ON-LINE RECOGNITION SYSTEM FOR FREE-FORMAT HANDWRITTEN JAPANESE CHARACTERS

HIROSHI MURASE
NTT Basic Research Labs
3-9-11 Midori-cho, Musashino-shi
Tokyo 180 Japan

This paper describes an on-line recognition system for free-format handwritten Japanese character strings which may contain characters with separated constituents or overlapping characters. The recognition method for the system, called 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 method used on the special hardware attained a segmentation rate of 98.8% and an overall recognition rate of 98.7% for 105 samples.

Keywords: Character recognition; Character segmentation; Graph search; Parallel processors.

1. INTRODUCTION

On-line recognition of handwritten Japanese character strings is widely applicable to areas such as word processing and CAD input. However, the methods proposed so far[1,2] impose restrictions on writers such as writing each character in a box (Fig. 1(a)), 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 handwritten Roman alphabets. 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 for automatically segmenting and recognizing Japanese character strings in order to deal with on-line free-format character strings which are written on blank paper without any indication of segmentation (Fig. 1(b)). The method uses character recognition and linguistic information to determine the most probable character string so that high recognition rates can be achieved. 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 dynamic programming (DP) the candidate character series corresponding to the

International Journal of Pattern Recognition and Artificial Intelligence Vol. 5 No. 1 & 2 (1991) 207–220

(a) Writing in a box.

(b) Free-format writing.

Fig. 1. Free-format writing.

shortest path under linguistic constraints is selected as the segmentation and recognition result.

The method requires a large amount of processing. Therefore, the author 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.

2. JAPANESE HANDWRITING

First, the structure of Japanese writing is introduced. Japanese writing has a hierarchical structure consisting of five layers as shown in Fig. 2. A sentence is usually composed of several words. The words are composed of characters. Here, characters in everyday use in Japan are the 1945 categories of Kanji characters, the 46 categories of Hiragana and some symbols. The Kanji characters are usually composed of one or two radicals and sometimes three radicals. Each radical is composed of

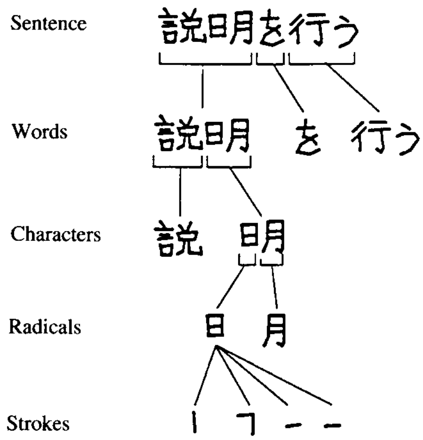


Fig. 2. Structure of Japanese writing.

several strokes. Here, the stroke is represented by the pen movement between pen-down and pen-up.

The following are the recognition problems due to the characteristics of Japanese handwriting.

(1) Handwritten characters may touch each other (Fig. 3(a)), 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 one part (e.g. left-hand radical and right-hand body) (Fig. 3(b)).

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 (Fig. 3(c)).

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 in daily use, so the maximum number of strokes was 23 in the 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 on-line recognition system. This problem can be solved by special hardware.

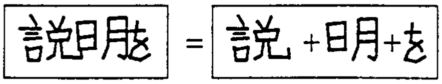
3. RECOGNITION ALGORITHM

This chapter introduces the candidate lattice method for on-line 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. It has a hierarchical structure as shown in Fig. 4. The algorithm contains the following six steps:

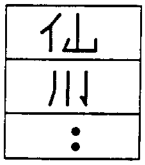
) handwriting data input,
(2) separation into basic segments,
(3) generation of candidate characters,
(4) matching of candidate characters,
(5) candidate lattice formation and shortest path search,
(6) linguistic processing.

Recognition objects were limited to Japanese characters. There are 1945 Kanji characters and 46 Hiragana characters in daily use, giving a total of 1991. The writing order of strokes in a character is free.

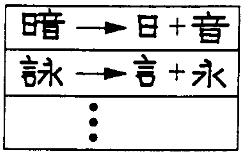
Japanese character strings can be written either vertically or horizontally. The method proposed here determines the direction of writing from the average direction of pen movement, i.e. top-to-bottom or left-to-right, in the preprocessing. The method



(a) Overlapping characters.



(b) Characters composed of separated parts.



(c) Characters composed of other characters.

Fig. 3. Problems for character segmentation.

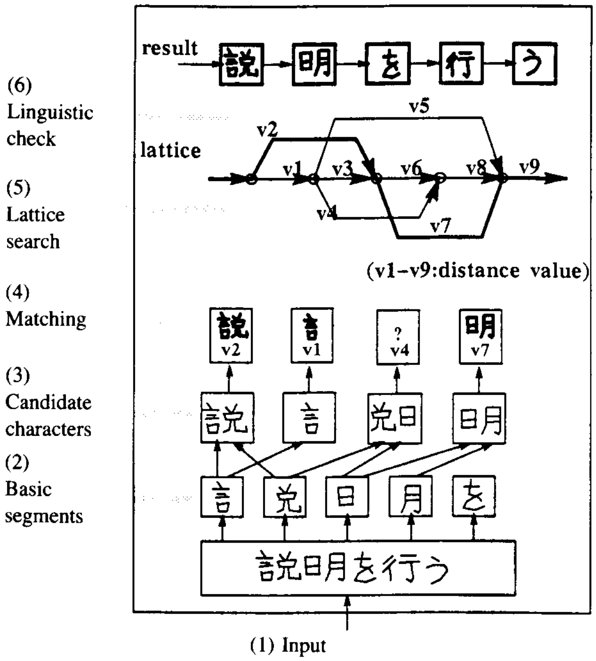


Fig. 4. Block diagram of candidate lattice method.

can deal with both vertical and horizontal writing, although only horizontal writing is treated in this paper.

3.1. Input of Strokes

An electromagnetic data tablet is used as an input device. The tablet samples pen movement in terms of x,y-coordinate values at the rate of 100 points per second. Pen-up and pen-down signals are detected by a microswitch in the pen. A 'stroke' refers to the writing from pen-down to the subsequent pen-up and is represented by the coordinate values of the pen movement between the pen-down and pen-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. Figure 5 shows the definition of profile overlapping. Figure 6 shows examples of segmentation for different Th values. 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. 6 ($Th = 0.0$). On the other hand, a small Th leads to over-segmentation and increases the amount of calculation in the next step. In the 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 according to the following conditions.

(1) The number of strokes to form a candidate should be less than 24 (see Sec. 2).
(2) The ratio of the height of the 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. Examples of candidate characters are shown in Fig.7.

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 (d_{ij}) between the ith stroke of the candidate character and the jth stroke of a standard character for all possible combinations of i and j, by

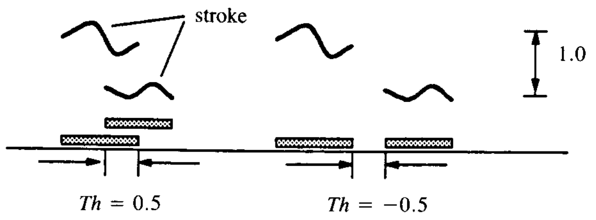


Fig. 5. Definition of profile overlapping.

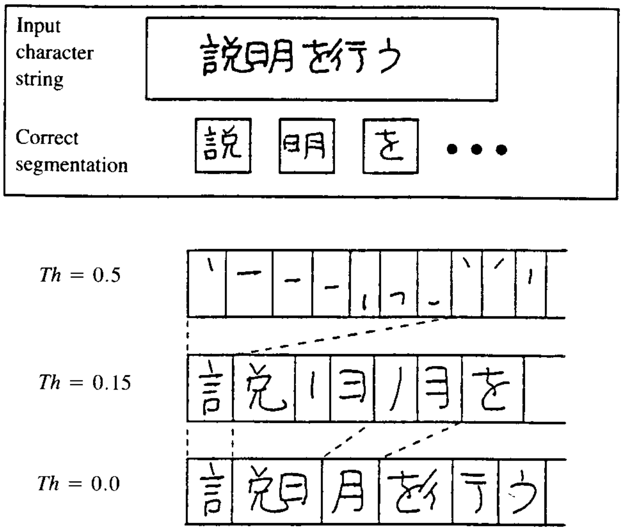


Fig. 6. Basic segments vs. parameter *Th*.

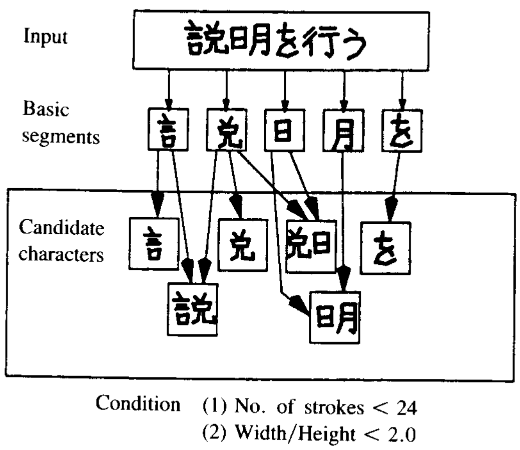


Fig. 7. Generated candidate characters.

DP matching based on inter-point Euclidean distances.

(3) Calculate the inter-character distance D by

$$D = \sum_i \left\{ \min_j (d_{ij}) \right\} .$$

Refer to Fig. 8. Step (3) follows the stroke-order-independent matching proposed by Odaka *et al.*[1] Here, if the stroke number of the candidate character is different from that of the standard character, the distance is set to zero.

(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 preliminary investigation.

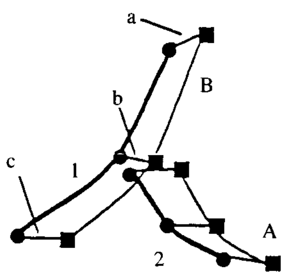
3.5. Candidate Lattice Forming and Path Search

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. 9).

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

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

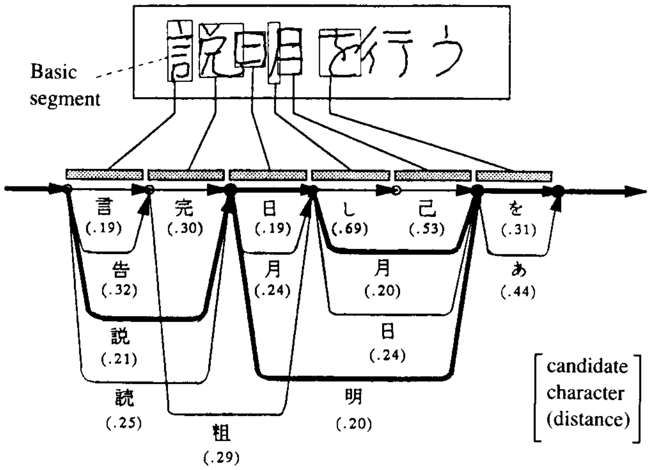
The sequence of candidate characters that corresponds to the determined path becomes the tentative recognition result.



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

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

Fig. 8. Stroke-order-independent matching.



Candidate character strings

(1) 説 明 を 行 う

(2) 説 日 月 を 行 う

Fig. 9. Example of candidate lattice.

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 phrase can form another noun phrase').[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.

Figure 10 shows the case where two similar candidates are extracted. Both candidates are valid from the standpoint of character recognition. However, the upper one is grammatically incorrect. By this method, the lower one, which is the correct answer, is selected.

4. RECOGNITION SYSTEM

This section shows the prototype system for free-format Japanese handwriting recognition and the specially designed hardware for the candidate lattice method.

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 recognition. For

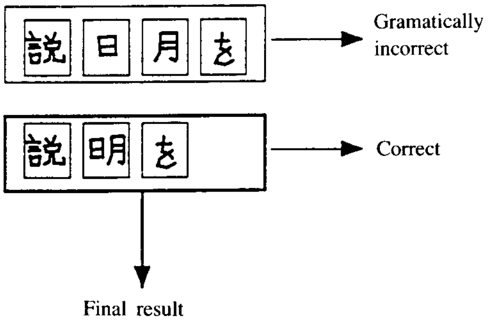


Fig. 10. Two similar tentative recognition results.

the 5-character string shown in Fig. 9, 51 candidate characters are generated. This means that 51 × 1991 (= 101 541) inter-character distance calculations are necessary.

In order to perform such a vast amount of calculation in real time, special hardware was developed. The hardware has 16 conventional microprocessors with multipliers linked in parallel. The block diagram of the hardware and its system is shown in Fig. 11. The top view of a printed circuit board for each processor element (1PE) and the hardware are shown in Figs. 12 and 13 respectively. The hardware performs inter-character distance calculations 16 times faster than one PE, by dividing the dictionary of 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. 5, whereas this special hardware takes only 6 seconds. Therefore, the recognition of

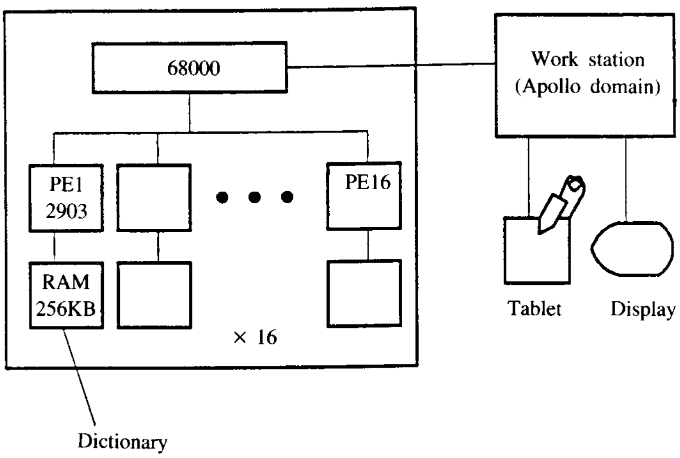


Fig. 11. Block diagram of the on-line recognition system.

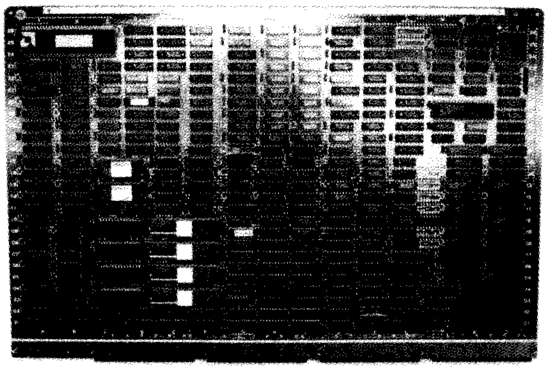


Fig. 12. Printed circuit board for each processor element.

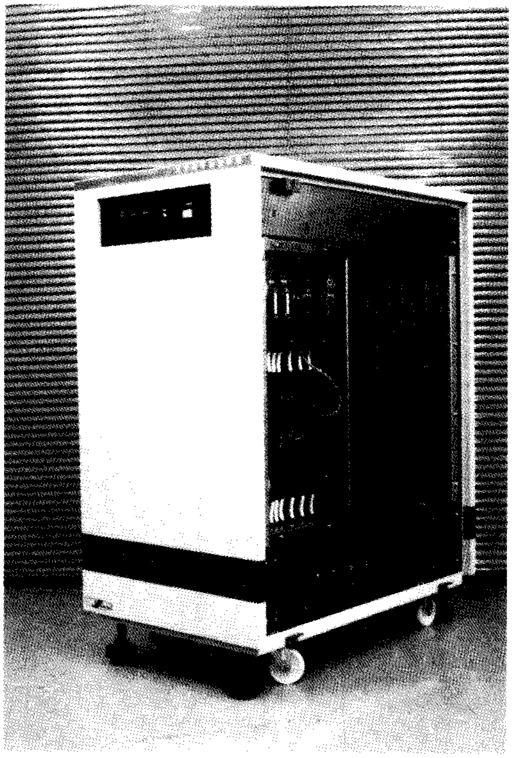


Fig. 13. Specially designed hardware.

Japanese character strings has become fast enough to keep up with the average Japanese writing speed of 3 seconds per character, thus giving a real-time system.

A simulation system was built on an Apollo Domain workstation with this hardware incorporated (Fig. 14). Figure 15 shows an example of recognition results displayed on this system. The upper window shows the input string, and the lower window the recognition result and Kth candidate path ($K = 1, 2, 3, 4$), which corresponds to the Kth shortest paths in the lattice.

5. EXPERIMENTAL RESULTS

Recognition tests were made on 105 samples written by 7 persons. The sentences were excerpts from technical reports, which satisfied the following two conditions:

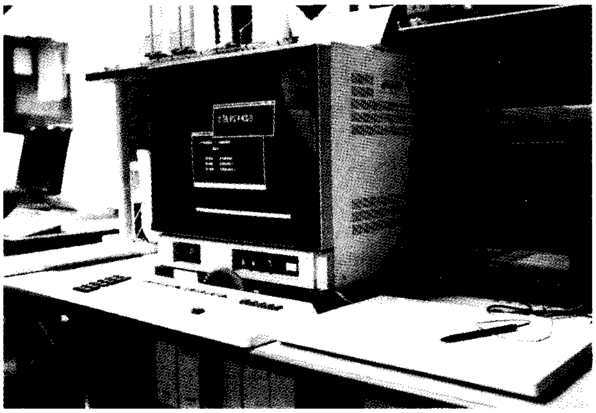


Fig. 14. Prototype system.

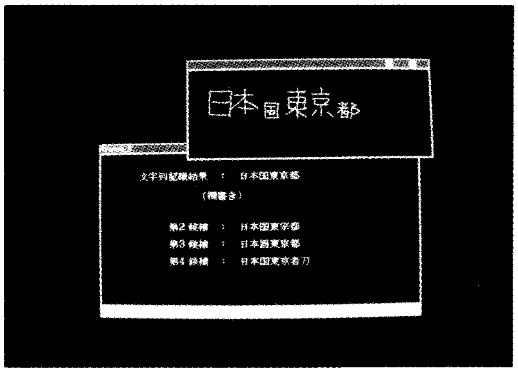


Fig. 15. An example of the result displayed on the workstation monitor.

(1) The number of characters in a sentence is about ten.
(2) Only words stored in the word dictionary are used (see Sec. 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 worthy to note 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.

6. DISCUSSION

6.1. Applicability

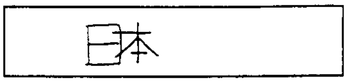
The candidate lattice method can be successfully applied to a variety of writing such as overlapping characters, and characters of different size or pitch (Fig. 16), because the method does not use pitch or size information.

6.2. Linguistic Check Iteration

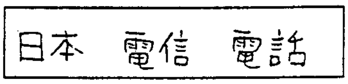
Linguistic checks improve the accuracy of segmentation and recognition. Figure 17 shows the distribution of the number of linguistic check iterations necessary for correct segmentation and recognition. It can be seen that five iterations are sufficient.

Table 1. Experimental results.

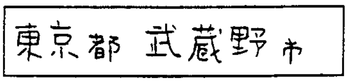
	Correct segmentation	Character recognition	Overall recognition
Gap information only	68.5%		
Without grammatical rules	96.7%	99.2%	95.9%
With grammatical rules	98.8%	99.9%	98.7%



(a) Overlapping character string.



(b) Different pitch character string.



(c) Different-sized character string.

Fig. 16. Correctly recognized character strings.

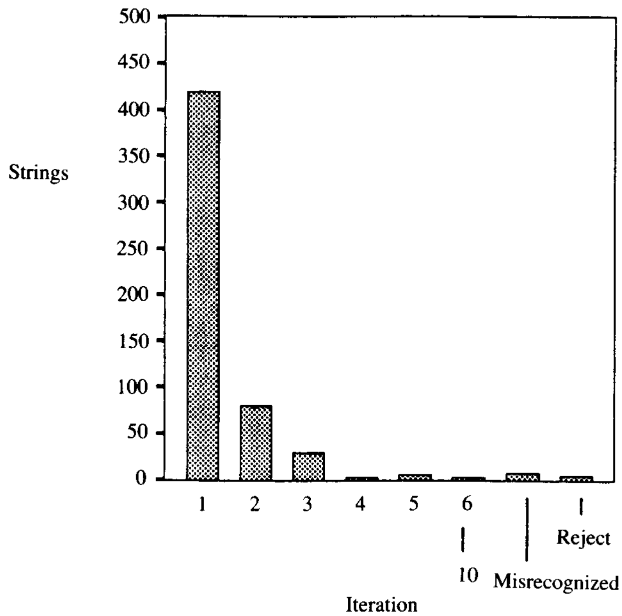


Fig. 17. Distribution of iterations.

6.3. Efficiency of Using Basic Segments

Use of basic segments instead of individual strokes lessens the total number of candidate character generations. In Fig. 6, 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 hardware specially designed 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 using 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 six seconds to recognize a 5-character string. The recognition has become fast enough to keep up with the average Japanese writing speed.

The recognition system based on the candidate lattice method has a great capability for handling a variety of free-format writing. However, the following problems still remain for future work. They are (1) refinement of linguistic processing, (2) further utilization of positional information about characters to improve performance and to reduce misrecognition, (3) application to handwritten Japanese word processing systems containing an editing function.

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.

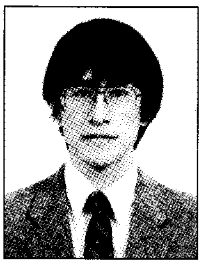
ACKNOWLEDGEMENTS

The author would like to thank Dr. Isao Masuda, Dr. Seiichiro Naito, Dr. Toru Wakahara and Mr. Mikio Shinya for valuable advice and encouragement, and Dr. Noriyoshi Osumi for his advice in preparing the paper. He is also grateful to Mrs. Yoko Murase for her linguistic collaboration.

REFERENCES

1. K. Odaka, T. Wakahara and I. Masuda, ''Stroke-order-independent on-line character recognition algorithm and its application'', *Rev. Electr. Commun. Lab.* **34** (1986) 79–85.
2. T. Wakahara and M. Umeda, ''On-line cursive script recognition using stroke linkage rules'', *Proc. 7th Int. Conf. on Pattern Recognition*, 1984, pp. 1065–1068.
3. G. F. Groner, ''Real-time recognition of handprinted text'', *AFIPS Proc. Fall Joint Computer Conf.*, 1966, pp. 591–601.
4. C. C. Tappert, ''Cursive script recognition by elastic matching'', *IBM J. Res. Dev.* **26** (1982) 765–771.
5. H. Murase and T. Wakahara, ''On-line hand-sketched figure recognition'', *Pattern Recogn.* **19** (1986) 147–160.
6. M. Shinya and M. Umeda, ''Evaluation of compound post-processing method in character recognition'', *Trans. IECE Japan* (in Japanese), **J98-D**, 5 (1986) 1118–1124.

Received 5 December 1989; revised 30 January 1990.



Hiroshi Murase received his B.E., M.E. and Ph.D. degrees in electrical engineering in 1978, 1980 and 1987 respectively. He has been engaged in pattern recognition and character recognition research at NTT Basic Research Laboratories in Tokyo since 1980. His current research interests lie in the areas of computer vision, pattern recognition, and human visual perception.