

RECOGNITION OF HANDPRINTED ALPHANUMERICS, SYMBOLS AND "KATAKANA" CHARACTERS

Yukio Hoshino¹, Hiroyuki Kami¹, Ritsuko Ohmori¹, Hideaki Ueda² and Yasuhumi Mitsuzawa²
¹Central Research Laboratories
Nippon Electric Company Ltd.
1-1, Miyazaki Yonchome, Takatsu-ku,
Kawasaki, Kanagawa 213, Japan

²Peripheral Engineering Division
Nippon Electric Co. Ltd.
10, Nisshin-cho Itchome,
Fuchu, Tokyo 183, Japan

This paper describes a new recognition system for 106 handprinted characters. The physical image of a written character is converted to an original binary pattern. This binary pattern is smoothed and thinned to a skeleton pattern. An input pattern feature string, described by symbols of singular points and direction codes of segments, is extracted from the skeleton pattern. The successful candidate categories are determined by comparing the input pattern feature string to the standard pattern feature strings. The desired category from the candidate categories is selected by the tournament structured decision tree. The system has been developed as the recognition system for a commercial optical character reader, OCR N6370, system. The recognition is carried out by custom ordered LSIs for preprocessing and a micro-computer for feature extraction and decision.

1. INTRODUCTION

In handprinted character recognition by feature extraction, measured features, such as concavities, loops, or stroke direction etc. are compared to the standard patterns. However, it is necessary to overcome the problems concerned with feature extraction and decision making technique in order to recognize the large number of symbols, as shown in Fig.1.

(A) The first problem is concerned with noise smoothing. There are many individual patterns, such as シ, ヲ, ッ, ネ, ホ, ン, " : ; . , and ? in Fig.1. Accordingly, if short lines or small points are erased by the noise smoothing operation, some misrecognitions will occur.

(B) The second problem is that there are many similar character pairs among the symbols shown in Fig.1. Especially, there are many pairs of topologically the same symbols, such as (0, □, °), (5, S), (7, 7, 7, 7), (9, 9), (I, 1), (ノ, ン), etc. Therefore, it is necessary to detect many detail features from the characters so as to discriminate properly between all these pairs.

(C) The third problem is the necessity to use the contextual information. Since symbol and character sizes depend on individual writers, for example, the size of numeral 0 is sometimes the same as the size of special symbol ° (P sound used in combination with katakana). Accordingly, it is difficult to discriminate between numeral 0 and symbol °.

Since the second problem was included, even in the recognition of freely handwritten numerals, auxiliary features were utilized to assist in emphasizing the main recognition features which

have a bearing on determining which symbol is being recognized. However, this recognition algorithm cannot conveniently handle more than 100 character symbols. The authors developed a new recognition system which is capable of solving the above mentioned problems for handprinted Japanese katakana characters. After that, the recognition system was extended for special symbols, the mixed use of alphanumerics and special symbols, and the mixed use of alphanumerics and katakana characters.

2. RECOGNITION ALGORITHM

The digitized character is represented by a binary matrix whose maximum dimensions are 64x48 bits. This is the original pattern. The original pattern is smoothed and transferred to a skeleton pattern. Each consecutive sequence of black dots (ones), between a terminal point and its connected terminal point or branch point, is called a "stroke". If the length of the stroke is shorter than a predetermined value, the short stroke is eliminated from the skeleton pattern. Instead of elimination, their position and direction are sequentially recorded in the memory. These features are mini features. Terminal points and branch points are easily detected from the skeleton pattern. Those positions are stored in the memory according to their detection order. The stroke is traced in order to detect inflection points, and divided into segments which are defined by consecutive sequence of black dots between the terminal points, the inflection points and the branch points on the stroke. The order of strokes is determined by the detection order of terminal point included in each stroke. Direction code,

as shown in Fig.2-1, is given to the segment. The origin of Fig.2-1 is corresponding to the preceding singular point detected by tracking. Figure 2-2 is an example of obtained segments, direction codes, and input pattern feature string. Each symbol, used in the input pattern feature string, has one of the following meanings. 0,1,2,-----7 : Direction code. E : Stroke tracking ended at a terminal point. K : Inflection point. B : Branch point.

First decision is performed by sequentially comparing the input pattern feature string (IPS) to the standard pattern feature strings (SPSs) with category name. The structure of each SPS is similar to that of IPS. After comparing, if no SPS is found, the character is rejected, if unique category SPSs are found, the character is recognized as the SPS category. If SPSs of more than two categories are found, second feature extraction and decision are actuated.

In the second stage, a category, which is most similar to the input character, is determined by a tournament structured decision tree. Each pair of two categories is sequentially combined from the candidate categories. When the winning category is determined by sequentially executing several statements regarding feature extraction and decision (hereafter a feature extraction and decision pair is expressed by FED), if there are remaining candidate categories, the winning category and first one of the remaining candidate categories become the next pair. When the winning category is not determined, these drawn categories are stored in the drawn categories list and deleted from the candidate categories. If only one candidate remains, the remainder is treated as the winner. Otherwise, the character is rejected. After all the sequential candidate category combinations, the winning category is confirmed to determine whether the winning category is more similar than all the drawn categories to the input character.

The configuration of the tournament structured decision trees depends on the input character shape. Also, the auxiliary feature to be detected and the parameters, such as the predetermined threshold, depend on the category pair. The auxiliary feature name and its parameters are described in an FED. The format of each FED is as follows.

C1 C2 F pl p2...pm dl d2...dn t1 t2 s nb nt

Each symbol has the following meaning.

C1: First category name. It is always necessary.
C2: Second category name or logical mark. The second category name is necessary when pair categories are distinguished by the current single FED or at the first FED for a logical combination

of FEDs. Logical mark is used when the current FED is logically combined with the preceding FEDs.
pl p2...pm: Singular point name (Bi, Ti, Ki, BI, TI). Bi, Ti, or Ki is i-th branch, terminal, or inflection point respectively. TI or BI is a terminal or branch point connected with another directly assigned point, respectively.

dl d2...dn: Parameters for feature extraction area size or directions for measuring the feature, etc.

F: Auxiliary feature name. Example of auxiliary features are as follows. a Distance between assigned two points vs. the distance between two other assigned points ratio. b Examine whether an assigned direction mini feature is in the assigned region. c The concavity or convex value in the area determined by two assigned points. d Current character height vs. preceding character height ratio. In execution time, a feature extraction routine is called by an auxiliary feature name in an FED. The measured feature value f is recorded in a temporary memory.

t1 t2 s: Functions of these parameters are defined in Table 1

nb nt: Numbers of branch points and terminal points respectively.

A series of FEDs for pair categories Ci and Cj, as shown in Table 2, constructs a decision tree as shown in Fig.3. Feature Fi value is represented by fi in Fig.3.

(3) HARDWARE IMPLEMENTATION AND CONCLUSION

The recognition algorithm was implemented in an actual OCR N6370 system. The original pattern is preprocessed in LSIs, which were developed for the OCR. Feature detection and decision are carried out by a micro computer. The feature detection program and decision program are recorded in 20k bytes of the micro program memory. The SPSs and the FEDs are recorded in 64k bytes of the data memory. Design work was laborious because the design concept depended on the heuristics of the designer and recognition experimentation. Instead, an economical OCR system has been successfully developed. The OCR can recognize 100 characters/sec.

REFERENCES

- (1) Suen, C.H., et al, Proc. 4th Int. Jt. Conf. Pat. Recog., pp30, Nov. 1978.
- (2) Hoshino, Y. and Kiji, K., Proceedings of the 1st Int. Jt. Conf. Art. Intel., ppl53, May, 1969.
- (3) Miyazaki, T. and Hoshino, Y., Institute of Electronics and Commun. Japan, PRL74-62, Feb., 1975. (in Japanese)
- (4) Yamamoto, I. et al, NEC Technical Journal, No. 11, pp29, Feb. 1977 (in Japanese)

0	1	2	3	4	5	6	7	8	9	-
---	---	---	---	---	---	---	---	---	---	---

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z -

アイウエオカキクケコサシスセソタチツテトナニヌネノ
A I U E O KA KI KU KE KO SA SHI SU SE SO TA CHI TSU TE TO NA NI NU NE NO

ハヒフヘホマミムメモヤユヨラリルロワヅン〃〇
HA HI FU HE HO MA MI MU ME MO YA YU YO RA RI RU RO WA WU ZU N N' O

+	-	*	/	=	<	>	€	%	^	(	)	¥	\$	.	┐	:	;	π	▮	?	#	ω
+	-	*	/	=	<	>	&	%	^	(	)	¥	\$	.	┐	:	;	π	▮	?	#	ω

These character shapes, without some special symbols depend on the Japanese handprint standard (draft)

When			If				Classified To	
S	$t_1 > t_2$	$t_1 < t_2$	$f > t_1$	$f < t_1$	$f < t_2$	$f > t_2$	C_1	C_2
1	1		1				1	
					1			1
		1		1			1	
						1		
2	1		1					1
					1		1	
		1		1				1
						1	1	
3	1		1				1	
		1		1			1	
4	1		1					1
		1		1				1

TABLE 1. Detected value f and parameters s , t_1 and t_2 .

::: ----- AND : ----- OR

Cl Cw F pl---pm dl---dn t1 t2 s nb nt

Ci	Cj	F1-----	7	2	4-----
Ci	::	F2-----	8	5	4-----
Ci	:	F3-----	2	6	4-----
Ci	::	F4-----	3	9	4-----
Ci	:	F5-----	1	3	3-----
Ci	Cj	F6-----	2	6	3-----
Ci	::	F7-----	8	6	3-----
Ci	Cj	F8-----	5	2	1-----

TABLE 2. A Series of FEDs

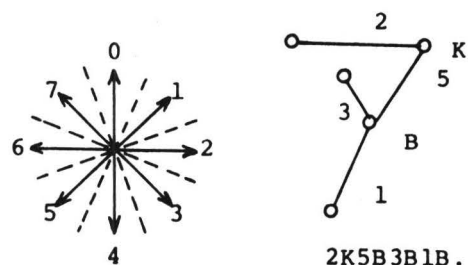


Fig. 2-1.

Fig. 2-2.

Fig.2-1. Direction code

Fig.2-2. Segments, direction codes and feature string for T(A)

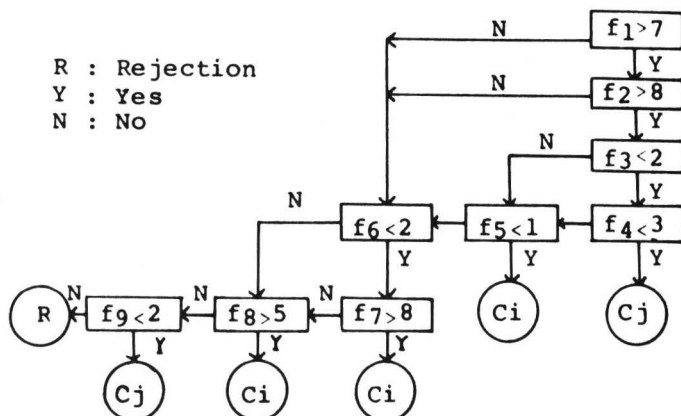


Fig.3. TABLE 2 diagrammatical illustration