A Hybrid Approach to Detect Texts in Natural Scenes by Integration of a Connected-component Method and a Sliding-window Method

Yojiro Tonouchi[1], Kaoru Suzuki[1] and Kunio Osada[2]

[1] Corporate Research and Development Center, Toshiba Corporation,
[2] IT Reseach and Development Center, Toshiba Solutions Corporation

Abstract. Text detection in images of natural scenes is important for scene understanding, content-based image analysis, assistive navigation and automatic geocoding. Achieving such text detection is challenging due to complex backgrounds, non-uniform illumination, and variations in text font, size, and orientation. In this paper, we present a novel hybrid approach for detecting text robustly in natural scenes. We connect two text-detection methods in parallel structure: (1) a connected-component method and (2) a sliding-window method and outputs basically both results. The connected-component method generates text lines based on local relations of connected components. The sliding-window method consisting of a novel Hough Transform-based method generates text lines based on global structure. These two text-detection methods can output complementary results, which enables the system to detect various texts in natural scenes.

Testing with the ICDAR2013 text localization dataset shows that the proposed scheme outperforms the latest published algorithms and the parallel structure consisting of the two different methods contributes to decreasing false negatives and improves recall rate.

1 Introduction

Text detection in natural scenes has a wide range of applications, such as augmented reality devices, image retrieval, and robotic navigation. Extraction of texts from natural scenes, however, is much more difficult than reading texts from scanned materials. Images which capture natural scenes are variable and subject to the influences of background clutter, lighting conditions, shadowing at different distances as well as the different perspectives, rotations, and skews of the image itself. Moreover, texts in natural scenes can appear anywhere in the image and can have different sizes and layouts.

A great number of works deals directly with detection of text from natural images. In general, the methods for detection text can be broadly categorized into three groups: Connected component methods, sliding-window methods and hybrid methods. Connected component methods (CC method) have recently grown more popular and have reported promising performance on the ICDAR2013 Robust Reading Competition [3]. These methods first detect individual connected components by using local properties (color, stroke-width, etc.) and assuming that the selected property does not change much for neighboring characters. The connected components are then grouped into higher structures such as words and text lines in subsequent stages. The advantage of CC methods is that the complexity typically does not depend on the parameters of the text (range of scales, orientations, fonts) and character segmentation is typically obtained, which can be used for optical character recognition (OCR). But they are sensitive to clutter and occlusions which change connected component structure. Sliding-window methods scan the image at a number of scales and the presence of text is estimated. Generally, a feature vector extracted from each local region is fed into a classifier which estimates the likelihood of text. Then, neighboring text regions are merged to generate text lines. These methods are generally more robust to noise in the image than other methods. On the other hand, a major limitation of these methods is the high computational complexity due to the need to scan the image at several scales. Additionally, these algorithms are typically unable to detect highly slanted text. Hybrid methods integrate CC methods and sliding-window method in order to take advantages of both methods.

In this paper, we present a new hybrid system which connects a CC method and a sliding-window method in parallel to detect text lines. The CC method generates text lines by means of a neighborhood graph. It generates lines based on local relations between connected components. On the other hand, in the sliding-window method, we adopt a new extended Hough transform-based text line generator. The extended Hough transform uses not only position information but also size information of text region in generation of text lines. It generates lines based on global structure that a text line consists of characters which have similar sizes and lie on a nearly straight line. This system outputs both text lines which two methods generate except in the case of overlapped regions. It can detect various text patterns in natural scenes by using these two types of results, which complement one another.

The rest of this paper is organized as follows. Previous work is presented in section 2 and our proposed algorithm is described in detail in section 3. Experimental results are presented in Section 4, and we conclude this paper in Section 5.

2 Previous work

The majority of recently published methods for text detection are CC methods [2, 12, 13, 8]. The methods differ in their approach to individual character detection, which can be based on edge detection [2, 12], character energy calculation [13] or extreme region (ER) detection [8]. Although these methods pay close attention to individual character detection, final segmentation (extractor of connected components) is done at a low level using only local features.

Sliding-window Methods [1, 6] use a window which is moved over the image, and the presence of texts is estimated on the basis of local image features. While these methods are generally more robust to noise in the image than other methods, their computational complexity is high because of the need to search with many windows of different sizes, aspect ratios and, potentially, rotations. Additionally, support for slanted or perspective distorted text is limited, and sliding-window methods do not always provide text segmentation information accurate enough to be used for character recognition.

A hybrid method was presented by Pan et al [10]. It consists of a text region detector and a connected components extractor based on an image local binarization in series. It makes use of size information obtained by the region detector in the image local binarization process and filter out non-text region. It connects the region detector with the connected components extractor directly to decrease false positives in the connected component extraction process.

3 Algorithm description

As mentioned above, both CC methods and sliding-window methods have weak points. Neither alone detects texts universally well. Figure 1 and 2 show some results of our two text detection methods. These figures show that the CC method and the sliding-window method output complementary results.

Hybrid methods which integrate CC methods and sliding-window method are likely to take advantages of both methods. Pan et al [10] connect them in serial. But we connect them in parallel with an aim to decrease false negatives in text detection. In this paper, we present a hybrid system consisting of two text detection methods (a CC method and a sliding-window method) and a method of integrating the results. The overall scheme of our algorithm is summarized in Figure 3. It proceeds by these steps:

1. CC-based text detection (CC method).
2. Sliding window-based text detection (sliding-window method).
3. Integration of text lines.

3.1 CC-based text detection

Our CC-based text string detection method has four steps:

1. Image binarization.
2. Connected component extraction.
3. Connected component verification.
4. Text-line generation from connected components.

Image binarization Niblack's binarization algorithm[9] is adopted. The formula is defined as

$$Niblack(x) = \begin{cases} black, & \mathbf{if} gray(x) < \mu(x) - k \cdot \sigma(x); \\ white, & \mathbf{if} gray(x) > \mu(x) + k \cdot \sigma(x); \\ gray, & \mathbf{other} \end{cases} \tag{1}$$

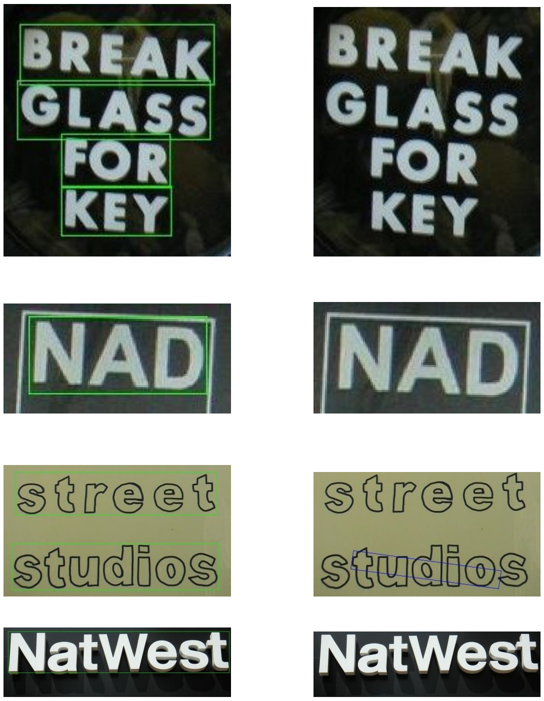


Fig. 1. Some texts in natural scenes which can be detected by the CC method **(left)** but not the sliding-window method **(right)**. Rectangles indicate detected texts.

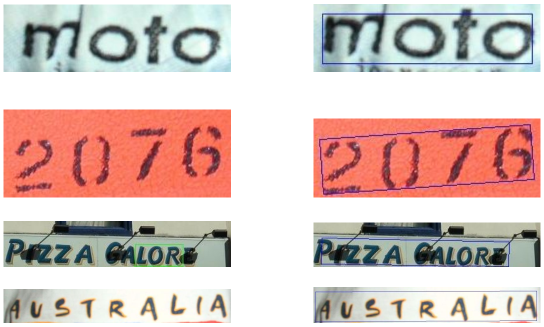


Fig. 2. Some texts in natural scenes which cannot be detected by the CC method **(left)** but can the sliding-window method **(right)**. Rectangles indicate detected texts.

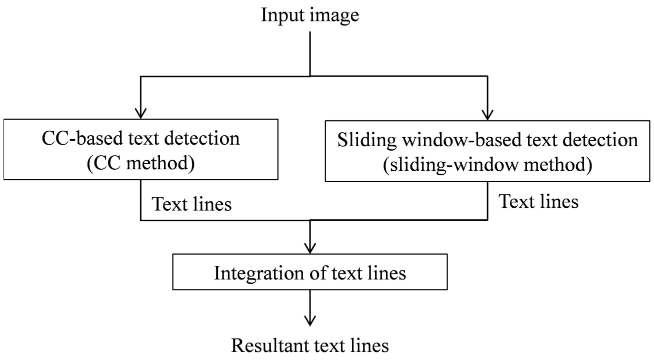

Input image
CC-based text detection (CC method)
Sliding window-based text detection (sliding-window method)
Text lines
Text lines
Integration of text lines
Resultant text lines

Fig. 3. Flowchart of our approach.

, where $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation of intensity within a constant-radius window centered on the pixel x; k is a smoothing term and gray() is gray-scale transform from color image. For a binarized image, black or white values are extracted as text candidate while gray values are not considered further.

Connected component extraction In binarized images, the method connects adjacent pixels which have same value (black or white) and similar stroke width in order to extract connected components. Stroke width can be calculated by the stroke width transform (SWT) [2].

Connected component verification To eliminate non-character connected components, the method rejects connected components which satisfy $N/L < T_{sh}$. The quantity N/L is sharpness and T_{sh} is a threshold parameter. Sharpness is the ratio of N, the number of pixels in the connected component, to L, the number of pixels where gradient magnitudes exceed another threshold parameter T_{gm}. The gradient magnitudes are extracted at each pixel by $\sqrt{v^2 + h^2}$ where v and h are vertical and horizontal gradient respectively, which are calculated by an edge detector such as Sobel filter.

Text line generation from connected components The text line generation has two steps. First, a neighborhood graph, in which nodes are connected components, is constructed, with two connected components sharing an edge if and only if they have similar color, brightness, position, size, and stroke width. Second, connected components are grouped according to whether they lie on a nearly straight line by searching each node and its attached edges in the neighborhood graph by a bottom-up approach. All pairs of connected components in the neighborhood graph are checked to see whether the pair can be grouped in order of increasing distance. To be grouped, a pair of connected components must satisfy any one of the following three conditions.

Case 1 Neither connected components belongs to a group.

Case 2 One of the connected components belongs to a group and the other connected component does not but lies on the line of the group which the first belong to.

Case 3 The connected components belong to different groups, and each lies on the line of the other's group.

Figure 4 is an example of text line generation from connected components.

3.2 Sliding window-based text detection

Our sliding windows-based text detection method has two steps:

1. Character candidate detection.
2. Generation of Text line from character candidates.

Character candidate detection First, the method detects character candidates. We use the detector proposed by Kozakaya et al. [5], which they used to detect the faces of cats. We trained a character dictionary with character samples and used this dictionary for detecting characters.

The system cascades (i) joint Haar-like features with AdaBoost [7] and (ii) co-occurrence histograms of oriented gradients (CoHOG) descriptors with a linear support vector machine (SVM) classifier [11]. The joint Haar-like feature improves discriminative performance by considering co-occurrence of multiple Haar-like features. It can be calculated very quickly due to the integral image technique. A strong classifier is learned by stagewise selection of the joint Haar-like features according to the AdaBoost algorithm. The CoHOG descriptor is based on gradient histogram. Gradient-based features are widely used in human detection and object detection. The CoHOG descriptor is calculated from histograms of paired gradient orientations, which we call a co-occurrence matrix. The co-occurrence matrices with the various orientation pairs are able to capture precise local shape information at multiple scales, and then the extracted CoHOG descriptors are evaluated with a linear classifier obtained by linear SVMs. The combination of these two distinct classifiers enables fast and accurate character detection as, Figure 5 shows.

Generation of Text line from character candidates Second, the method generates text line from the character candidates. We use the extended Hough transform proposed by Kohno et al.[4] to detect text lines which consist of an array of similar size character candidates. The normal Hough transform detects

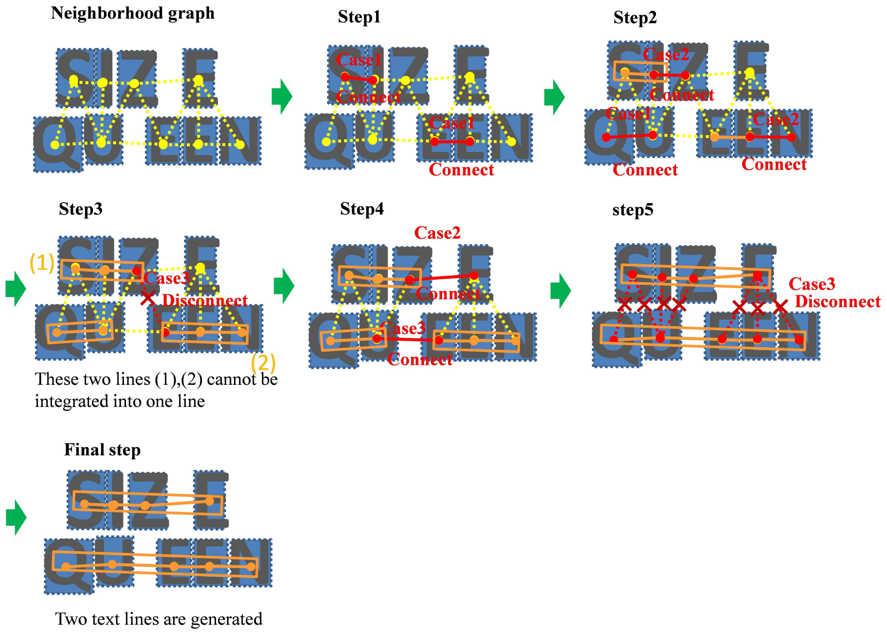


Fig. 4. An example of text line generation from connected components.
(Upper left) Yellow broken lines represent edges of the neighborhood graph.
(From upper middle to lower right) Text line generation proceeds in steps. Yellow broken lines represent potential groupings. Red solid lines represent current accepted groupings and red broken lines represent rejected groupings. Orange solid lines represent groupings which had been accepted and orange rectangles represent the resultant text lines.

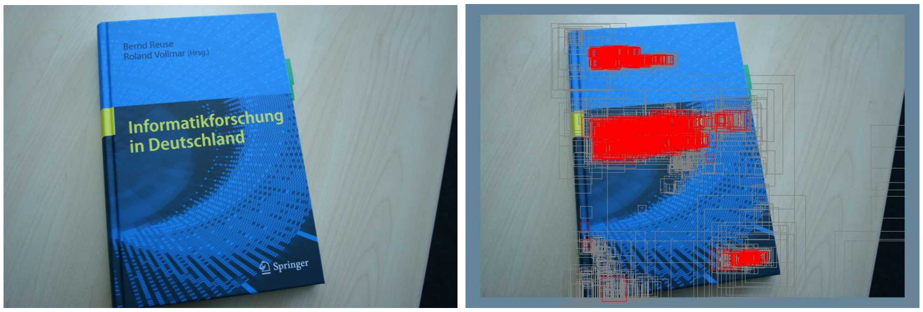


Fig. 5. An example of character candidate detection. **(Left)** Original image, **(Right)** Character detection results by (i) joint Haar-like features with AdaBoost and (ii) Co-HOG descriptors with a linear SVM. Gray rectangles represent regions detected by (i) but rejected by (ii). Red rectangles represent regions detected by (i) and accepted by (ii).

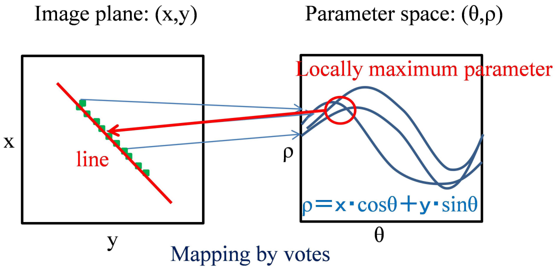


Fig. 6. Normal Hough transform. **(Green point)** pixels which are to be voted.

line consisting of pixels (Figure 6). The parameter space is two-dimensional, with parameters ρ and θ determining the line:

$$\rho = x \cos\theta + y \sin\theta. \tag{2}$$

Kohno et al.[4] extends the Hough transform to generate text lines, which consist of characters and have width (Figure 7). In the extended Hough transform, the parameter space is three dimensional consisting of the previous parameters (ρ and θ) and the new parameter s. s represents text-line width as obtained from the character candidate's height. In the extended Hough transform, character candidate regions (x, y, s) are voted for, instead of pixels. Here, (x, y) is the center coordinate of a character candidate. The method can detect text lines consisting of character candidates which have similar sizes and lie on a nearly straight line. It can detect not only text line position parameter (ρ, θ) but also line width s. Figure 8 shows an example of detection by the extended Hough transform. This example shows that it can generate neighboring text lines which have different region and different sizes even if character candidate regions overlap each other. We summarize differences between the two Hough transforms in Table 1.

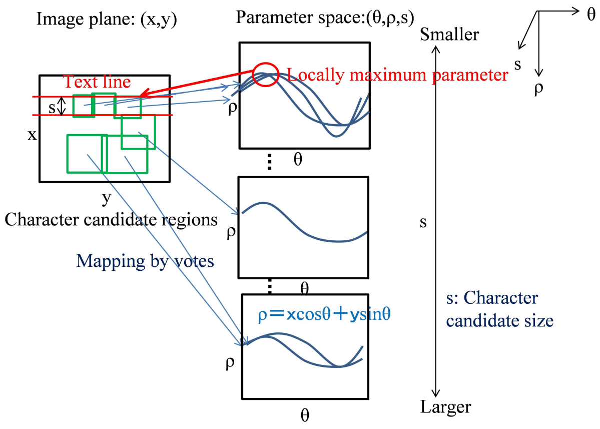


Fig. 7. The extended Hough transform. **(Green rectangles)** character candidate regions which are voted.

3.3 Integration of two text string detection

The proposed method integrates results from two types of text string candidates. We will briefly remind readers of the results' characteristics here. Figure 9 shows results from each type of text detection method. The CC method tends to output

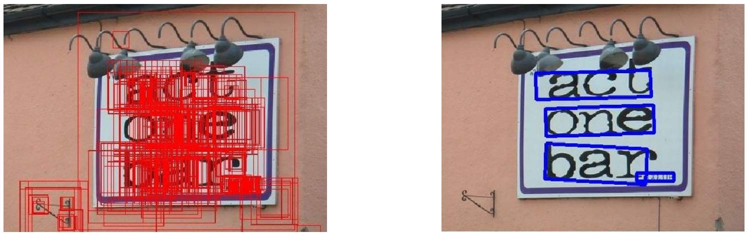


Fig. 8. Text detected by the extended Hough transform. **(Left)** Character candidate regions, **(Right)** text lines generated by the extended Hough transform.

Table 1. Differences between the normal Hough transform and the extended Hough transform.

	The normal Hough	The extended Hough
Voting unit	Pixel (dot)	Region (character candidate)
Voting value	x, y	x, y, s
Parameters space	ρ, θ	ρ, θ, s

candidates which are more accurately positioned than those detected by the sliding-window method, and so we give higher weight to results from the CC method.

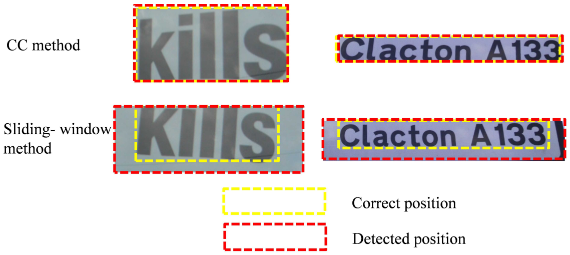


Fig. 9. Positions detected by CC method **(upper)** and sliding-window method **(Lower)**. The CC method gives rectangular results. The sliding-window method gives distortion-corrected quadrilaterals. Yellow broken rectangles represent correct text positions. Red broken rectangles represent detected text positions.

The integration of results from the two methods is performed in the following three cases as shown Figure 10.

Case 1: Isolation All regions are accepted.

Case 2: Inclusion Outer regions are accepted. Inner regions are rejected.

Case 3: Overlap Regions generated by the CC method are accepted. Regions generated by the sliding-window method are rejected.

This has the obvious effect of prioritizing results from the CC method. Figure 11 shows an example initial candidates and the integrated result. This text string integration procedure can be summarized as follows.

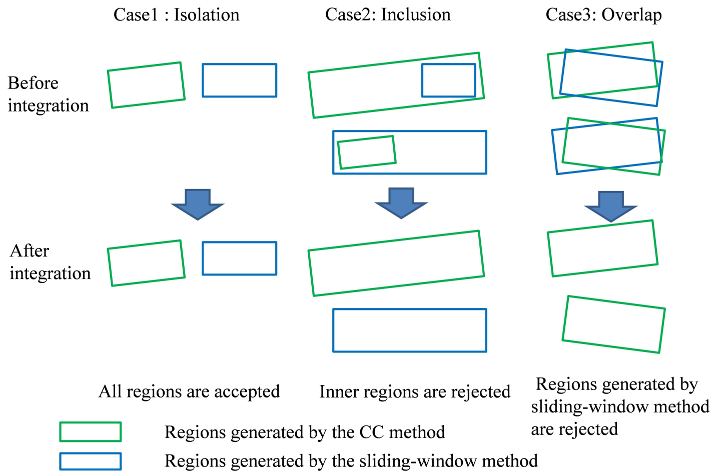

Case1 : Isolation
Case2: Inclusion
Case3: Overlap
Before integration
After integration
All regions are accepted
Inner regions are rejected
Regions generated by sliding-window method are rejected
🟩 Regions generated by the CC method
🟦 Regions generated by the sliding-window method

Fig. 10. Integration of two text string detection. Green rectangles represent regions generated by the CC method. Blue rectangles represent regions generated by the sliding-window method. Black rectangles represent integrated regions.

```
$L := \emptyset$
for all $t_i^c \in$M, $t_j^c \in$N
    if $t_i^c \supset t_j^s$ then $L := L \cup \{t_i^c\}$
    elseif $t_i^c \subset t_j^s$ then $L := L \cup \{t_j^s\}$
    elseif region$(t_i^c \cap t_j^s) >$ Threshold then $L := L \cup \{t_i^c\}$
    else $L := L \cup \{t_i^c, t_j^s\}$
endfor
return $L$
```

In this procedure, M represents results from the CC method, N represets results from the sliding-window method and L is resultant text lines.

4 Experiments

We evaluate the performance of the proposed method on a public dataset:The ICDAR 2013 Robust Reading competition test dataset[3]. It contains 1095 words in 233 images. We use 61 images which we collected as training dataset for the character candidate detector. Under the ICDAR 2013 competition evaluation

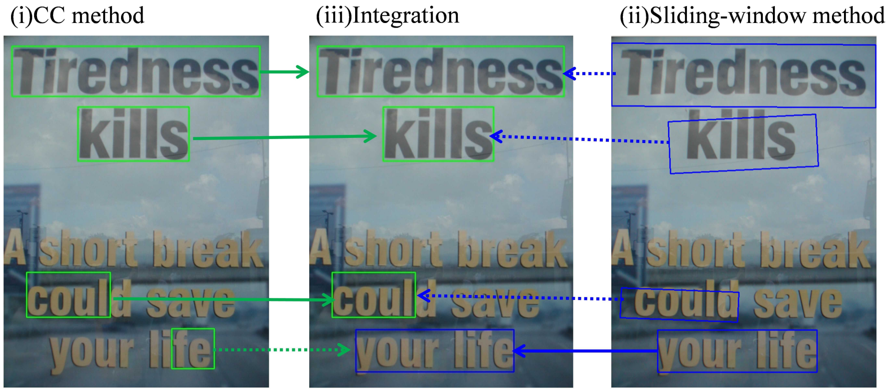


Fig. 11. An example of the integration method. **(Left)** Results of the CC method. **(Right)** Results of the sliding-window method. **(Middle)** Results of the integration. Rectangles are detected text regions, colored by method.

scheme [3], the method achieves the recall of 67.85% , precision of 86.80% and an F-score corresponding to 76.17% in text localization This F-score is better than that of the winner of the ICDAR 2013 Robust Reading competition (75.89%) [1]. Table 2 show the performances of our three methods: 1)our CC method alone,2)our sliding window method alone and 3)our total system(the proposed method) and the winner of ICDAR2013 Robust Reading competition. These results show that the integration process improves recall rate.

Table 2. The text detection results.

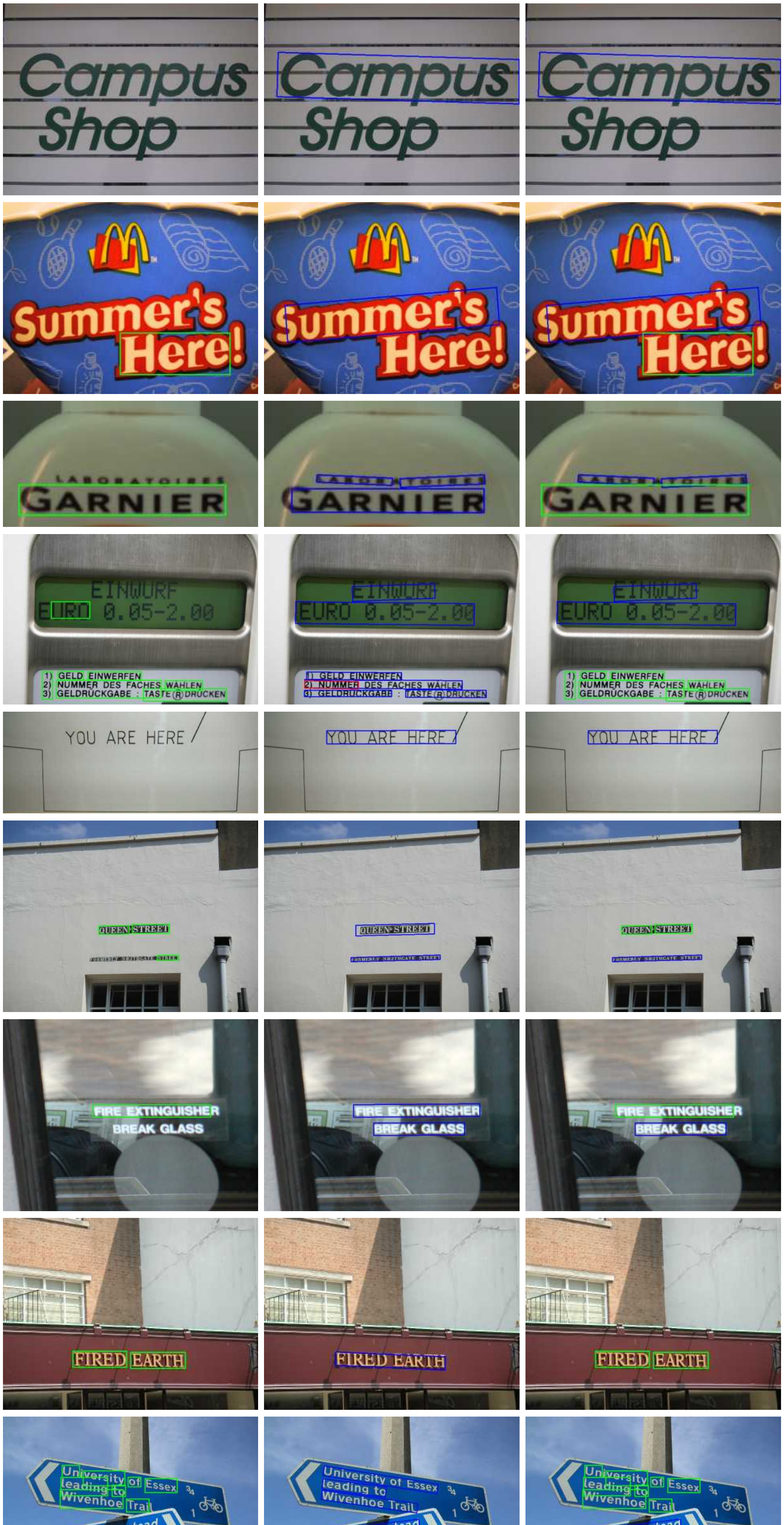
Method	Recall (%)	Precision (%)	F-score
CC alone	61.52	81.81	70.23
Sliding window alone	58.54	85.19	69.39
Integration(Proposed)	67.85	86.80	76.17
Winner of ICDAR2013	66.45	88.47	75.89

Figure 12 shows some examples of text detection by our CC method (left), our sliding-window method (center), and the proposed method (right).

5 Conclusion

We presented a hybrid text detection algorithm which is effective on natural scene images. We built two text detection methods : a CC method and a sliding-window method.

[1] In the ICDAR 2013 Robust Reading competition on-line web-page[3], two methods(anon2014, SWT) reaches 77.03% and 76.80%(F-score). These two methods exceeds the winner of ICDAR2013


Campus Shop
Summer's Here!
LABORATOIRES GARNIER
EINWURF
EURO 0.05-2.00
1) GELD EINWERFEN
2) NUMMER DES FACHES WÄHLEN
3) GELDRÜCKGABE : TASTE R DRÜCKEN
YOU ARE HERE
QUEEN STREET
FORMERLY NORTHGATE STREET
FIRE EXTINGUISHER
BREAK GLASS
FIRED EARTH
University of Essex
leading to
Wivenhoe Trail

In the sliding-window method, we adopt the extend Hough transform to generate text lines based on global text line structure that a text line consists of characters which have similar sizes and lie on a nearly straight line.

By connecting it to the CC method based on local relations between connected components in parallel, the method produces fewer false negatives and improves recall rate. Extensive testing on the ICDAR2013 test dataset shows that the proposed scheme outperforms the latest published algorithms.

We plan to combine the text detection algorithm with a text recognition algorithm to build a complete text detection and recognition system. Text recognition would also help to reduce false positives and improve the precision rate.

References

1. Chen, X., Yuille, A. L.: Detecting and reading text in natural scenes. CVPR (2004)
2. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. CVPR (2010)
3. ICDAR 2013 Robust Reading Competition,
   http://dag.cvc.uab.es/icdar2013competition (2013)
4. Kohno, Y., Aoki, Y., Hamamura, T., Irie, B.: A method for license plate recognition using relative location of extracted Numbers (in Japanese). Vision Engineering Workshop I-37 (2007)
5. Kozakaya, T., Ito, S., Kubota, S., Yamaguchi, O.: Cat face detection with two heterogeneous features. IEEE ICIP (2009)
6. Lee, J.J., Lee, P.H., Lee, S.W., Yuille, A.L., Koch, C.: AdaBoost for text detection in natural scene. ICDAR (2011)
7. Mita, T., Kaneko, T., Hori, O.: Joint haar-like features for face detection. ICCV (2005)
8. Neumann, L., Matas, J.: Real-time scene text localization and recognition. CVPR (2012)
9. Niblack, W.: An introduction to digital image processing. Prentice-Hall (1986) 115-116
10. Pan, Y.F., Hou, X., Liu, C.L.: A hybrid approach to detect and localize texts in natural scene images. IEEE Trans. on Image Processing, vol. 20, no. 3 (2011) 800-813
11. Watanabe, T., Ito, S., Yokoi, K.: Co-occurrence histograms of oriented gradients for pedestrian detection. PSIVT (2009)
12. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. CVPR (2012)
13. Zhang, J., Kasturi, R.: Character energy and link energy-based text extraction in scene images. ACCV (2010)