Automated Text Detection and Text-Line Construction in Natural Images

Chih-Chang Yu, Ying-Nong Chen, Wang-Hsin Hsu and Thomas C. Chuang

Abstract This work develops an automated system to detect texts in natural images captured by the cameras embedded on mobile devices. Unlike former researches which focus on detecting with straight texts, this work proposes a text-line construction algorithm which is able to extract curved text-lines in any orientations. An image operator called the Stroke Width Transform is adopted to exploit connected components which have stroke-like properties. Text components are classified into two types: active and passive. The links of active components are considered the initial orientation of text-lines. Complete text-lines are constructed by linking active and passive components. The system is implemented on the Android platform and the experimental results demonstrate the feasibility and validity of the proposed system.

Keywords Text detection · Stroke width transform · Mobile application

1 Introduction

Employing text detection algorithms on mobile devices assists users in understanding or gathering useful information. Today, many advertisements embed specific QR-codes, allowing users to capture the code using the camera on

C.-C. Yu (✉) · W.-H. Hsu · T. C. Chuang
Department of Computer Science Information Engineering,
Vanung University, 32061 Zhongli, Taiwan, Republic of China)
e-mail: tacoyu@mail.vnu.edu.tw

Y.-N. Chen
Department of Computer Science Information Engineering, National Central University,
32001 Zhongli, Taiwan, Republic of China.)

S.-S. Yeo et al. (eds.), *Computer Science and its Applications*,
Lecture Notes in Electrical Engineering 203, DOI: 10.1007/978-94-007-5699-1_68,

smartphones and then direct them to a specific webpage. Others may ask users to type in keywords on browsers. A system incorporating with text detection and Optical Character Recognition (OCR) techniques provides a more convenient, flexible, and intuitive approach for users. Mobile applications with text detection techniques can greatly assist people. For example, Ezaki et al. [1] proposed a text-to-speech system to assist visually impaired people to understand the content of an image.

Algorithms used for detecting texts are categorized into region-based method and connected component-based method. Region-based methods adopt features such as edges [2], DCT or wavelet coefficients [3], and histograms of oriented gradients [4] are used for analyzing whether the features in a sliding window reveal text-like properties. However, this kind of algorithm is not suitable for applications in mobile devices due to the high computational complexity. Another approach is the connected component-based method. Text candidates are extracted by thresholding firstly. Then, non-text components are filtered out based on certain geometric rules. The connected component-based method is usually computationally inexpensive than the region-based method. Therefore, applying a connected component-based method to mobile devices is a reasonable option. In terms of connected component-based method, Epshtein et al. [5] proposed an image operator called the Stroke Width Transform (SWT). However, the authors in [5] did not explicitly explain how to construct text-lines. Moreover, most text detection algorithms [6, 7] focus on detecting horizontal or straight texts. However, texts may be curved in photographs (see Fig. 3a). Therefore, built upon the effort of SWT, this work proposes a text-line construction algorithm. The advantage of the proposed method is that it can obtain text orientation, regardless of how the text is displayed. Moreover, the proposed method is able to detect texts in different colors. That is, the system does not assume that letters in the same text should have the same color.

2 Proposed System

2.1 Stroke Width Transform

The SWT operator manifests image regions containing parallel edges. Non-edge pixels inside the stroke are filled with the stroke width value. Pixels are then grouped to form components according to their stroke width value. Because the width of strokes should be a fixed value ideally, components which have small variances of SWT values are regarded as text candidates. More detail can be found in [5].

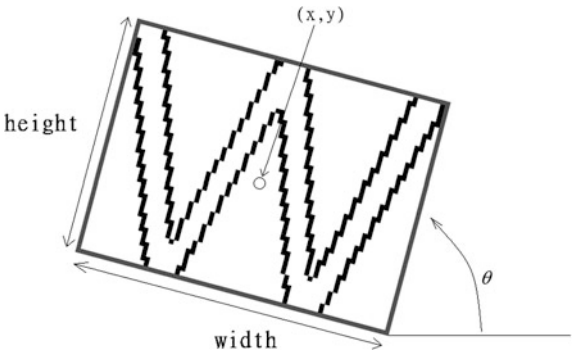

Fig. 1 Minimum rectangle which encloses letter 'W' and its properties

2.2 Text Components Classification

The idea of SWT is very simple, which is to find parallel edges in images. However, many objects in natural images also contain parallel edges, such as fences or stripes. These objects often cause false alarms. To remedy this problem, this work classifies text components into two types, active type and passive type. First, extracted components after performing SWT are discarded if they meet the following criteria: (i) the size of the CC is too small. (ii) The variance of SWT value is larger than a specific threshold. (iii) the CC is overlapped with more than three other CCs. (iv) The average SWT value of the CC is too large.

After eliminating most non-text components, the rest CCs are then enclosed by a minimum rectangle as shown in Fig. 1. For each CC, the following properties are obtained based on the minimum rectangle: the height and width of the rectangle, the centroid of CC, the orientation θ, and the number of contours.

A CC is discarded if it satisfies the following criteria: (i) The number of contours is larger than a specific value. This is because the maximum number of contours of alphanumeric is 3 (e.g., 'B' and '8'). This work set the value to 4 to prevent some characters may stick together due to imperfect SWT. (ii) The height/width ratio of CC is too large. (iii) The fill rate, which is the number of pixels of CC over the area of the rectangle, is too small. CCs which pass the above criteria are classified into active and passive components. A CC is defined as a passive component if it meets the following criteria: (i) the number of contour is 1 and the height/width ratio is larger than a specific threshold. Characters such as 'I' and '1' will be classified as passive components according to this criterion. (ii) The height/width ratio of CC is larger than a threshold *th* (*th* is set to 4 in our experiments). (iii) The fill rate is larger than 0.8. A CC which does not satisfy these criteria is classified as active type. The active components try to connect with other CCs while the passive components are only connected by active components. A problem arises in estimating the orientation of character "O/o" because the letter is nearly round, and thus, allows for multiple orientations. Therefore, before estimating the orientation of the rectangle, a CC is first examined if it is character "o" before computing its orientation. The characteristic of "o" is stated as follows:

(i) the height/width ratio is close to 1. (ii) It contains exactly two contours. (iii) The difference of height/width between the inner contour and outer contour is close to the stroke width. When a component is determined as character "o", it is classified as active component but its orientation is not computed.

In the proposed design, many non-text CCs will be classified as passive components. This work does not discard these components directly because true text components may be broken or merged together due to imperfect SWT extraction. The orientations of active CCs are used to search nearby active CCs.When such things happen; the properties of these components are very similar to that of noises. Therefore, this work classifies them as passive components so that they can be recovered by connecting with active components.

2.3 Text-Line Extraction

According to the orientation of an active component C_i, the directions of two orthogonal rays are considered (see Fig. 2a). Both rays are given scores. The score of the ray is increased if a component C_j that is passed by the ray meet the following criteria: (i) C_i and C_j have similar stroke width. (ii) C_i and C_j have similar orientation or C_j is recognized as character 'O/o'. (iii) The height ratio or width ratio of C_i and C_j do not exceed a threshold. The length of R_1 and R_2 is three times of the diagonal of C_i in this work. The ray with the higher score is considered the potential text-line direction of C_i. C_i will connect to another active component C_j which is closest to C_i according to the potential text-line direction. Figure 2b shows the connection results of active components.

Afterwards, the system starts to extend/merge these text-lines. For each connected set, the end CCs, C_1 and C_2, can be found. According to the link direction of C_1 or C_2, the opposite directions are extended within a certain distance until it encounters another component C_k. C_k is connected to the set if the following criteria are satisfied: (i) C_k is an active component. (ii) C_k is a passive component and the stroke width value of C_k is similar to C_1/C_2. (iii) C_k is a passive component and the height ratio or width ratio of C_k and C_1/C_2 is similar. The main difference to the above step is that the passive component is considered in this step and the orientation information of CCs is discarded. Finally, those connected sets with less than two components are eliminated. The extension results are shown in Fig. 2c.

3 Experiments

To verify the feasibility and validity of the system, the proposed system is developed on the Android platform. 181 images were collected using the smartphone HTC Desire, which has a CPU speed 1 GHz with 576 MB RAM. The resolution of the embedded camera on the smart phone is 500 megapixels. In this

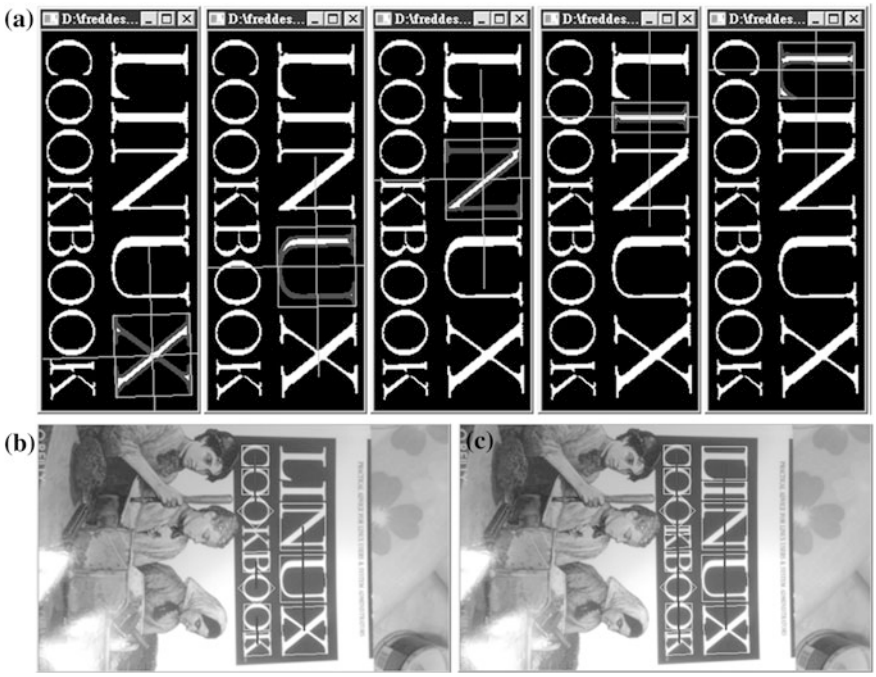

Fig. 2 Example of text-line construction: **a** directions of text-line seeking with each CC, **b** connected active components, **c** constructed text-lines after extension

work, the resolution of the input images is fixed at 640-by-384. Although the camera on HTC Desire is capable of taking higher resolution images, the processing time is too long for practical mobile applications. Test images may contain some backgrounds to test the robustness of the system.

The performance of the proposed system is evaluated using two forms of notation, precision and recall, which are calculated based on the area of the rectangles (see Fig. 3). The definition of precision and recall are listed as follow:

$$\text{Precision} = \frac{C}{E} \tag{1}$$

$$\text{Recall} = \frac{C}{T} \tag{2}$$

E is called the estimate, which is the total area of the rectangles detected by the proposed algorithm. T is the area of the rectangles of true texts, which is manually labeled. C is the intersected area between E and T. In general, the standard f-measure is used for evaluating the quality of the algorithm:

$$f = \frac{1}{\frac{\alpha}{precision} + \frac{1-\alpha}{recall}} \tag{3}$$

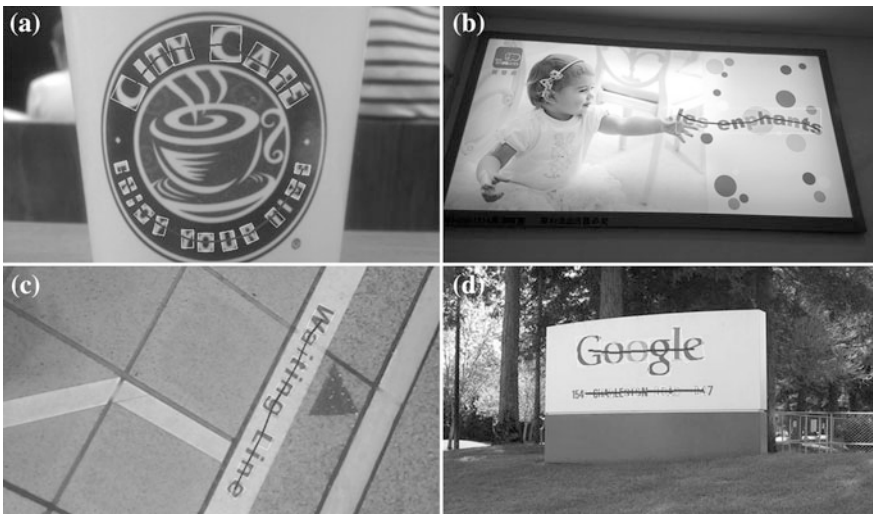

Fig. 3 Detection results with different situations. **a** Curved texts, **b** curved texts and horizontal texts, **c** slanted texts, **d** texts with different colors

α is set to 0.5 to give equal weight to the precision and recall. In this work, the precision rate is 89.2 % and the recall rate is 64.14 %, and the corresponding *f*-score is 0.75. Figure 3 shows some detection results. It is noticeable that the proposed system is able to detect text-lines in any direction or curved text-lines even the colors of characters inside a text-line are different.

4 Conclusions

This paper proposes a system which automatically finds texts in natural images. The image operator called stroke width transform is employed to extract possible text components in images. To detect curved text-lines, a two-stage framework is proposed. Text components are classified into two types: active and passive. Active text components are first linked each other based on some rules to form strong text sets. The orientations of these links are regarded as initial text-lines. These text-lines are further extended to link passive components to form the refined text-lines. This work only considers the detection performance of alpha-numeric letters. This work can be further improved by detecting characters of other languages (e.g. Chinese) in the future.

References

1. Ezaki, N., Bulacu, M., Schomaker, L.: Text detection from natural scene images: towards a system for visually impaired persons. In: 17th International Conference on Pattern Recognition, vol. II, pp. 683–686 (2004)
2. Chen, X.R., Yuille, A.L.: Detecting and reading text in natural scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, Washington, USA, pp. 366–373 (2004)
3. Gllavata, J., Ewerth, R., Freisleben, B.: Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In: 17th International Conference Pattern Recognition, pp. 425–428 (2004)
4. Ma, L., Wang, C., Xiao, B.: Text detection in natural images based on multiscale edge detection and classification. In: 3rd International Congress on Image and Signal Processing, pp. 1961–1965 (2010)
5. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
6. Ferreira, S., Garin, V., Gosselin, B.: A Text detection technique applied in the framework of a mobile camera-based application. In: Workshop of Camera-Based Document Analysis and Recognition (2005)
7. Subramanian, K., Natarajan, P., Decerbo, M., Castañòn, D.: Character-stroke detection for text-localization and extraction. In: International Conference on Document Analysis and Recognition (2005)