Chapter 28
Interactive Scene Text Detection on Mobile Devices

Jinlong Hu, Baihua Xiao, Chunheng Wang, Cunzhao Shi and Song Gao

Abstract With the increasing resolution and availability of digital cameras, text detection in natural scene images receives a growing attention. When taking pictures using a mobile device, people generally only concerned with interesting texts instead of all of the text in the image. In this paper, we propose an interactive method to detect and extract interesting text in natural scene images. We first draw a line to label a region which contains the texts we want to detect. Then a coarse-to-fine strategy is adopted to detect texts in this label region. For coarse detection, we apply Canny edge detection and connected component (CC)-based approach to extract coarse region from the label region. For fine detection, some heuristic rules are specially designed to eliminate some non-text CCs and then to merge the remaining CCs in the coarse region. To better evaluate our algorithm, we collect a new dataset, which includes various texts in diverse real-world scenarios. Experimental results on the proposed dataset demonstrate very promising performance on detecting text in complex natural scenes.

Keywords Text detection · Interactive · Coarse-to-fine · Connected component analysis

J. Hu · B. Xiao (✉) · C. Wang · C. Shi · S. Gao
The State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences,
Beijing, China
e-mail: baihua.xiao@ia.ac.cn

J. Hu
e-mail: jinlong.hu@ia.ac.cn

C. Wang
e-mail: chunheng.wang@ia.ac.cn

C. Shi
e-mail: chunzhao@ia.ac.cn

S. Gao
e-mail: Song.gao@ia.ac.cn

A. A. Farag et al. (eds.), *Proceedings of the 3rd International Conference on Multimedia Technology (ICMT 2013)*, Lecture Notes in Electrical Engineering 278,
DOI: 10.1007/978-3-642-41407-7_28, © Springer-Verlag Berlin Heidelberg 2014

28.1 Introduction

Text detecting in natural scene images plays a very important role in content-based image analysis. However, this is a challenging task due to the wide variety of text appearances, such as variations in font and style, geometric and photometric distortions, partial occlusions, and different lighting conditions.

Text detection has been considered in many recent studies and numerous methods are reported in the literature [1–6]. Most of the existing methods of text detection could be roughly classified into two categories: region-based and CC-based. Region-based methods need to scan the image at multiple scales and use a text/non-text classifier to find the potential text regions. Chen et al. [1] proposed a fast text detector base on a cascade AdaBoost classifier. As opposed to region-based method, CC-based methods first use various approaches such as edge detection, color clustering or stroke width transform to get the CCs, and heuristic rules or classifiers are used to remove non-text CCs. Pan et al. [7] adopted region-based classifier to get the initial CCs and use the CRF to filter non-text components.

Most of previous methods focus on detecting all of the text in the image. However, when taking pictures using a mobile device, we are only interested in certain text in the image. Moreover, detecting all of the text in the image requires more computation and storage capacity. Therefore, most of these methods are not suitable for use in real-time applications and on mobile devices. In this paper, we propose an interactive method to detect texts in natural scene images. The overall process of the text detection is illustrated in Fig. 28.1. First, Canny edge detection and CC-based approach are applied to quickly extract coarse region from the label region (Fig. 28.1c, d). Then specially designed heuristic rules are used to eliminate the non-text CCs [8]. Finally, the remaining CCs in the coarse region are merged to get the fine region (Fig. 28.1e, f).

In comparison to previous text detection approaches, our algorithm offers the following major advantages. First, interactive text detection only concerned with the region we want to detect. Further, our method provides a reliable binarization for the detected text, which can be directly passed to OCR for text recognition.

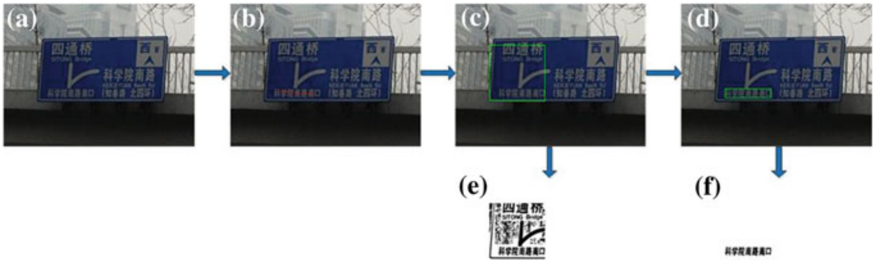

Fig. 28.1 Overview of our method

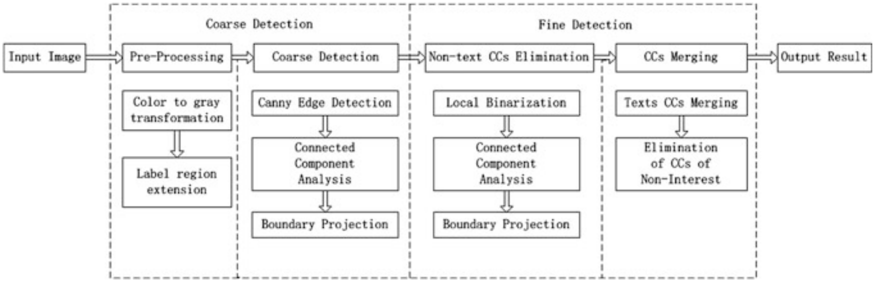

Fig. 28.2 The flowchart of the proposed method

28.2 Text Detection Method

The flowchart of our text detection algorithm is shown in Fig. 28.2. The algorithm works on a gray scale image and can be separated into three main steps: (1) Extract coarse region from the label region; (2) Eliminate the non-text CCs using heuristic rules; (3) Merge the rest CCs to get the fine region.

28.2.1 Coarse Region Extraction

The aim of this step is to extracts coarse region from the label region.

Connected Component Extraction To extract CCs from the image, we use Canny edge detector [9] to produce an edge map (Fig. 28.3c) from the extended image (Fig. 28.3b). This edge detector is efficient and to provide accurate results which makes it suitable for our purpose. With the result of Canny detection, we obtain a binary image where the foreground CCs are considered as text candidates.

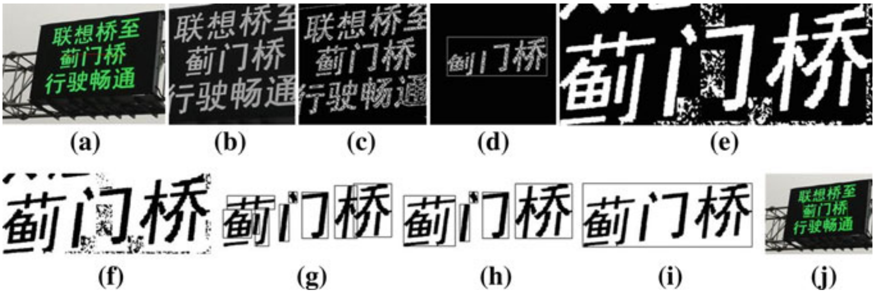

Fig. 28.3 Text detection process. **a** Original image with line labeled by user. **b** Extended image. **c** Edge map. **d** Boundaries of coarse region. **e** Coarse region. **f** Component filtering result. **g** CCs before merged. **h** CCs after merged. **i** Fine region. **j** Detected result

Connected Component Analysis The purpose of component analysis is to identify and eliminate the CCs that are unlikely belong to part of text. Toward this end, we devise a filter which consists of a set of heuristic rules. As in most state-of-art text detection systems, we perform a set simple and flexible geometric checks on each CCs to filter out non-text objects. First of all, very large and very small CCs are rejected. Then, since most of characters have aspect ratio being close to 1, we reject CCs with very large and very small aspect ratio. A conservative threshold on the scale is selected to make sure that some separated strokes are not discarded. The components with one or more invalid properties will be taken as non-text regions and discarded. This preliminary filter proves to be both effective and efficient. A large portion of obvious non-text regions are eliminated after this step.

Boundary projection According to the results of connected component analysis, we apply the projection method to find the boundaries [10]. To find height boundaries, we scan pixels along horizontal line with lengths equals to the width of label line. We scan these lines and calculate the ratio between foreground pixels and background pixels. The procedure is applied until the ratio is less than a certain threshold. As an outcome of this procedure we obtain the top and bottom boundaries. To find the width boundaries, we scan pixels along vertical line with heights equals to the height of top and bottom boundaries. We scan these lines following the same pixel criteria used earlier. The algorithm moves the lines toward left and right until this criteria are fulfilled (Fig. 28.3d).

The combination of these three procedures computes a rectangular bounding box that encloses the text (Fig. 28.3e). However, the produced bounding box may be slightly larger than the minimum bounding box due to noise present in the image. This bounding box is a coarse region, we will extract fine region in this smaller region.

28.2.2 Non-text CCs Elimination

The main purpose of this step is to eliminate non-text CCs using heuristic rules [8].

Local binarization In order to reduce the impact of lighting conditions and complicated background, we propose a local binarization approach to binarize the coarse region (Fig. 28.3e). Our technique is essentially based on Otsu's binarization method [11]. The coarse region is divided in nonoverlapping blocks of in 100*100 pixels, and each block is binarized using Otsu algorithm. Then we obtain the anti-color binarization image via exchanging the foreground and background pixels of the binarization image.

Connected component analysis For binarization image and anti-color binarization image, we use connected component labeling on them separately. And then we perform a set of simple heuristic rules on each CCs to filter out non-text CCs (Fig. 28.3f). First, we remove the CCs which are connected with the image boundary. Then, very large and very small CCs are rejected. Finally, we remove the isolated CCs which are far away from its surrounding CCs. We select the threshold following the same criteria used earlier to make sure that some separated strokes are not discarded.

According to the number of remaining CCs in two kinds of binarization images, we can determine the polarity of the image and choose the appropriate one as the binarization image.

28.2.3 Rest CCs Merging

We found there are still some narrow non-text CCs in the remaining CCs because these CCs are very similar to separated strokes of some characters. Therefore, we propose an algorithm to merge that separated strokes which belong to the same character.

For every remaining CCs (Fig. 28.3g), we scan each CCs around it and decide whether these two CCs are merged based on the heights, widths, and position of their bounding boxes. If one CCs contains another, or two components have overlapping region, or two components are close enough, they are merged into one component (Fig. 28.3h). Each rectangle represents a CC. This step also serves as a filtering step because the CCs which very small and cannot be merged are taken as components casually formed by noises or background clutters, and thus are discard. Once the non-text CCS are eliminated and text CCs are merged, the CCs close to the label line are preserved as the final text region and we obtain the fine region (Fig. 28.3i, j).

28.3 Experimental Results

28.3.1 Dataset and Experiment Setting

For evaluating the performance of the proposed methods, we introduce a dataset for evaluating text detection algorithm, which contains images of real-world complexity.

Although widely used in the community, the ICDAR dataset [12, 13] has two major drawbacks. First, most of the text lines (or single characters) in the ICDAR dataset are horizontal. In real scenarios, however, text may appear in any orientation. The second drawback is that all the text lines or characters in this dataset are in English. These two shortcomings are also pointed out in [7, 14]. In this work, we generate a new multilingual image dataset with horizontal as well as skewed and slant texts. This dataset contains 250 natural images in total. These images are taken from indoor (office and mall) and outdoor (street) scenes using a mobile camera. Some typical images from this dataset are shown in Fig. 28.4a.

The dataset is very challenging because of both the diversity of the texts and the complexity of the backgrounds in the images. The texts may be in different languages (Chinese, English, or mixture of both), fonts, sizes, colors, and orientations.

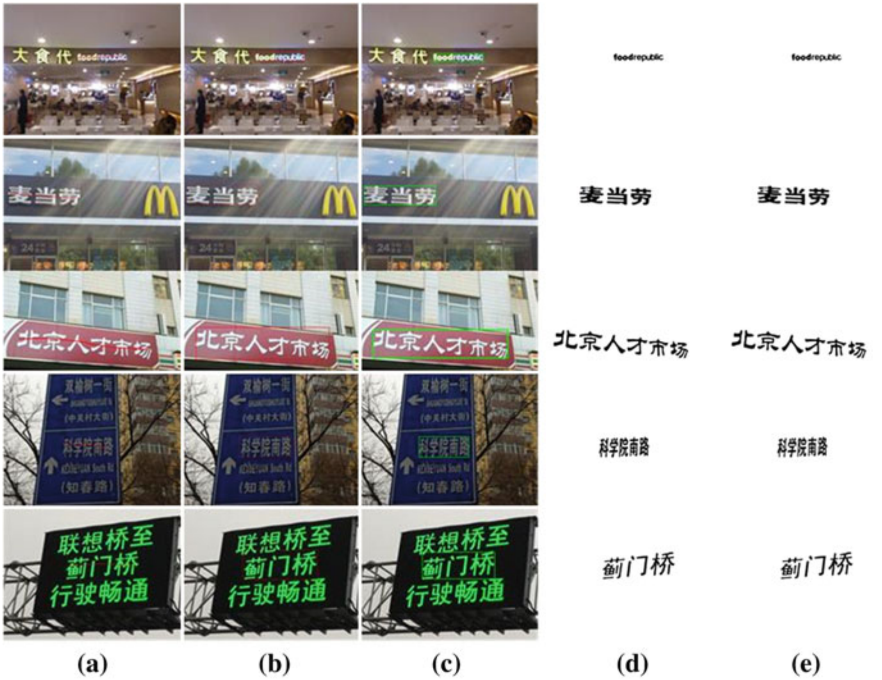

Fig. 28.4 Selected results of our coarse-to-fine method on the proposed dataset. **a** Original image with the line labeled by user. **b** Manually labeled text areas (*red rectangles*). **c** Detected text by our method (*green rectangles*). **d** Manually extracted text. **e** Extracted text by our method

Our evaluation method is inspired from the one used in the ICDAR2003 competition, but it is much simpler. The definitions of precision and recall are:

$$\text{precision} = |\text{TP}|/|E| \quad \text{recall} = |\text{TP}|/|T| \tag{28.1}$$

For text detection, E and T are the sets of estimated rectangles and ground truth rectangles. For text extraction, E is the text pixels extracted by our algorithm, T is the manually labeled text pixels. Where TP is their intersection. There is usually a trade-off between precision and recall for a given algorithm. It is therefore necessary to combine them into a single final measure of quality f:

$$f = 2\text{pr}/(p + r) \tag{28.2}$$

28.3.2 Results and Analysis

To evaluate our coarse-to-fine method for text detection, according to Eq. (28.1) we can compute precision and recall using image areas. Some text detection examples of the proposed algorithm are presented in Fig. 28.4d. The algorithm can

Table 28.1 Performances of text detection method evaluated on the proposed dataset

Step	Precision	Recall	F-measure
1st step	73.80	81.37	77.40
2nd step	86.78	77.20	81.71
3rd step	93.24	76.74	**84.19**

Table 28.2 Performances of text extraction method evaluated on the proposed dataset

Step	Precision	Recall	F-measure
1st step	49.33	70.03	57.88
2nd step	61.82	69.58	65.47
3rd step	91.05	66.20	**76.66**

handle several types of challenging scenarios, e.g., variations in text font, color and size, as well as repeated patterns and background clutters. The results of this experiment are reported in Table 28.1.

To evaluate our coarse-to-fine method for text extraction, according to Eq. (28.1) we can compute precision and recall using image areas expressed in terms of number of pixels. Examples of our algorithm on this dataset are shown in Fig. 28.4e. The results of this experiment are reported in Table 28.2.

From Tables 28.1 and 28.2, we observe that our algorithm achieves significantly enhanced performance when detecting texts of arbitrary orientations. It demonstrates the effectiveness of the proposed method. The images in Fig. 28.5 are some typical cases where our algorithm failed to detect the texts or gave false positives. The misses are mainly due to strong highlights, blur, and low resolution.

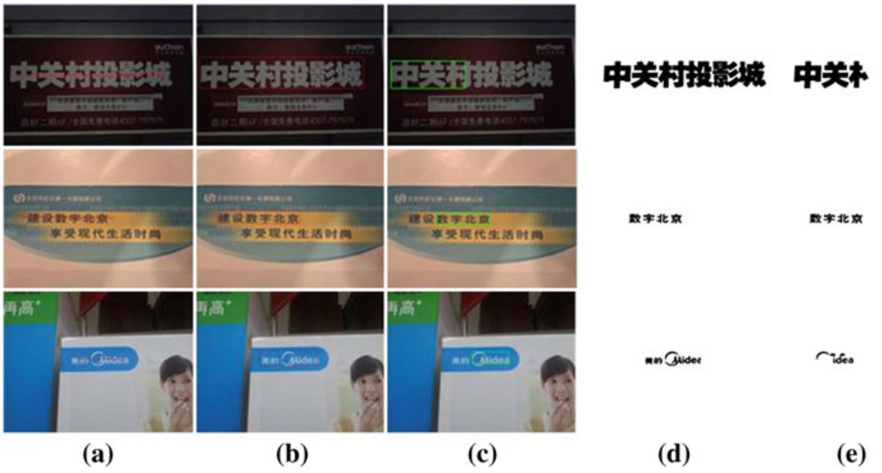

Fig. 28.5 Examples of failure cases. **a** Original image with the line labeled by user. **b** Manually labeled text areas (*red rectangles*). **c** Detected text by our method (*green rectangles*). **d** Manually extracted text. **e** Extracted text by our method

28.4 Conclusions

In this paper, an interactive method was proposed to detect and extract interesting text in natural scene images. We first draw a line to label a region which contains the texts we want to detect. Canny edge detection and CC-based approach are applied to quickly extract coarse region from the label region, then some specially designed heuristic rules are used to eliminate the non-text CCs and the remaining CCs in the coarse region are merged to get the fine region. Then use the high complexity precise approach to detect the small amount of candidate regions can greatly accelerate the speed of text detection and localization. Experimental results on the proposed dataset demonstrate very promising performance on detecting text in complex natural scenes.

Acknowledgments This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No.60933010, No.61172103 and No.61271429.

References

1. Chen X, Yuille AL (2004) Detecting and reading text in natural scenes. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 2, pp 366–373
2. 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), pp 2963–2970
3. Jung K., Kim K. I., Jain A. K (2004) Text information extraction in images and video: a survey. Pattern Recognit 37(5):977–997
4. Liang J, Doermann D, Li H (2005) Camera-based analysis of text and documents: a survey. IJDAR 7(2–3):84–104
5. Shivakumara P, Phan TQ, Tan CL (2011) A laplacian approach to multioriented text detection in video. IEEE Trans Pattern Anal Mach Intell 33(2):412–419
6. Shi C, Wang C, Xiao B, Zhang Y, Gao S, Zhang Z (2013) Scene text recognition using part-based tree-structured character detection. In: IEEE conference on computer vision and pattern recognition (CVPR)
7. Pan Y, Hou X, Liu C (2011) A hybrid approach to detect and localize texts in natural scene images. IEEE Trans Image Process 20(3):800–813
8. Chen H, Tsai SS, Schroth G, Chen DM, Grzeszczuk R, Girod B (2011) Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In: 18th IEEE international conferences on Image processing (ICIP), pp 2609–2612
9. Canny J (1986) A computational approach to edge detection. In: IEEE transaction on pattern analysis and machine intelligence, vol 6, pp 679–698
10. Petter M, Fragoso V, Turk M, Baur C (2011) Automatic text detection for mobile augmented reality translation. In: IEEE international conferences on computer vision workshops (ICCV workshops), pp 48–55
11. Otsu N (1975) A threshold selection method from gray-level histograms. Automatic 11(285–296):23–27
12. Lucas SM (2005) Icdar 2005 text locating competition results. In: Proceedings of the eighth international conference on document analysis and recognition, IEEE, pp 80–84

13. Sosa LP, Lucas SM, Panaretos A, Sosa L, Tang A, Wong S, Yound R (2003) Icdar 2003 robust reading competitions. In: Proceedings of the seventh international conference on document analysis and recognition, Citeseer
14. Yi C, Tian Y (2011) Text string detection from natural scenes by structure-based partition and grouping. IEEE Trans Image Process 20(9):2594–2605