AN ALGORITHM FOR LOCATING CHARACTERS IN COLOR IMAGE USING STROKE ANALYSIS NEURAL NETWORK

Anto Satriyo Nugroho, Susumu Kuroyanagi, Akira Iwata

Department of Electrical and Computer Engineering,
Nagoya Institute of Technology
Gokiso-cho Showa-ku, Nagoya, 466-8555 Japan
E-mail: anto@mars.elcom.nitech.ac.jp

ABSTRACT

Character segmentation is a significant part in a character recognition system. Particularly, when the system is assumed to work in a color image with multi-segment characters such as Japanese Kanji characters, the complexity of the characters and the background properties bring the difficulties to the segmentation problem. Discussion in this paper is focused on designing an automatic system for locating text regions, assuming that the texts are composed by Kanji characters. The principle of the proposed model is the inclusion of recognition phase to give a feedback in controlling the segmentation task, yielding a robust algorithm to solve the complexity of the characters. The algorithm is assumed to work with color images, which makes it suitable for practical applications. The evaluation of the model shows that the algorithm promises an appropriate approach to deal with the complexity of Kanji character segmentation.

1. INTRODUCTION

Character recognition is one of popular topics in pattern recognition problem, and it has been widely implemented in many applications. One of the most popular application is Optical Character Reader (OCR). In the first step of OCR, the scanning result of an image is presented to a text segmentation phase, then the objects in the image are separated into text regions and non-text ones. The text regions are presented to a character classifier which can be constructed of neural network based method as well as the other statistical ones. The input image of OCR is often assumed to be black-white. However, if the segmentation algorithm is improved to be able working with color image, many other advantages could be obtained. This problem has attracted many attentions on the development of segmentation algorithm that works with color image.

Progress in this topic has been reported in [1][2]. Jain has introduced a modified version of connected component analysis, and applied the algorithm for video indexing, web search and color image databases [1]. Wu et al. proposed a robust algorithm to extract text from general backgrounds [2]. Most of these efforts used alphabets as inputs. When alphabets are used, the text region has several distinctive features that make it easy to be distinguished from the non-text regions. The text region should be composed by a string of characters with spatial cohesion. Thus, this property can be used to extract the text region without inviting any recognition tasks.

However, these algorithms have limitations in practical application, particularly when they are applied to symbol-based character domain. The popular examples of the family of characters with this kind of property are the Japanese and Chinese Kanji characters. In this case, many Japanese words consist of only few characters, and often one or two characters can form a word. The real examples can be seen from the name of subway station, traffic signs in Japan or China. Therefore, the assumptions used in the alphabet based algorithms are not appropriate. The other property of the Japanese and Chinese characters is that they are constructed of many segments, in contrast to the alphabet characters those are composed by single segment. These features make the segmentation problem of Kanji characters has been considered to become one of the most difficult tasks in developing a character recognition system.

In this study, we present a new algorithm to extract text regions from color images that can be used for multi-segment characters, such as Japanese and Chinese Kanji characters. The principle of the model is that the segmentation problem should be supported by a feedback obtained from recognition-based stroke analysis. The proposed model is a combination of competitive learning neural network and multilayer perceptron (MLP) trained by backpropagation algorithm. The first part works as stroke selection and the second one analysis the probability of the strokes combination in forming a character.

The paper is organized as follows : Section 2 provides the explanation of the system, Section3 presents the experimental results and discussion. The conclusion of the study is provided in Section 4.

2. SYSTEM OVERVIEW

Figure 1 depicts the overview of the system. The system consists of: a camera to capture environmental image, color reduction, multivalued image decomposition, following by connected component extraction, adopted the scheme proposed by Jain [1]. In this study, we modified the algorithm by following the process with: segment selection, encoding and supervised single link clustering, that make the system applicable for multi-segment characters.

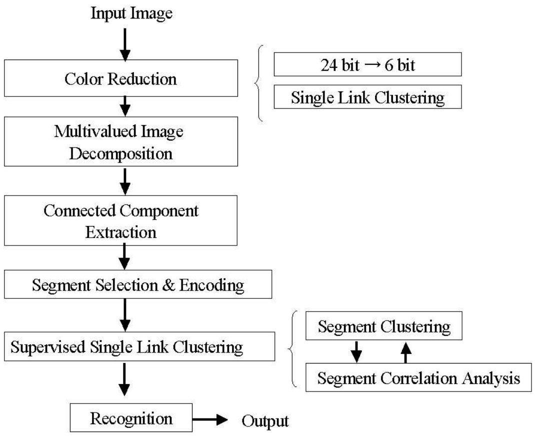


Fig.1 Overview of the system

2.1. From Color Reduction to Connected Component Extraction

The color reduction is performed by implementing (i)bit-dropping for RGB color bands and (ii) color quantization. A 24-bit color image consists of three 8-bit red, green and blue images. In this work, we used only the highest 2 bits for each band image to compress 24-bits formatted image into 6-bits expression. In a bit dropped image, text may be presented in several colors which hopefully are close in the color space. Therefore in the next steps we perform color quantization to generate a small number of meaningful color prototypes. We employ the single-link clustering method for quantizing the color space [3].

In multivalued image decomposition stage, the image is decomposed into several images based on the foreground color. Let us denote the pixel value in the image as $u \subset \Psi$, where $\Psi = \{0,1,\cdots,U-1\}$.The image is then decomposed into U images, each of the image is constructed of pixel with the same value: $u \subset \Psi$,where

$$\bigcup_{i=0}^{U-1} I_i = I, \quad I_i \bigcap_{i \neq j} I_j = \phi \tag{1}$$

Figure 2 shows the example of image decomposition, with U=6.

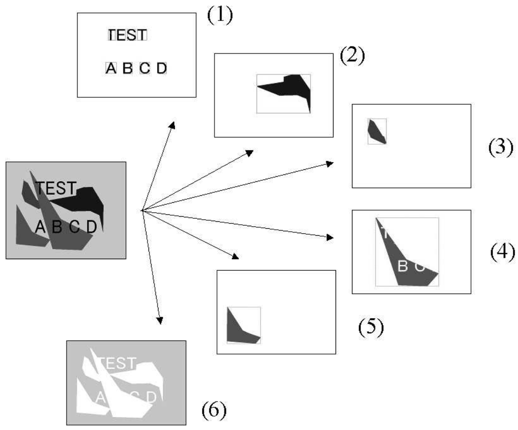


Fig.2 Image decomposition.

After decomposition of a multivalued image, we obtain a lookup-table of foreground identifications for pixel values according to foreground images. This information will be used for finding connected components in gray level. This step used Block Adjacency Graph (BAG) that has been used for efficient computation of connected components since it can be created by a one-pass procedure. The one-pass BAG algorithm for multivalued image is explained in [4].

2.2. Segment Selection and Encoding

As the results of the previous stages, we obtained several connected components from each image. The next stage is selecting part of the components as candidates of the characters' segments. The selections are carried out by measuring the similarity between the components and a set of templates. The templates were prepared by applying clustering algorithm to a database of characters' segments. Let us denote the component as x, the templates as μ_i ($i = 1,2,\cdots,M$), and the number of templates as M . The components are assigned with the label of the most similar template. If the similarity exceeds a certain threshold s_{th} , then x is consider as the segment of character, and it is assigned with the label of the most similar template, denoted as $L(x)$.

$$\max_{p=1,\cdots,M} \{sim(x,\mu_p)\} = sim(x,\mu_k) \tag{2}$$

$$If \quad sim(x,\mu_k) > s_{th} \quad \rightarrow \quad L(x) = k \tag{3}$$

$$sim(x,\mu_p) = \frac{x.\mu_p}{|x||\mu_p|} \tag{4}$$

Otherwise, x is considered as non-text part and is removed from the candidates. In this study, we used self-growing algorithm [5] to obtain the templates. This algorithm is stated as follows:

1. One training vector is randomly chosen as the reference vector for the first neuron.
2. **Data Presentation**
   Each datum is then presented to all of the neurons and the similarity with their reference vectors are calculated. A neuron with the highest similarity is determined as the winner, then the datum is recorded as a member of a cluster corresponding to this neuron.
3. **Neuron Creation**
   If the similarity level of the input datum and the reference vector of a neuron is less than a similarity threshold (r_{th}), then a new neuron is created with reference vector as same as the input datum vector.
4. **Neuron Division**
   The inner potential of a neuron is defined by the number of members belonging to the cluster corresponding to the neuron. If the inner potential of a neuron exceeds the inner potential threshold (h_{th}), divide the cluster into two new clusters. The two clusters are separated by a hyperplane that divides the old cluster into two parts with similar inner potential.

After all of the patterns are presented, the first process of this algorithm is completed. The process is then repeated until there is no significant change of the neuron's reference vectors.

2.3 Stroke analysis using modified single-link clustering and neural network

As result of the selection and encoding stage, a set of character stroke candidates are obtained from each foreground image. Then, the strokes are analyzed whether they are parts of character by a simple 3-layered Multilayer perceptron (MLP), symbolized by F . The output of this network is denoted by $F(y)$, where y is a vector constructed by the label of a subset of segments. The algorithm to obtain y is explained in Fig.3. This model works as a decoder-network that convert the vector segment-label level expression onto the character-level expression. The proposed algorithm is a modification of single-link clustering algorithm by including the feedback of stroke analysis obtained from MLP, yielding a supervised fashion. The principle used in the algorithm is that the distance of strokes in a character is smaller than the distance of two adjacent strokes from different characters. The algorithm is stated as follows:

[Step-1] Find two components x_1, x_2 with minimum distance d

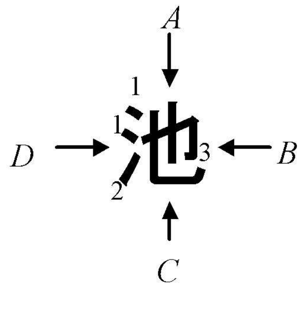
[Step-2] Connect all components where the distance is within d with a line, as a new cluster y

[Step-3] Analyze y using neural network model F . If $F(y) > q_{th}$ then register y as a character candidate and compare this new score $F(y)$ to those of the subsets in the previous stage. If the new score is less then the scores that have been assigned to each component in the previous stage, then destroy the new link and the components are not merged. The purpose of this step is to find the combination of strokes that have the highest probability to form a character, by the control of threshold q_{th} .

[Step-4] If $F(y) \le q_{th}$ then return to Step-1

Repeat the procedure until there is no generation of new cluster.

One important issue in this algorithm is the method to create a feature vector y from a character. In this study we used the peripheral information of the character, hence the characters were scanned from 4 directions (A:top-down, B:right-left, C:bottom-up and D:left-right), and the labels of each segment were recorded, sequentially. This method brings an advantage that the vector represents the relational-position of each segment in the character. Figure 3 illustrates the feature extraction algorithm used in this study. The character scanning from A-direction produces the order of the segment-labels from top to down: 3,1,1,2. In case the scanning depth is set at 5, thus the A-direction representation is given by "31120". The same procedures are applied to the B,C and D directions, yielding a vector representation as shown in Fig.3.



$$[\underbrace{31120}_{A}\ \underbrace{31210}_{B}\ \underbrace{23110}_{C}\ \underbrace{21130}_{D}]^T$$

Fig.3 Feature extraction of the characters

3. EXPERIMENTAL RESULTS

In this section, we present the experiment to evaluate the dynamic of the system. The database was created by a set of 43 characters used as the name of subway stations "Higashiyama Line", located in Nagoya city. The characters were typed, printed and the images were captured using digital camera. Each character was normalized to the size of 24x24. From this data, we obtained a database of character strokes. Templates μ_i are prepared by performing self-growing algorithm to this stroke-dataset, thus a set of 30 templates were generated (similarity threshold r_{th} =0.55). The stroke-analyzer F was prepared by training a 3 layered multilayer perceptron using the stroke-database, whereas the vector representation of each pattern was obtained by performing feature extraction as illustrated in Figure 3. The input vector had a dimensionality of 20 (depth:5, 4 directions). The number of neurons of input, hidden and output layer of MLP were 20, 50 and 43, respectively. The network was trained until the MSE error descended below 10^{-4}or the maximum iteration 20,000 epochs.

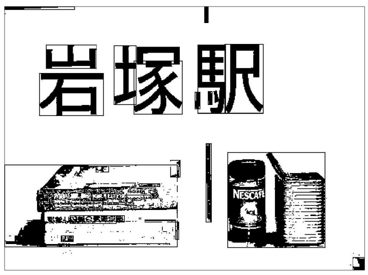
The experimental results is shown in Fig. 4 (a)-(h). The original figure is shown in Fig.4(a), showing three characters: "IWA-TSUKA-EKI", with the meaning: Iwatsuka Station. The algorithm worked to extract the three characters and ignore the non character regions (books, pen box and bin). Firstly, the color of the image was reduced, then was decomposed into several foreground images. One of the foreground image is shown in Figure 4(b).

It was followed by connected component extraction of which the result is shown in the same figure. Each of the connected component is shown inside a box. Figure 4(c) shows the result of segment selection and labeling phase. Note that only the boxes are shown without the picture of the objects. The segments were selected by calculating their similarity with the templates μ_i . Segments with low similarity were considered as the non-text region, thus they were discarded and not displayed. The figures inside the boxes in Fig.4(c) show the label assigned to each component. The next step was the merging of the components with the minimal distance. As the result, the second ("TSUKA") and third ("EKI") characters were obtained with F scores: 0.9 and 0.6, respectively, as shown in Fig.4(d). In the same figure, we also show the score for the other components as references for the next steps.

The next step was finding the components with new minimum distance, then as a result the components were merged as shown in Fig.4(e), resulting the segmentation of the first character "IWA". The new F score of the merged part was 0.9, which is higher than the individual scores of each segment (0.4 and 0.2). In the next step, the new minimum distance was obtained and the components were merged as shown in Fig.4(f) and Fig.4(g), respectively. However, the score of the new part was lower than the previous score of each component, thus they were not merged to form a character. The final result is shown in Fig.4.(h), that the three characters were successfully extracted. The figures next to the boxes show the highest scores obtained during the process.

The experiment shows the dynamic of the proposed method to extract text regions and ignoring the others. Since the model utilizes the inter-stroke distance characteristic and a feedback of stroke-analysis neural network, it has an advantage that there is not any necessity to assume the text to be composed by a certain number of characters. This property becomes important, when the model is implemented for practical application, since many words in Japanese language consist only a very few number of characters. The future work of this study is addressed to the evaluation of the model using scene image.

(a)



(b)

Fig.4 The progress of the algorithm in finding the characters (a,b)

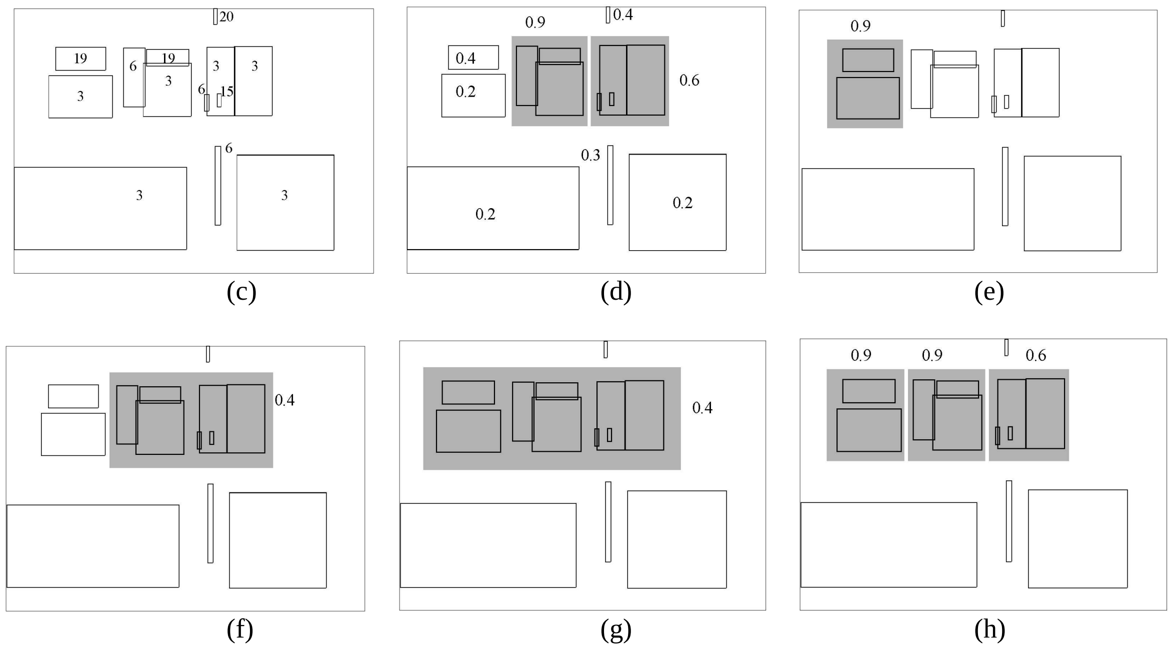


Fig.4 The progress of the algorithm in finding the characters (c-h)

4. CONCLUSION

In this study, we discuss the problem of character segmentation from a color image. The segmentation of characters, particularly those are composed by several segments such as Japanese Kanji is known as one of the most difficult problem in developing a character recognition system. To overdue this problem, we propose an algorithm with the inclusion of neural network to analyze the correlation of strokes. The principle of the model is that the segmentation of a multisegment characters is supported by the feedback of stroke analysis. One advantage of the model to the others is that there is not any necessity to assume that the text region should be constructed by several characters, thus yielding an appropriate and robust approach to the characteristic of Japanese Kanji characters. The evaluation of the model shows that the algorithm can extract the multisegment characters in an environment that is composed by text and non text regions.

The future work of this study is addressed to the evaluation of the algorithm using scene picture, thus opening the possibility of its implementation in many practical applications.

5. ACKNOWLEDGEMENT

Part of this study was supported by the grant from the Hori Information Science Promotion Foundation.

6. REFERENCES

[1] A.K.Jain, B.Yu , “Automatic Text Location in Images and Video Frames”, *Pattern Recognition*, vol.31, no.12, pp.2055-2076, 1998

[2] V.Wu, R.Manmatha, E.M.Riseman , “TextFinder : An Automatic System to Detect and Recognize Text In Images”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.21, no.11, pp.1224-1229, Nov.1999

[3]A.K.Jain and R.C.Dubes , “*Algorithms for clustering data*”, Englewood Cliffs: NJ: Prentice Hall, 1988

[4] B.Yu and A.K.Jain , “A generic system for form dropout”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.18, pp.1127-1134, 1996

[5] K. Hotta, A. Iwata, H. Matsuo, and N. Suzumura, ``Large Scale Neural Network “CombNET-II””, *IEICE Trans. Inf. & Syst*., vol.J75-D-II, no.3, pp.545-553, 1992.

[6] Y.Liu, T.Yamamura, N.Ohnishi, N.Sugie, “Extraction of Character String Regions from a Scene Image”, *IEICE Trans. Inf. & Syst*, vol.J81-D-II, No.4, pp.641-650, April 1998