MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

A Method for Character String Extraction from Binary Images Using Hough Transform

Osamu Shiku *
Sasebo National College of Technology

Akira Nakamura†
Kumamoto Institute of Technology

Hideaki Takahira ‡
Sasebo National College of Technology

Hideo Kuroda §
Department of Electrical
Engineering and Computer Science
Nagasaki University

Abstract

This paper describes a method for extracting character strings which have various directions and sizes. In this method, we regard a character string as a line segment which width is equal to the character size, and extract the line segment (character string) using a robust line detection method, Hough Transform. The voting scheme of Hough transform is improved from black pixel-base to line segment-base, which contributes to improvement of extracting accuracy and reduction of processing time.

In order to estimate performance of our method, it was applied to 15 free formed images (512 × 512 pixels) involving about 60 strings. As a result, almost character strings were correctly extracted.

1 Introduction

In order to change paper images into data for computers automatically, it is necessary to extract character strings from the images. However it is difficult to extract strings from such images as free-formed documents which involve strings in various directions and sizes, and characters touched or overlapped with background figures.

Several methods for character string extraction have been reported in the literature. In these methods, connected components or skeletons are grouped into strings using the Hough transform [1, 2] or figure features of a string [3, 4, 5].

We regard a character string as a line segment which width is equal to the character size, and extract the line segment (character string) using a robust line detection method, Hough Transform. In our method, short line segments extracted from input images are grouped into strings using Hough transform which is improved from black pixel-base voting to line segment-base voting.

*Address: 1–1 Okishin, Sasebo, Nagasaki 857–11 Japan. E-mail: shiku@cc.sasebo.ac.jp

†Address: 4–22–1 Ikeda, Kumamoto, Kumamoto 860 Japan. E-mail: akira@ee.kumamoto-it.ac.jp

‡Address: 1–1 Okishin, Sasebo, Nagasaki 857–11 Japan. E-mail: takahira@cc.sasebo.ac.jp

§Address: 1–14 Bunkyo, Nagasaki, Nagasaki 852 Japan. E-mail: kuroda@ec.nagasaki-u.ac.jp

2 Target Images and Strings

A document print is digitized via an image scanner. Input images involve character strings and background figures. Every character string in the images has its own direction and size. The string consists of the same size printed characters which lie along the same straight line.

3 String Extraction Algorithm

Fig.1 shows how to extract the strings from the input image by our method. The following sections describe each process in detail.

3.1 Character Candidate Line Segment Extraction

Fig.1(a) is an example of the input images. The thinning and the line segment approximation are applied to the input image.

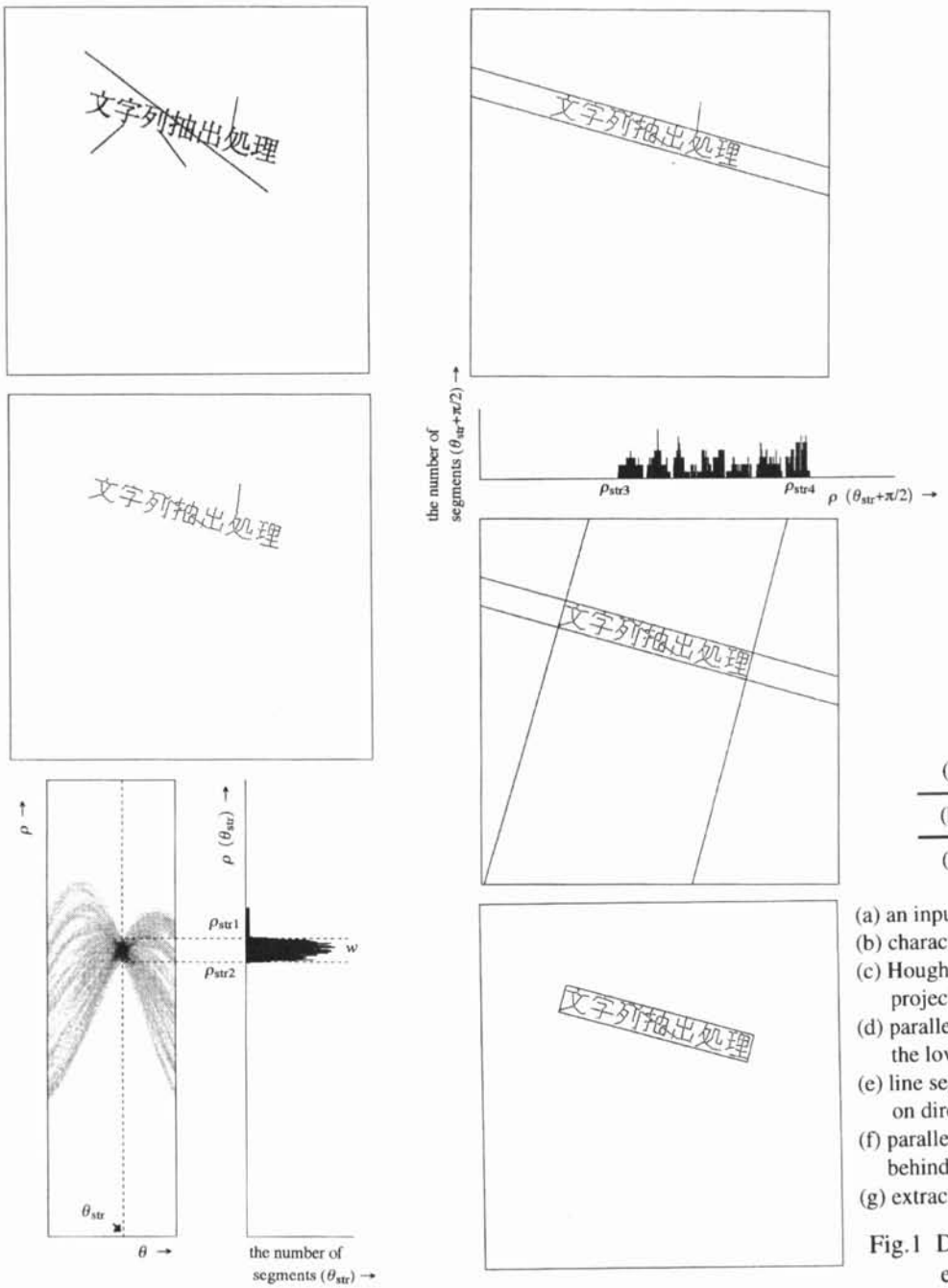
In general, a character has shorter line segments than a figure. Therefore, we extract short line segments from the image as character candidate line segments. Through this process, background figures, which are composed of long line segments, and touch or overlap with characters, are removed.

Fig.1(b) shows examples of the extracted character candidate line segments.

3.2 Hough Transform

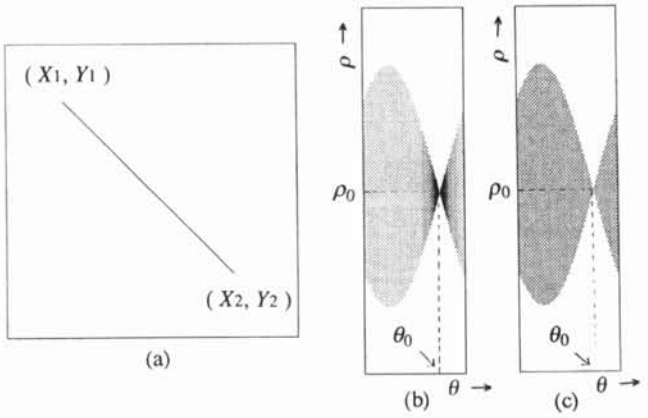
In order to group character candidate line segments into strings, the Hough transform is applied to the extracted line segments.

First, one of the line segments, expressed by the equation ($\rho_0 = X\cos\theta_0 + Y\sin\theta_0$), is picked up, as



(a) an input image
(b) character candidate line segments
(c) Hough domain and line segment projection on direction θ str
(d) parallel lines (the upper line and the lower line of the string)
(e) line segment projection on direction (θ str + $\pi/2$)
(f) parallel lines (in front of and behind the string)
(g) extracted string region

Fig.1 Demonstration of string extraction.



(a) an input image
(b) black pixel-base voting : traditional method
(c) line segment-base voting : our method

Fig.2 Hough domain.

Fig.2(a). The edge points (X_1, Y_1), (X_2, Y_2) of the line segment are transformed into two curve lines in the Hough domain by the following formula(1).

$$\rho = X\cos\theta + Y\sin\theta \quad (1)$$
$$0 \leq \theta < \pi$$

The resolution along the θ direction is set to one degree. The resolution along the ρ direction is set to one pixel.

These two curves corresponding to the edges intersect each other at a point (ρ_0, θ_0) in the Hough domain, as Fig.2(b),(c). The area enclosed by these two curves is named "enclosed area". Every point (ρ, θ) within the enclosed area fulfills the following condition,

if $(X_1\cos\theta + Y_1\sin\theta) \leq (X_2\cos\theta + Y_2\sin\theta)$

then $(X_1\cos\theta + Y_1\sin\theta) \leq \rho \leq (X_2\cos\theta + Y_2\sin\theta)$

else $(X_2\cos\theta + Y_2\sin\theta) \leq \rho \leq (X_1\cos\theta + Y_1\sin\theta)$.

Then, value "1" is voted for every points (ρ, θ). Fig.2 (b) and (c) show the results of two kinds of voting scheme, the black pixel base (traditional method) and the line segment base(our method) respectively. In the black pixel base, value "1" is voted for each black pixels on the curves in the Hough domain which correspond to every black pixels on the line segment of the input image, Fig.2(a). As those curves intersect at the same point (ρ_0, θ_0), the total voted score (i.e. curve lines density) amounts to higher value at the neighbor of the point (ρ_0, θ_0), as Fig.2(b). On the other hand, in the line segment base, value "1" is voted for every point (ρ, θ) in the enclosed area. Consequently, the enclosed area has uniformly same voted value "1" regardless near or far from the intersected point (ρ_0, θ_0) for the single corresponding line segment.

Next, for all the other character candidate line segments in the image, the Hough transform and voting mentioned above are carried out repeatedly. Fig.1(c) shows an example of the resultant Hough domain.

Here, we compare our method with the traditional method. In the traditional method, the voted value of a point (ρ, θ) in the Hough domain means a number of black pixels on a line $\rho = X\cos\theta + Y\sin\theta$ in the image. On the other hand, in our method, the voted value of a point (ρ, θ) means a number of line segments which cross a line $\rho = X\cos\theta + Y\sin\theta$.

The merits of improving the voting scheme from black pixel-base to line segment-base are :

(1) high accuracy of extraction : The number of line segments (our method) is more available feature than the number of black pixels (traditional method).

(2) reduction of processing time : In our method, formula(1) is calculated only two times per a line segment independent of the length of line segment. On the other hand, in the traditional method, formula(1) is calculated repeatedly depending on the length of line segment.

3.3 String Direction Extraction

A string in the image exists between parallel lines ($\rho_{str1} = X\cos\theta_{str} + Y\sin\theta_{str}$ and $\rho_{str2} = X\cos\theta_{str} + Y\sin\theta_{str}$) as shown in Fig.3. The interval of the parallel lines w represents the size of the character. That is, we extract the string by detecting the parallel lines (parameter ρ_{str1}, ρ_{str2} and θ_{str}).

Through the Hough transform, the enclosed areas corresponding to the line segments which exist between the parallel lines intersect at the points between $(\rho_{str1}, \theta_{str})$ and $(\rho_{str2}, \theta_{str})$ in the Hough domain. Thus, the string in the image corresponds to a ridge of the voted value in the Hough domain as shown in Fig.1(c). The width of the ridge w is equal to the string height, and its direction θ_{str} is equal to the string direction. The edge points $(\rho_{str1}, \theta_{str})$, $(\rho_{str2}, \theta_{str})$ of the ridge in the Hough domain represent the parallel lines, the upper line and the lower line of the string in the image.

Then, we detect the edge points of the ridge in the Hough domain. The detail is as follows. Fig.4 shows how to detect the edge points.

1. The Hough domain is smoothed through median filtering (7×1 pixels; $\rho \times \theta$). Fig 4 (b) shows the result of smoothing Fig 4 (a).

2. An edge point is detected from the smoothed Hough domain using an edge detection filter in Fig.5. The edge points of the ridge have both a plus and a minas edge intensities. Fig 4 (c) shows the result of edge detection.

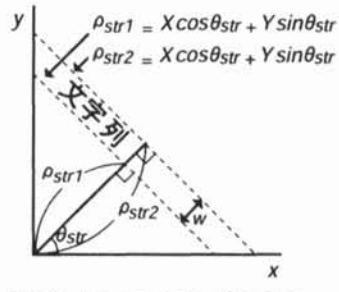


Fig.3 An example of string.

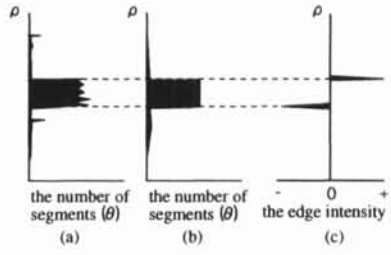


Fig.4 Edge ditection.

Fig.5 Edge detection filter.

3. A couple of edge points which has the same direction θ and the plus/minus edge intensities is detected.

4. The couple of edge points which satisfy the following two conditions is regarded as the edge points $(\rho_{str1}, \theta_{str}), (\rho_{str2}, \theta_{str})$ of the ridge.

   condition 1 : The edge intensities of the couple of points are greater than a fixed threshold.

   condition 2 : The average voted value of the ridge is greater than a fixed threshold.

   Here, condition 1 and 2 mean a string region has many line segments.

3.4 String Extraction

All character candidate line segments between the extracted parallel lines are extracted. And these extracted line segments are projected on one of the parallel lines as shown in Fig.1(e). Then, both end points $(\rho_{str3}, \theta_{str} + \pi/2), (\rho_{str4}, \theta_{str} + \pi/2)$ of the string are determined by the line segment projection. These points $(\rho_{str3}, \theta_{str}+\pi/2), (\rho_{str4}, \theta_{str}+\pi/2)$ represent parallel lines in front of and behind the string in the image as shown in Fig.1(f).

The region surrounded with the two pairs of the parallel lines extracted through step **3.3** and **3.4**, is regarded as a string region as shown in Fig.1(g).

In case of extracting another string, the character candidate line segments regarded as the string are deleted from the Hough domain. And the step **3.3** and **3.4** are repeated.

4 Preliminary Experiment

The proposed method has been implemented on a SUN SPARCstation 10 workstation in C language.

A preliminary experiment was carried out for 15 original images (image size : 512 × 512 pixels) involved 60 character strings (character size : 20-50 pixels). A image involves about 1 - 8 strings.

Fig.6 shows examples of experimental results (input images (left), character candidate line segments and string regions (right)). The strings in various directions and sizes are extracted correctly. But, a few string is mis-extracted (Fig.6 string A), because the edge of the ridge are mis-detected from the Hough domain.

5 Conclusion

This paper proposes a method for character string extraction based on Hough transform.

The merits of this method are:

(1) the voting scheme is improved from black pixel-base to line segment-base, which contributes to improvement of extracting accuracy and reduction of processing time.

(2) strings in various directions and sizes can be extracted.

The method was applied to images which involve strings in various directions and sizes. As a result, almost strings were correctly extracted.

References

[1] L.A.Fletcher and R.Kasturi, "A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images," IEEE Trans. PAMI, Vol.10, No.6, pp.910-918, 1988.

[2] L.L-Sulem, A.Hanimyan and C.Faure, "A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents," Proc 3rd ICDAR, Vol.II, pp.774-777,1995.

[3] K.Takizawa, D.Arita, M.Minoh and K.Ikeda, "Extraction of Character Strings from Unformed Document Images," Proc 2nd ICDAR, pp.660-663,1993.

[4] F.Hones and J.Lichter, "Text String Extraction within Mixed-Mode Documents," Proc 2nd ICDAR, pp.655-659,1993.

[5] H.Hontani and S.Shimotsuji, "Character Detection based on Multi-Scale Measurement," Proc 3rd ICDAR, Vol.II, pp.644-647,1995.

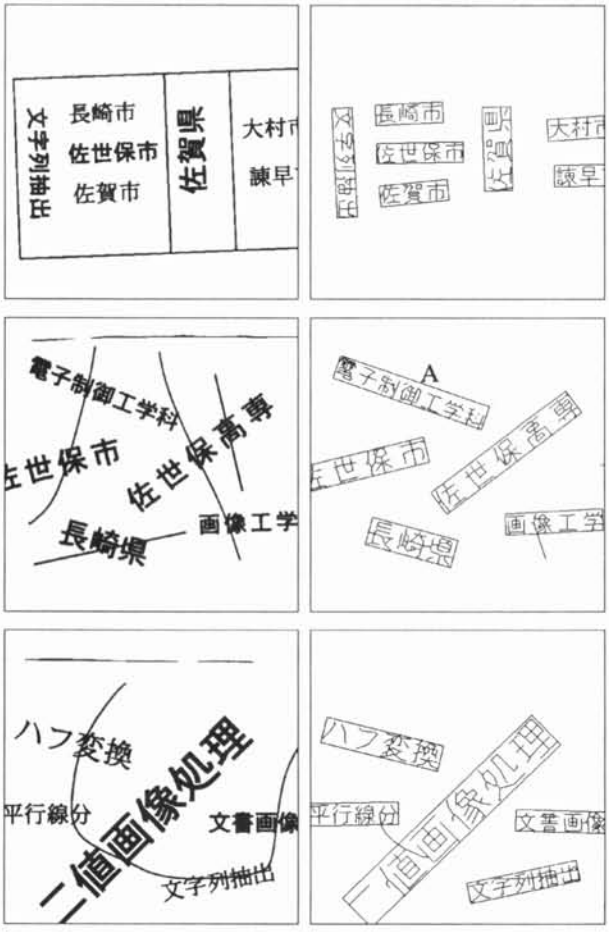


Fig.6 The results of extracting character strings.