Scene Text Detection based on Enhanced Multi-channels MSER and a Fast Text Grouping Process

Jin Dai, Zu Wang, Xianjing Zhao, Shuai Shao
Chongqing University of Posts and Telecommunications
China
daijin@cqupt.edu.cn, S151201008@stu.cqupt.edu.cn
S151231016@stu.cqupt.edu.cn, S151231005@stu.cqupt.edu.cn

ABSTRACT: *Scene text detection has become a popular research in computer vision and pattern recognition field in recent years because of the accurate and rich information carried by scene text. Now component-based methods have become the trend, and the detection result is largely determined by the success of filtering text-like non-text regions. The main task of this paper is to reduce the time complexity without a big fall in recall. First an enhanced multi-channels MSER model is introduced. Before extracting MSER, the image is sharpened by using the Laplacian and Gaussian blur and multi-channel is utilized, then the step of the threshold used in MSER algorithm is set to the minimum in order to get add the more refined MSERs. Second, two novel scene text features local contrast and boundary key points are proposed to better distinguish text regions from non-text regions. Finally, a fast text grouping algorithm is achieved which reduces the time complexity from* $O(n^2)$ *to* $O(n\log_2 n)$*. Experiments on both ICDAR 2011 and ICDAR 2013 show that the recall of the proposed method is improved by 3%.*

Keywords: MSER, Multi-channels, Scene Text, Text Detection

Received: 14 September 2017, Revised 20 October 2017, Accepted 28 October 2017

DOI: 10.6025/jcl/2018/9/2/47-59

1. Introduction

The texts in the natural scene images have accurate and rich information, and it is very important to image analysis, translation based on images, image searching and so on. Over the past two decades, researchers have proposed a number of ways to detect text in natural scene images. There are three main types of methods.

Component-based methods [1-9] treat text as connected components which are first extracted by various means, such as color clustering or extreme area extraction, and then the non-text components are filtered using manually designed rules or automatically trained classifiers. In general, these methods are more efficient because the number of components to be processed is relatively small. In addition, these methods are insensitive to rotation, scale and font. In recent years, component-based method has become the mainstream of scene text detection. The Maximally Stable Extremal Regions (MSER) proposed in [8] is robust to

the affine variation of the image, and can effectively extract the text region in the image. In paper [9], the extraction algorithm of MSER is improved to make the time complexity of the algorithm reach the linear time. The key task of these methods is to find some rules or features that can best distinguish text regions from non-text regions.

Texture-based methods [10-17] treat text has a special type of texture and use their texture properties, such as local intensities, filter responses and wavelet coefficients to distinguish between text and non-text regions of an image. These methods are usually computationally expensive as all locations and scales should be scanned. In addition, these methods deal mainly with horizontal text and are sensitive to rotation and scale.

Hybrid method [18-20] takes advantage of both texture-based methods and component-based ones. Fabrizio et al. [19] proposed a hybrid and multi-scale text detection algorithm that can better handle "challenging text" such as multi-size, multi-color and multi-orientation; but these methods are time consuming and need a pre-set lexicon for every image.

Though many researchers have been done on scene text detection, there are still many challenging problems which can be divided into the following two types. (1) Text in a document image has regular fonts, similar color, uniform size and even arrangement, but text in a natural scene may have different fonts, color, scale and orientation even in the same scene. (2) The background of natural scene images can be very complex. Signs, fences, bricks and grass are difficult to distinguish from real text; so it is easy to cause confusion and errors.

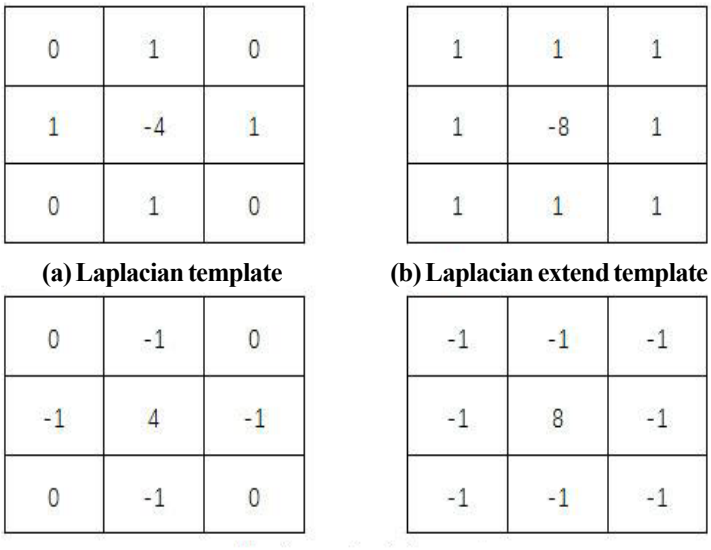
In this paper text candidate is extracted by the MSER algorithm combined with multi-channels and Laplacian in order to get more refined text regions, but more non-text regions are extracted as well. Then the novel scene text features local contrast and boundary key points are introduced to better distinguish text regions and non-text regions. The SVM is trained to filter the text candidate and finally a fast two-layer text grouping algorithm is proposed, which reduces the time complexity from $O(n^2)$ [26, 29, 30] to $O(nlog_2 n)$.

The remainder of this paper is organized as follows. Some basic theory is introduced in Section 2. The proposed method is described in Section 3. Experiment and conclusion is given in Section 3 and Section 4 respectively.

2. Basic Theory

2.1 Image Sharpening with Laplacian

Laplacian is the simplest isotropic differential operator, and it can be expressed in the form of a template. Figure 1 (a) shows the



(a) Laplacian template **(b) Laplacian extend template**

(c) other two Laplacian template

Figure 1. Laplacian templates

template of the discrete Laplacian, Figure 1 (b) shows its extended template (considering the diagonal) and figure 1 (c) shows the other two Laplacian templates (more commonly used in practice). It is difficult for general enhancement techniques to determine the edge lines of steep edges and slowly changing edges. However, this operator can detect those challenging edges by the zero-crossing points between the positive and negative peaks of secondary differential, and it is more sensitive to the isolated points or endpoints, so it is particularly applicable to highlight the isolated points, isolated lines and line endpoints. Like the gradient operator, the Laplacian also enhances the noise in the image. Sometimes the image can be smoothed first.

As Laplacian is a differential operator, it can enhance regions of intensity mutation and weaken the regions of slow-changing intensity. Therefore, the sharpening process can select the Laplacian to process the original image, and then add the Laplacian image to the original image to produce sharpened image. The basic method of the Laplacian sharpening can be represented by the following equation:

$$g(x,y) = f(x,y) + c\,[\nabla^2 f(x,y)] \tag{1}$$

$g(x,y)$ and $f(x,y)$ represent the sharpened image and the input image respectively. If the template in figure 1(a) or figure 1(b) is used, then c = -1, otherwise c = 1.

2.2 Maximally Stable Extremal Regions

The Maximally Stable Extremal Regions(MSER) was first proposed in [8] for wide baseline matching. It is pointed out that MSER can detect the text regions in the image and can be applied to scene text detection. The basic principle of MSER is to take a threshold value from 0 to 255 in turn and get the corresponding binary images. In all the binary images obtained, some of the connected regions change little or even stay the same, then these regions are called the Maximally Stable Extremal Regions. Its mathematical definition is:

$$q(i) = |Q_{i+\Delta} - Q_{i+\Delta}| / |Q_i| \tag{2}$$

Q_i denote the connected region of threshold i, Δ denotes the tiny change of grey value and $q(i)$ is the changing rate of region Q_i with threshold i. When the $q(i)$ is the local minimal, the Q_i is the Maximally Stable Extremal Regions.

The above method can only detect the black area of the grey image, so the image is reversed, and then we can get the binary images with threshold changing from 0-255 of reversed image. These two operations are called MSER+ and MSER- respectively. MSER has the following characteristics:

(1) Invariance to affine transformation of image intensities.

(2) Stability: only regions whose support is nearly the same over a range of thresholds is selected.

(3) Multi-scale detection without any smoothing, both fine and large structure is detected.

In paper [9], the algorithm of MSER is improved to make the time complexity reach the linear time. The formula is defined as follows:

$$q(i) = |Q_i - Q_{i-\Delta}| / |Q_{i-\Delta}| \tag{3}$$

In paper [21], the algorithm is also improved to extract the MSER from the color image. But its efficiency is much lower than that of grey image.

3. The Proposed Method

Matas et al. [8] first proposed the MSER that can effectively detect the text of the scene image. In paper [22], it is proved that the

use of multi-channels (such as R, G, B, etc.) can improve the detection effect of MSER, thus more text regions can be detected but more non-text regions are extracted as well. The most important task of MSER-based methods is to classify those extracted components into text regions and non-text regions, and according to Eq.9 the precision of these methods will be reduced a lot due to non-text regions, and according to Eq.10 more text regions can improve the recall. The aim of the paper is to extract more MSER (containing both text regions and non-text regions) and filter out these interferences and improve the text grouping algorithm to reduce the time complexity.

The improvement of this paper has three parts: (1) The extraction of MSER is enhanced, so more challenging text regions can be detected. (2) Two new scene text features, local contrast and boundary key points are introduced, and the MSER regions are classified by trained SVM [23] with a RBF kernel [24]. (3) A fast grouping of the text regions is realized by the two-layer algorithm (get the initial text lines in the vertical direction and group the word region in the horizontal direction), and the time complexity is reduced from $O(n_2)$ [26,29,30] to $O(nlog_2n)$. The overall algorithm flowchart is shown in Figure 2.

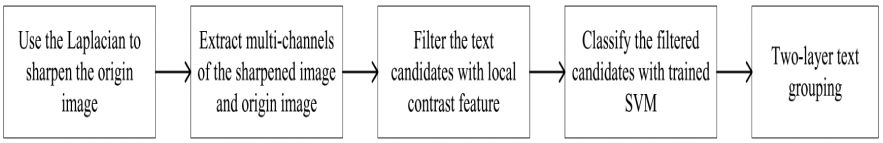


Figure 2. Flowchart of the proposed method

3.1 Enhanced Multi-Channels MSER

The original MSER algorithm is only for grey images, and color information is not taken into account, but there is usually a significant difference between text and non-text regions in aspect the of color. Neumann et al. [22] used the multi-channels processing to exploit the color information for better extracting of text MSER. Based on the multi-channels MSER, we first apply the Gaussian blur to the original image (Gaussian blur before sharpening can effectively reduce the influence of noise [25]), then use the Laplacian template (Figure 1(c)) to preprocess the image and then the sharpened image is obtained according to Eq.1, which enhances the contrast between text regions and their background, so the proposed method can better extract the text in a complex background.

After the sharpening process, we extract the R, G, B, H, S, V and grey channels of the origin and sharpened image, thus totally 16 input images are produced. MSER is extracted from those images as text candidate. In the following table, some comparative experiments are shown to support that the proposed method can extract more text regions.

Method	Precision	Recall
Origin MSER	0.43	0.76
Multi-channels MSER	0.40	0.79
Enhanced MSER (the proposed method)	0.36	0.83

Table 1. Result of extracted MSER

According to Table 1 (the performance is only considered at the level of letter, not the final result of word level), after the preprocess of Laplacian and multi-channels, more text regions can be extracted (the recall has increased), but more non-text regions are extracted as well (the precision has decreased). The continuing work is to filter those unexpected extracted non-text regions.

3.2 Scene Text Features

After MSER is detected, some features that can better distinguish text regions from non-text regions should be introduced. Six scene text features are exploited containing 4 that is used in paper [26] (aspect ratio (w/h), compactness ($\sqrt{a}/p$), hole area ratio (a_c/a) and stroke area ratio (w_s/a)) and two new novel features, local contrast and boundary key points.

Local Contrast *lc*. It is obviously that text which may be recognized by people must have some contrast against its background.

In a local area, the non-text region extracted by MSER algorithm has a low contrast against the background, some non-text regions are shown in Figure 3.

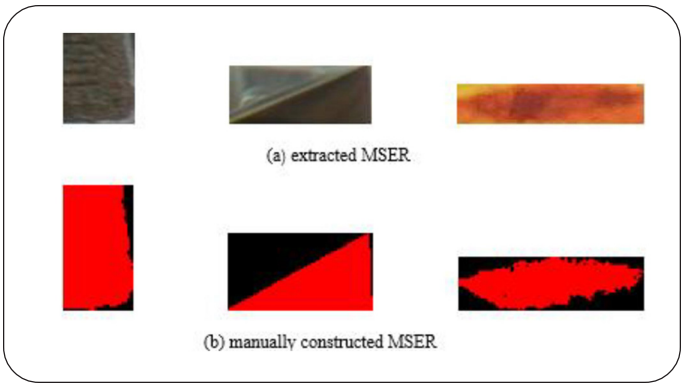


Figure 3. Non-text MSER

For better understanding, Figure 3(b) manually constructed the MSER corresponding to the real extracted MSER in Figure 3(a). The red part of the rectangle is the MSER region and the black part is the background. There is a common feature among these non-text regions, that is the contrast between MSER region and its background is low and based on this feature, the local contrast feature is added to filter the non-text regions. For better utilization of color information, the *R*, *G* and *B* channels are extracted for every single text region and its neighbor or corresponding background. The equation is given as follows:

$$lc = \frac{\left|\sum_{i=1}^{n}\left(R_i + G_i + B_i\right) - \sum_{j=1}^{k}\left(R_i + G_i + B_i\right)\right|}{\max\left(\sum_{i=1}^{n}\left(R_i + G_i + B_i\right), \sum_{j=1}^{k}\left(R_i + G_i + B_i\right)\right)} \tag{4}$$

n denotes the number of pixels of the MSER region, *k* denotes the number of pixels of the background. R, G, B are the three-color channels of the image. Through the experiments, it is found that *lc* of text regions is always bigger than 0.35.

Boundary Key Points *k*. By connecting some points of the region's outer boundary in certain order, the origin region can be approximately restored. The main task of this paper is to find a minimal set of those points. An example of the boundary key points (marked as a circle) is illustrated in Figure 4.

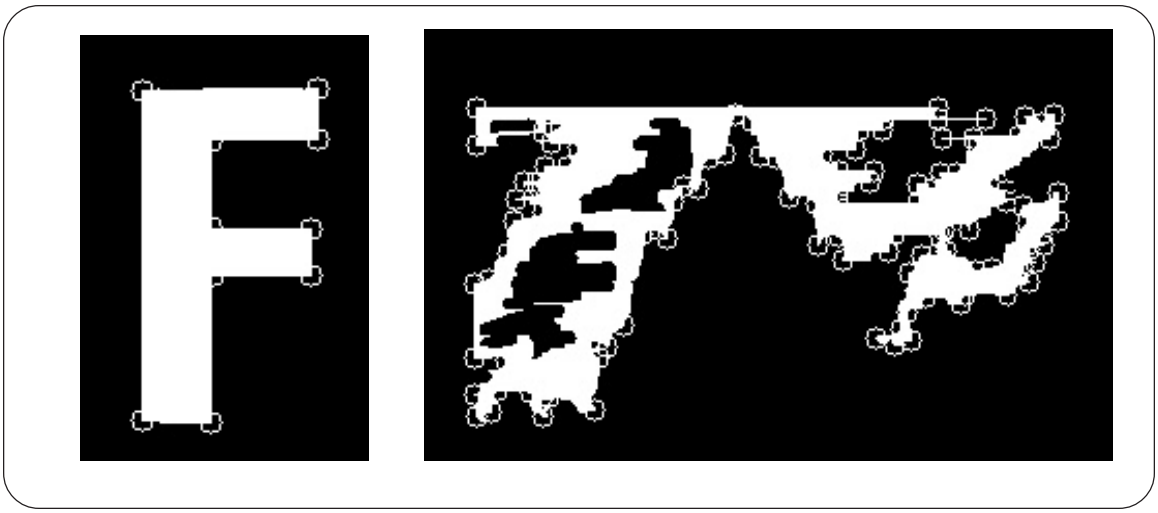


Figure 4. Boundary key points

Process of calculating the k:

Step 1: Construct binary image. Set the grey value of those pixels that belong to MSER to 255, meanwhile set the grey value of the remaining pixels of the rectangle region to 0.

Step 2: Calculate the contour points. Iterate all the pixels of the binary image. If $p(x, y) = 255$ and one of $p(x+1, y)$, $p(x-1, y)$, $p(x, y+1)$, $p(x, y-1)$ has the value 0, the pixel $p(x,y)$ belongs to the contour points.

Step 3: Calculate the k. Use the Douglas–Peucker algorithm to compress the points of the contour, and the remaining points are the boundary key points.

The Characteristic of *k*: The number of the k is invariant to rotation and scale. Through experiment, it is found that the number of k of letter is limited in 5 to 16, and the number of k of non-text regions is more than 16 or less than 5. The training set of the letters is chosen from Chars74K [27] which contains 7705 images of single letter '0'-'9','a'-'z','A'-'Z'(different orientation). Training set of the non-text region is from ICDAR 2013 [28] which contains 229 images, and totally 56523 non-text regions are extracted.

3.3 Text Candidate Construction based on Local Contrast and SVM

The time complexity of the algorithm to calculate the local contrast is according to Eq.4, and the calculation of the other features used in section 3.2 is more computationally expensive, so the construction process of the text candidate set is divided into two stages. The first stage uses only the feature lc, if $lc <= 0.35$ the region is directly filtered out. This stage can quickly filter out the vast majority of the background area (non-text area). After the first stage of the construction, there are still some non-text areas (Figure 5) which are difficult to distinguish from the text area by the local contrast, so a trained SVM (using all the features in section 3.2 except lc, training set is from ICDAR 2013) is used to classify the MSER for subsequent text grouping.

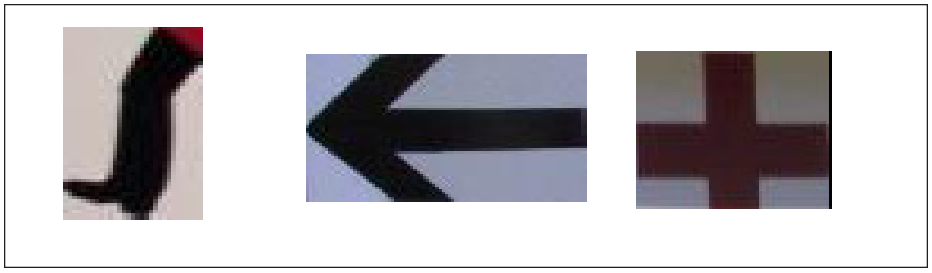


Figure 5. Non-text regions

3.4 Two-layer Text Grouping

Because text regions detected by MSER are almost single letters and text candidate obtained from section 3.3 need to be grouped in order to detect the final location of each word. Existing text grouping is usually done by calculating the degree of association between regions (like spatial position relations [29] and text line constraints [30], as shown in Figure 6) and then iteratively clustering those regions according to experimental thresholds. However, these methods need to consider the relationship between each two text regions or among every triple, so the time complexity is $O(n^2)$ (n is the number of extracted regions). Based on these methods, this paper improves the text region grouping algorithm and reduces the time complexity to $O(n\log_2 n)$.

Figure 6. Text line constraints

In this paper, the text grouping is divided into two stages. The first stage detects the initial text line in the vertical direction, as

shown in Figure 7 (a). In documents, words in different rows can never overlap, but in scene, the images overlaps may always happen between words in different rows. Through experiments, it is found that if only considering the vertical direction, the overlaps can be constrained to a certain value, so the words can be grouped into different rows. In the second stage, the region of a word is detected in the horizontal direction (see Figure 7 (b)). In printed documents, the distance d_1 between adjacent letters in a word is fixed, and the distance d_2 between words is also fixed, and there is an obvious difference between d_1 and d_2. According to this feature, words in the same line can be grouped. But the text in scene images generally lie with an irregular arrangement, so d_1 and d_2 do not have a fixed value like texts in documents. Experimentally found, in the local context there is a constraint between the two distance, and according to the constraint the final text detection can be accomplished.

(a) Vertical grouping (b) Horizontal grouping

Figure 7. Two-layer text grouping

First Stage:

(1) Sort the text candidate by the *Y* coordinate in ascending order (choose the top-left corner as the coordinate origin).

(2) Iterate the sorted text candidate, calculate d_v of each two adjacent regions. The equation is given as follows:

$$d_v = (b_2 - t_2) / h_2 \tag{5}$$

b_1 denotes the Y-axis coordinate (the maximum Y-axis coordinate) of the bottom of the first text region, t_2 denotes the top Y-axis coordinate of the second text region (minimum Y-axis coordinate), h_2 denotes the height of the second text region (Figure. 8). If $d_v > 0.62$(experimentally set), the two text regions are classified as the same class (belong to same text line), if $d_v <= 0.62$, the two text regions are not the same class, and the second text region is regarded as new class (A new text line is split in the Y-axis direction).

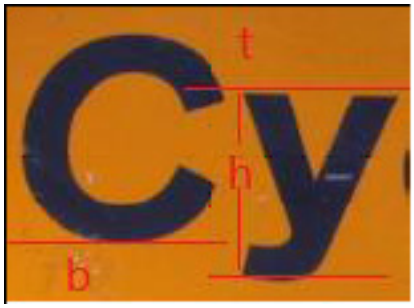


Figure 8. Constraint of the text line

Second Stage:

(1) Take the output of first stage as the input, sort each region in each text line respectively by X-axis coordinate in ascending order.

(2) Iterate text regions in each text line, calculate distance d_h between adjacent regions. The equation is given as follows:

$$d_h = \overline{w} / \Delta d \tag{6}$$

Δd denotes the distance difference (interval) between two adjacent letters in the X-axis. There are two types of Δd. Let d_1 denotes the distance between letters within the same word, and let d_2 denotes the distance between words. In printed documents, d_1 and d_2 always have a fixed value so the words can be grouped, but for the scene text the rule does not work. Experiments found that (Figure 9, the training set comes from ICDAR 2013 with 229 pictures, containing 1226 words), attaching a coefficient $\overline{w}$, which denotes the average width of all the letters within a text line, to Δd a threshold can be found to separate words. d_h is the ratio of letter width and interval. If $d_h < 2.33$, then the two regions belong to the same class (the same word), if $d_h >= 2.33$, the two do not belong to the same class (the two regions do not belong to the same word) and take the second text region as the start of a new word.

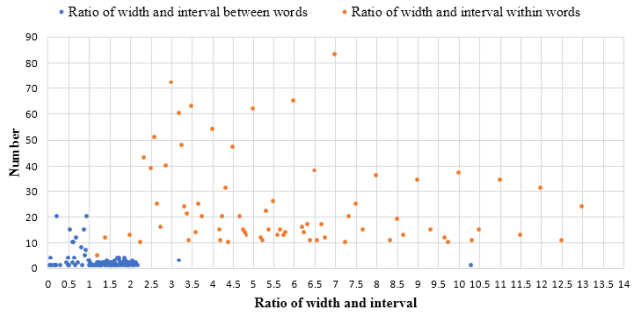


Figure 9. Statistics of ratio of letter width and interval

4. Experiment

In order to verify the correctness and validity of the algorithm proposed in this paper, comparison experiment was conducted on ICDAR 2011 [31] and ICDAR 2013 dataset. The detail of experiment steps is described as follows:

Step1: Prepare the dataset. Use the test set in ICDAR 2011 and ICDAR 2013.

Step2: Text detection. Using the algorithm proposed in this paper and other methods of the same kind we can detect the text in test set containing 233 pictures.

Step3: Text detection time comparison. Compare the text detection time between the proposed method and some similar ones.

Step4: Text detection effect evaluation. Precision (p), recall (r) and f-measure are calculated using the evaluation index presented in [32]. Compare and analyze similar methods in the above three evaluation index.

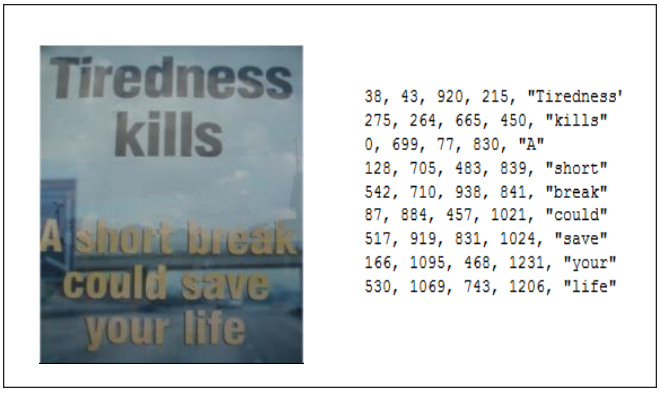


Figure 10. Example of the test set

4.1 Test Set and Evaluation Index

The ICDAR 2011 test set contains 255 images and the ICDAR 2013 test set contains 233 images. Each image corresponds to a text document, which records the specific coordinates of the text that need to be detected, it is shown in Figure 10.

In Figure10, each line denotes a word that need to be detected, the first four items denote the co-ordinates of the word's rectangle region (the first two represent the *X*-axis and *Y*-axis coordinates of the top-left corner of the rectangle, the latter two represent the *X*- axis and *Y*- axis coordinates of the right-bottom corner of the rectangle). The evaluation of the detection effect is mainly calculated by the coincidence between the detected text region and the actual text region. For each rectangle to be evaluated, the maximum matching value is used. The formula is as follows:

$$m\left(r,r^{'}\right)=\frac{2a\left(r\cap r^{'}\right)}{a\left(r\right)+a\left(r^{'}\right)} \tag{7}$$

$$m\left(r;R\right)=\max\left\{m\left(r,r^{'}\right)\middle|r^{'}\in R\right\} \tag{8}$$

r denotes the detected text region, *a (r)* denotes the rectangular area of the text region *r*, and *R* denotes the region set for matching. Get the maximum area matching, and then calculate the precision, recall and *f*-measure. The formula is as follows:

$$precision=\frac{\sum_{r_e\in E}m\left(r_e;T\right)}{\left|E\right|} \tag{9}$$

$$recall=\frac{\sum_{r_t\in T}m\left(r_t;E\right)}{\left|T\right|} \tag{10}$$

E denotes the set of text regions that need to be detected, and *T* denotes the set of rectangles to be evaluated. *F* is the combination of precision and recall. The relative weights of the precision and recall are controlled by the parameter α, which is usually set to 0.5, so that the precision and the recall have the same weight:

$$f=\frac{1}{\frac{\alpha}{precision}+\frac{1-\alpha}{recall}} \tag{11}$$

4.2 Experiment Result and Analysis

In the experiment, we mainly compare the time complexity and the effect of text detection (precision, recall and f-measure).

(1) Experiments on Time Performance of Scene Text Detection

Because of the extra preprocess in section 3.1(sharpening and multi-channels), more inputs are produced (16 new inputs per image) and meanwhile more MSER are extracted. It is very important to take the time performance into consideration. The proposed method uses a two-layer text grouping algorithm to reduce the time complexity. The experiments are conducted on both ICDAR 2011 and ICDAR 2013.

Method	Average detection time on all images	Average detection time on 640*480 images
Neumann and Matas [30]	6.9s	1.8s
Liu et. al [29]	6.3s	1.4s
Proposed method	**2.5s**	**1.0s**

Table 2. Time performance of scene text detect on ICDAR 2011

Method	Average detection time on all images	Average detection time on 640*480 images
Neumann and Matas [30]	6.6s	1.6s
Liu et. al [29]	5.8s	1.2s
Proposed method	**2.4s**	**0.9s**

Table 3. Time performance of scene text detect on ICDAR 2013

The minimum resolution of the dataset is 640*480 (the vast majority), the maximum resolution is 3888*2592. The higher the resolution, the more accurate the MSER is detected and the more time-consuming the process of extracting features, classification and grouping, so the average detection time of all images on both datasets is longer than the resolution of 640*480 images. In this paper, through the two-layer grouping algorithm (initial grouping in the vertical direction and the second grouping in horizontal direction) the final rectangle area of the words is detected. The proposed method only need two sorting operations, and the iterative clustering process of comparison between each two regions is avoided. Experimental results from two datasets show that the two-layer text grouping algorithm can reduce the time of scene text detection. On the ICDAR 2011 dataset, the average detection time of all images was reduced by 60.3% compared with the similar algorithm, and the time on images with 640*480 resolution was reduced by 28.6%. On the ICDAR 2013 dataset, the average detection time for all images was reduced by 58.6%, and the detection time was reduced by 25.0% for images with 640*480 resolution.

(2) Experiments on Scene Text Detection Effect

Scene text detection effect is evaluated by three indexs: precision, recall and f-measure. Comparison experiments are conducted on ICDAR 2011 and ICDAR 2013 respectively. The results are shown in the following tables.

Method	Recall	Precision	f-measure	Year
Proposed method	**0.71**	0.83	**0.77**	-
Wu et. al [14]	0.68	0.82	0.75	2016
Yu et. al [17]	0.65	**0.84**	0.73	2015
Yi and Tian [33]	0.68	0.76	0.67	2013

Table 4. Scene text detection effect on ICDAR 2011

Method	Recall	Precision	f-measure	Year
Proposed method	**0.77**	0.85	**0.81**	-
Zhu et. al [2]	0.74	**0.86**	**0.80**	2016
Neumann and Matas [30]	0.72	0.82	0.77	2015
Gomez and Karatzas [5]	0.73	0.77	0.75	2014

Table 5. Scene text detection effect on ICDAR 2013

From Table 4, it is obvious that the proposed method is better than other methods in recall and f-measure on ICDAR 2011 dataset compared with similar methods. It can be seen from Table 5 that the method of this paper is also better than the similar methods in recall and f-measure on ICDAR 2013 dataset. But the proposed method does not have the best performance on precision index. The main reason for this problem is caused by the preprocess which produces an additional increase in the number of input images. More text MSER are extracted but more non-text MSER are detected as well, so which finally leads to the decrease in the precision.

Some detection results (success and failure) are shown in Figure 11. In Figure 11(a) the scene text is successfully detected. In Figure 11(b) some interference factors like low contrast and irregular letters cause the failure of scene text detection.

5. Conclusion

This paper proposes a scene text detection algorithm based on enhanced multi-channels MSER. By sharpening and extracting multiple channels of the image, some text in complex background can be detected, but more non-text regions are introduced as well. In order to filter these non-text regions, this paper introduces two scene text features: local contrast and boundary key

(a) Sucess

(b) Failure

Figure 11. Results of scene text detection

points. SVM is used to classify the text candidate. Finally, the text grouping algorithm is improved. The text region grouping is divided into two stages. First, the text candidate is grouped in the vertical direction. Then, the regions in each text line are grouped into the final word-based rectangle area, and the time complexity is reduced to $O(nlog_2n)$. Experimentally found, the proposed method has a good performance compared to some similar methods on both ICDAR 2011 and ICDAR 2013 dataset.

The proposed method can only deal with horizontal or nearly horizontal text, and it can only deal with English words. In the follow-up work, we need to improve the classification or filter of the text candidate set, because the experimental results show that the proposed method has not achieved the best performance in the precision index compared with the similar methods. Finally, convolution neural network can be introduced to get more robust text features in order to filter the text-like non-text regions.

References

[1] Wu, Y., Shivakumara, P., Lu, T., et al. (2016). Contour Restoration of Text Components for Recognition in Video/Scene Images, *IEEE Transactions on Image Processing*, 25. 5622-5634.

[2] Zhu, A., Gao, R., Uchida, S. (2016). Could scene context be beneficial for scene text detection?, *Pattern Recognition*, 58. 204-215.

[3] Yi, C., Tian, YL. (2011). Text string detection from natural scenes by structure-based partition and grouping, *IEEE Transactions on Image Processing*, 2011, 20 (9), 2594–2605

[4] Huang, W., Lin, Z., Yang, J., et al. (2013). Text localization in natural images using stroke feature transform and text covariance descriptors, *In:* Proceedings of the IEEE International Conference on Computer Vision. 2013: 1241-1248.

[5] Gomez, L., Karatzas, D. (2014). A fast hierarchical method for multi-script and arbitrary oriented scene text extraction, 2014, 19 (4) 1-15.

[6] Neumann, L., Matas, J. (2016). Real-time Lexicon-free Scene Text Localization and Recognition, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38 (9) 1872-1885.

[7] He, T., Huang, W., Qiao, Y., et al. (2015). Text-Attentional Convolutional Neural Network for Scene Text Detection, *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 2015, 25 (6) 2529-2541.

[8] Matas, J., Chum, O., Urban, M., et al. (2004). Robust wide-baseline stereo from maximally stable extremal regions, *Image and vision computing*, 22 (10) 761-767.

[9] Nistér D, Stewénius H.(2008). Linear Time Maximally Stable Extremal Regions/ Computer Vision - ECCV 2008, *In:* European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings.183-196.

[10] Chen X, Yuille AL. Detecting and reading text in natural scenes, /Computer Vision and Pattern Recognition, 2004. CVPR 2004. *In:* Proceedings of the 2004 IEEE Computer Society Conference on. IEEE, 2004, 2: II-366-II-373 Vol. 2.

[11] Zhong, Y., Karu, K., Jain, A K. (1995). Locating text in complex color images, Document Analysis and Recognition, 1995. *In: Proceedings of the Third International Conference on. IEEE*, 1995, 1. 146-149.

[12] Kim KI, Jung K, Kim JH. (2003). 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*, 25 (12) 1631-1639.

[13] Gllavata, J., Ewerth, R., Freisleben, B. (2004). Text detection in images based on unsupervised classification of high-frequency wavelet coefficients, Pattern Recognition. ICPR 2004. *In:* Proceedings of the 17th International Conference on. *IEEE*, 1: 425-428.

[14] Wu, H., Zou, B., Zhao, YQ. (2016). Natural scene text detection by multi-scale adaptive color clustering and non-text filtering, *Neurocomputing*, 214. 1011–1025.

[15] Chen, YL., Yu, PT. (2016). An evidence-based model of saliency feature extraction for scene text analysis, *International Journal on Document Analysis and Recognition* (IJDAR), (3) 1-19.

[16] Bai, X., Yao, C., Liu, W. (2016). Strokelets: A Learned Multi-Scale Mid-Level Representation for Scene Text Recognition, *IEEE Transactions on Image Processing*, 25 (6) 2789-2802.

[17] Yu, C., Song, Y., Zhang, Y. (2015). Scene text localization using edge analysis and feature pool, *Neurocomputing*, 175. 652-661.

[18] Pan, YF., Hou, X., Liu, CL. (2011). A hybrid approach to detect and localize texts in natural scene images, *IEEE Transactions on Image Processing,* 20 (3) 800-813.

[19] Fabrizio, J., Robert-Seidowsky, M., Dubuisson, S., et al. (2016). TextCatcher: a method to detect curved and challenging text in natural scenes, *International Journal on Document Analysis and Recognition (IJDAR)*, 2016, 19 (2) 99-117.

[20] Tian, Z, Huang, W, He, T, et al. (2016). Detecting Text in Natural Image with Connectionist Text Proposal Network, Computer Vision – ECCV 2016. *Springer International Publishing*, 2016.

[21] Liu, T., Chen, J., Wang, C. (2010). Improved Maximally Stable Extremal Region detector in color images, *IEEE International Conference on Information and Automation. IEEE*, 1711-1716.

[22] Neumann, L., Matas, J. (2010). A method for text localization and recognition in real-world images, *Asian Conference on Computer Vision*. *Springer Berlin Heidelberg*, 770-783.

[23] Nosov, AV. (2005). An introduction to support vector machines. China Machine Press.

[24] Muller, KR., Mika, S., Ratsch, G., et al. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12 (2) 181-201.

[25] Gonzalez, RC. (2006). Woods RE. Digital Image Processing (3rd Edition)/ *Digital Image Processing* (3rd Ed). 182.

[26] Neumann, L., Matas, J. (2012). Real-time scene text localization and recognition/ IEEE Conference on Computer Vision and Pattern Recognition. *IEEE Computer Society*, 3538-3545.

[27] Campos TED., Babu BR., Varma, M. (2009). *Character Recognition in Natural Images*/VISAPP (2) 273-280.

[28] Karatzas, D., Shafait, F., Uchida, S. (2013). ICDAR 2013 Robust Reading Competition. 2 (2-3)1484-1493.

[29] Liu, J., Su, H., Yi, Y, (2016). Robust text detection via multi-degree of sharpening and blurring, *Signal Processing*, 124. 259-265.

[30] Neumann, L., Matas, J. (2015). Efficient Scene text localization and recognition with local character refinement, *In:* International Conference on Document Analysis and Recognition. IEEE, 746-750.

[31] Shahab, A., Shafait, F., Dengel, A. (2011). ICDAR 2011 Robust reading competition challenge 2: Reading text in scene images/2011 International Conference on Document Analysis and Recognition. *IEEE*, 1491-1496.

[32] Lucas, SM., Panaretos, A., Sosa, L., et al. (2003). ICDAR 2003 Robust Reading Competitions/ICDAR, 682.

[33] Yi, C., Tian, Y. (2013). Text Extraction from Scene Images by Character Appearance and Structure Modeling, *Computer Vision & Image Understanding*, 117 (2) 182-194.