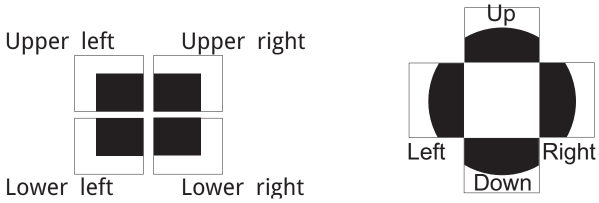
Isolated Character Recognition by Searching Feature Points

Abstract

Conventional segmentation technique cannot extract a isolated character and touching characters. In this paper, we propose a novel character recognition method which executes segmentation and recognition simultaneously. This method enable to extract such difficult characters to extract. The effectiveness of the proposed method is confirmed by experiments.



(a) Corners.

(b) Curves.

Figure 1. Features.

1. Introduction

Accuracy of character recognition for segmented character image is enough for practical use. However, conventional segmentation method can be applied only *well-defined* problems requiring that the character images are arranged on a line, completely separated from other characters, and so on. Therefore, *isolated character* which is apart from lines of characters, and touching characters, which is parts of more than two characters are connected, are hard to extract. In this paper, we propose a novel character recognition method which executes segmentation and recognition simultaneously. The proposed method allow segmentation of a isolated character, touching characters and so on.

2. Preparation

2.1. Input image and reference image

"Input image" is a relatively larger image including target characters. "Reference image" is a relatively smaller image among 62 characters of alphabets and numerals.

2.2. Histogram of edge direction

At first, the edge direction and the edge intensity of each pixel is calculated from horizontal and vertical intensity by the Sobel filter. Then, the edge direction is quantized into n bins. Each bin has $\theta = \frac{2\pi}{n}$ radian. To eliminate the effect of noise, a pixel whose intensity is less than a threshold is regarded as a *no edge pixel*. The n bins do not include no edge pixels. The histogram are normalized so that the sum of the histogram values to be 1. This normalized histogram is a feature vector of n dimensions.

2.3. Feature point

Corner, curve, branch, intersection and bending point are known as effective features in character recognition [1]. In the proposed method, 4 corners and 4 curves are used as feature points 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 desired. Therefore, θ depends on feature types: $\theta = \pi/2$ and $n = 4$ for corners, and $\theta = \pi/8$ and $n = 16$ for curves.

To describe the detail, four intervals of bins of a corner are defined as

$$\left[-\frac{\pi}{4} + \frac{\pi}{2}n, \frac{\pi}{4} + \frac{\pi}{2}n\right), \quad 0 \le n \le 3. \tag{1}$$

Such a loose feature can extract both regular font such as Arial and oblique font auch as Franklin in a feature as shown in Fig. 2. Four intervals of bins of a curve are defined as

$$\left[-\frac{\pi}{16} + \frac{\pi}{8}n, \frac{\pi}{16} + \frac{\pi}{8}n\right), \quad 0 \le n \le 15. \tag{2}$$

(a) Arial. (b) Franklin.

Figure 2. Robust feature point extraction with the upper left feature.

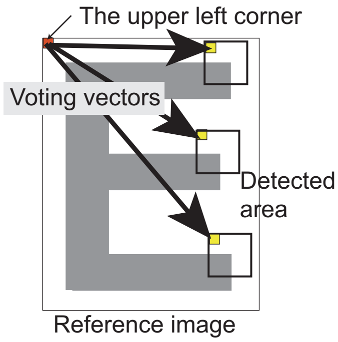


Figure 3. Feature area extraction from reference image.

2.4. Similarity

The proposed method search for feature points. A rectangle in the input image or the reference image, which is a target of the search, is called a search area. To evaluate how much a search area and a feature point are close, similarity proposed by Swain et al. [2] is used. Let H be the histogram of the input image or the reference image, M be the histogram of a feature point. Then, the similarity between H and M are given as

$$S_{HM} = \sum_{i=1}^{Q} \min(H_i, M_i), \tag{3}$$

where H_i and M_i are the elements of H and M respectively. Since both H and M are normalized histogram, the maximum value of S_{HM} is 1 and the minimum one is 0. If the similarity of a search area is more than threshold T_1 ($0 \leq T_1 \leq 1$), it is feature point.

3. Segmentation and recognition of isolated characters

3.1. Feature point extraction from reference image

First of all, all the feature points are detected from reference image, and the positions of each feature are obtained. The information of the position is represented by *voting vector* (see Fig. 3).

Define a coordinate system: the origin is the upper left corner of the reference image, and x-axis is leftward, y axis is downward. Let $M_i^{(k)}$ be the number of features area detected by i-th feature of character k for $i = 1 \sim 8$. Let (x_{ij}, y_{ij}) be the coodinate of j-th feature point detected by i-th feature of character k. The voting vector is defend as

$$\mathbf{f}_{ij}^{(k)} = (x_{ij}, y_{ij}), \quad 1 \leq i \leq 8,\ 1 \leq j \leq M_i^{(k)}. \tag{4}$$

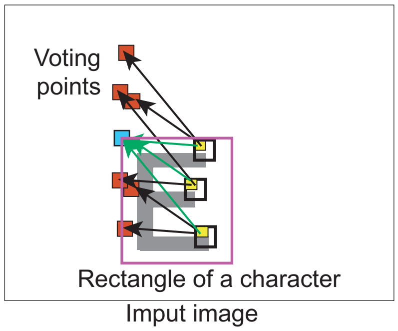


Figure 4. Feature area extraction from input image, voting and segmentation.

Furthermore, the total number of features points extracted from the reference image is given by

$$F_S^{(k)} = \sum_{i=1}^{8} M_i^{(k)}. \tag{5}$$

3.2. Segmentation

Fot segmentation, all the feature points are detected from input image, and candidates of character type and position are voted at the *voting point* given as

$$\mathbf{P}_{ij}^{(k)} = \mathbf{x}_i - \mathbf{f}_{ij}^{(k)}, \quad 1 \leq j \leq M_i^{(k)}, \text{ for all the } k, \tag{6}$$

where $\mathbf{x}_i$ is a feature point detected. In short, the voting point is a candidate of the upper left corner of a character image.

Next, the voting rule near the voting point is described. Let $v_{ij}^{(k)}(x, y)$ be *voting table*, which stands for the voting value at (x, y) in the input image. In the beginning, it is initialised as $v_{ij}^{(k)}(x, y) = 0$. The voting table near $\mathbf{P}_{ij}^{(k)}$ is modified as

$$v_{ij}^{(k)}(x, y) = 1, \quad \text{for } (x, y) \text{ satisfying } \left|(x, y) - \mathbf{P}_{ij}^{(k)}\right| \leq r. \tag{7}$$

Note that if the same point was voted more than once, the voting value is 1.

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

$$V^{(k)}(x, y) = \sum_i \sum_j v_{ij}^{(k)}(x, y). \tag{8}$$

$V^{(k)}(x, y)$ stands for the possibility that (x, y) is the upper left corner of the image of character k. Let T_2 be the threshold of character segmentation where $0 \leq T_2 \leq 1$. Since $F_S^{(k)} = \max V^{(k)}(x, y)$, if

$$V^{(k)}(x, y) \geq F_S^{(k)} T_2 \tag{9}$$

is satisfied, (x, y) is determined to be the image position.

4. Experiments

4.1. Feature point extraction

Feature points extracted by the proposed method are illustrated in Fig. 5. The figures are eight feature points of “0”, “E”, “M” and “S” of Arial. In the figures, each rectangle represents a feature point. Because feature points detected were so crowded, rectangles look filled. 5×5 was used as the size of the search area (same as the size of rectangle of the figures). $T_1 = 0.75$.

In the figures, other features than pure corners and curves were extracted. However, the most important thing to recognize correctly is that the same points is extracted from the reference image and the input image. Therefore, they were used without eliminating.

4.2. Recognition experiments

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

The proposed method ignores no edge pixels, and does not use the information of them for segmentation and recognition. However, an area which is not feature point can be detected due to the combination of many no edge pixels and a few edge pixels in the search area. Therefore, the search area which contains 70% of no edge pixels is ignored.

Figure 6. Correct: the reference image is “A”.

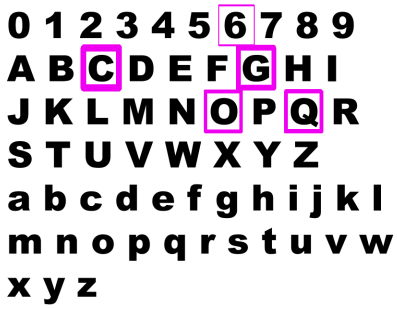


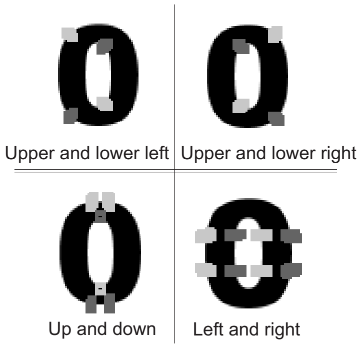
Figure 7. Match: the reference image is “C”.

Experiment results are classified into three categories for evaluation: “Correct”, “Match”, and “Miss”. If only the same character image(s) as the reference image is(are) extracted, it is classified into “Correct” (See Fig. 6). If character images extracted contains the same character image(s) as the reference image, it is classified into “Match” (See Fig. 7). If the same character image(s) as the reference image is(are) not extracted, it is classified into “Miss” (See Fig. 8).

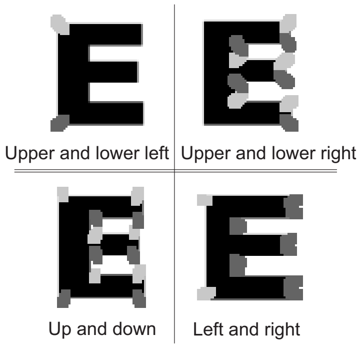
4.2.1 Against different fonts

The reference images of several fonts, similar to Figs. 6 to 8, were prepared. When the fonts of the reference image and the input image are the same, recognition accuracy was 100%. Therefore, experiment results against different fonts are shown here. Arial was used as the reference image and Arial Black, Franklin and Maru Gothic were used as the input images: Arial Black is bold, Franklin is oblique, Maru Gothic has rounded corners. Two thresholds are determined as $T_1 = 0.75$ and $T_2 = 0.8$.

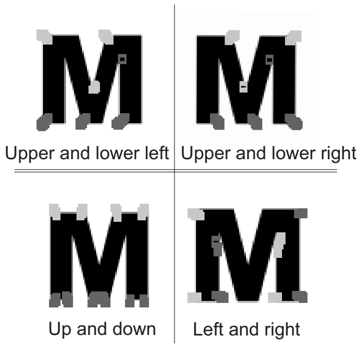
Recognition results are summarized in Table. 1. The ta-



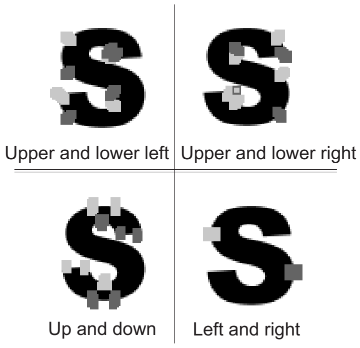
(a) “0”.



(b) “E”.



(c) “M”.



(d) “S”.

Figure 5. Feature points. Corners are in the upper row, and curves are in the lower row.

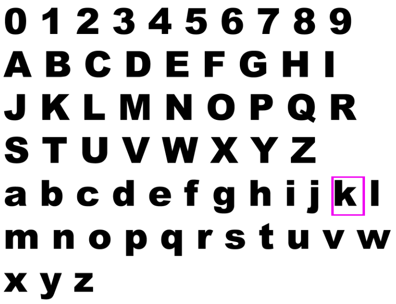


Figure 8. Miss: the reference image is “K”.

Font		Correct	Match	Miss
Arial Black	# of occurence	20	41	1
	Ratio	32%	53%	2%
Franklin	# of occurence	20	39	3
	Ratio	32%	63%	5%
Maru Gothic	# of occurence	24	20	16
	Ratio	39%	36%	26%

Table 1. Recognition result of different fonts.

ble shows that almost one third were classified into “Correct”. Also, more than 95% of Arial Black and Franklin images, and more than 75% of Maru Gothic were classified into either “Correct” or “Match”. Though images of “Match” include other images, they can be recognized by conventional methods for character images extracted. Therefore, these results shows the effectiveness of the proposed methods.

Then, we consider the causes of “Miss”. 26% of Maru Gothic were classified into “Miss”. One of the reason is thresholds. When $T_2 = 0.75$ were employed instead of $T_2 = 0.8$, about 85% of images can be extracted. Therefore, this problem can be solved by determining the proper thresholds for the images.

Also, “K” of Arial Black, “5”, “7”, “K” of Franklin were

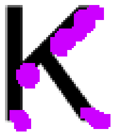


Figure 9. Cause of error: too many feature points of the lower right were extracted.

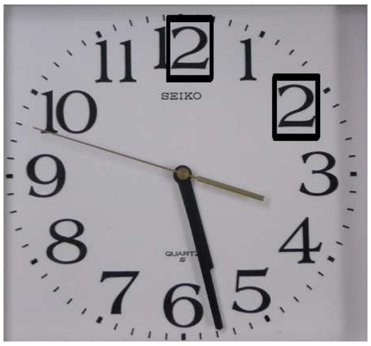


Figure 10. Recognition of numerals of clock.

	Correct	Match	Miss
# of occurence	9	1	0
Ratio	90%	10%	0%

Table 2. Recognition result of numerals of clock.

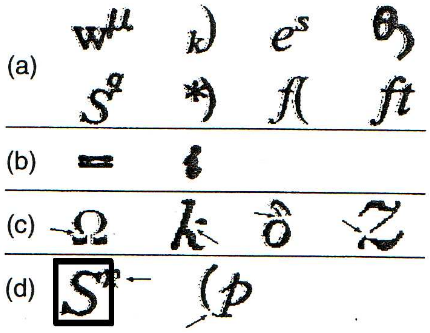


Figure 11. Recognition of touching characters: the reference image is "S".

classified into "Miss". An image of feature points extracted from "K" of Arial Black is shown in Fig. 9. In the figure, too many feature points were extracted. This is the main cause of "Miss".

4.2.2 Against isolated characters

To confirm the effectiveness of the proposed method against isolated characters, a photograph a clock including numerals were used. Numerals of Arial ware used as the reference images. $T_1 = 0.75$ and $T_2 = 0.85$ were used. Recognition result of "2" are shown in Fig. 10, and summarized in Table. 2. Only "0" was "Match" and the others were "Correct". In the picture, there are several "1"s and "2"s. Since they all were successfully recognized, they were classified into "Correct". Segmentation of characters in the photo is very hard for conventional methods. This is a novelty of the proposed method.

4.2.3 Against touching characters

An image including touching characters [3] 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 larger "S" in the input image. The recognition result in Fig. 11 shows that "S" was correctly extracted ("Correct"). In addition, "e" was also extracted ("Match").

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

5. Conclusion

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

References

[1] K. Mori. *Pattern Recognition*. IEICE, Tokyo, 1988.

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

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