Online recognition of free-format Japanese handwritings

Hiroshi MURASE

Naito Group, NTT Basic Research Labs,
3-9-11 Midori-cho, Musashino-shi, Tokyo 180, JAPAN

Aabstract

This paper describes an online recognition method for free-format written Japanese character strings, which may contain characters with separated constituents or overlapping characters. This method (Candidate Lattice Method) conducts segmentation and recognition of individual character-candidates, and applies linguistic information to determine the most probable character string in order to achieve high recognition rates. Special hardware designed to realize a real-time recognition system is also introduced. The new method used on the special hardware attained a segmentation rate of 98.8% and an overall recognition rate of 98.7% for 105 samples.

1. Introduction

Online recognition of handwritten Japanese character strings is widely applicable to such areas as wordprocessing and CAD input. However, methods proposed so far [1][2] impose such restrictions on writers as writing each character in a box, or indicating the end of each character. These restrictions lead to limited application of those recognition systems.

Several methods have been reported for automatic segmentation of Roman alphabets writing. Some of these methods utilize time and spatial information such as gaps between characters or words [3], or character shape information provided by the character recognition process [4]. However, Japanese characters are essentially different from alphabets. For example, Japanese characters in general contain more strokes than the letters of alphabets, and quite often consist of more than two separate groups of strokes. Therefore, the alphabet methods are not directly applicable to Japanese character string recognition.

This paper proposes a method of automatically segmenting and recognizing Japanese character strings in order to deal with online free-format character strings which are written on blank paper without any indication of segmentation. The method uses character recognition and linguistic information to determine the most probable character string in order to achieve high recognition rates. In this method, a character string is described as a directional graph (called a candidate lattice) representing possible segmentation and character plausibilities. Then, by means of DP (dynamic programming) the candidate character series corresponding to the shortest path under linguistic constraints is selected as the segmentation and recognition result.

The method requires a large amount of processing. Therefore, I have developed dedicated hardware composed of 16 processors linked in parallel. A simulation system incorporating the hardware has proved to be capable of recognizing Japanese character strings in real time.

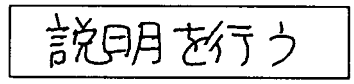


Fig.1 Example of free-format Japanese writing

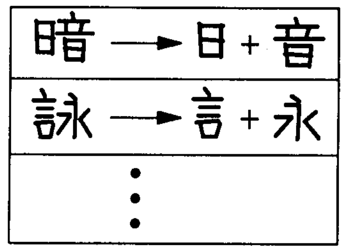


Fig.2 Characters composed of other characters

2. Japanese handwriting

An example of a handwritten Japanese character string is shown in Fig.1. The following are the characteristics of Japanese handwriting and their related recognition problems.

(1) Handwritten characters may contact each other, and their sizes vary considerably. Therefore, spatial information such as character pitch and gaps between characters does not help much in determining character segmentation.

(2) About 30% of Japanese characters consist of more than 2 parts (e.g., left-hand radical and right-hand

body). This characteristic also makes segmentation difficult.

(3) Any of those parts can be an individual character in itself (Fig.2). The characters having such separate parts account for 13% of the total. Such characters might be mistaken for a sequence of two or more characters, if only knowledge of character shapes is used in the recognition.

(4) The number of strokes forming one character ranges from 1 to 23 for the characters for daily use, so the maximum numbers of strokes was 23 in my experiments.

(5) There are about 2000 different Japanese characters.

Characteristics (4) and (5) lead to a vast amount of matching calculation, while real-time response is necessary in an online recognition system. This problem can be solved by a special hardware.

3. Recognition algorithm

This chapter introduces the candidate lattice method for online recognition of Japanese character strings. The method uses character recognition and linguistic information in order to solve the first three problems mentioned in the previous section, and has a hierarchical structure shown in Fig.3. There are six layers in the structure: (1) handwriting data input, (2) separation into basic segments, (3) generation of candidate characters, (4) matching of the candidate characters, (5) candidate lattice formation and shortest path search, and (6) linguistic processing.

I limit recognition objects to Japanese characters. There are 1945 Kanji characters daily use and 46 Hiragana: a total of 1991. The writing order of strokes in a character is free.

Japanese character strings can be written either vertically or horizontally. The method proposed here determines the direction of writing from the average direction of pen movement top-to-bottom or left-to-right in the preprocessing. The method can deal with both vertical and horizontal writing, although only horizontal writing is treated in this paper.

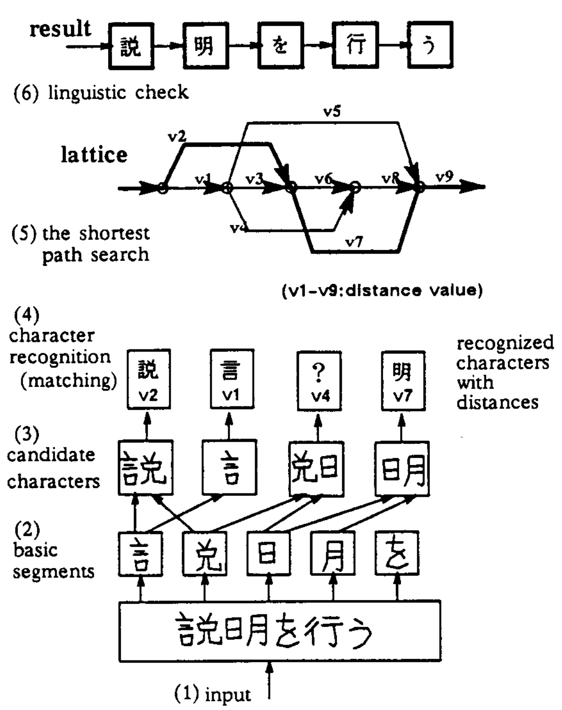


Fig.3 Block diagram of candidate lattice method

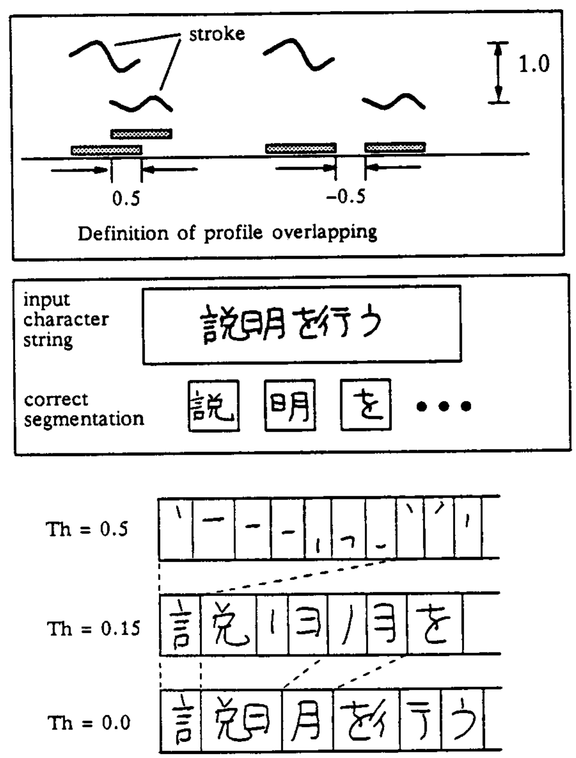


Fig.4 Basic segments vs. parameter Th

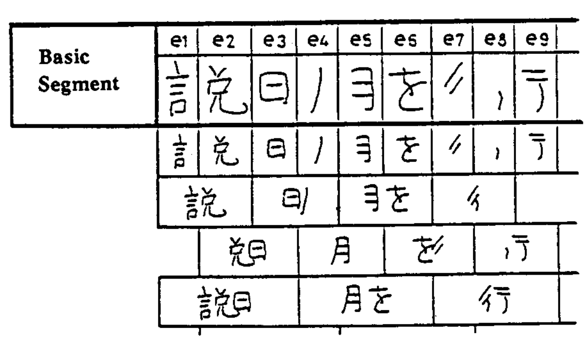


Fig.5 Generated candidate characters

3.1 Input of strokes

An electro-magnetic data tablet is used as an input device. A "stroke" is represented by the x,y-coordinate values of the pen movement between the pen down and up.

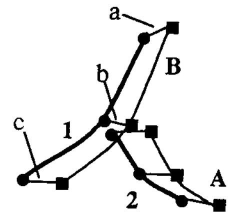
3.2 Separation into basic segments

The first step is to divide the input stroke series into basic segments. A basic segment should correspond to a character or a character component. The strokes do not usually continue from one character to another in Japanese writing. Therefore, the basic segments are generated by the following processing.

Using the projection of all strokes on the X-axis, parts that have more than a certain constant of profile overlapping (Th) are extracted as segments (See Fig.4). Examples of segmentation for different Th values are shown in Fig.4. Th is a parameter denoting the ratio of profile overlapping to the average height of the string. The larger Th is, the more strokes a segment can contain. This leads to possible segmentation error, since a character segment might extend to other characters as shown in Fig.4 (Th=0.0). On the other hand, a small Th leads to over-segmentation and increase in the amount of calculation in the next step. In my experiments, the optimum Th was 0.15.

3.3 Generation of candidate characters

A character might be composed of several basic segments. So candidate characters are generated by combining one or more basic segments (Refer to Fig.5) according to the following conditions. (1) The number of strokes to form a candidate should be less than 24 (See Section 2). (2) The ratio of candidate character to the average height of the character string should be less than 2.0. These two conditions are sufficient to compose probable candidate characters.



$$d_{1B} = a + b + c$$

$$D = \sum_i \left(\min_j d_{ij} \right) = d_{1B} + d_{2A}$$

Fig.6 Stroke-order-independent matching

3.4 Matching of candidate characters

The distances between each candidate character and all of the standard characters in the dictionary are calculated in the following steps.

(1) Approximate each candidate character stroke with equidistant sampling points on the stroke.

(2) Calculate the distances between the i-th stroke of the candidate character and the j-th stroke of a standard character (dij) for all possible combinations of i and j, by DP matching based on inter-point Euclidean distances.

(3) Calculate the inter-character distance by

$$\sum_i \{ \min_j (d_{ij}) \}$$

(Refer to Fig.6.) This step follows the stroke-order-independent matching proposed by Odaka[1].

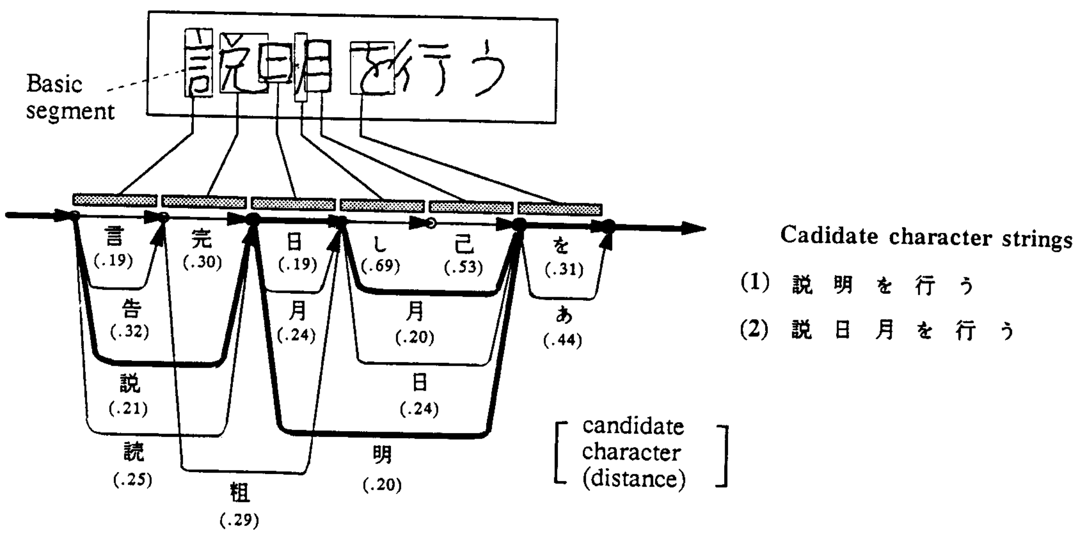


Fig.7 Example of candidate lattice

(4) Choose the first N standard characters that have the shortest inter-character distances. Then, the names and distances of these characters are stored for use in the next stage.

Here, the value of N is set to 3, since 99% of the correct choices were included in the top 3 candidates according to my preliminary investigation.

3.5 Candidate lattice forming and path search

The result of segmentation and candidate character recognition is described in terms of a two-terminal weighted directional graph (candidate lattice) where a node is defined by a boundary between basic segments, a branch by a candidate character, and weight by calculated distance (Fig.7).

Next, the shortest path is searched for over the candidate lattice by DP. The shortest path is the choice of branches, which minimizes the following objective function, S.

$$S = \Sigma\ (\text{distance}) \times (\text{no. of segments}).$$

The sequence of candidate characters that corresponds to the determined path becomes the tentative recognition result.

3.6 Linguistic processing

The sequence of characters in the tentative result is checked grammatically using morphological analysis and a dictionary containing about 8000 words and a basic Japanese grammar composed of about 130 rules (e.g. "adjective + noun can form another noun")[6]. When any contradiction to those rules occurs, the tentative character sequence is rejected and the next shortest path in the lattice is selected. The above process is performed iteratively until a grammatically correct sequence is selected to give the final recognition result. Fig.7 shows the case where two similar candidates are extracted :

[説 日 月], [説 明].

Both candidates are valid from the standpoint of character recognition. However the former one is grammatically incorrect. By this method, the latter one, which is the correct answer, is selected.

4. Specially Designed Hardware

In the candidate lattice method, the recognition process requires a large amount of distance calculation, not only because there are a great number of standard characters (1991), but also because the system has to perform a lot of character recognitions. For the 5-character string shown in Fig.5, 51 candidate characters are generated. This means that 51×1991 (=101541) inter-character distance calculations are necessary.

In order to perform such a vast amount of calculation in real time, I have developed special hardware that has 16 conventional microprocessors (2903) with multipliers linked in parallel. It performs inter-character distance calculations 16 times faster, by dividing the dictionary with 1991 standard characters into 16 parts and loading each part on a different processor. The processors work in parallel.

The candidate lattice method is easily implemented by parallel architecture. A 1-MIPS mini-computer takes 125 seconds to recognize the 5-character string in Fig.7, whereas this special hardware takes only 6 seconds. Therefore, the recognition of Japanese character strings has become fast enough to keep up with the average Japanese writing speed of 3 seconds per character, thus making a real-time system.

5. Experimental Results

Recognition tests were made on 105 samples written by 7 persons. The sentences were excerpts from technical reports that satisfy the following two conditions: (1) The number of characters in a sentence is about 10, (2) Only words stored in the word dictionary are used (See 3.6). The results are shown in Table.1. The method achieved a correct segmentation rate of 98.8% and an overall recognition rate of 98.7%. It is noteworthy that the low segmentation rate of 68.5% was obtained when using only gap information.

Misrecognitions are mainly due to seriously deformed patterns and weak grammatical rules.

Table.1 Experimental results

	correct segmentation	character recognition	overall recognition
Gap information only	68.5%		
Without grammatical rules	96.7%	99.2%	95.9%
With grammatical rules	98.8%	99.9%	98.7%

6. Discussion

(1) Applicability The candidate lattice method can be successfully applied to such a variety of writing as overlapping characters, and characters of different pitch or size, (Fig.8), because the method does not use pitch or size information.

(2) Linguistic check iteration Linguistic checks improve the accuracy of segmentation and recognition. Fig.9 shows the distribution of the number of linguistic check iterations necessary for correct segmentation and recognition. It can be seen that 5 iterations are sufficient.

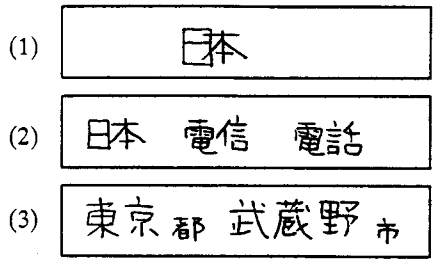


Fig.8 Correctly recognized character strings

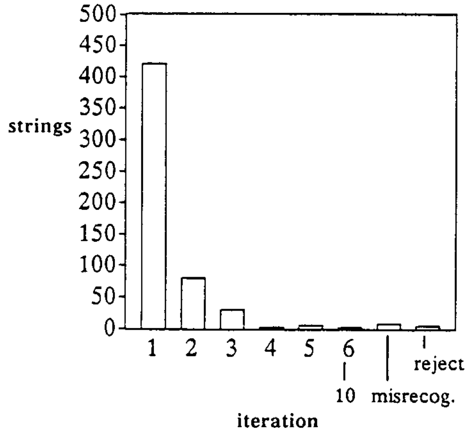


Fig.9 Distribution of iterations

(3) **Efficiency of using basic segments** Use of basic segments instead of individual strokes lessens the total number of candidate character generations. In Fig.5, only 51 candidates are generated using basic segments, whereas 419 candidates are generated using individual strokes. This means that the efficiency of pattern matching and path search is greatly improved.

7. Conclusion

This paper has introduced the candidate lattice method which segments and recognizes free-format handwritten Japanese character strings, and has described dedicated hardware for this method.

The candidate lattice method has greatly improved segmentation accuracy and has achieved the high overall recognition rate of 98.7%. In the method, all possible segmentations and recognitions of a character string are described in terms of a two-terminal graph, and the graph is searched for the optimum path under linguistic constraints. Thus, the processes of segmentation, recognition and linguistic check can be conducted together.

The method also introduced the idea of basic segments to perform efficient stroke clustering. This leads to a considerable reduction in calculation time.

Japanese character string recognition by the candidate lattice method requires a vast amount of calculation. Dedicated hardware with 16 processors linked in parallel solved this problem. The experiment using the hardware has shown that it takes only 6 seconds to recognize a 5-character string. The recognition has become fast enough to keep up with the average Japanese writing speed.

The candidate lattice method has a great capability of handling a variety of free-format writing. However, problems still remain for future work.

(1) Refinement of linguistic processing.

(2) Further utilization of positional information about characters to improve performance and accuracy.

Further research is also necessary to make the candidate lattice method applicable to character strings written with less care, such as in run-on style.

Acknowledgment

I would like to thank Dr. I. Masuda, Dr. S. Naito, Dr. T. Wakahara, Dr. N. Hagita and Mr. M. Shinya for valuable advice and encouragement, and Dr. N. Osumi for his advice in preparing the paper. I are also grateful to Mrs. Y. Murase for her linguistic collaboration.

References

[1] K.Odaka, T.Wakahara, and I.Masuda, Stroke-order-independent online character recognition algorithm and its application. Review Electrical Communications Laboratories, 34, pp.79–85, 1986

[2] T.Wakahara and M.Umeda, Online cursive script recognition using stroke linkage rules, 7th ICPR, pp.1065–1068, 1984.

[3] G.F.Groner, Real-time recognition of handprinted text. Proc. FJCC, pp.591–601, 1966.

[4] C.C.Tappert, Cursive script recognition by elastic matching. IBM J. Research and Development, 26, pp.765–771, 1982.

[5] H.Murase and T.Wakahara, Online hand-sketched figure recognition, Pattern Recognition, 19, pp.147–160, 1986.

[6] M.Shinya and M.Umeda, Evaluation of compound post- processing method in character recognition, IECE Japan (in Japanese), J68-D,5, pp.1118–1124,1986.