Isolated Character Recognition by Searching Features in Scene Images

Kazuya NEGISHI[*], Masakazu IWAMURA[†], Shinichiro OMACHI[*], and Hirotomo ASO[*]
[*]Tohoku University, 6–6–05 Aoba, Aramaki, Aoba-ku, Sendai-shi, 980–8579 Japan
[†]Osaka Prefecture University, 1–1 Gakuencho, Sakai-shi, Osaka, 599–8531 Japan

Abstract

Conventional segmentation technique cannot extract an isolated character and a touching character. In this paper, to utilize information of such characters, we propose a novel character recognition method based on extracting feature points and voting. The voting algorithm of the proposed method is similar to the generalized Hough transform. This method enables us to extract and recognize such troublesome characters in relatively shorter computational time. The effectiveness of the proposed method is confirmed by experiments.

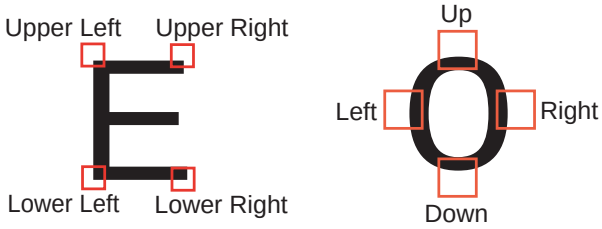
1 Introduction

Accuracy of character recognition for segmented character image is enough for practical use. However, a conventional segmentation method can be applied to only *well-defined* problems such that the characters constitute a string and a character is completely separated from other characters and so on. Therefore, an *isolated character* which does not constitute a character string, and a touching character which is connected to other characters are hard to extract. In this paper, we propose a novel character recognition method which executes segmentation and recognition of a character simultaneously. The proposed method extracts features and then votes. The voting algorithm is similar to the generalized Hough transform [1, 2, 3]. The proposed method allows segmentation of an isolated character, a touching character and so on in relatively shorter computation time.

2 Preparation

2.1 Input image and reference image

"Input image" is a relatively large image such as a photograph including isolated characters. "Reference image" is a relatively small image of one of the 52 alphabet letters and 10 numerals.



(a) Corners. (b) Curves.

Figure 1. Features.

2.2 Histogram of edge directions

The proposed method searches for eight kinds of features described in Sec. 2.3 from the square area of 5×5 pixels in the input image and the reference images. The square area is called *searching area*. Each feature is characterized by a histogram of edge direction in a searching area.

A histogram is constructed as follows. In advance, the edge direction and the edge intensity of each pixel is calculated with the Sobel filter. The edge direction is quantized into n bins whose width is $\theta = \frac{2\pi}{n}$ radian. Namely, n intervals of bins are defined as

$$\left[-\frac{\pi}{n} + \frac{2\pi}{n}t, \frac{\pi}{n} + \frac{2\pi}{n}t\right), \quad 0 \leq t \leq n-1. \qquad (1)$$

Then, a histogram whose bin width is $\theta = \frac{2\pi}{n}$ radian is constructed in a searching area. To eliminate the effect of noise, a pixel whose edge intensity is less than a threshold is regarded as a *no edge pixel*. The histogram does not contain such *no edge pixels*. Finally the histogram is normalized so that the sum of all the bin values to be 1. This normalized histogram is used as a feature vector of n dimensions for the searching area.

2.3 Features

A corner, a curve, a branch, an intersection and a bending point have been considered as effective features in character recognition [4]. In the proposed method, four corners and four curves are used as features because they are easy to extract by search and seem to be effective. Four corners are "upper left", "upper right", "lower left" and "lower right" (See Fig. 1(a)). Four curves are "up", "down", "left" and "right" (See Fig. 1(b)).

To extract features robustly, larger θ is desired. However, to extract complex features, smaller θ is required. Therefore, θ depends on features: $\theta = \pi/2$ (i.e., $n = 4$) was used for corners, and $\theta = \pi/8$ (i.e., $n = 16$) for curves. A loose feature of four bins can extract corners of both regular font such as Arial and oblique font such as Franklin.

2.4 Similarity

To evaluate how much a searching area is similar to each feature, a similarity measure proposed by Swain et al. [5] is used. Let H be the histogram of the searching area, M be the histogram of a corner feature or a curve feature. Then, the similarity between H and M are

$$S_{HM} = \sum_{t=1}^{Q} \min(H_t, M_t), \tag{2}$$

where H_t and M_t are the t-th bin value of H and M respectively, and Q is the number of bins of the feature (i.e., 4 or 16). Since both H and M are normalized histograms, $0 \leq S_{HM} \leq 1$. If the similarity of the searching area is larger than a threshold T_1, the searching area is regarded as a feature point. The range of T_1 is $0 \leq T_1 \leq 1$.

3 Segmentation and recognition of isolated characters

The proposed method is similar to the generalized Hough transform. Therefore we explain the generalized Hough transform at first, then we explain the proposed method.

3.1 The generalized Hough transform [3]

The generalized Hough transform is used to detect a figure that cannot be analytically easily expressed although it has the particular silhouette. We explain the generalized Hough transform without considering rotation and expansion or reduction below.

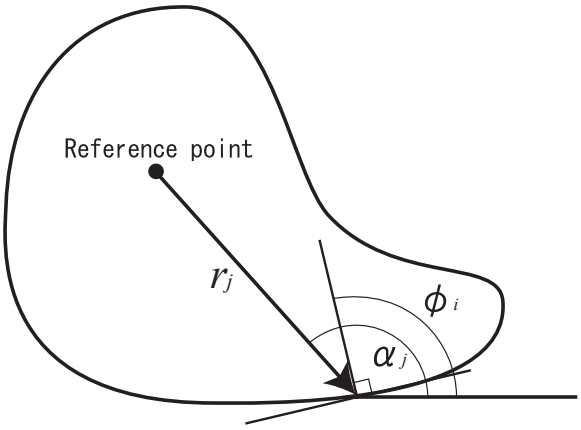


Figure 2. The generalized Hough transform.

Edge direction	Locations of feature points from the reference point
ϕ_1	(r_{11}, α_{11}), (r_{12}, α_{12}) ,...
ϕ_2	(r_{21}, α_{21}), (r_{22}, α_{22}) ,...
⋮	⋮

Table 1. R table of the generalized Hough transform.

3.1.1 Description of a figure

In the generalized Hough transform, a target figure is registered by feature points. This process equals to construct an R table as shown in Table 1. At first, a reference point in the target figure is set as shown in Fig. 2 and each edge point of the target figure (we call this feature point) is also set. A vector from the reference point to the j-th feature point (which corresponds to a combination of r_j and α_j in Fig. 2) is calculated, and the edge direction of the feature point (ϕ_i in Fig. 2) and the vector is registered in the R table. The edge direction is used as an index.

3.1.2 Detection of a figure

The following procedure detects the target figure. First, all the edge directions are examined. For convenience of explanation, the edge direction at coordinate (X, Y) is assumed to be ϕ. Secondly, candidate locations of the reference point are calculated from the edge direction ϕ by referring the R table. The candidate of the reference point calculated is called *voting point*. Then the voting value at the voting point is increased by one degree. Here, if $\phi = \phi_2$, the voting points will be $(X - r_{21}cos\alpha_{21}, Y - r_{21}sin\alpha_{21})$ and $(X - r_{22}cos\alpha_{22}, Y - r_{22}sin\alpha_{22})$. After all the votes

are finished, figures whose reference point have large voting values are detected.

3.2 The proposed method

The difference between the proposed method and the generalized Hough transform is summarized as follows.

(1) Although a vote in the generalized Hough transform is based on edge directions, a vote in the proposed method is based on types of the detected features.

(2) Although a vote in the generalized Hough transform increases the voting value only at the reference point, a vote in the proposed method increases not only the reference point but also the points around the reference point. This enables to detect figures robustly even if fonts are different.

(3) Although the maximum number of the voting value is undecided in the generalized Hough transform, it is described by $F_s^{(k)}$ defined in Sec. 3.2.1 in the proposed method.

3.2.1 Description of a figure: detection of the feature points from the reference image

First of all, all the corner and curve feature points are detected from the reference images. The information on the positions and types of the feature points are used for description of a figure. However, since the feature points tend to be detected too much, these points are narrowed down to a representative point as illustrated in Fig. 3. The representative point is decided to have the highest similarity among the points around it within q pixel distance. $q = 5$ is used in this paper.

Then the number of the points represented by a representative point is used as the *voting weight*. Let $W_{ij}^{(k)}$ be the voting weight of the j-th representative point of the i-th feature of character k. The position of the representative point is determined by *voting vectors* as illustrated in Fig. 4. In the generalized Hough transform, the vector corresponds to a combination of r_j and α_j. It is defined as follows. Let the origin be the upper left corner of the reference image of a character. Let $N_i^{(k)}$ and $R_i^{(k)}$ be the number of occurrences of the i-th feature in the reference image and that of their representative points respectively. Let $(x_{ij}^{(k)}, y_{ij}^{(k)})$ be the coordinate of the j-th representative point of the i-th feature in the character k. The voting vector is defined as

$$\boldsymbol{f}_{ij}^{(k)} = (x_{ij}^{(k)}, y_{ij}^{(k)}), \quad 1 \leq i \leq 8,\ 1 \leq j \leq R_i^{(k)}. \tag{3}$$

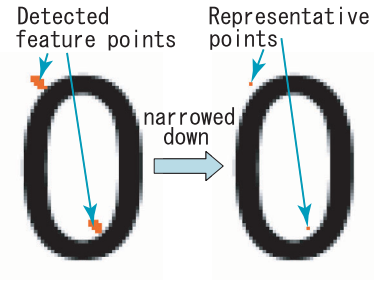


Figure 3. Detected feature points of the upper left corner feature from the reference image (left) and their representative points (right).

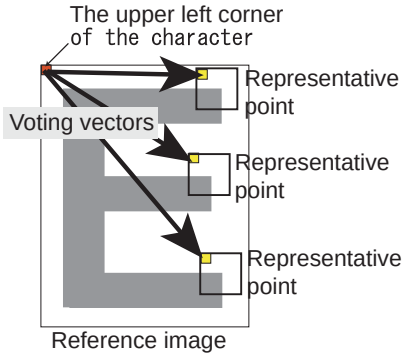


Figure 4. Feature point extraction from a reference image. Three representative points represent three clumps of feature points extracted as the upper right corner. A voting vector is a vector from the upper left corner of the character image to a representative point.

The total number of occurrence of eight features in the reference image of character k is defined by

$$F_S^{(k)} = \sum_{i=1}^{8} N_i^{(k)}. \tag{4}$$

This value is used in Sec. 3.2.2.

3.2.2 Detection of a figure: segmentation

At first, all the feature points are detected from the input image, and these feature points are also narrowed down into representative points as in Sec. 3.2.2. Then, figures are detected by voting. The categories of the characters and their locations are also determined by voting at the *voting point*

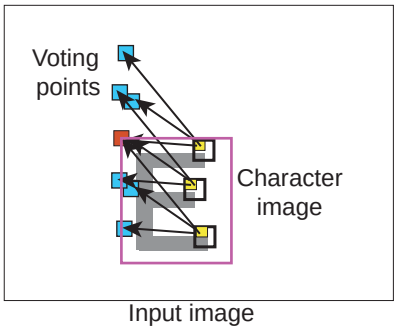


Figure 5. Feature point extraction and segmentation from the input image. There are three voting points for each representative point in this example.

$\boldsymbol{P}_{ijl}^{(k)}$. $\boldsymbol{P}_{ijl}^{(k)}$ is given as

$$\boldsymbol{P}_{ijl}^{(k)} = \boldsymbol{x}_{il} - \boldsymbol{f}_{ij}^{(k)}, \quad 1 \leq j \leq R_i^{(k)}, \text{ for all } k, \tag{5}$$

where $\boldsymbol{x}_{il}$ is the l-th representative point of the i-th feature in the input image. In short, the voting point is a candidate of the upper left corner of a character image as illustrated in Fig. 5.

To allow deformations of a character image, the neighbors of the voting point are also voted. The voting rule is as follows: Let $v_{ij}^{(k)}(x,y)$ be the *voting table*, which stands for the voting value at (x,y) of the input image. In advance, it is initialized as $v_{ij}^{(k)}(x,y) = 0$ for all (x,y). The neighbors within r pixels distance from the voting point are voted as

$$v_{ij}^{(k)}(x,y) = W_{ij}^{(k)}, \quad \text{for } (x,y) \text{ s.t. } \left|\left|(x,y) - \boldsymbol{P}_{ijl}^{(k)}\right|\right| \leq r. \tag{6}$$

In this paper, $r = 10$ is used. Note that even if the same point is voted more than once, the voting value is $W_{ij}^{(k)}$ as long as the same voting vector is used.

After voting, let $V^{(k)}(x,y)$ be the sum of the voting tables $v_{ij}^{(k)}(x,y)$. Namely,

$$V^{(k)}(x,y) = \sum_i \sum_j v_{ij}^{(k)}(x,y). \tag{7}$$

$V^{(k)}(x,y)$ stands for the possibility that the upper left corner of the character image of category k exists at (x,y). Let T_2 be the threshold of character segmentation where $0 \leq T_2 \leq 1$. If

$$V^{(k)}(x,y) \geq F_S^{(k)} T_2 \tag{8}$$

is satisfied, (x,y) will be determined to be the upper left corner of an image of character k. Note that, $F_S^{(k)} = \max V^{(k)}(x,y)$.

3.2.3 Comparison with the traditional template matching method

Compared with the simple traditional template matching method, the proposed method can detect a deformed character more robustly. Therefore the computational cost of the proposed method is a little larger than that of a simple template matching method.

The details of the computational cost are described below. When the size of the reference image is $M \times M$, and that of the input image is $N \times N$, the computational cost of the simple template matching method is $O(M^2N^2)$. In the proposed method, for the feature extraction, it is $O(8 \times 5^2N^2)$ because it searches 8 kinds feature of 5×5 with template matching method. $O(8 \times 5^2N^2)$ is less than $O(M^2N^2)$ for $M > 10\sqrt{2}$, i.e. $M \geq 15$. For the voting algorithm, let $X_i^{(k)}$ be the number of representative point of the i-th feature of character k in the input image. Then, the computation cost of the voting algorithm of the proposed method is $O(r^2 R_i^{(k)} X_i^{(k)})$. Since the voting algorithm is added to template matching method, the computation cost of the proposed method is larger than the simple template matching method.

4 Experiments

To confirm the effectiveness of the proposed method, three experiments were carried out against (1)different fonts, (2)touching characters, (3)isolated characters. As the preliminary to all the experiments, both the reference and input images were converted into gray scale and smoothing filter was applied to them.

The proposed method ignores *no edge pixels*, and does not use the information of them for segmentation and recognition. However, combination of many *no edge pixels* and a few edge pixels in a searching area can cause miss detection of the feature points. Therefore, a searching area in which over 70% pixels are *no edge pixels* is ignored.

Recognition results are classified into three categories for evaluation: “Correct”, “Match”, and “Miss”. If the extracted character images consist of only the same category as the reference one, it is classified into “Correct”. If the extracted images include the same category as the reference one and more than one other characters, it is classified into “Match”. If all the character images of the same category as the reference one are not extracted, it is classified into “Miss”.

4.1 Against different fonts

Input images of several fonts were prepared. When the fonts of the reference image and the input image were the

Font	Correct	Match	Miss
Arial Black	18 (29%)	43 (69%)	1 (2%)
Franklin	15 (24%)	44 (71%)	3 (5%)
Maru Gothic	22 (35%)	23 (37%)	17 (27%)

Table 2. The number of occurrences and its ratio fot different fonts. The sum of ratios is not always 100% because the ratios are rounded.

same, recognition rate was 100%. Therefore, experimental results against different fonts are shown here. Arial was used for the reference images, and Arial Black, Franklin and Maru Gothic were used for the input image: Arial Black is bold, Franklin is oblique, Maru Gothic has rounded corners. Two thresholds $T_1 = 0.75$ and $T_2 = 0.85$ were used. Recognition results are summarized in Table. 2.

The table shows that almost 30% were classified into "Correct". Also, more than 95% of Arial Black and Franklin, and more than 70% of Maru Gothic were classified into either "Correct" or "Match". Though images of "Match" contain images of other categories, they can be recognized by conventional charactrer recognition methods for extracted character images. Therefore, sum of "Correct" and "Match" shows the effectiveness of the proposed method.

Then, we consider the causes of "Miss". 27% of Maru Gothic were classified into "Miss". One of the reason is thresholds. When $T_2 = 0.75$ were employed instead of $T_2 = 0.85$, all categories were correctly extracted (in detail, 6 categories were "Correct" and the rest were "Match"). Therefore, this problem can be solved by determining the proper threshold for the image.

4.2 Against touching characters

An image including touching characters [6] was used as the input image. "S" of Arial was used as the reference image. $T_1 = 0.75$ and $T_2 = 0.8$ were used. The size of the reference image was changed to fit the largest "S" in the input image. The recognition result in Fig. 6 shows that "S" was correctly extracted ("Correct"). In addition, "e" was also extracted ("Match").

In this experiment, since the size of the reference image fit the largest "S" in the input image, smaller "S"s were not extracted. Therefore, an invariant method for the size of the input image are required.

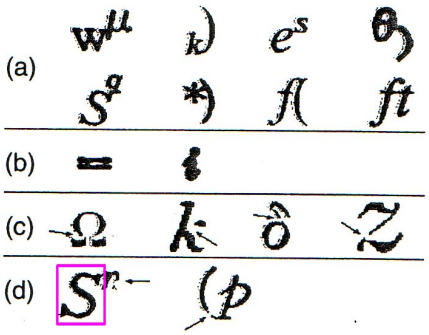


Figure 6. Recognition of touching characters: the reference image was "S".

4.3 Against isolated characters

To confirm the effectiveness of the proposed method against isolated characters in scene images, experiments were carried out. For all images, $T_1 = 0.75$ and $T_2 = 0.8$ were used. The size of an input image was reduced so that the size of the character in the input image is equal to that of the reference one.

The proposed method utilized the Sobel filter for extracting the edge direction in the input and the reference images. However, there is a problem that the extracted edge direction is different in the case of the black lettering on a white background and the white lettering on a black one, since the proposed method determines the edge direction that based on the black lettering on a white background. Therefore, we used reversed color to avoid this problem if the image includes white lettering on a black background.

The experimental results are shown in Figs. 7 to 19. Arial font wes used for the reference image. For each figure, the character used as the reference image is shown in the caption. In summary, the following two results were obtained.

(1) Over 90 percents of the characters in the input images were segmented by the proposed method. This result confirms the effectiveness of the proposed method for isolated characters in scene images. Though images of "Match" contain images of other categories, they can be recognized by conventional charactrer recognition methods for extracted character images. Therefore, hereafter we consider the causes of "Miss" such as "3" in the number plate of a car in Fig. 18(a) and "C" in the thermometer in Fig. 15(b). Two main causes look to be (i) threshold T_2, and (ii) essential difference between character shapes of an input image and a reference one. For the cause (i), change in threshold T_2 helps recovering from the failure of extraction. For example, the characters in Fig. 19 were not extracted when

$T_2 = 0.8$. However, when T_2 was changed into 0.7, both images were extracted completely. This indicates an automatic selection method of the threshold is required. For the cause (ii), it is difficult for the proposed method to extract figures. For example, the shape of "1" in the number plate in Fig. 18(b) is similar to "I" of Arial font rather than "1". When "I" was used as the reference image, "1" was extracted successfully. Here is another example. Fig. 21 includes "3" of the reference image and that in the number plate in Fig. 18(b). The two figures show that there is great difference in extracted feature points.

(2) The average computation time of the proposed method was 5.2 seconds, while that of the simple template matching method was 0.73 seconds. The machine used is 2.4GHz Pentium 4 with 1024MB memory. The proposed method takes more time than the template matching method because the neighbors within 10 pixels distance from the voting point are voted to allow deformations of a character image. The proposed method can detect a deformed character by this procedure, while the simple template matching method cannot.

5 Conclusion

In this paper, we proposed a novel character recognition method which executes segmentation and recognition simultaneously. The proposed method is based on histogram of edge directions. Experimental results showed the effectiveness of the proposed method against (1)different fonts, (2)touching characters, and (3)isolated characters. Developing automatic selection method of thresholds and invariant method for the size of the input image are future works.

Acknowledgment This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 16500096, and Young Scientists (B), 17700205, 2005.

References

[1] D. Ballard, Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, vol.13, no.2, pp111–122, 1981.

[2] D.H. Ballard, C.M.Brown, Computer vision. *Japan computer association*, 1987. Written in Japanese.

[3] T. Matsuyama, Y. Kuno, A. Imiya, Computer vision :technical criticism and future perspective, *Shingijutsu communications*, 1998. Written in Japanese.

[4] K. Mori. *Pattern Recognition*. IEICE, Tokyo, 1988. Written in Japanese.

[5] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[6] S. Uchida, A. Nomura, and M. Suzuki. Quantitative analysis of mathematical documents. *IEICE Technical Report*, 2004. Written in Japanese.

[7] K. Negishi, M. Iwamura, S. Omachi, and H. Aso. Isolated Character Recognition by Search of the Partial Regions. *Forum on Information Technology*, pp37–38, 2004. Written in Japanese.

Figure 7. A recycle mark: "1".

(a) "6".

(b) "P".

Figure 8. Signboards of a bus stop and parking.

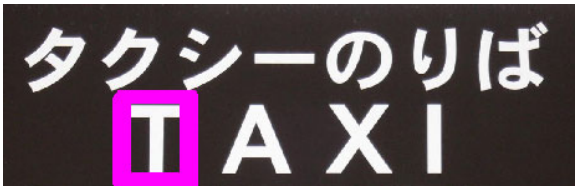


Figure 9. Signboards of a taxi stand: "T".

Figure 10. A tachometer of a car: “4”.

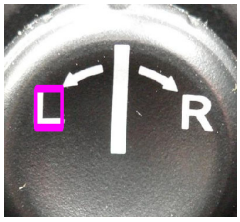
Figure 11. A speed meter of a car: “0”.

(a) “D”.

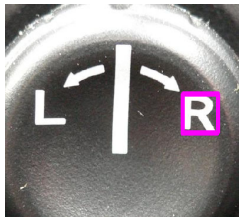
(b) “N”.

(c) “P”.

Figure 12. A shift indicator of a car.



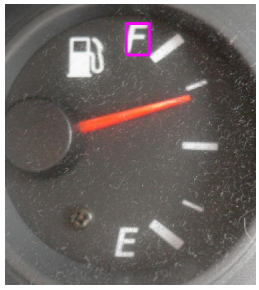
(a) “L”.



(b) “R”.

Figure 13. A mirror adjuster of a car.

(a) “E”.



(b) “F”.

Figure 14. A fuel gauge of a car.

(a) “H”.

(b) “C”.

Figure 15. A thermometer of a car.

Figure 16. A traffic sign: “4”.

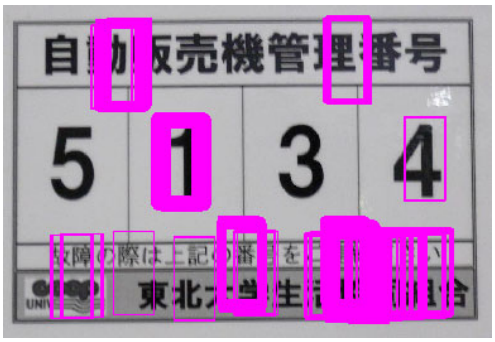


Figure 17. A management number of a vending machine: “1”.

(a) “3”.

(b) “I ”. Thereshold T_2 = 0.95.

Figure 18. A number plate of a car.

(a) “2”.

(b) “5”.

Figure 19. Signboards of an address and a number. Thereshold T_2 = 0.7.

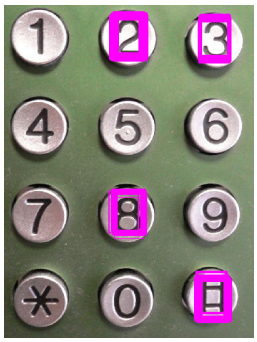
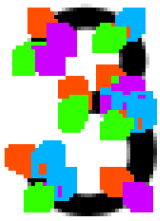
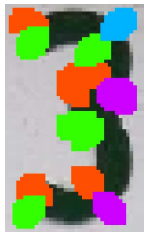


Figure 20. Push buttons of a telephone: “2”.



(a) “3” of the Arial font.



(b) “3” in the number plate in Fig. 18.

Figure 21. Extracted feature points.