An Efficient Detection Method for Text of Arbitrary Orientations in Natural Images

Lanfang Dong, Zhongdi Chao and Jianfu Wang

Abstract Due to the high complexity of natural scenes, text detection is always a critical yet challenging task. On the basis of existing character detection method, a novel text line detection method is proposed in this paper, which can localize text of arbitrary orientation by using related information of character regions in candidate text line. First, inspired by the Hough transform, text line detection problem is regarded as line detection problem in candidate characters set obtained by Most Stable Extremal Regions (MSERs). Second, in order to find out the relationship of adjacent candidate regions, a graph model is built based on some constraints and adjacent candidates are linked into pairs to obtain search domain. Then, to avoid repeated calculation of the same line, some strategies need to be used. Finally, as some of the potential text lines are incorrect, we use a new text line descriptor to exclude the non-text areas. Experimental results on the ICDAR 2013 competition dataset and MSRA-TD500 show that the proposed approach is favorable no matter for non-horizontal text or horizontal text.

Keywords MSERs · Graph model · Text line detection · Text line descriptor

1 Introduction

Wearable device refers to a portable device that can be directly worn on the body or integrated into the clothes or accessories. In order to realize user interaction, life entertainment, human monitoring, and other functions, this kind of equipment

L. Dong (✉) · Z. Chao · J. Wang
School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China
e-mail: lfdong@ustc.edu.cn

Z. Chao
e-mail: chaozhd@mail.ustc.edu.cn

J. Wang
e-mail: wangjf55@mail.ustc.edu.cn

C. Yang et al. (eds.), *Wearable Sensors and Robots*, Lecture Notes in Electrical Engineering 399, DOI 10.1007/978-981-10-2404-7_35

integrates various types of recognition, sensing, connection, cloud services, and other interactive technology and storage technology to replace the handheld and other devices. With the development of computer technology, it is possible to apply image processing and computer vision technology into wearable devices, such as portable intelligent navigation glasses for the blind, automatic translation and explanation tools for tourists. The key issue of these applications is extracting information from the scene accurately and efficiently. As a form of information with high expressiveness, texts in natural images get more and more attention in recent years.

Because of the complexity of background and randomicity of text color, font, orientation, and position, the recognition accuracy of OCR system is relatively low in natural image or in video image. In order to localize texts accurately, three problems need to be resolved: (1) texts in scene image may appear in any form, such as font varies, color changeable, etc. (2) texts can be embedded in any background, such as landscape, architecture, window, etc. (3) text may be similar to the background, such as handwritten characters and comics, characters and columns, pencils, trunks, etc. Therefore, how to solve these problems is the main difficulty of existing methods.

In this paper, we present an unconstrained text localization method as Fig. 1 shows. Rather than deleting all noise character regions in the phase of character detection, we use text line descriptor to exclude incorrect text lines. In the first phase, character regions are detected by MSERs. As some candidate regions only contain part of a character, a preprocessing is executed first. Then noise regions are removed by character descriptor. As it is almost impossible to discriminate text regions from non-text regions completely just by using character descriptor, post-processing needs to be considered. In most previous papers, text lines are detected by pruned exhaustive search or some heuristic rules. The exhaustive method is computational as it detects every potential line in the candidate set, and methods based on heuristic rules are more applicable for horizontal text extraction. In this paper, a quick text line extraction method is proposed, which can localize text of arbitrary orientation by detecting lines in the graph model built by adjacent relationship. Non-text areas are excluded by descriptor that integrates information (like size, center, color, etc.) of regions in candidate text lines.

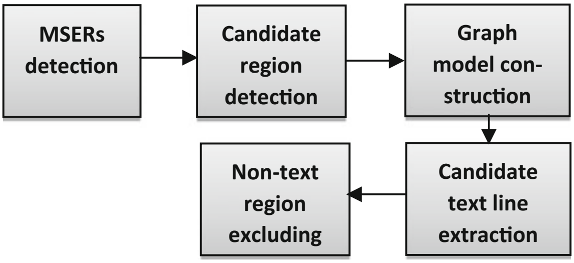
MSERs detection
Candidate region detection
Graph model construction
Candidate text line extraction
Non-text region excluding

Fig. 1 Block structure of the proposed method

The rest of this paper is organized as follows. An overview of previous published methods is given in Sects. 2, 3 details the proposed method. Experiments and results are presented in Sect. 4 and conclusions are drawn in Sect. 5.

2 Previous Work

Present text localization approaches can be roughly divided into two categories: texture-based and component-based. The texture-based approaches assume that compared to non-text area, there are some unique textures in the text areas. Based on this assumption, Xiaodong Huang thought that text area was rougher than background (Huang 2012). In order to predict the location of text, an edge map was calculated by wavelet transform, then text rows were detected based on statistic coarseness of the edge map. Shekar B H, Smitha M L detected text by using discrete wavelet transform and gradient difference (Shekar et al. 2014). They calculated the gradient map with Laplacian template and MGD on the edge map extracted by discrete wavelet transform. Yatong Zhou distinguished text and non-text areas by discrete cosine transform. First image was divided into N * N pieces, and then based on the gradient information of each piece, the text areas were obtained. Although above approaches perform perfectly in image with regular texts, such as video frame, their detection results are not ideal in scene images because of the morphological diversity and randomness of text. Jung-Jin Lee, Kai Wang, Xiangrong Chen, etc., scanned images with sliding windows of different scales and distinguished text and non-text regions by descriptors (Mancas-Thillou and Gosselin 2006, Wen et al. 2009, Wang et al. 2011). As text areas are texture-unique, these methods are more favorable, but the amount of computation is also considerable.

In contrast to texture-based method, region-based approaches pay more attention on features of characters. Yuning Du, Shihong Lao, etc., detected dot text based on FAST points (Chen and Yuille 2004); Mingcheng Wan, Fengli Zhang, etc., thought that the distribution of corners on character edge were special and regular, which could be used to discriminate the non-text edges (Du et al. 2011). Wonder Alexandre Luz Alves and Ronaldo Fumio Hashimoto used Ultimate Attribute Opening to extract the set of characters candidates. Neumann L localized text with oriented stroke detection (Yao et al. 2012). Although various methods have been proposed, SWT (Wan et al. 2008, Epshtein et al. 2010, Huang et al. 2013) and MSERs (Neumann and Matas 2013, Iqbal et al. 2014, Neumann and Matas 2012, Matas et al. 2004) are by far the most popular methods as they are more favorable in the case of various complex scenes.

Although using MSERs to detect candidate character regions can achieve good result in majority cases, one considerable factor is that the cardinality of the MSERs set is exponential in the number of pixels in the image, so that it is time consuming for subsequent processing. In order to reduce the unnecessary regions, Neumann L designed a character descriptor to evaluate the region. He recorded the value of descriptor when the regions changed, and the region with biggest value is regarded as best rectangle for a character. In Matas' paper (Matas et al. 2004), a MSERs lattice is built by the inclusion relationship. The character region is seen as the region that satisfies some specific connecting relationship. Although these methods are effective, the computation is complex and rules in them are difficult to be obtained.

3 The Proposed Method

3.1 MSERs Detection

We assume that each character is a continuous region and has similar color, and the optimal candidate region just covers outer boundary of a character. As the MSER lattice (Fig. 2) induced by the inclusion relation shows, smaller MSERs are imbedded into bigger ones. In other words, one MSER will experience several changes (like embedded into another, or become larger or barely budged). In order to obtain the optimal MSER, growth rate of MSER must be calculated in every phase. We think that, the region with the lowest growth rate is set as the optimal MSER. The Fig. 2 shows the results of proposed approach. As smaller MSERs will be embedded into bigger ones eventually, some small and stable MSERs without containing any characters cannot be excluded (as show in Fig. 2), otherwise, the one just covering character also should be removed, which plays the same role as its successor node in MSERs lattice.

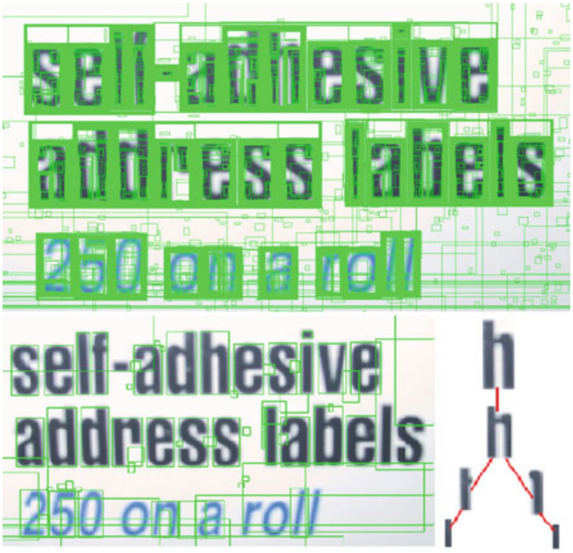

Fig. 2 Process instance of character detection method

3.2 Two-Level Character Descriptor

As the high diversity of characters and backgrounds, there are many noise regions after MSERs detection, and it is hard to exclude all of them just through one-level classifier. In this paper, a two-level character descriptor is designed.

Considering that most of irrelevant regions are of color-closing overall or rough details, and these features can be quantified easily, a coarse classifier constituted by some constraints is designed to remove part of noise regions in the first level. A rule of thumb holds that the area of character cannot exceed a certain percentage of the total area, a character is always a continuous region, and the edge points and connected domain are generally finite. Based on this, three factors are taken into account: (1) color distribution after binarization; (2) number of connected domains; (3) number of edge points. We set l as MSER, $p(i)$ denotes the proportion of pixels have value i, conn(l, i) be the number of connected domains of pixels of value i, edge(l) be the number of edge points and w and h be the width and height of MSER. Then following constraints need to be satisfied.

$$P(i) < 0.78 \ (i = 0, 1) \tag{1}$$

$$(w + h) < \text{edge}\,(l) < (w + h) * 5 \tag{2}$$

$$\sum_i \text{conn}\,(l, i) \leq 7 \ \&\& \ (\text{conn}\,(l, 0) \leq 3 \,||\, \text{conn}\,(i, 1)) \leq 3 \tag{3}$$

Constraint (1) can be applied to exclude flat areas like whitespace between characters in Fig. 3. Constraint (2) and constraint (3) are used to eliminate MSERs with rough details. Figure 3 shows the results after the coarse classifier that integrates these constraints. We can see that majority of non-text regions are removed.

In the second level, a neural network classifier is adopted. Different from the non-text region, text regions always possess following features: (1) containing two main colors; (2) high contrast between character region and background; (3) pixels with the same gray value always located in the same connected domain; (4) smooth edge, etc. Based on these characteristics, following features are extracted:

(1) HOG feature of 8 orders;
(2) Color difference. The color difference measures roughness of the candidate region, and it is calculated as follows:

$$\sum_{i=0}^{N} (i - \text{average Gray})^2 \times p(i) \tag{4}$$

(3) Entropy of histogram
(4) Ratio of edge and the sum of width and height

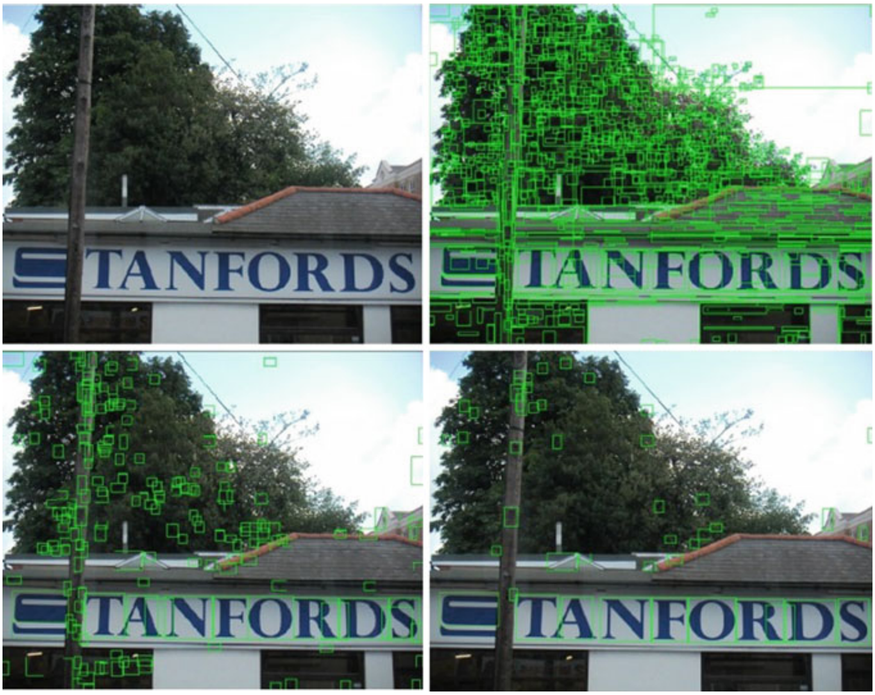

Fig. 3 Result after character descriptor (from the *left* to *right* are original map, map after MSERs detection, result of coarse classifier, map after neural network classifier)

(5) Ratio of edge and area

(6) Proportion of foreground in the region

(7) Edge difference. The edge difference describes the roughness of edges in a region. As most edges of character are smooth, the values of them are smaller than regions with rough details. The calculation formula is as follows:

$$\nabla X = \sum_{x=i-1}^{i+1} x; \quad \nabla Y = \sum_{y=j-1}^{j+1} y;$$
$$\mathrm{eCom} = \sum_{i,j} \sqrt{\left(\frac{\nabla X}{\mathrm{count}} - i\right)^2 + \left(\frac{\nabla Y}{\mathrm{count}} - j\right)^2} \tag{5}$$

∇X, ∇Y , are the sum of X-axis and Y-axis of edge points in the 3 * 3 window with center (i, j), count is the number of edge points in the window. When the edge is straight, the edge difference tends to zero. With the increase of rough degree, the value will be bigger.

3.3 Text Line Detection

Although above method is effective to eliminate noise regions, there are still a lot of non-text regions existing. In order to solve this problem, a new text line extraction method is proposed. It not only can find out the text line quickly, but also can exclude non-text lines easily.

Characters in a text line are always neatly arranged and in a straight line (as Fig. 4). Inspired by Hough transform, the following hypothesis is put forward: candidate text line detection can be regarded as the straight line detection in the image. As all lines with different slopes need to be checked, the amount of computation is the square of number of candidate regions. It is very time consuming when the quantity of candidates is tremendous, moreover, as text lines cannot be neglected just by length like Hough transform, a lot of candidate lines will be extract. In order to decrease unnecessary computation, search space in the proposed

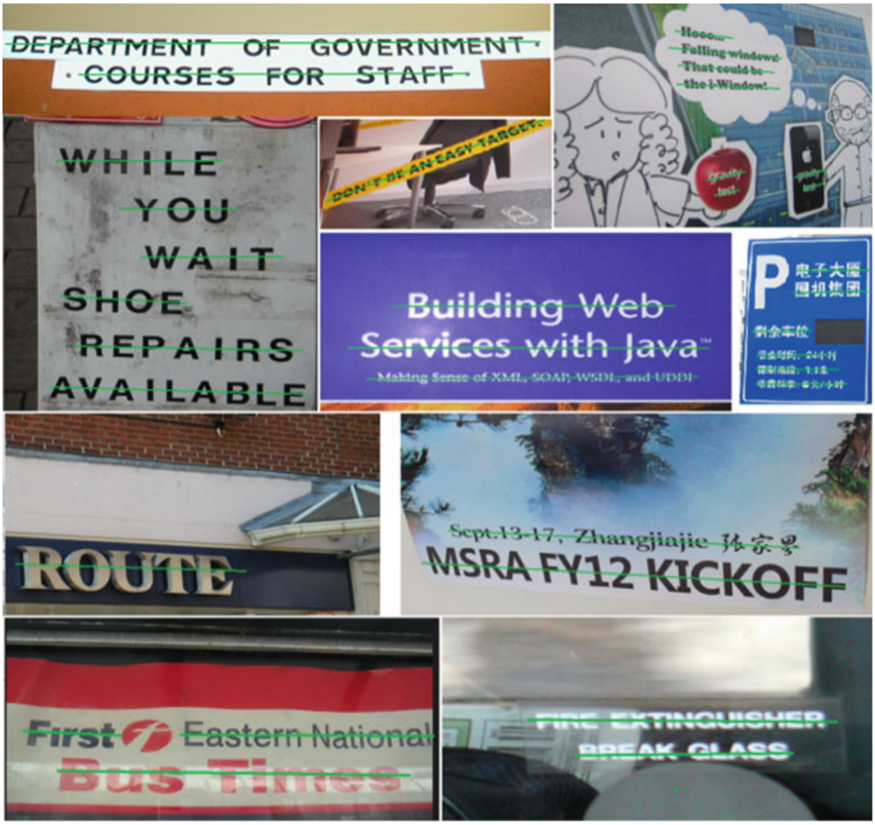

Fig. 4 Instances of pictures containing text line

method is narrowed by preset search paths. As regions in a text line are similar in some aspects, we can restrain search paths by constructing a graph model.

3.4 Graph Model Construction

The key of constructing graph is definite of neighboring relations. As to make the search path proceed along those regions most likely in the same line, in this module, the neighboring relation is not only determined by Euclidean distance, but also affected by size of regions. Concretely, in the graph, each node will point to no more than two nodes which are successor nodes with the shortest distance from their precursor, and have to satisfy following constraints.

$$\begin{gathered}\left|W(l_i) - W(l_j)\right| < 0.5 * \min\left(W(l_i),\ W(l_j)\right) \\ \left|H(l_i) - H(l_j)\right| < 0.5 * \min\left(H(l_i),\ H(l_j)\right) \\ \text{distance}\,(l_i, l_j) < 2 * \min\left(\&\,\text{diag}\,(l_i),\ \&\,\text{diag}\,(l_j)\right)\end{gathered} \tag{6}$$

where $L(l_1, l_2, l_3, l_4, \ldots, l_n)$ denotes the set of MSERs, n is the size of the set, $W(l_i)$ and $H(l_i)$ are the width and height of l_i. diag (l_i) is the length of diagonal of l_i.

The node with no successors will be treated as a sub graph. As two nodes are probably the nearest node to each other, that makes the retrieval unable to continue, like Fig. 5. Some measures need to be taken to avoid this.

3.5 Text Line Detection

After the construction of graph model, the retrieval of text lines only needs to be carried out along the line in the graph. Two lists (CN and LN) must be created first, CN is used to save candidate nodes need be checked and LN is used to save nodes in the search line. The search algorithm contains three steps

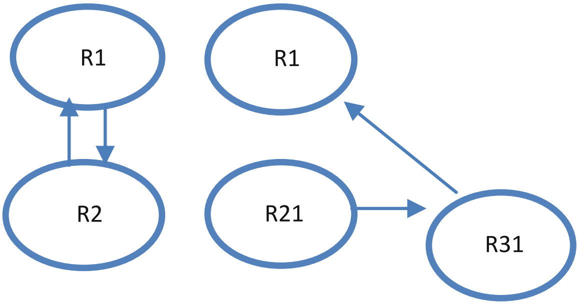

Fig. 5 Structures interrupting search process

(1) Traverse node pairs comprised by adjacent nodes in the graph.
(2) Calculate the line constituted by the nodes pair, and put the nodes pair into the LN and successor nodes of them into CN.
(3) If CN is not empty, get a node in CN and checked it, else if it is in the line, put the node into LN and its successor nodes are added into CN. Repeat this step until there is no nodes in CN.
(4) The candidate text line constructed by nodes in LN is added to the candidate line list.

To avoid repeated calculation for the same line, we stipulate that if two nodes have the relation of precursor-successor in the graph and are in a straight line that has been extracted, the line constructed by them will not be considered. As to avoid adding the irrelevant regions into the line, region that will be searched must be the successor of one region in the line. Result of this text line detection method in Fig. 6 shows that most text lines were seek out. It proves that this method can extract text line effectively. However, a lot of non-text regions were also extracted. A text line classifier trained by SVM is designed to eliminate these noise text lines.

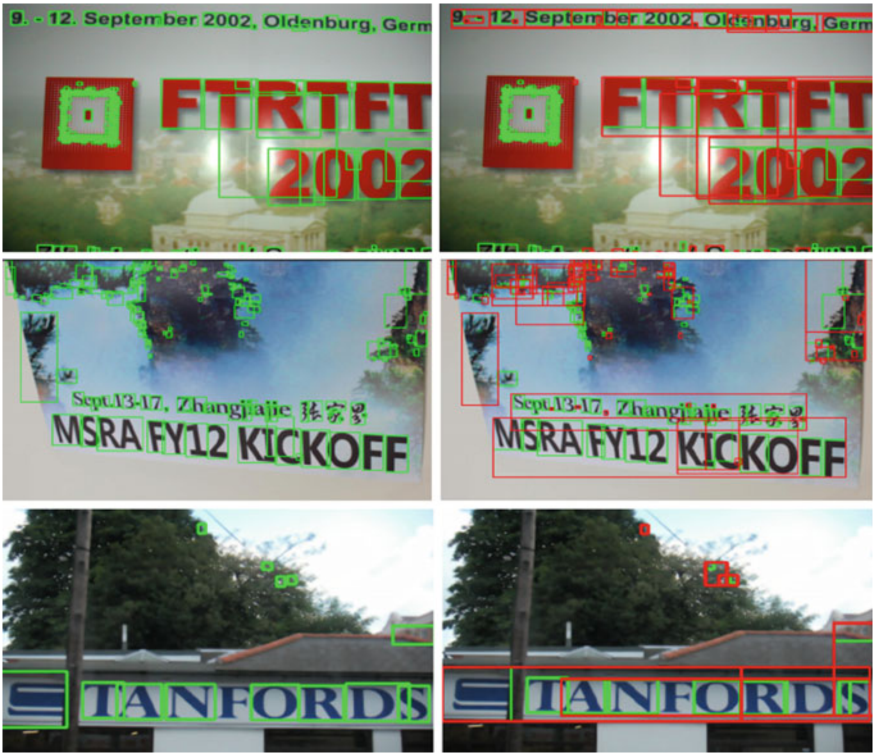
9. - 12. September 2002, Oldenburg, Germ
FTRTFT
2002
Sept.13-17, Zhangjiajie 张家界
MSRA FY12 KICKOFF
TANFORDS

Fig. 6 Result of text line search method

Because regions in a text line are similar to each other, the comprehensive features are extracted, like

(1) Distance difference: the variance of distances between adjacent regions in the line.
(2) Height difference: the variance of height of regions in the line.
(3) Difference of ratio of foreground after binarization.
(4) Difference of average gray value of foreground. Due to color of regions in a text line are almost the same, the value will be smaller than regions have variance colors.
(5) Difference of the ratio of edge pixels to foreground pixels.
(6) Weighted color variance: the variance of $\mathrm{WP}(l_i)$. As characters in a text line always have similar color distribution, this feature value of a real text line is lower than others theoretically, and non-text regions always have multicolor and irregular distributions. Let $P(i)$ be the proportion of pixels of value i, then $\mathrm{WP}(l_i) = \sum_{i=0}^{255} i * P(i)$.
(7) Difference of LBP feature.
(8) Difference of wavelet energy.
(9) Hog features of text line.

4 Experimental Results

The performance of the proposed method was evaluated on ICDAR 2013 and MSRA-TD500 dataset. There are totally 509 fully annotated text images in ICDAR, of which 258 images are used for training and others for testing. There are 500 images with inclined text lines in MSRA-TD500 dataset with 300 images for training and 200 for testing. The testing procedure is split into three parts in this paper. The first part is testing the performance of character detection method; the second part is to test text line extraction method and test lines screening method is tested in the third part.

4.1 Character Regions Detection

Candidate character regions are firstly generated by MSERs, and then a character descriptor is designed as there are a lot of non-text regions. The character descriptor in this paper is a two-level classifier. In the first level, majority of noise regions are removed by constraints, and in the second level, a neural network classifier is used. Samples used to train neural network classifier in the paper are extracted in images used for training in ICDAR and MSRA-TD500. 12093 samples are produced,

Table 1 Result of character regions detection

Dataset	Samples	Recall (%)	Precision (%)
ICDAR	209	85.4	43.3
MSRA-TD500	200	83.3	50.6

including 5324 positive samples and 6769 negative samples. We test our extraction approach on ICDAR and MSRA-TD500, the results are shown in Table 1.

As most pictures in ICDAR are scene image and character regions in them are obvious, majority of characters can be extracted. By contrast, pictures in MSRA-TD500 are mostly sign images, and backgrounds are always simple. Although the recall is lower than ICDAR, the precision is higher.

4.2 Text Lines Retrieval

The text lines extraction algorithm is introduced in Sect. 3.5. In order to test this algorithm, we carried on the experiments on these two datasets above and result is shown in Table 2.

4.3 Text Lines Screening

The text line classifier used in this paper is trained by samples generated by the text line retrieval algorithm we proposed. As relevant information is utilized, we compute these nine features after extracting candidate text lines, and pick out 1400 samples (includes 600 positive samples and 800 negative samples) to train the text line classifier. Figure 7 shows the result after screening. We can see that part of text regions are extracted repeatedly. In order to avoid this, the text regions are confined to the region that has larger value generated by text line descriptor. The relative features of text lines that consist of single character are assigned as average values of corresponding values of samples.

4.4 Performance Analysis

As is known to all, the MSERs is of high efficiency, so the computational complexity of the method in this paper hinges on the complexity of character descriptor

Table 2 Result of text lines detection

Dataset	Samples	Recall (%)	Precision (%)
ICDAR	209	82.4	44.4
MSRA-TD500	200	76.3	48.3

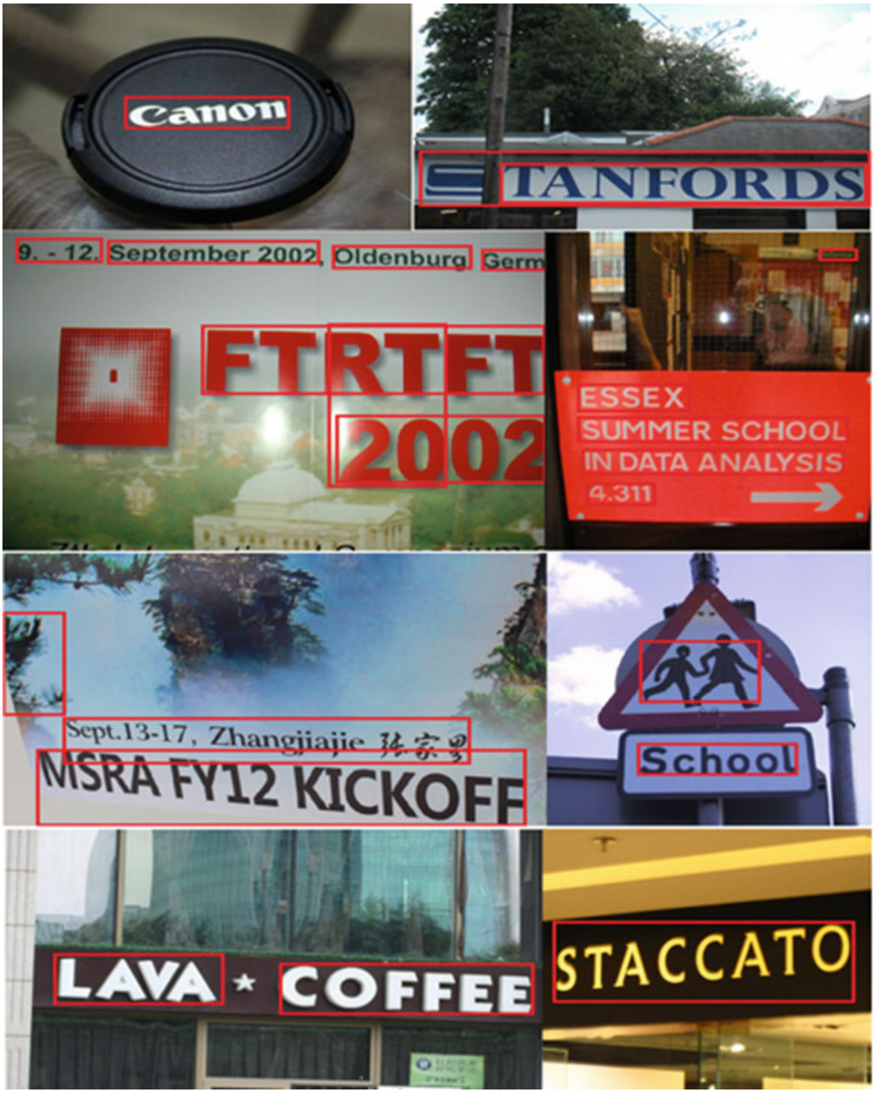

Fig. 7 Text locating results

and text line detection. According to the description of this paper, we know that the computation amount of character descriptor is $O\left(n^2\right)$ (n is the number of pixels in an image), and complexity of text line retrieval is far lower than $O\left(N^2\right)$ (N is the number of candidate character regions), so it is much quicker than methods detecting text lines by exhaustive search or other search methods.

5 Conclusion

A new text line detection method is proposed in this paper which not only works for horizontal text, but also performs well in detecting text of arbitrary orientations in complex scenes. With constructing a graph model, a lot of unnecessary computation is omitted and the detection time is shortened significantly. In order to exclude non-text regions, a two-level character descriptor is designed and employed in character detection module, and relative features are extracted to remove noise text lines. Experimental results show that the method we prosed can extract text lines effectively no matter for horizontal or inclined texts.

References

Chen X, Yuille AL (2004) Detecting and reading text in natural scenes. In: Computer vision and pattern recognition. CVPR 2004. Conference on Proceedings of the 2004 IEEE computer society. IEEE 2004, vol 2, pp II-366-II-373. doi:10.1109/CVPR.2004.1315187

Du Y, Ai H, Lao S (2011) Dot text detection based on fast points. In: International conference on document analysis and recognition (ICDAR). IEEE 2011, pp 435–439. doi:10.1109/ICDAR.2011.94

Epshtein B, Ofek E, Wexler Y (2010) Detecting text in natural scenes with stroke width transform. In: IEEE Conference on computer vision and pattern recognition (CVPR). IEEE 2010, pp 2963–2970. doi:10.1109/CVPR.2010.5540041

Huang W, Lin Z, Yang J, Wang J (2013). Text localization in natural images using stroke feature transform and text covariance descriptors. In: IEEE international conference on computer vision (ICCV). IEEE 2013, pp 1241–1248. doi:10.1109/ICCV.2013.157

Huang X (2012) Automatic video text detection and localization based on coarseness texture. In: Fifth international conference on intelligent computation technology and automation (ICICTA). IEEE 2012, pp 398–401. doi:10.1109/ICICTA.2012.106

Iqbal K, Yin XC, Hao HW, Asghar S, Ali H (2014) Bayesian network scores based text localization in scene images. In: International joint conference on neural networks (IJCNN). IEEE 2014, pp 2218–2225. doi:10.1109/IJCNN.2014.6889731

Matas J, Chum O, Urban M, Pajdla T (2004) Robust wide-baseline stereo from maximally stable extremal regions. Image Vis Comput 22(10):761–767

Mancas-Thillou C, Gosselin B (2006) Natural scene text understanding. na, Ann Arbor

Neumann L, Matas J (2012) Real-time scene text localization and recognition. In: IEEE conference on computer vision and pattern recognition (CVPR). IEEE 2012, pp 3538–3545. doi:10.1109/CVPR.2012.6248097

Neumann L, Matas J (2013) Scene text localization and recognition with oriented stroke detection. In: IEEE international conference on computer vision (ICCV). IEEE 2013, pp 97–104. doi:10.1109/ICCV.2013.19

Shekar BH, Smitha ML, Shivakumara P (2014) Discrete wavelet transform and gradient difference based approach for text localization in videos. In: 2014 fifth international conference on signal and image processing (ICSIP). IEEE 2014, pp 280–284. doi:10.1109/ICSIP.2014.50

Wan M, Zhang F, Cheng H, Liu Q (2008) Text localization in spam image using edge features. In: International conference on communications, circuits and systems. ICCCAS 2008. IEEE 2008, pp 838–842. doi:10.1109/ICCCAS.2008.4657900

Wang K, Babenko B, Belongie S (2011) End-to-end scene text recognition. In: IEEE International Conference on Computer Vision (ICCV). IEEE 2011, pp 1457–1464. doi:10.1109/ICCV.2011.6126402

Wen W, Huang X, Yang L, Yang Z, Zhang P (2009) An efficient method for text location and segmentation. In:. WRI world congress on software engineering. WCSE'09. IEEE 2009, vol 3, pp 3–7. doi:10.1109/WCSE.2009.292

Yao C, Bai X, Liu W, Ma Y, Tu Z (2012) Detecting texts of arbitrary orientations in natural images. In: IEEE Conference on computer vision and pattern recognition (CVPR). IEEE 2012, pp 1083–1090. doi:10.1109/CVPR.2012.6247787