Extraction of Character String Regions from Scenery Images Based on Contours and Thickness of Characters

Tomohiro Nishino, Kimitoshi Yamazaki, Kei okada, Masayuki Inaba
The University of Tokyo
Rm.73B2, Engineering Building No.2, The University of Tokyo,
7-3-1, Hongo, Bunkyo-ku, Tokyo, JAPAN, 113-8656
{tomohiro, yamazaki, k-okada, inaba}@jsk.t.u-tokyo.ac.jp

Abstract

This paper describes a method to find strings of characters from natural scene images. It finds character candidates by taking particular note of closed contours in an image, and detects character strings by evaluating their size and line consistency. In the case of low resolution images, some disadvantageous condition occurs, for instance, only a part of actual contour may be able to be found or two closed edges are merged between two characters. To cope with such difficulties, character candidates are first selected, and then the uniformity of thickness of them is evaluated. Character strings can be found through this process. After that, additional character candidates are searched again around the already detected candidates. Moreover, the detection results of character strings are used to find rotation parameters to transform image for reading the character string with propriety.

1 Introduction

A lot of objects on which characters are written exist in our living environment. We humans get much information from these texts. It is expected that robots act in our living environment and support us in the future. If robots can read text on objects such as packages and signs, robots can get information from them, and they can use it in their activation and support for us.

Owing to the progress of OCR, computers have been able to read text in images. However, images have many non-character textures, and they make it difficult to read text by OCR. To cope with that problem, we need to extract character string regions from images.

The algorithms of text extraction from images can be broadly classified under three types[1]. They are gradient feature based[2][3], color segmentation based[4][5], and texture analysis based[6][7]. The gradient feature based algorithm is based on the idea that pixels which have high gradient are the candidates of characters since edges exist between a character and background. One of the researches based on this algorithm uses closed contours[8]. However, with low resolution images, not all of contours of characters are extracted as closed contours, and cannot be extracted as characters.

In this paper, we also use closed contours, however, we calculate thickness of characters from them additionally. By calculate thickness, unclosed contours may be extracted as characters, and character string regions can be revised. Moreover, to cope with inclined character strings, we calculate a line which passes centers of characters in a character string region using Hough transform, and modify the direction of searching characters.

2 Character candidates selection by closed contour extraction

One of the characteristics of common characters in real images is that most of them are capable of producing closed contour when edge extraction process is applied. So we take an approach to detect closed contours from images. Moreover, it is assumed that a character string consists of characters which lie on a straight line. From these assumptions, character string should be found from regions where closed contours are arranged with regularly.

2.1 Closed contour extraction

Fig.1 shows the procedure. First, canny edge detection is applied to an input image in 'Canny edge detection' and edge segments are extracted in 'Extract edge segment'. From this result, short edge segments the lengths of which are less than θ_{l1} are removed in 'Remove short edge (1)'. On the other hand, in some parts which have steeply changes of edge direction, edges are often disconnected in the detection process. So edge re-connection between neighborhoods is added. Endpoints of two edges the distance of which is less than θ_{cn} are connected in 'Connect edge'. After that, short segments the lengths of which are less than θ_{l2} are again removed in 'Remove short edge (2)'. Remaining edges are selected as closed edges if the distances of their end-points are less than θ_{cl} in 'Extract closed edge'. In this extraction process, closed contours which are included in another closed contour are also removed in 'Remove included closed edge'. However, these removed contours in 'Remove included closed edge' are registered in relation to its external contour.

In this process, we set thresholds $\theta_{l1} = 2$, $\theta_{cn} = 2$, $\theta_{l2} = 12$, $\theta_{cl} = 6$.

Fig.2-(a),(b),(c) show one of the results.

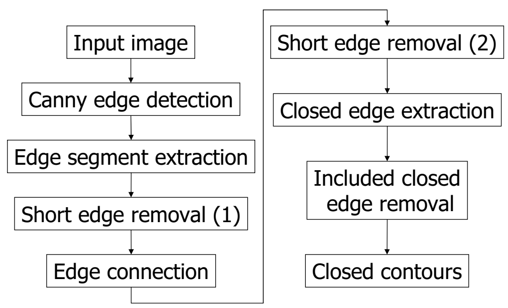

Figure 1. The process of closed contour extraction

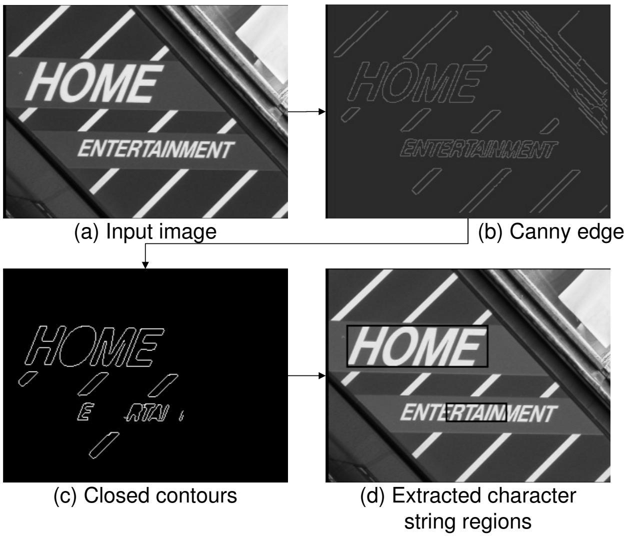

Figure 2. The result of character string extraction

2.2 Character string region extraction

Assuming that characters included in a character string are aligned horizontally, string regions can be extracted by detecting horizontally aligned closed contours. However, characters may not be aligned horizontally, so we should cope with that problem. We explain a method to cope with that problem in Section.4.

We explain how to detect the horizontally aligned closed contour. First, a circumscribed rectangle of a closed contour is calculated. Next, the rectangle is slid to right by some pixels as much as the width of the rectangle. If the rectangle includes the center of a circumscribed rectangle of another closed contour, these two closed contours are assumed to be aligned horizontally and to be included in the same character string. Closed contours which are isolated are assumed not to be characters. Circumscribed rectangles of each character string are assumed to be string regions.

Fig.2-(d) shows one of the results. By this process, string regions of horizontally aligned closed contours are extracted.

In this experiment, since the distances of each character in this input image are short, some characters are not able to be extracted as closed contours, and some of them are not included in character string region. In the next section, we explain the process which adds these characters to character string regions.

3 Revision of character string regions based on thickness of characters

3.1 Calculation of characters' thickness in each character string region

Since closed contours of characters are extracted, thickness of characters included in each character string region can be calculated.

First, some points on closed contours are sampled. Next, vertical lines of the contours which pass the sampled points are calculated, and the intersection points of those line and closed contours are also calculated. The distance between a sampled point and an intersection point is one of the candidates of the character's thickness. Finally, all candidates of each character string region are sorted respectively.

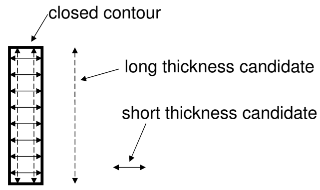

Figure 3. Candidates of a character's thickness

Fig.3 shows that short and long candidates exist in a closed contour. However, in most cases, thickness of characters is calculated from sampled points on longer side. Therefore, the middle of sorted candidates can be assumed to be the thickness of character in each character string region.

3.2 Extraction of similar thickness contours

After the process of calculating the thickness of characters, character string regions are revised based on it.

Both closed and unclosed contours which lie left or right of character string regions are extracted, and each thickness is calculated. These contours are added to the character string region if they have the similar thickness to that of characters in that region.

Fig.4 shows the sorted candidates of characters' thickness in upper region of Fig.2-(d), and Fig.5 shows those in lower region. These two graphs show that the gradient of the sorted candidates is low around the middle, so thickness of characters can be detected safely by selecting the middle. Fig.6-(a) shows the contours which have the similar thickness to character string regions. Fig.6-(b) shows the revised character string region based on thickness of characters.

By this process, both closed and unclosed contours are able to be included in character string region properly.

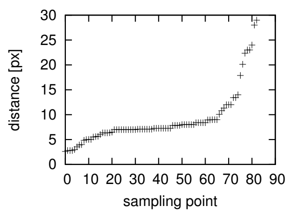

Figure 4. Thickness of characters in upper region of Fig.2-(d)

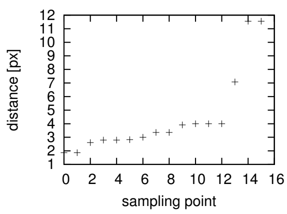

Figure 5. Thickness of characters in lower region of Fig.2-(d)

4 Inclined string detection based on Hough transform

The method described in section 2 and 3 assumes that a character string is captured at horizontal line. This may prevent getting right result when character

Figure 6. Revision of character string regions

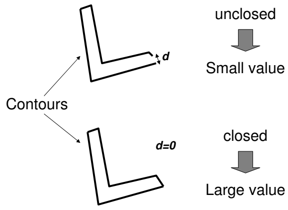

Figure 7. The voted value

strings are inclined because proposed character detection does not work well. However, character strings may practically be written with inclined line, or daily objects which have target strings will be placed with arbitrary direction from a camera. To cope with such various conditions, we take an approach to estimate the inclination angle based on finding a line of characters.

The method complies with Hough transform. Common Hough transform is used to find simple geometrical feature such as line segment or circle. In our assumption, because a string is constructed by characters arranged with line, this principle can also be used in our case. With this method, we detect the lines which pass the center of contours in character string. We use both closed and unclosed contours.

A group of straight lines which intersects at one point can be represented as a sine curve in (r, θ) space. Now we consider two points A and B in (x, y) space. If a sine curve in (r, θ) space, which represents the group of straight lines which cross at point A in (x, y) space, intersects with a sine curve derived from point B, the crossing point indicates a line which connects point A and B. In our case, a center of closed contour is regarded as a point, and each point draws a sine curve in (r, θ) space. The (r, θ) space is divided into lattice arrangement, and grids which include sine curves acquire a vote. If a certain grid has many votes, it indicates that there are a reliable straight line exists in (x, y) space.

In the voting process, a weighting factor based on the completeness of closed contour is considered. That is, the distance between two end-points of a contour is used. If the contour is entirely closed, large value is voted. Otherwise, the voted value is adaptively defined depending on the distance. Fig.7 shows the difference of voted value based on the distance d. If the distance equals to d, the value val is defined by

$$val = \frac{1}{d^2 + 1}. \qquad (1)$$

As a result of this line detection, some character strings are able to be detected. In our implementation, the grid size of (r, θ) space is decided so that each grid has $\pi/60$ radians width between $-\pi \le \theta \le \pi$ area.

Fig.8 shows the input image. Fig.9 shows the target contours. Fig.10 shows the (r, θ) space. Fig.11 shows the result line of Hough transform. Fig.12 shows the extracted inclined character string regions.

In this process, the line which passes the characters in a character string can be calculated. Since the slope of this line is assumed to be the slope of the character string, inclined character string can be extracted by detecting the neighbor closed contours along this angle.

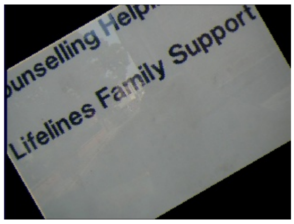

Figure 8. Input image

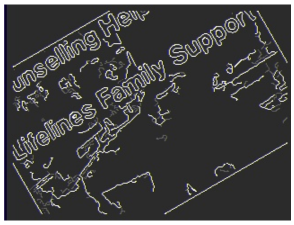

Figure 9. Target contours

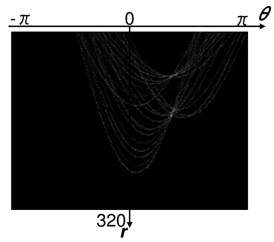

Figure 10. (r, θ) space

Figure 11. Result line of hough transform

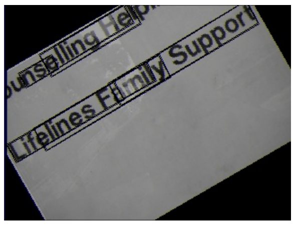

Figure 12. Extracted inclined character string regions

5 Experimental results

Character string regions of scenery images can be extracted in these processes. We experiment with them using some images. All images are 320-by-240 resolution.

5.1 Experiment on ICDAR 2003 dataset

We evaluate this system on the dataset which was used for the ICDAR 2003 robust reading competitions. Many of the images of this dataset have higher resolution than 320-by-240. We resize all images keeping aspect ratio so as to be included in 320-by-240 resolution. In this experiment, the process for detecting slope of character strings is not done.

Fig.13 shows the result. In this figure, (a) shows the extracted character string regions before revision based on the thickness of characters, and (b) shows the region after revision. Character string regions are revised properly considering the thickness of characters.

Table.1 shows the results of precision rate, recall rate, and f measure. The row of “no thickness” is the result of the experiment in which revision of character string regions based on thickness of characters is not done, and the row of “thickness”is the result with revision based on thickness of characters. This valuation is done according to [9]. ICDAR database is annotated at word level, so the system in this paper for extracting character string regions does not get high points. However, considering thickness of characters gives higher points.

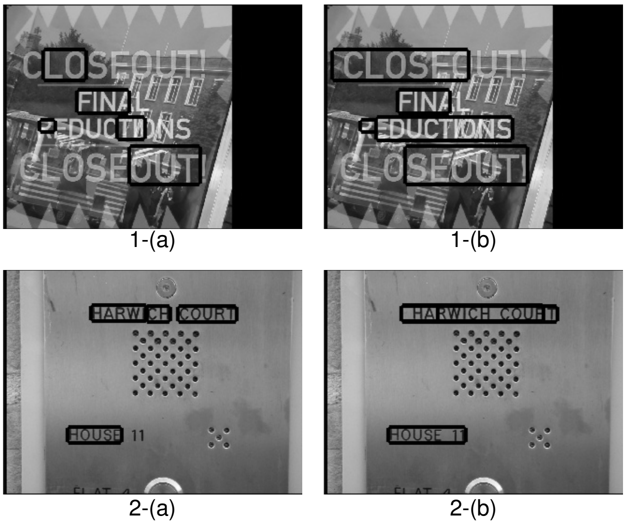

Figure 13. Results of ICDAR images

Table 1. Result precision and recall and f measure

	Precision	Recall	F
no thickness	0.25	0.23	0.22
thickness	0.27	0.26	0.24

5.2 Experiment on inclined character strings

We experiment on inclined character strings. Not all of objects are upright in the environment in which robots act, so inclined character strings should be extracted. Fig.14 shows the results. The slope of character strings can be detected by Hough transform. As a result, character strings on inclined objects can be extracted.

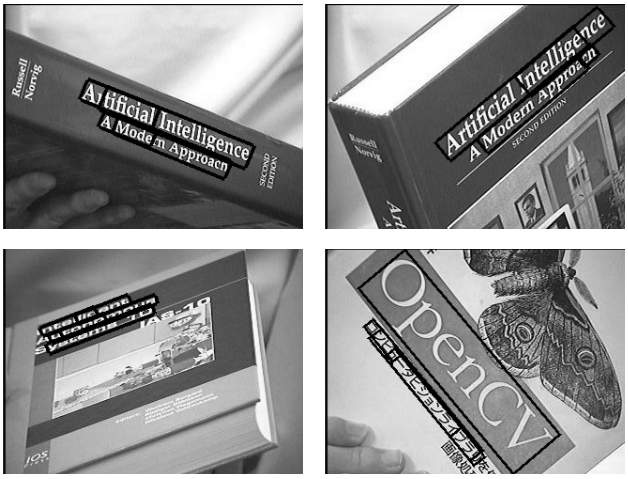

Figure 14. Results of inclined character strings

6 Conclusions

In this paper we described the methods for extracting character string regions from scenery images. Characters have closed contours, and a character string consists of characters which lie on a straight line in most cases. Therefore, by extracting closed contours and search neighbors of them, character string regions can be extracted. However, not all of characters’ contours are extracted as closed contours, so these unclosed contours and isolated closed contours are added to character string regions based on the thickness, and character string regions are revised. To deal with inclined character strings, the line which passes the centers of each contour in a character string region is calculated by Hough transform, and the direction of searching neighbor contours is modified to the slope of this line.

However, texture of non-character is sometimes included in extracted character string regions, so we need more process to remove them.

References

[1] J. Liang, D. Doermann, H. Li: “Camera-based analysis of text and documents: a survey” *IJDAR*, vol.7, no.2-3, pp.84-104, 2005.

[2] Sunghoon Kim, Daechul Kim, Younbok Ryu, Gyeonghwan Kim: “A robust license-plate extraction method under complex image conditions” In *Proceedings of ICPR*, vol.3, pp.216-219, 2002.

[3] Min Cai, Jiqiang Song, Lyu M.R: “A new approach for video text detection” In *Proceedings of ICIP*, vol.1, pp.117-120, 2002.

[4] Lienhart R, Stuber F: “Automatic text recognition in digital videos” In *Proceedings of SPIE*, vol.2666, pp.180-188, 1996.

[5] Miene A , Hermes Th , Ioannidis G: “Extracting textual inserts from digital videos” In *Proceedings of ICDAR*, pp.1079-1083, 2001.

[6] P. Clark, M. Mirmehdi: “Finding text regions using localised measures” In *Proceedings of 11th BMVC*, pp.675-684, 2000.

[7] Keechul Jung, Kwang In Kim, JungHyun Han: “Text extraction in real scene images on planar planes” In *Proceedings of ICPR*, vol.3, pp.469-472, 2002.

[8] Takahashi Hiroki, Kasai Kouichi, Kim Daewoo, Nakajima Masayuki: “Extraction of text regions from scenery images using multiple features” *ITE Technical Report*, vol.25, no.11, pp.39-44, 2001.

[9] Simon M. Lucas: “Text Locating Competition Results” In *Proceedings of ICDAR*, pp.80-85, 2005.