**8**(4):147-163

[16] Acharjya PP, Das R, Ghoshal D. A study on image edge detection using the gradients. International Journal of Scientific and Research Publications. 2012;**2**(12):1-5

[17] Yogamangalam R, Karthikeyan B. Segmentation techniques comparison in image processing. International Journal of Engineering and Technology. 2013;**5**(1):307-313

[18] Canny J. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986;**PAMI-8**(6):679-698

[19] Lakshmi S, Sankaranarayanan V. A study of edge detection techniques for image segmentation computing approaches. Special Issue on Computer Aided Soft Computing Techniques for Image and Biomedical Applications. 2010:35-41

[20] Nikolaou N, Papamarkos N. Color reduction for complex document images. International Journal of Imaging Systems & Technology. 2009;**19**:14-269

[21] Manjula KA. Role of image segmentation in digital image processing for information processing. International Journal of Computer Science Trends and Technology. 2015;**3**(3):312-318

[22] Liu D, Ma L, Chen H, Meng K Medical image segmentation based on improved fuzzy C-means clustering In: Proceedings of the International Conference on Smart Grid and Electrical Automation. 2017. pp. 406-410

[23] Fabijańska A Corneal endothelium image segmentation using feedforward neural networkIn: Proceedings of the Federated Conference on Computer Science and Information Systems; 2017. pp. 629-637

[24] Liu Y, Cao F, Zhao J, Chu J. Segmentation of white blood cells image using adaptive location and iteration. IEEE Journal of Biomedical and Health Informatics. 2017;**21**(6): 1644-1165

[25] Liu W, Qing X, Zhou J. A novel image segmentation algorithm based on visual saliency detection and integrated feature extraction. In: Proceedings of the International Conference on Communication and Electronics Systems; 2016. pp. 1-5

[26] Swarnalatha A, Manikandan M Review of segmentation techniques for intravascular ultrasound (IVUS) images In: Proceedings of the Fourth International Conference on Signal Processing, Communication and Networking. 2017. pp. 1-4

[27] Wangchamhan T, Chiewchanwattana S, Sunat K. Multilevel thresholding selection based on chaotic multi-verse optimization for image segmentation. In: Proceedings of the 13th International Joint Conference on Computer Science and Software Engineering; 2016. pp. 1-6

[28] Zhou X, Li Y, Shen L. A novel adaptive local thresholding approach for segmentation of HEp-2 cell images. In: Proceedings of the IEEE International Conference on Signal and Image Processing; 2016. pp. 174-178

[29] Mirah Kasturi M, Sutha P, Jayanthi VE. efficient edge detection method for diagnosis of 2D and 3D lung and liver images. In: Proceedings of the International Conference on Inventive Systems and Control; 2017. pp. 1-5

[30] Hien NM, Binh NT, Viet NQ Edge detection based on fuzzy c means in medical image processing systemn: Proceedings of the International Conference on System Science and Engineering; 2017. pp. 12-15

[31] Günen MA, Atasever ÜH, Beşdok E. A novel edge detection approach based on backtracking search optimization algorithm (BSA) clustering. In: Proceedings of the 8th International Conference on Information Technology; 2017. pp. 116-122

[32] Albano JA, Ziemann AK, Messinger DW. Assessing the impact of the edge-weighting function in a graph-based approach to anomaly detection. In: Proceedings of the 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing; 2013. pp. 1-4

[33] Bi C, Yuan Y, Zhang R, Xiang Y, Wang Y, Zhang J. A dynamic mode decomposition based edge detection method for art images. IEEE Photonics Journal. 2017;**99**:1-1

[34]   Liu Y, Cheng M, Hu X, Wang K, Bai X. Richer convolutional features for edge detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 5872-5881

[35]   Manisha, Radhakrishnan B, Padma Suresh L. Tumor region extraction using edge detection method in brain MRI images. In: Proceedings of the International Conference on Circuit, Power and Computing Technologies; 2017. pp. 1-5

[36]   Grinias, Tziritas G. Region-level moving object segmentation by graph labeling. In: Proceedings of the 13th European Signal Processing Conference; 2005. pp. 1-4

[37]   Liu Z, Wu X, Zhao B, Peng Q. Clothing extraction using region-based segmentation and pixel-level refinement In: Proceedings of the IEEE International Symposium on Multimedia 2014. pp. 303-310

[38]   Wang S, Fu C, Li Q. Text detection in natural scene image: A survey. In: Proceedings of the International Conference on Machine Learning and Intelligent Communications; 2017. pp. 257-264

[39]   Liang J, Doermann D, Li H. Camera-based analysis of text and documents: A survey. International Journal of Document Analysis and Recognition. 2005;**7**(2–3):84-104

[40]   Matsuda Y, Omachi S, Aso H. String detection from scene images by binarization and edge detection. Institute of Electronics, Information and Communication Engineers, D. 2010;**J93-D**(3):336-344

[41]   Hase H, Yoneda M, Sakai M, Maruyama H. Consideration of color segmentation to extract character areas from color document images. Institute of Electronics, Information and Communication Engineers, DII. 2000;**J83-D-II**(5):1294-1304

[42]   Ashida K, Nagai H, Okamoto M, Miyao H, Yamamoto H. Extraction of characters from scene images. Institute of Electronics, Information and Communication Engineers, DII. 2005;**J88-D-II**(9):1817-1824

[43]   Saitoh S, Goto H, Kobayashi H. Analysis and comparison of frequency features for scene text detection. In: Technical Report of IEICE, PRMU2004-128. 2004. pp. 31-36

[44]   Crandall D, Antani S, Kasturi R. Extraction of special effects caption text events from digital video. International Journal on Document Analysis and Recognition. 2003;**5**:138-157

[45]   Kalai Selvi U, Anish Kumar J. Camera based assistive text reading system using gradient and stroke orientation for blind person. International Journal of Latest Trends in Engineering and Technology. 2014;**4**(1):325-330

[46]   Yao C, Bai X, Liu W, et al. Detecting texts of arbitrary orientations in natural images. In: Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition; 2012. pp. 1083-1090

[47] Yin XC, Yin X, Huang K, et al. Robust text detection in natural scene images. IEEE Transaction on Pattern Analysis and Machine Intelligence. 2014;**36**(5):970-983

[48] Shivakumara P, Phan TQ, Tan CL. New Fourier-statistical features in RGB space for video text detection. IEEE Transactions on Circuits and Systems for Video Technology. 2010;
20(11):1520-1532

[49] Neumann L, Matas J. A method for text localization and recognition in real-world images. In: Proceedings of the Asian Conference on Computer Vision; 2010. pp. 770-783