

MACHINE RECOGNITION OF TYPEWRITTEN
CHARACTERS BASED ON SHAPE DESCRIPTORS

by

Eugene J.A. Kanciar

B.A.Sc., University of Western Ontario, 1972

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in the Department

of

Electrical Engineering

We accept this thesis as conforming to the
required standard

THE UNIVERSITY OF BRITISH COLUMBIA

October, 1974

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study.

I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver 8, Canada

Date Oct 17/79

ABSTRACT

An optical character recognition technique for typewritten letters was developed with application to a personal reading machine for the blind. The feature extraction process defined a character in terms of lines and shapes which are closely related to a person's description of form. The system was developed to identify all upper and lower-case typewritten characters in the alphabet. A letter was described by any combination of seven basic features, usually in a 3 x 3 feature matrix. The extraction of topological (or structural) properties had several advantages; a very small feature dictionary with about 100 code-word entries; quick and simple training procedure for a new font; and, a strong capability to handle character deformities. A separate technique, based on edge examination, was developed to identify characters with prominent diagonal features. Sequential classification was employed throughout the entire system so that recognition was made once a sufficiently unique measure was satisfied.

Tests on both repeated characters and typewritten passages produced approximately 97% accuracy when the system was applied to three fonts which varied from a stylized to a serifless print. For a scanning rate of 60 wpm, a recognition speed of two characters per second was achieved. The system was developed on a PDP-12 computer and is fully compatible for realization on a PDP-8 computer with 8K of memory.

TABLE OF CONTENTS

	Page
Abstract	ii
Table of Contents	iii
List of Illustrations	v
Acknowledgement	vi
I. INTRODUCTION	1
1.1 Purpose of Research	1
1.2 Perspective	1
1.3 Existing Schemes	2
1.3.1 Template Matching	2
1.3.2 Contour Tracing	3
1.4 A New Approach	4
II. THE PATTERN RECOGNITION SCHEME	6
2.1 Introduction	6
2.2 Input	6
2.3 Preprocessing	6
2.4 Feature Extraction, Part 1	7
2.4.1 Introduction	7
2.4.2 Basic Concepts	7
2.4.3 String Identification	8
2.4.4 Four Level Resolution	9
2.4.5 Basic Feature Assignment	10
2.4.6 Vertical Compression	12
2.4.7 Serif Deletion	13
2.4.8 Horizontal Feature Integration	14
2.4.9 Horizontal Compression	16
2.4.10 An Example with Curvature	17
2.5 Feature Extraction, Part 2	19
2.5.1 Characters with Diagonals	19
2.5.2 Special Cases	20

2.6	Classification	20
2.6.1	Introduction	20
2.6.2	Code-Word Generation	21
2.6.3	Subgroup Assignment	21
2.6.4	Identification	22
2.7	Overall System Configuration	23
III	TESTS AND RESULTS	24
3.1	Description of Test Material	24
3.2	Tests and Results with Repeated Letters	25
3.3	Tests and Results with Typewritten Passages	25
3.4	Sources of Error	26
3.5	Training Procedure	28
3.6	Speed	28
3.7	System Storage Requirements	29
3.8	System Flexibility	29
IV	CONCLUSION	30
4.1	Summary	30
4.2	Recommendations for Further Research	31
	References	33
APPENDIX A	Confusion Matrices	35
APPENDIX B	Examples of the Character Recognition Process	40
APPENDIX C	Examples of Errors	43
APPENDIX D	Examples of System Flexibility	45
APPENDIX E	Basic Flowchart for the Diagonal Checking Routine.	47

LIST OF ILLUSTRATIONS

	Page
1. The template matching scheme	2
2. The contour tracing scheme	4
3. The closure criteria	8
4. String identifications in the letter 'R'	9
5. Four level resolution of string information for the letter 'R'	10
6. Basic feature assignments for the letter 'R'	12
7. Vertical compression to three levels for the letter 'R' ...	12
8. Serif deletion in the letter 'R'	13
9. Horizontal feature integration for the letter 'R'	16
10. Horizontal compression for the letter 'R'	17
11. Feature integration applied to a character with curvature, the letter 's' as an example	18
12. Recognition of diagonal characters by selective edge examination	19
13. The three code-words for the letter 'R'	21
14. Subgrouping the alphabet	22
15. Character identification, the letter 'c' as an example ...	23
16. Block diagram of the system	23
17. The three fonts used in the tests, reproduced in full size	24
18. Suggested closure criteria to replace the criteria in Figure 3	32

ACKNOWLEDGEMENT

The author would like to thank Dr. M.P. Beddoes for his constant interest and input of ideas into this project. The availability of his time to my inquiries is gratefully appreciated. Thanks are also due to Rodney G. George for all his assistance with software implementation as well as his practical insight into problem areas. Finally I would like to extend my appreciation to all the graduate students, faculty and staff with whom I shared my experiences.

Financial assistance was provided by the National Research Council of Canada, the Medical Research Council of Canada, and the Vancouver Foundation.

I. INTRODUCTION

1.1 Purpose of Research

The purpose of this research was to : (1) devise an optical character recognition scheme that would be applicable to a personal reading machine for the blind; (2) develop a feature extraction technique based on shape analysis and comparable to human description of form; (3) obtain preliminary recognition results and determine the sources of errors with typewritten characters.

1.2 Perspective

The basic approach to reading aids for the blind has been to develop direct translating devices that produce facsimile reproductions of characters with either tactile [4] or auditory [3], [20] output. The advantages achieved in this approach are compactness, technical simplicity, and economy. The disadvantages of this approach are that the reader must be trained to decode the output and the reading rates are low.

The next order of complexity in a reading machine has been to employ OCR (optical character recognition) with spelled or spoken speech as output. Character recognition schemes have been in commercial use for a number of years. Present users of OCR include the post office, government, and industrial concerns. In these cases OCR is used with applications requiring large volume, high speed, and accuracy. Usually these schemes utilize a well defined set of characters. By contrast, relaxation of many performance criteria may be applied to a personal reading machine; but, at the same time this machine must cope with a wider range of fonts and printing quality than those which are encountered in commercial applications.

Two schemes have been developed over the last ten years for this particular purpose and these are discussed in the next section.

1.3 Existing Schemes

1.3.1 Template Matching

One solution to the OCR problem has been Smith-Mauch's template matching scheme [20]. The locations of black areas, encountered as the template scans across the character, are processed to identify it. This system, illustrated in Figure 1, consists of a hand held probe that focuses the printed character onto a two-dimensional photocell array of 12 sensors. The discriminating routine is applied in a multiple snapshot sequence. As a character is scanned across the optical input, it stimulates a trigger cell (TC). Every time a white to black or black to white transition occurs in the trigger cell, a snapshot of all the cells in template is taken. Output from these cells is converted into a five-bit code that specifies the character. The objective of this scheme is an accuracy of 90% to 95% at 80 wpm to 90 wpm but this has yet to be achieved. The present reading speed is about 20 wpm to 25 wpm. The recognition logic of this machine is designed with integrated circuits.

For this scheme to work well, several critical parameters must be satisfied. The exact positioning of the template over the letter is important. Also, the threshold for information when a cell on the template is only partially in a black field is a relevant consideration. This scheme, in extracting spatial information in a snapshot manner, does not make good use of geometric or structural properties which require more continuous measurements. Thus the system's performance is limited because the extracted information is not complete enough to handle problems such as character distortions or ambiguities.

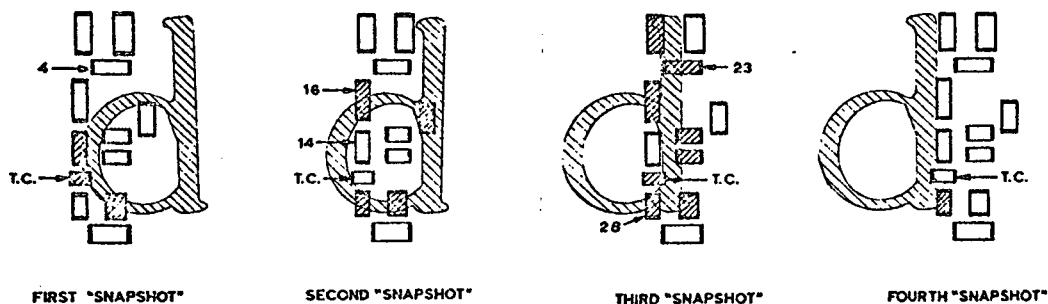


Fig. 1 The template matching scheme.

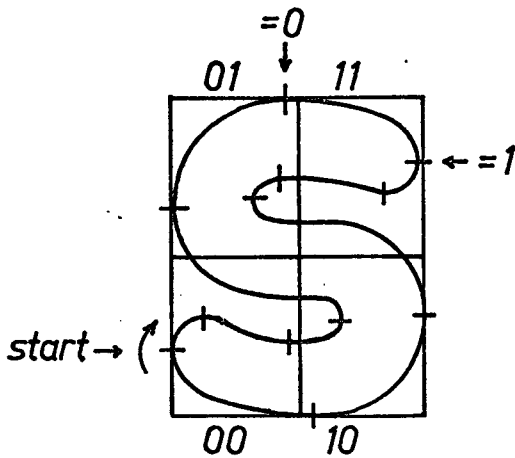
1.3.2 Contour Tracing

Mason [16] and Clemens [6] proposed an OCR scheme which utilized contour tracing. This technique, illustrated in Figure 2, requires three pieces of information: a code word that specifies the character's geometric extrema in the x and y directions; a coord word which stores the location of each extremum; and, the height-to-width ratio (H/W) that classifies the character into one of four subgroups. From these measurements a 30-bit code word is derived. The lookup table contains three to five possibilities for every character. The original scheme, as developed by Clemens [6] achieved 1% to 3% error at 100 wpm and was tested on ten fonts. Further improvements upon the system have reduced the error to 0.1% but this has been achieved on work with only one font [13], [16]. This system uses a PDP-9 type of computer and requires a flying spot scanner for the data input.

The code word is a good topological description of a character's shape. However both the coord word and the height-to-width ratio are rigid geometric measurements which do not make this scheme readily adaptable to other fonts. The coord word is a sensitive function of position and as such, it will alter as the extrema are located in new regions. The height-to-width ratios also vary among the different fonts.

The print material used by Mason and Clemens to test their recognizer was free from breaks. As the recognizer deduces from the print not only the print information but also the position to explore next, a break produces a doubly catastrophic error. These break errors are minimized by using high resolution (60 x 60 points were used by Lee [13]).

One last basic limitation with the Mason-Clemens recognizer is that only the outside contour of a character is used. With ambiguous pairs of characters such as 'D' and 'B', 'c' and 'e', a special technique, such as taking a vertical slice through the character's centre and noting the number of intersections (two or three) was used, because the outside contour was an insufficient measure.



1. code word: 1 0 0 1 1 ...
2. coord word: 00 00 00 10 01 ...
3. H/W ratio

Fig. 2 The contour tracing scheme.

1.4 A New Approach

The pattern recognition techniques that have been developed on large computers are not realizable on small, economical processors. Statistical and probabilistic approaches [2], [11] are not applicable for this purpose because of their complexity. The two schemes discussed in Section 1.3 represent one approach to the problem; that is, discrete spatial information was extracted and no attempt was made to integrate the measurements into more general and higher order information. In this thesis such a new approach was undertaken.

The process of identifying a letter by means of features is well known [1], [14]. However, results show that a complex machine is needed to recognize one font [16]. This work set out, as a main aim, to invent a set of features to identify a letter simply. A further consideration has been to relate these features to descriptors of shape which would appear natural to the human reader. For example, with the letter 'b' we could use the description: "a straight vertical line at the left which carries a little flag (serif) at the top; this line intersects two horizontal lines, one at the bottom and one half way up, which are traced to the right and, moving together, form a junction at the right." This compact description of form is strongly topological and gives the gist of how this OCR scheme interprets a character. Our technique requires seven features to characterize a letter. Usually the letter was reconstructed within a 3 x 3 feature matrix. A special technique was used for characters with prominent diagonal features, such

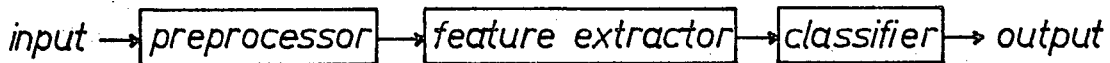
as 'W', whereby the character's edge was used for its identification. This scheme should be less sensitive to distortion and more flexible in its recognition capability than the existing schemes.

This thesis is based on original work and is conceptually most similar to work done by Genchi [12] and Marko [15]. Genchi has developed an OCR scheme which is used in the Japanese postal system for recognition of handprinted numerals. The numerals were recognized by extracting a sequence of features in horizontal zones. A 3 x 3 scanning window identified each 9 element array as one of 7 possible stroke segments such as blank, vertical, slanted, etc. In the second step, each horizontal string of stroke segments was classified into one of 16 horizontal features. The third step was to connect vertically all the individual horizontal features by using a transition table. The character was recognized by this table. With Marko's system, spatial filtering was performed on the character in four directions (horizontal, vertical, and two diagonal directions). From this processing, higher order features such as angles, curvatures, and endpoints were extracted. Information was then integrated into a very compact feature matrix. Finally, a weighting scheme was applied to these features which were then processed by the classifier.

II. THE PATTERN RECOGNITION SCHEME

2.1 Introduction

In order to describe the techniques employed in the character recognition system it is useful to employ a general model for a pattern recognition system with each step in the processing being identified:



The scanner is the transducer, a photo-electric device, which converts the printed character into a two-state (binary) input. The preprocessor, if present, employs techniques such as data normalization and stroke thinning. The purpose of the feature extractor is to obtain a set of characteristic measures and statements upon which a decision as to the most probable identity of the pattern sample can be made. The classifier, on the basis of the information provided by the feature extractor, applies a decision criterion to the pattern measurements to make a decision as to which, if any, of the allowable classes the pattern belongs. The output relays the identification in an appropriate form.

2.2 Input

A 64 vertical element Reticon RL-64P solid-state line scanner provided the input for a PDP-12 computer. The speed of the scan corresponded to approximately 60 words per minute. The typical picture frame for a single upper case character contained 15 x 30 elements and 12 x 20 elements for a small lower case letter.

2.3 Preprocessing

Preprocessing refers to data manipulation in the primary stages of a process so that redundancy and distortion can be appreciably reduced. In the OCR scheme that was developed, there was no formal preprocessing. It was believed that the successive stages of information handling contained enough sophistication to be able to cope with the inherent noise in the binary data. The system was designed with this premise in mind. A preprocessor [24] can have the following objectives: a character should ideally be of unit

thickness; this skeleton must resemble the line pattern a human would draw, that is, no information other than the line width is lost; breaks in the character must be searched for and, where missing, the continuity should be restored. As useful as these objectives may be, for this specific application, where the cost of machine implementation must be kept low and an error rate of a few percent can be tolerated, this preprocessing is not justified.

2.4 Feature Extraction, Part 1

2.4.1 Introduction

Feature extraction in OCR has received considerable attention because the effectiveness of this stage predicates the recognition ability of the entire system. It consists of one or a combination of three possible techniques [21] : geometrical, topological and mathematical (or statistical) feature extraction. Topological feature extraction has been considered in the recognition of handwritten characters. Eden and Halle [9] found that only 18 different strokes were needed to construct any English character. By partitioning the perimeter around a character into eight sections, Tou [22] was able to extract topological features such as bays, inflections and curvatures from each octant. Interesting work on computational topology has been reported [23] , however, this approach is yet to be applied.

2.4.2 Basic Concepts

Important concepts and terms are introduced. A string is a continuous segment of binary 1's within a column and has the property length, L . Each string is characterized by one of three descriptions: a vertical (V), a closure (C), or a centre cell (CC). For a character of height H , $H/4 = T + R$ (where the remainder, R , is $0 \leq R < 3$) defines the $1/4$ height threshold T .

Two types of verticals are defined: a full vertical and a minor vertical. A string with $T < L < 3T$ was assigned a minor vertical. A string with $L > 3T$ was assigned a full vertical.

Whenever two branches of a character converged to or diverged from a string common to both, that string was classified as a closure. In the letter 'c' in Figure 3, column three is identified as a closure because it initiates the divergence of the two branches in column four. For closure to exist all the conditions in Figure 3, must be satisfied. If columns three and four are interchanged the test would indicate convergence. Any string which was classified as a minor vertical was tested for closure. If the string was identified as a full vertical, the closure criterion was not applied.

A string with $L < T$ was assigned its centre cell. However, a string with $T/2 < L < T$ was also checked for a closure. Occasionally a closure did occur with $L < T/2$ and in these cases the closure was missed but these errors were infrequent.

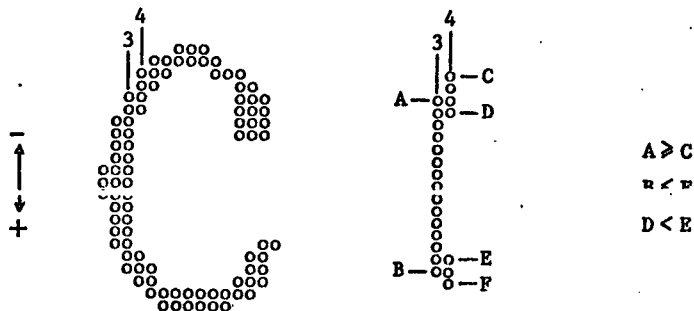


Fig. 3 The closure criteria.

2.4.3 String Identification

The letter 'R' was chosen to demonstrate the OCR technique that was developed. A typical binary picture for the letter 'R' is shown in Figure 4A.

The first information to be noted about the character was its height and width (28×17) and, $T (28/4=7)$. The strings in columns one through three were less than T . Thus each string was completely described by its CC. For the string in column four $L > 3T$, so a full

each column were quantized into one of four possible vertical quadrants. For the example treated in Section 2.4.3, the information so processed appears in Figure 5. The blank quadrants are zero entries.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	1	V	1	1	1	1	1	1	1	2	4	C	27	28	28
27	27	27		13	13	14	14	14	15	15	C	12	27			
				20	27	27	28					V				
				26												

1	1	1	V	1	1	1	1	1	1	1	2	4	C			
			V	13	13	14	14					12	C			
			V	20					15	15	C	V				
27	27	27	V	26	27	27	28				C	V	27	27	28	28

Fig. 5 Four level resolution of string information for the letter 'R'.


2.4.5. Basic Feature Assignment

Each column entry of data is compared to the information stored in the next column. This comparison results in one of six basic features being assigned :

- (/) U - Up
- (\) D - Down
- (—) H - Horizontal
- (|) V - Vertical
- (X) C - Closure
- (.) Z - Zero

For two adjacent columns, a pair of adjacent elements (X,Y) can be obtained. Figure 6A summarizes the feature selection process for two adjacent columns. The assignment of basic features is illustrated in Figure 6B. When the first column was compared to the second, the first (X,Y) pair was (1,1) which was represented by an H. Likewise for the second pair (27,27). The second and third columns produced identical results. When the third and fourth columns were studied, the pairs (1,V) and (27,V) resulted in V entries in the first and fourth vertical quadrants. These entries, although they may seem

superfluous, are necessary for the continuity of the feature selection process. The reasoning behind such entries will become evident when columns eleven and twelve are discussed. A four level vertical was assigned to column four. In columns five and six, the first two pairs (1,1) and (13,13) produced two H features. The third pair was (20,0). In such a case a check was performed on the paired elements immediately above (13,13) and below (26,27). Thus (13,13) was compared to a possible combination of (20,13); similarly, (26,27) was compared to (20,27). In both cases the (X,Y) combination with $X = 20$ was the poorest choice. The comparison therefore remains with (20,0) followed by the last column pair (26,27). These pairs result in Z and D features being entered respectively. The second pair in columns nine and ten (14,0) required a similar search; however, in this case a more appropriate pair (14,15) was found during the search and thus a D feature was stored in the second level. The selected basic feature was entered on the same vertical level as the X in the (X,I) pair. The first pair (1,2) in columns eleven and twelve produced a D feature. For the second pair (15,C) C was assigned. The reason for storing C was to maintain the continuity of the columns without introducing false information. Since the first pair (1,2) spanned both columns, a Z entry for a no comparison for the pair (15,C) would have broken column continuity in the second row. Whereas, possibly entering an H feature to indicate this continuity would have introduced new information into the letter. Now, assume C is entered for (15,C). The next time this same C is encountered between columns twelve and thirteen, it will appear as (C,V) and, from Figure 6A, another C will be entered. Thus two C's will have been entered for the same datum. As it will be shown later, continuous horizontal entries of C or V can be reduced to only one C or V without any loss of information.

Present Column	Next Column			
	CC	C	V	Z
	UDH	C	V	
	C	C	C	C
	V	V	V	V
	Z	Z	Z	Z


 search for best y

Figure 6A

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	1	V	1	1	1	1	1	1	1	2	4	C			
			V	13	13	14	14	14				12	C			
			V	20					15	15	C	V				
27	27	27	V	26	27	27	28				C	V	27	27	28	28

--	--			--	--	--	--	--	--	--	\	\	X	X		
				--	\	--	\						X	X		
										--	X	X				
--	--			\	--	\						X		--	\	--

Figure 6B

Fig. 6 Basic feature assignments for the letter 'R'.

2.4.6 Vertical Compression

After the assignment of basic features, the character was compressed vertically to three horizontal rows with the resulting advantages of compactness and better level continuity. Rows two and three were compressed together. The result of compression between two basic features is summarized in Figure 7A. The following order or superpositioning dominance was imposed : (U,D,H)>C>V>Z.

Rows two and three are compressed in Figure 7B.

2nd Row	3rd Row			
	UDH	C	V	Z
	UDH	UDH	UDH	UDH
	C	UDH	C	C
	V	UDH	C	V
	Z	UDH	C	V


 no compression

Figure 7A

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	--			--	--	--	--	--	--	\	\	X	X		
				--	\	--	\					X	X		
									--	X	X				
--	--			\	--	\					X		--	\	--

--	--			--	--	--	--	--	--	\	\	X	X		
				--	\	--	\	--	X	X	X	X			
--	--			\	--	\				X		--	\	--	

Figure 7B

Fig. 7 Vertical compression to three levels for the letter 'R'.

2.4.7 Serif Deletion

Serifs, which are the cross-lines finishing off letters, tend to increase a letter's legibility. However in the machine recognition process, the serifs were confusing. Due to serifs' short lengths, the printing quality, and the resolution of the scanner, a serif could be described by zero, one, or more basic features. This introduced an unnecessary amount of serif variability.

To increase system reproducibility, horizontal serif deletion was implemented. Serifs are found in the top and bottom quadrants of a letter and are associated with verticals. By assigning an appropriate serif length for the font being read, searching the top and bottom quadrants only, and noting the fact that a serif begins from an empty space and ends in a vertical (and vice versa), selective deletion of serifs was achieved. The serifless character is presented in Figure 8. In addition to the serifs being deleted, the two verticals in column three were also eliminated. A minor routine deleted any column which contained only extraneous verticals with no U, D or H information present. For this example, column four could suitably represent all the information contained in both columns three and four.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-	-			-	-	-	-	-	-	\	\	X	X		
				-	\	-	-	\	-	X	X	X	X		
-	-			\	-	\					X		-	\	-

	-	-	-	-	-	-	\	\	X	X
	-	\	-	-	\	-	X	X	X	X
							X			

Fig. 8 Serif deletion in the letter 'R'.

2.4.8 Horizontal Feature Integration

After a letter was compressed vertically into three rows, horizontal feature integration (or compression) was applied along each row. The state transition table in Figure 9A was used to select the most appropriate integrated shape that would describe a continuous row of individual U, D and H basic features. The programme took the present state of the feature integration process and compared it to the next basic feature to generate the new state. The negative sign in the table represented a terminal condition whereby the system stored the present state of the integration process. As well, the state transition table was reinitialized with the next basic feature upon encountering the negative sign.

Essentially, the table encoded a continuous row of U, D and H individual elements into one generalized feature. At least two (or any combination of) U, D or H consecutive features had to occur before any integrated feature could be assigned. Similarly, a minimum of two entries were necessary for a generalized Z.

A significant difference occurred with the V and C features. For this situation, one V or C entry was enough to consider them as the generalized features. This was the case because these two features had greater significance than any single U, D, H or Z element. Continuous horizontal row entries of either V or C were generalized into only one entry of the appropriate feature. Since C was a more selective occurrence than V, in a continuous string with both V and C entries, C was chosen over V to be representative of that row of features. The reason for this masking of V by C is illustrated by the letter 'c' in Figure 3. Column two was classified as a minor vertical; column three was the closure. Column two is expanding towards and developing the closure in column three and can be represented by the more significant feature, the closure.

The above discussion is summarized as follows:

for single entries:

$$Z = U = D = H = 0$$

$$V = V, C = C$$

for repeated horizontal entries:

$$X^n = (<)^{n-1} X \text{ where } X = (Z, U, D, H, V, C) \\ n = 2, 3, 4, \dots$$

for a row of V and C entries:

$$V^m \cdot C^n \cdot V^0 = (<)^{m + 0 + n - 1} C$$

The symbol (<) indicated the continuity across columns of the feature to its right. Integrated features other than the one listed above are also possible. These occurred with combinations of U, D, and H features (Section 2.4.10).

Figure 9B applies the state transition table to the example being developed. In both rows one and two a generalized H feature was stored rather than the D. This was the result of an additional check which was performed on the terminal feature state of the integration process to ensure that the most appropriate selection had been made. In this instance, a simple check on the first row such as noting that there were six H and two D elements disqualified a D entry. Instead an H entry was deemed most suitable. The same criterion was applied to the second row. To devise a larger and more complex state transition table to handle every possibility was found to be experimentally impractical. Instead, satisfactory feature integration performance was achieved with a two-step procedure. This was accomplished by using a transition table of moderate size in conjunction with a secondary verification routine.

		Next Basic Feature							
		D	H	U	Z	V	C		
Present State of Integration	IZ	ID	IH	IU	Z	V	C	initial zero	
	IU	H	HH	U	IZ	V	C	initial up	
	ID	D	LH	H	IZ	V	C	initial down	
	IH	LH	H	HH	IZ	V	C	initial horizontal	
	V	-V	-V	-V	-V	V	C	vertical	
	C	-C	-C	-C	-C	C	C	closure	
	LH	D	LH	PU	-H	-H	-H	low horizontal	
	HH	PD	HH	U	-H	-H	-H	high horizontal	
	D	D	D	CU	-D	-D	-D	down	
	U	CA	U	U	-U	-U	-U	up	
	CA	CA	CA	U	-CA	-CA	-CA	cap	
	CU	D	CU	CU	-CU	-CU	-CU	cup	
	PD	CA	PD	H	-H	-H	-H	peaked and downward	
	PU	H	PU	CU	-H	-H	-H	peaked and upward	
	H	LH	H	HH	-H	-H	-H	horizontal	
	Z	-Z	-Z	-Z	Z	-Z	-Z	zero	

Figure 9A

		1	2	3	4	5	6	7	8	9	10	11
			-	-	-	-	-	-	\	\	X	X
			-	\	-	-	\	-	X	X	X	X
									X			
Row 1			-	-	-	-	-	-	\	\	X	X
State		V	IH	H	H	H	H	H	LH	D	C	C
Store		V								H		C
Row 2			-	\	-	-	\	-	X	X	X	X
State		V	IH	LH	LH	LH	D	D	C	C	C	C
Store		V						H				C
Row 3									X			
State		V	IZ	Z	Z	Z	Z	Z	Z	C	C	IZ
Store		V							Z		C	

Figure 9B

Fig. 9 Horizontal feature integration for the letter 'R'.

2.4.9 Horizontal Compression

Once the integration process was completed, horizontal compression was applied to the character. The integrated features allowed for a compact representation of the letter. A feature matrix representation was obtained in the following manner. The columns were examined in pairs, starting on the right side and ending at column one. Adjacent horizontal quadrants in each pair of columns were checked for continuities. If there existed at least one continuity in each pair of horizontal quadrants then the two columns were compressed into one. This process continued until two adjacent integrated features were encountered. At this point the compressed column, which contained all

the features to the right of it, was saved. Column compression was then reinitialized with the next column. Referring to Figure 10, columns eleven and ten and nine through two were reduced to only one column each.

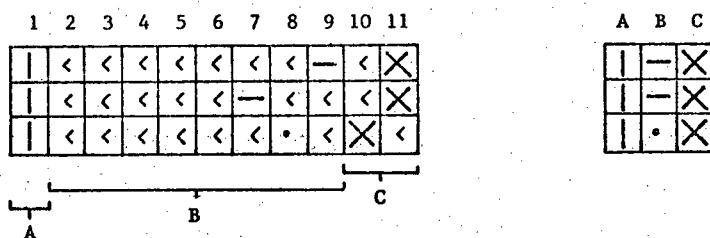


Fig. 10. Horizontal compression for the letter 'R'.

2.4.10 An example with Curvature

To demonstrate the full capabilities of the integration process, especially in the identification of curvatures, the letter 's' was selected. In Figure 11, the letter has been reduced to its three level form. Each row was integrated with the state transition table and then horizontally compressed. The cap (\cap) and cup (\cup) descriptions were well suited to represent the curvatures along the horizontal levels.

This example also illustrates the generality of the closure symbol (X). A closure can represent either divergence or convergence. Experimentally it was found that the distinction between a convergence and a divergence was not critical to the letter's identification. The assignment of an all inclusive closure (X) was a convenient simplification.

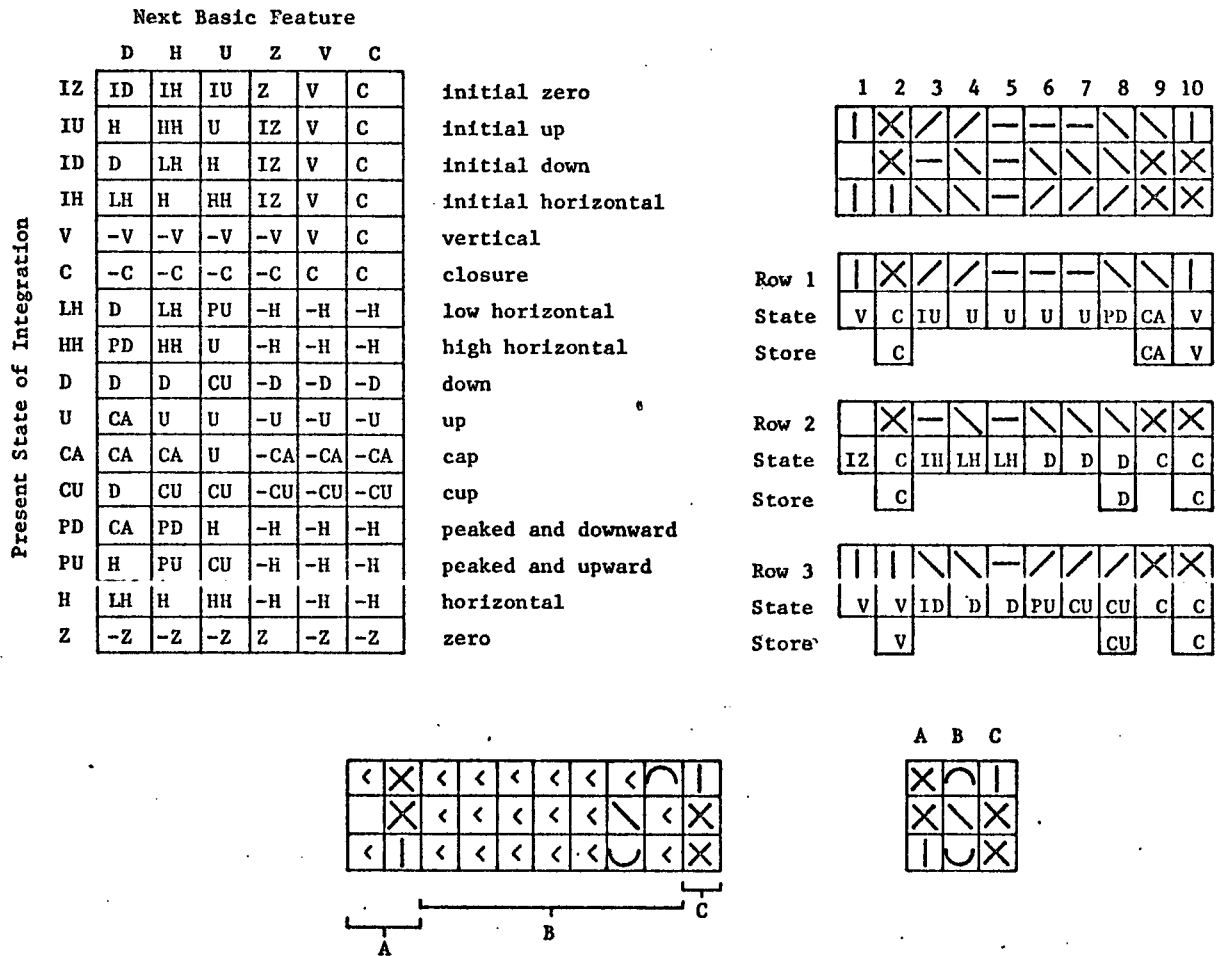


Fig. 11 Feature integration applied to a character with curvature with the letter 's' as an example.

2.5 Feature Extraction, Part 2

2.5.1 Characters with Diagonals

The letters 'A, k, K, v, V, w, W, x, X, y, Y,' are comprised basically of diagonals and require a separate approach for their identification. An alternate method of feature extraction is used. In Figure 12, the majority of the strings which form the diagonals are long enough to satisfy L , where $T < L < 3T$ and are identified as minor verticals; thus, the directional quality of the diagonals is lost.

The most descriptive features in these characters are their continuous edge transitions. By looking at the side of each character from the directions indicated by the arrows in Figure 12, the unobstructed (serifless) and unique diagonal features are clearly identified. Using the programme in Appendix E, each of the diagonal characters was viewed along its unique side(s): the top side for 'A'; the bottom side for 'v, V, w, W'; and, both the left and right sides for 'k, K, x, X, y, Y'. The specifications of a diagonal length (number of increments) and the up-down sequence for a particular side were used for this subgroup's identification. The location of this particular routine in the overall system is shown in Section 2.7.

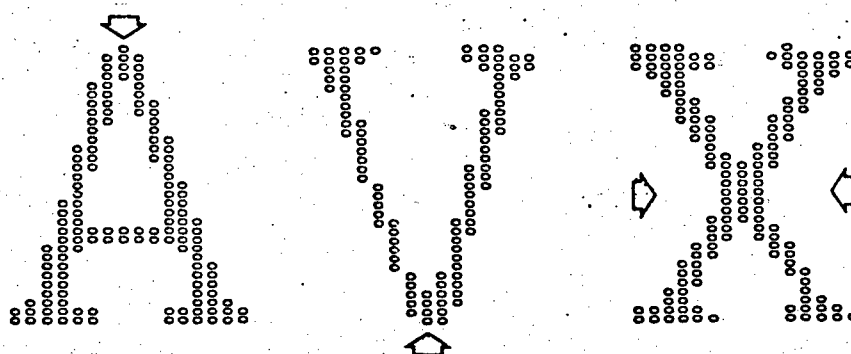


Fig. 12 Recognition of diagonal characters by selective edge examination.

2.5.2 Special Cases

The letters 'g, i, j, m' did not require the entire feature extraction process for their identification. In these special cases the identification was made once a unique measurement was satisfied. The distinctive characteristic of the letters 'i' and 'j' is the fact that they are dotted. To detect the dot all the columns were inclusively OR'ed together. An 'i' or 'j' was assumed if a break existed in the upper half of the OR'ed character. The length of the section below the dot was used to distinguish between the two; 'i' and 'j' without their dots have the same height as a small and tall character, respectively. The letter 'g' is the only four level character in the alphabet. The contents of levels two and three were always examined before vertical compression (Section 2.4.7) to three levels was applied. If both the second and third rows were filled with U, D, and H information then the unknown character was identified as a 'g'. The letter 'm' is unique because it contains three full height verticals. When this condition was satisfied in the string identification routine (Section 2.4.3) an 'm' was assigned.

These measurements were simple and effective. The savings realized in computational time and in the reduction of entries in the look-up table made this an attractive approach.

2.6 Classification

2.6.1 Introduction

If the cost of taking feature measurements is high, then sequential decision procedures (classification) should be applied. This is especially pertinent in the design of the recognition logic for a personal reading machine. Fu [8] emphasizes that a trade-off between the error and the number of features to be measured can be obtained by taking measurements sequentially and terminating the process when a sufficiently unique measurement is satisfied. The measurements must be ordered in such a way that the extracted features will cause the terminal decision as early as possible. These principles were incorporated in this classifier.

2.6.2 Code-Word Generation

A code-word was assigned to every horizontal level in the compressed character. This code-word does not contain continuities (<). Furthermore, only a Z bounded on both sides by features other than a continuity or another Z was stored in the code-word. In Figure 13, the three code-words for the letter 'R' are generated. Since there are seven possible values for a feature, a minimum of three bits is required to specify any one of these features. Thus in a twelve-bit code-word, up to four features can be packed.

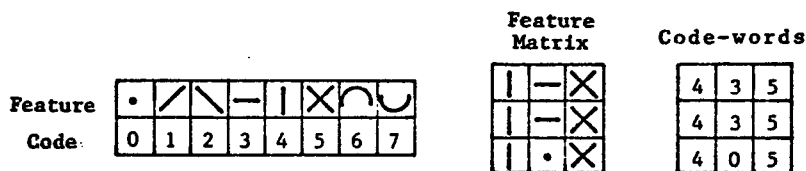


Fig. 13 The three code-words for the letter 'R'.

2.6.3 Subgroup Assignment

Rather than compare the code-words to all the alphabetic possibilities, a classification technique was implemented which confined the recognizer to search within a limited subgroup of characters. The separation of characters into classes was based on two physical properties of a letter. First, for a random selection of characters, 'a, B, c, E, G, H, N, O', the height is used to separate small letters from tall ones: 'a, c' and 'b, E, H, N, O'. Secondly, the number of full character height verticals, whether zero, one, or two, further subgroups the characters: 'a, c' (small letters) and 'G, O' (no verticals) and 'b, E' (one vertical) and 'H, N' (two verticals). In this manner four distinct classes are identified. The alphabet is subgrouped in Figure 14. The special cases and the two forms 'g' and 'g' are also included.

Small Letters	a	c	e	n	o	r	s	u	z
No Verticals	C	G	O	Q	S	Z			
One Vertical -lower case	b	d	f	g	h	l	p	q	t
-upper case	B	D	E	F	I	J	L	P	T
Two Verticals	H	M	N	U					
Diagonal Characters -lower case	k	v	w	x	y				
-upper case	A	K	V	W	X	Y			
Special Cases	g	i	j	m					

Fig. 14 Subgrouping the alphabet.

2.6.4 Identification

Three code-words were assigned to an unknown character, one for each of the three feature levels. The first code-word was compared to all the stored code-words for that level, in its appropriate subgroup. Each code-word was associated with a second twelve-bit word which stored all the possible characters with that code-word in common. Since there were less than twelve characters in any upper or lower case subgroup, each bit of the second twelve-bit word was associated with a specific character. The entry of a '1' or '0' for a particular bit indicated whether the code-word was common to that character or not. If the code-word was unique to only one character then the identification was made at that point. If the code-word was shared by several characters, then these possibilities were recoded and the process was repeated for the next feature level code-word. If no unique code-word matches were found after searching the three levels, then the identification was based on the character with the maximum number of code-word matches.

Let us assume that the unknown character in Figure 15 is the letter 'c'. The code-word for the first level is common to both 'c' and 's'. However, the second level in 'c' is unique to it and so the identification as 'c' is made. For the example of the letter 'R',

there are no unique code-words and thus the identification is based on the maximum number of code-word matches. This character recognition process is illustrated in Appendix B.

a	c	e	n	o	r	s	u	z
Subgroup a c e n o r s u z								
1st CW Matches 0 1 0 0 0 0 1 0 0								
2nd CW Matches 0 1 0 0 0 0 0 0 0 Unique								
3rd CW Matches 0 1 1 0 0 0 0 0 0								

Fig. 15 Character identification, the letter 'c' as an example.

2.7 Overall System Configuration

The entire system is represented in block diagram form in Figure 16. The pattern recognition process has been broken down into its various stages. Each stage is further identified as to the routines which are performed within it. The numbered routines indicate processes that occur during one major programme. The lettering refers to discrete routines that are classified according to their common function.

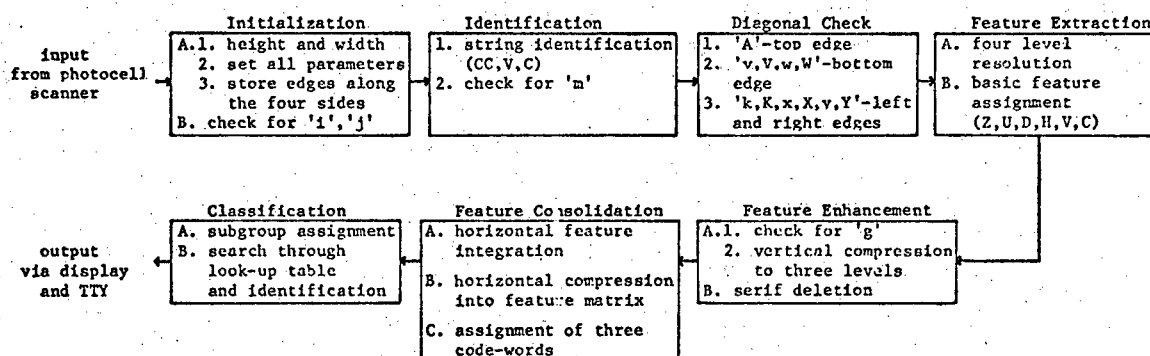


Fig. 16 Block diagram of the system.

III. TESTS AND RESULTS

3.1 Description of the Test Material

Three different typewriter fonts were used in the testing of the system: Hermes Ambassador (HA), IBM Delegate (D), and IBM Gothic (G). These fonts were selected because of their large print which ensured good machine resolution. About one-half of all typewriter fonts are this size. This thesis was typed with the IBM Prestige-Elite font; its size is typical of the smaller sized fonts. The three fonts shown in Figure 17, were selected because they cover the range of print styles, excluding italic and script, from the most stylized HA, through a moderately stylized D, to a bold and serifless G. The height of the tall and small letters is approximately the same for the three fonts although, the widths vary. The test material included all upper- and lower-case characters. Carbon ribbon was used for all typing.

Hermes Ambassador, pitch 10

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
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

IBM Delegate, pitch 10

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
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

IBM Gothic, pitch 12

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
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

Fig. 17 The three fonts used in the tests, reproduced in full size.

3.2 Tests and Results with Repeated Letters

The typewritten material was presented in the following manner. Each upper and lower case character was typed 10 times in a row. After scanning one row, the recognition results were printed out on the TTY. Testing was performed first with the HA look-up table on all three fonts. Then the system was trained on each of the D and G fonts, individual look-up tables were developed, and the tests were repeated on the newly trained fonts. The lighting intensity was adjusted only once at the beginning of each font's test for maximum resolution.

The following results were obtained from these tests :

font	HA	D	G
untrained	--	66.8	68.9
trained	88.2	85.2	89.0

Over 65% of the HA look-up table was general enough to be applicable to the other fonts without training. All three fonts achieved similar accuracies. Since the training procedure was simple and accomplished quickly, the system demonstrated good multifont adaptability. Confusion matrices have been tabulated for the test results and are included in Appendix A.

3.3 Tests and Results with Typewritten Passages

This test was performed to determine the recognition capability of the system on multicharacter text under more realistic conditions than in Section 3.2. A passage from a magazine article was typed, exclusive of punctuation, with the three fonts. The tests were performed with the trained look-up tables for each font. The system was not trained beforehand on this passage. The resolution (illumination) of the scanner was adjusted only once at the beginning of each font's

text. Character identifications were printed on the TTY at the end of each line.

The recognition results for a typewritten passage which contained 293 characters, are given below :

font	HA	D	G
results	89.9%	89.1%	87.7%

The passages as recognized by the system are included in Appendix A. In this test material characters rarely touched and no segmentation scheme [5] was implemented.

3.4 Sources of Error

In this section the major contributors to the error are identified. Typical errors are illustrated in Appendix D. From the confusion matrices which were tabulated for HA (Appendix A), the errors have been categorized as follows :

misinterpretation of a closure	25.4%
misclassified features (blotted or broken characters)	23.8%
missing a diagonal character	15.9%
indiscriminate serif deletion	12.7%
confusion among similar characters	11.1%
improper subgroup assignment	11.1%

A significant source of error was due to misinterpretation of a closure as a minor vertical and vice versa. A closure required certain constraints (Section 2.4.2) to be satisfied, but these measurements proved to be incomplete and a reformulation of the closure criteria is suggested (Section 4.2).

A broken string describing a minor vertical or a closure resulted in an error. A break in a full vertical could be checked for and corrected by counting the number of black cells in a column and, if the tally exceeded the height requirement for a full vertical, then one was assigned. However, with minor verticals and closures no such criterion existed to detect a break and thus an error results. Blotted features in a character also resulted in error. For example, a blotted area can change a short string into a longer one, thus wrongly changing its identity from a CC to a V.

The performance of the routine that checked for diagonal characters was troublesome. The problem was that the programming was inefficient to deal with all the variations encountered in following the incremental changes along an edge.

Certain groups of characters were difficult to distinguish. For example, the letters 'V' and 'Y' were difficult to separate. These misclassification errors were confined to one or two similar characters and usually occurred within the same subgroup. Thus misclassification was confined to possibly one of three characters rather than one in fifty-two.

The serif deletion routine, besides reducing the variability of the information, was sometimes indiscriminantly applied. For example, when serifs are eliminated from the letters 'l' and 'I' identical results are obtained.

Another type of error was improper subgroup assignment. In the case of such letters as 'J' and 'U', the height criterion for a full vertical was occasionally missed because of the rounding on the verticals' bottoms, thus confining the code-word search to the wrong subgroup.

3.5 Training Procedure

A simple training procedure was used. A row of ten identical characters was scanned. The operator then observed the computer displays for each character, which were similar to those in Appendix B, and noted the most common feature matrix representations. The new look-up table for either D or G was developed from the HA table by simple alteration of the existing code-words. In most cases not all three code-words for a character required modification. Rather, one or two code-words were in need of modification and, usually only part of the code-word. For example, with the characters 'C' (HA) and 'C' (G), only the code-word for the top level needed to be changed from closure-cap-vertical (feature code 564) to closure-cap (feature code 56). The code-words for the two remaining feature levels in this letter were identical. The system required approximately 4 hours to be trained to a new font.

3.6 Speed

The speed of character recognition was in the range of 376 ms. to 668 ms. This research was exploratory and, as such, no attempt was made to optimize the programming. The programme was written in such a way that each step of processing could be readily followed; as well, the ease of modification was important. Thus only one operation was performed on the data at a time and, as a result, much duplication of effort exists. The speed of the present system can be increased by approximately 150 ms. using more efficient programming techniques.

For a practical system, the separate routines can be incorporated together. For instance, string identification need occur only one column ahead of basic feature assignment, which in turn should be only one column ahead of the feature integration process. This type of system configuration should be able to achieve a reading rate of 10 char/sec.

3.7 System Storage Requirements

This OCR system was allocated 8K of memory on the PDP-12 computer. The recognition programme itself required only 4K of memory. The second 4K was occupied by the I/O and display routines as well as the storage of all the processed information. The programming was written in PDP-8 language and therefore the system can be implemented on a PDP-8 minicomputer.

Approximately 100 code-word entries were necessary for the look-up table of a particular font. About 40% of these code-words were unique entries; or, 60% of the code-words were shared by two or more characters.

3.8 System Flexibility

The ability of this system to accomodate poor image quality, distortion, and complex shapes was highly satisfactory. The system's flexibility is illustrated in Appendix D. A break in a character or a blotted printing could be accomodated. If damage was localized, then only one code-word was affected and, the other two could still be used for identification. Elongation and compression of a character was tolerated by the system. This type of rubber-sheet distortion of a plane is known as a topological mapping [7] and establishes the strongly topological nature of this scheme. The system's ability to handle carefully handprinted characters was noted. The scheme was also able to resolve complex shapes into more simplified structures.

IV. CONCLUSION

4.1 Summary

This thesis has developed an OCR system with a potential application to a personal reading machine for the blind. A topological feature extraction technique based on line and shape descriptors was implemented. A vertical photocell array, scanning at 60 wpm, was used to obtain a binary image. A continuous string of black cells was characterized by one of three possibilities: a single point to represent the centre cell of a short string; a vertical of appropriate length to characterize a long string; and, an area of closure to define the convergence or divergence of two branches within a letter. A four level resolution of column data was adopted and each column was compared to its next neighbour to generate continuous features across the columns. Feature enhancement, such as serif deletion and vertical compression to three levels, was implemented. A transition table integrated horizontal features along the three feature levels. After horizontal compression, a character's shape was described by any combination of seven basic features, usually within a 3 x 3 feature matrix. Each of the three horizontal levels was assigned a code-word.

The classifier confined the unknown character into a subgroup of the alphabet; this was based on the character's height and the number of full length verticals which it contained. Identification was made when a unique code-word in the letter was encountered; or, if the code-words were all common within the subgroup, then the recognition was on the basis of the character with the maximum number of code-word matches. A special technique was developed to identify characters with prominent diagonal features; the transition of an edge along a specific side of a character was used for this subgroup's recognition. Sequential classification was employed throughout the system so that identification was made once a sufficiently unique measure was satisfied.

The system was tested on upper- and lower-case characters of three different typewriter fonts which varied from a stylized to a serifless print. Results indicated that the level of recognition achieved was approximately 87% accuracy for a recognition speed of two characters per second. The training of the system to a new font was a simple procedure that required about four hours of effort. Approximately 100 code-word entries were stored in the look-up table.

An OCR method has been proposed which uses shape descriptors which in a sense parallel descriptions that a man would use. Transformations which will leave the letter's identity unaltered to man should likewise be used by the machine. The pilot studies indicate several advantages of this approach. Rubber-sheet changes of scale produce invariable results. Breaks in characters produced localized uncertainty which in most instances allowed the character to be correctly identified. The presence or absence of stylization, such as serifs, in the font seem to have little effect on identification. A few carefully drawn, hand-printed characters were tried with the OCR and were correctly identified.

4.2 Recommendations for Further Research

The detection of a closure as stated in Section 2.4.2 is weak. Referring to Figure 18, a more rigorous reformulation of the closure criteria is now presented: 1. the column (B) at which the closure occurs must have two strings in the adjacent column (C), that are located at its extremities; 2. to be certain of the identity of the two strings as actually two branches that converge to form the closure, the next column (D) over from these two strings must be checked for their continuation; 3. the column (A) preceeding the closure should be smaller than the closure string to indicate expansion towards the closure. Such a new criteria can correct the closure errors in the letters 'e, C, U' in Appendix C.

The classifier is underdeveloped. For example in Appendix C the feature matrices for both 'P' and 'O' are good character descriptions despite the misclassifications. The errors were confined to only one column; however, in the horizontal direction several rows were affected.

Thus more than one code-word was altered by the same error. We suggest that three code-words for the vertical direction in addition to the three code-words for the horizontal direction will produce a more powerful classifier. This will double the size of the present look-up table to about 200 entries, but this is still a modest size. An improvement to the system is anticipated because the six code-words for each character will result in a better distinction among characters, double the possibilities for unique code-words, and introduce a high degree of error tolerance. Such a classifier will be able to recognize the letters 'P' and 'O' in Appendix C from their vertical code-words.

From the discussion on errors (Section 3.7), it was noted that there exist small sets of characters which the basic system finds some difficulty in separating. An advanced system will have to include a small number of sorting routines whenever these latent ambiguities are encountered.

The routine that checks for diagonal characters has to be improved. Specifically, the ability to accomodate localized distortions when following an edge must be incorporated into the programming.

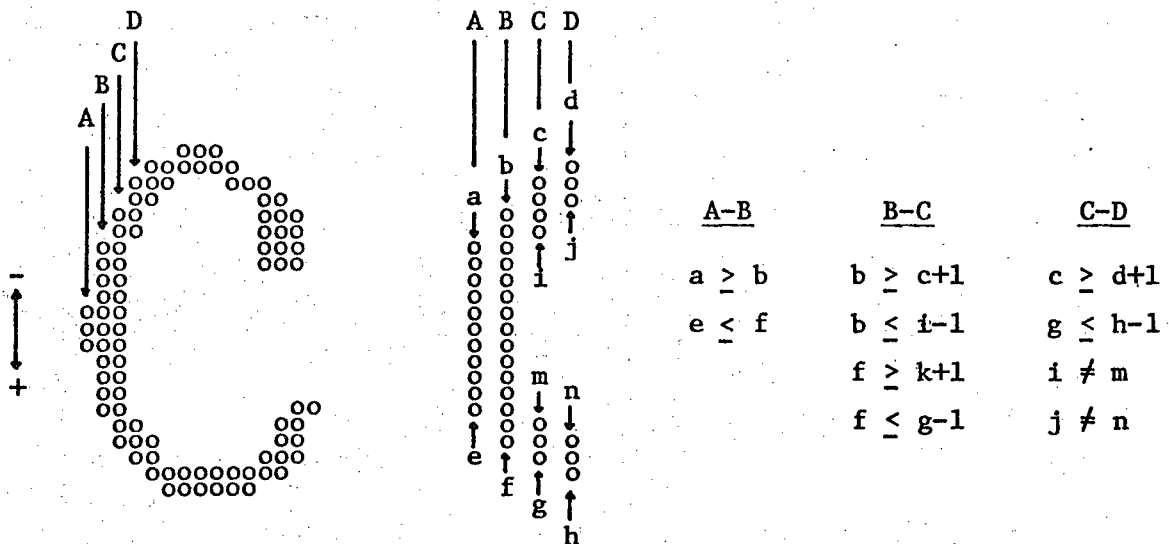


Fig. 18 Suggested closure criteria to replace the criteria in Figure 3.

REFERENCES

- [1] S.K. Abdali, "Feature Extraction Algorithms", Pattern Recognition, 3, pp. 3-23, April, 1971.
- [2] H.C. Andrews, Mathematical Techniques in Pattern Recognition, New York: Wiley, 1972.
- [3] M.P. Beddoes, "An Inexpensive Reading Instrument for the Blind", IEEE Trans. Bio-Med. Eng., Vol. BME-15, pp. 70-79, April, 1968.
- [4] J.C. Bliss, "A Relatively High-Resolution Reading Aid for the Blind", IEEE Trans. Man-Mach. Syst., Vol. MMS-10, pp. 1-9, March, 1969.
- [5] Clayden et al., "Letter Recognition and the Segmentation of Running Text", Information and Control, 9, pp. 246-264, 1966.
- [6] J.K. Clemens, "Optical Character Recognition for Reading Machine Applications", Ph.D. thesis, M.I.T., September, 1965.
- [7] R.O. Duda and P.E. Hart, Pattern Recognition and Scene Analysis, pp. 327-378. New York: Wiley, 1973.
- [8] M. Eden, "The Application of Character Recognition Techniques to the Development of reading machines for the Blind", Image Processing in Biological Science, ed. D.M. Ramsey, U.C.L.A. Press, pp. 35-56, 1968.
- [9] M. Eden and M. Halle, "Characterization of Cursive Handwriting", Proc. 4th London Symp. Inform. Theory, C. Chery ed., London: Butterworths, 1961.
- [10] K.S. Fu, Sequential Methods in Pattern Recognition and Machine Learning. New York: Academic Press, 1968.
- [11] K. Fukunaga, Introduction to Statistical Pattern Recognition. New York: Academic Press, 1972.
- [12] H. Genchi et al., "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting", Proc. IEEE, Vol. 56, no. 8, pp. 1292-1301, August, 1968.
- [13] F. Lee, "A Reading Machine: From Text to Speech", IEEE Trans. Audio and Electroacoustics, vol. 17, no. 4, pp. 275-282, December, 1969.
- [14] M.D. Levine, "Feature Extraction: A Survey", Proc. IEEE, vol. 57, no. 8, pp. 1391-1407, August, 1969.
- [15] H. Marko, "A Biological Approach to Pattern Recognition", IEEE Trans. Systems, Man, and Cybernetics, vol. SMC-4, no. 1, pp. 34-39, January, 1974.

- [16] S.J. Mason and J.K. Clemens, "Character recognition in an Experimental Reading Machine for the Blind", Recognizing Patterns, eds. P.A. Kolars and M. Eden, M.I.T. Press, Cambridge, Mass., pp. 156-167, May, 1968.
- [17] G. Nagy, "State of the Art in Pattern Recognition", Proc. IEEE, vol. 56, pp. 836-862, May, 1968.
- [18] P.W. Nye and J.C. Bliss, "Sensory Aids for the Blind: A Challenging Problem with Lessons for the Future", Proc. IEEE, vol. 58, no. 12, pp. 1878-1898, December, 1970.
- [19] A. Rosenfeld, "Figure Extraction", Automatic Interpretation and Classification of Images, ed. A. Grosselli, Academic Press, New York, pp. 137-153, 1969.
- [20] G.C. Smith and H.A. Mauch, "Summary Report on the Development of a Reading Machine for the Blind", Bull. Prosthetics Res., vol. BPR 10-12, pp. 243-271, Fall, 1969.
- [21] J.T. Tou, "Figure Extraction in Pattern Recognition", Pattern Recognition, vol. 1, pp. 3-11, July, 1968.
- [22] J.T. Tou and R.C. Gonzalez, "Recognition of Handwritten Characters by Topological Feature Extraction and Multilevel Categorization", IEEE Trans. Computers, vol. C-21, pp. 776-785, July, 1972.
- [23] G. Tzoumakis and J. Mylopoulos, "Some Results in Computational Topology", J. ACM 30, 3, pp. 439-455, July, 1973.
- [24] E.E. Triendl, "Skeletonization of Noisy Handdrawn Symbols Using Parallel Operations", Pattern Recognition, vol. 2, pp. 215-226, 1970.

APPENDIX A

Confusion Matrices

The first three pages contain the confusion matrices that were tabulated from tests with repeated characters under the conditions of constant illumination and trained look-up tables for each font. The top figure is the lower case results and the bottom is for the upper case. The question mark (?), which is one of the possibilities of identification, indicates that no code-word matches were found for this character. An 'X' represents complete recognition (10/10). A slash (/) through a number indicates that, if this is the lower-case matrix, then the confusion is with the upper-case character, and vice versa.

The recognition results for the typewritten passages are included on the fourth page of this appendix. An asterisk (*) is located above each error.

Identification

		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	?
Test Input	a	X																										
	b		9																			1						
	c			X																								
	d				X																							
	e					X																						
	f						X																					
	g							X																				
	h		2						8																			
	i									X																		
	j										X																	
	k		2									7														1		
	l		2										8															
	m													9												1		
	n	1													7							2						
	o																X											
	p																	X										
	q												1						8								1	
	r																			X								
	s				1																8		1					
	t																					X						
	u																						X					
	v																							X				
	w			1																1					8			
	x			3																						7		
	y																									1	9	
	z	1																									8	1

Identification

		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	?
Test Input	A																											
	B	X																										
	C		8										2															
	D			9															1									
	E				X																							
	F	2				7														1								
	G						8															2						
	H							8																			2	
	I					2			7																		1	
	J						1			9																		
	K	1									9																	
	L								2			8																
	M												7	1													2	
	N														X													
	O															X												
	P												2					8										
	Q			1																7	1						1	
	R					1														1	8							
	S																					X						
	T													2									7					
	U																							7	1			2
	V																								X			
	W																									X		
	X																										8	2
	Y																											
	Z																						1				9	7

Confusion matrices for Hermes Ambassador font, on which the system was developed.

Identification

	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	?
a	7																								2	1	
b		9																								1	
c			9	1																							
d		1		9																							
e			1		8											1											
f						X																					
g							8													2							
h		1						9																			
i									X																		
j										X																	
k		1									9															1	
l		3										7															
m													8	1												1	
n														9								1					
o																X											
p		1															9										
q																		X									
r																			X								
s																				7				2		1	
t					1						1										8						
u																						X					
v																							X				
w																								X			
x			1																						9		
y																									7	6	1
z		1																									8

Identification

	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	?
A																											
B		8																2									
C			X																								
D				X																							
E					7	2														1							
F						8												2									
G			1				9																				
H		1						9																			
I									6											4							
J				1						8																	1
K											9							1									
L		1										6								2							1
M													9	1													
N															X												
O			2													8											
P						1												9									
Q																			9								1
R																				7							
S											3										9						
T			1																			5					
U																						9					1
V																							X				
W																								X			
X																									9		1
Y																							6		4		
Z																										7	3

Confusion matrices for IBM Delegate font (trained).

Identification

	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	?
a	8																				1			1			
b		X																									
c			9															1									
d				8																2							
e					X																						
f						X																					
g							X																				
h								X																			
i									X																		
j										X																	
k		2									8																
l												X															
m													8													2	
n														9			1										
o															28												
p																	9	7									
q				1														9									
r			2																7							1	
s																				5		1		4			
t																					X						
u																						X					
v																							X				
w																								X			
x																									X		
y																										X	
z																										X	

Identification

	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	?
A	1																										
B		9						1																			
C			8																		1					1	
D	1			8																						1	
E				1	9																						
F						X																					
G						1	9																				
H		2							8																		
I										8		2															
J											X																
K												X															
L													X														
M														7	2											1	
N															9											1	
O			1				1									8											
P																	X										
Q																		9								1	
R												4							6								
S			1																	9							
T												2									8						
U																						8				2	
V																							X				
W																								X			
X																									9	1	
Y																					7				3		
Z																										X	

Confusion matrices for IBM Gothic font (trained).

After our documentary The Fifth Estate was aired by the CBC
 Kevin O'Neill who was revealed as director of Canada's largest
 intelligence agency the Communications Branch of the National
 Research Council CBNRC was asked by a newspaper reporter who
 it was that he reports to O'Neill told him that he had spent
 most of the morning trying to determine just that

After our documentary The Fifth Estate was aired by the CBC
 Kevin O'Neill who was revealed as director of Canada's largest
 intelligence agency the Communications Branch of the National
 Research Council CBNRC was asked by a newspaper reporter who
 it was that he reports to O'Neill told him that he had spent
 most of the morning trying to determine just that

After our documentary The Fifth Estate was aired by the CBC
 Kevin O'Neill who was revealed as director of Canada's largest
 intelligence agency the Communications Branch of the National
 Research Council CBNRC was asked by a newspaper reporter who
 it was that he reports to O'Neill told him that he had spent
 most of the morning trying to determine just that

Reading results from a continuous passage. Top to bottom: Hermes Ambassador, IBM Delegate,
 and IBM Gothic fonts.

After our documentary The Fifth Estate was aired by the CBC
 Kevin O'Neill who was revealed as director of Canada's largest
 intelligence agency the Communications Branch of the National
 Research Council CBNRC was asked by a newspaper reporter who
 it was that he reports to O'Neill told him that he had spent
 most of the morning trying to determine just that

After our documentary The Fifth Estate was aired by the CBC
 Kevin O'Neill who was revealed as director of Canada's largest
 intelligence agency the Communications Branch of the National
 Research Council CBNRC was asked by a newspaper reporter who
 it was that he reports to O'Neill told him that he had spent
 most of the morning trying to determine just that

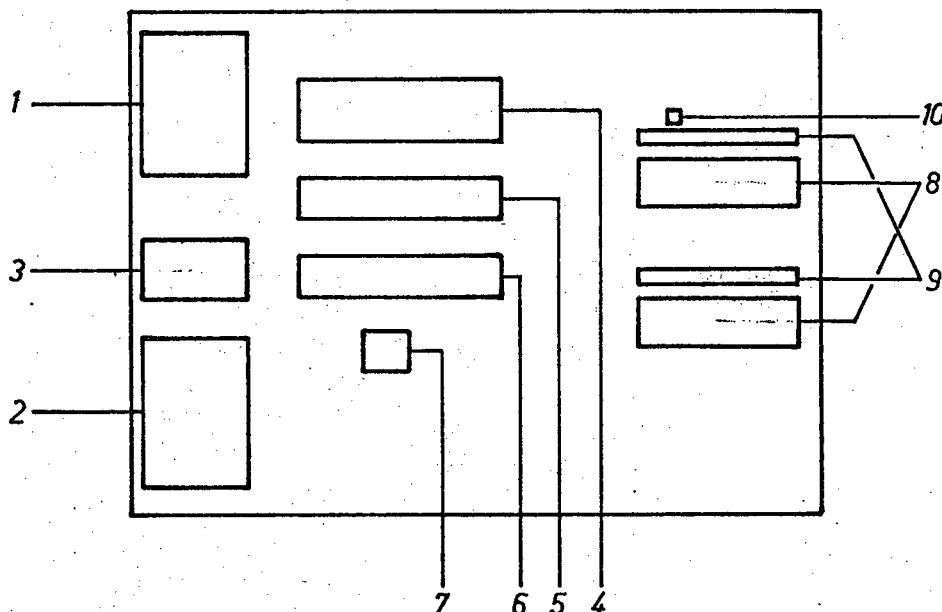
After our documentary The Fifth Estate was aired by the CBC
 Kevin O'Neill who was revealed as director of Canada's largest
 intelligence agency the Communications Branch of the National
 Research Council CBNRC was asked by a newspaper reporter who
 it was that he reports to O'Neill told him that he had spent
 most of the morning trying to determine just that

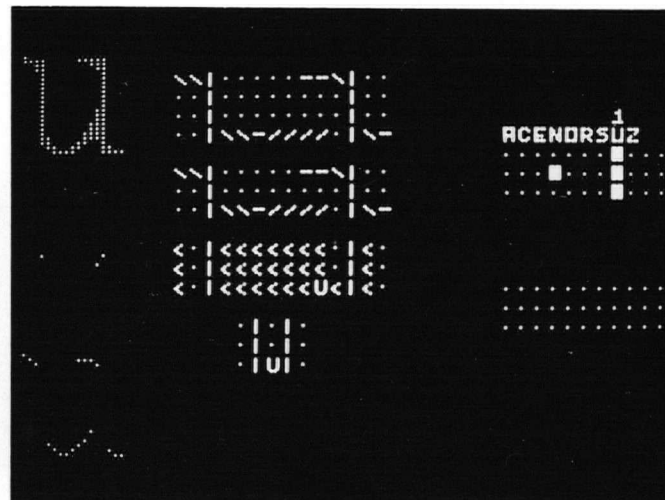
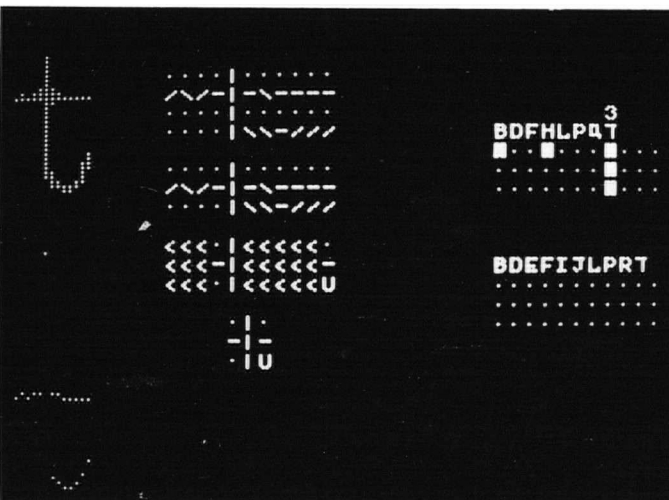
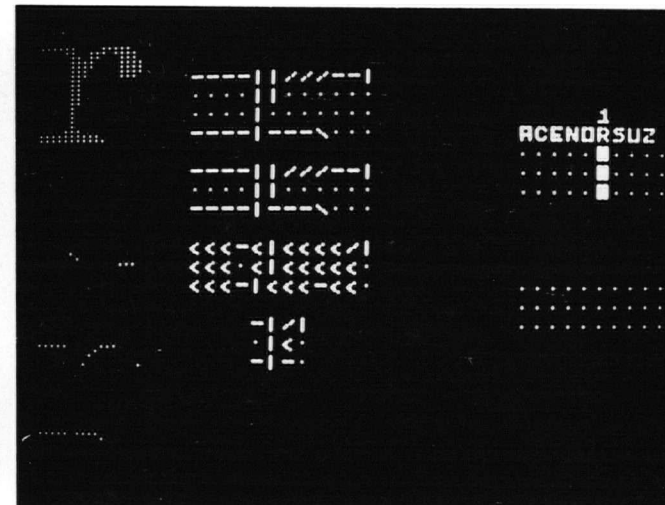
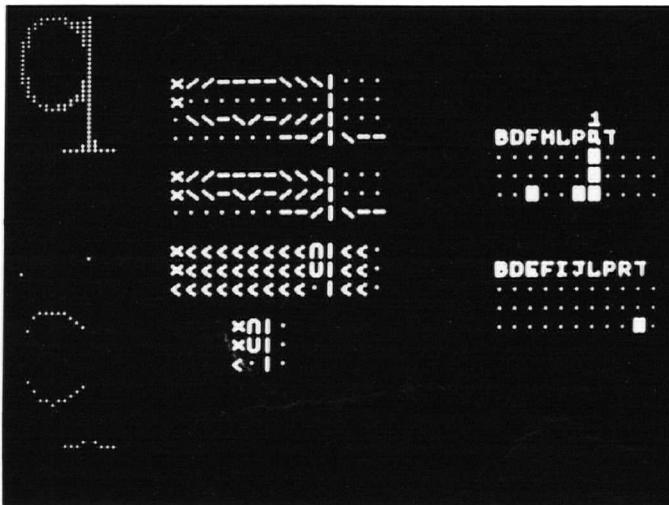
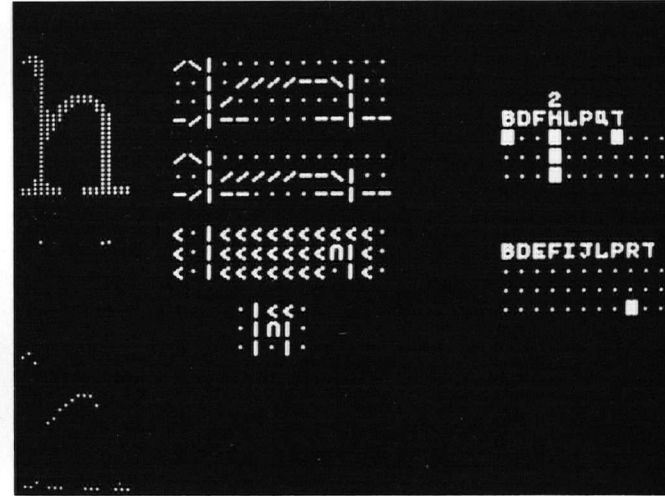
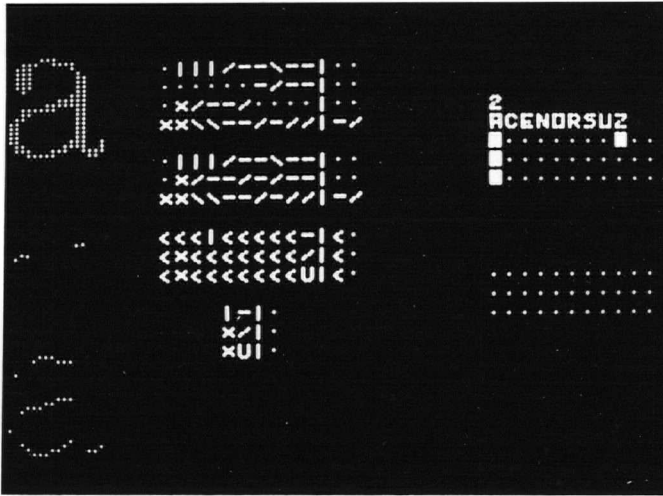
APPENDIX B

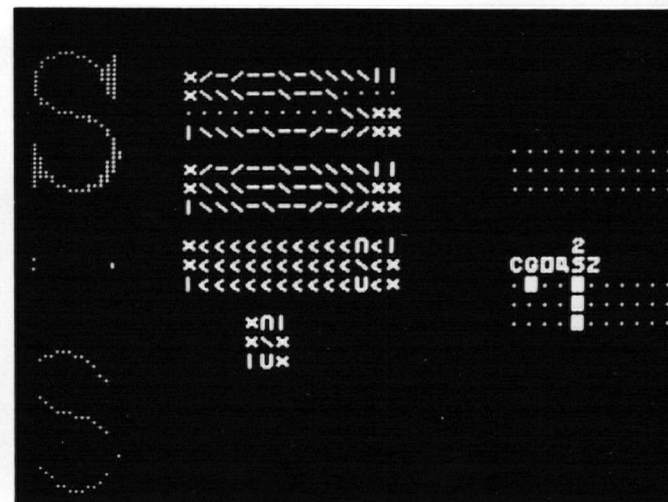
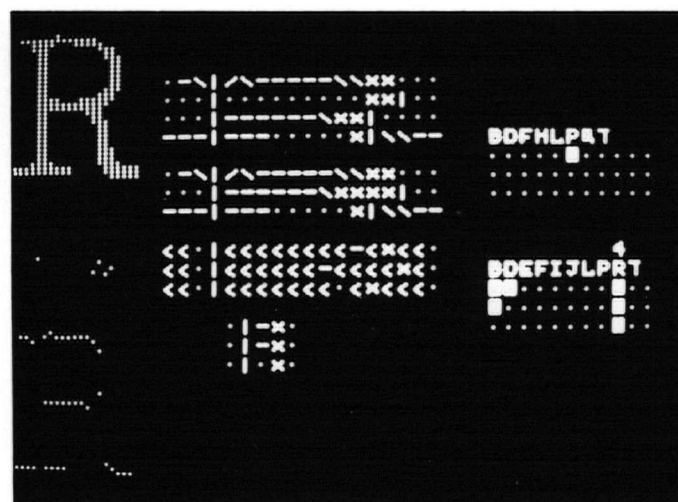
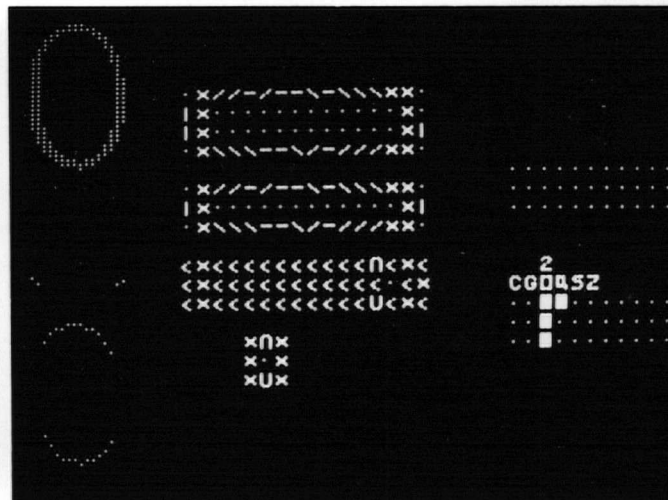
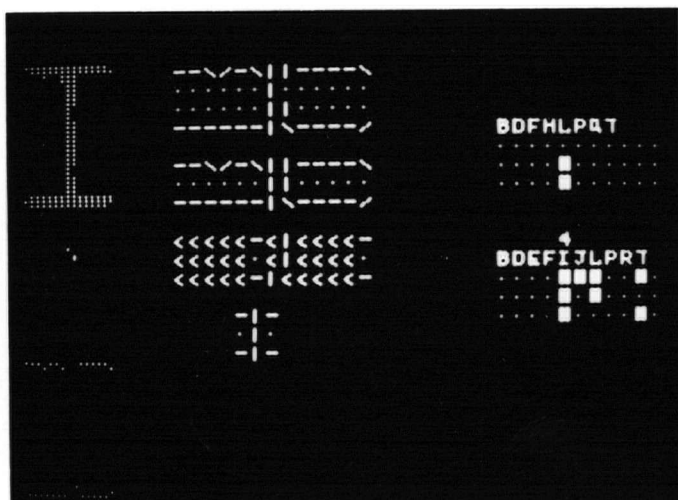
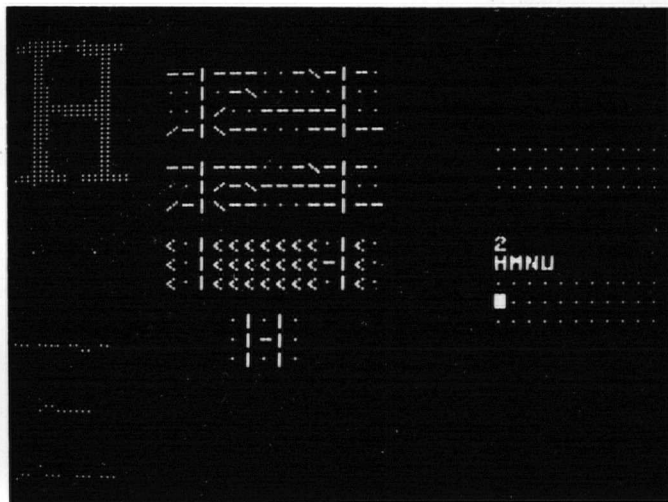
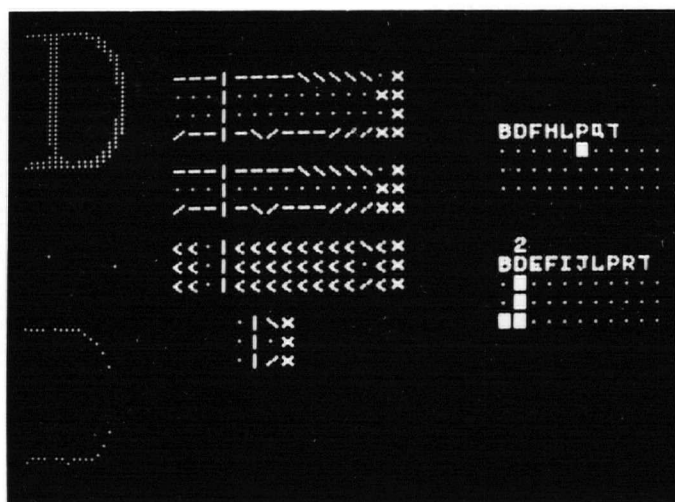
Examples of the Character Recognition Process

The numbered statements refer to areas within each picture, as illustrated in the figure at the bottom of the page.

1. input to the system from the photocell scanner.
2. the points represent the centre cells of the short strings.
3. the points represent verticals and closures and are located directly above the column to which they belong.
4. the character is quantized into four levels and each string is assigned its appropriate feature.
5. vertical compression to three levels; serif deletion.
6. horizontal feature integration along each level.
7. final feature matrix; a code-word is assigned for each level.
8. the three levels correspond to the three code-words; markers identify code-word matches.
9. subgroup to which the classification is confined; top and bottom areas represent the lower and upper case characters, respectively.
10. a number above a character indicates the identity of the unknown character; the numbers one through three specify at what feature level the identification is made; a four indicates that the identification is based on the character with the most code-word matches.







APPENDIX C

Examples of Errors

Top Left: The closure on the right side was missed for this character because the string at which the two branches converged was too short.

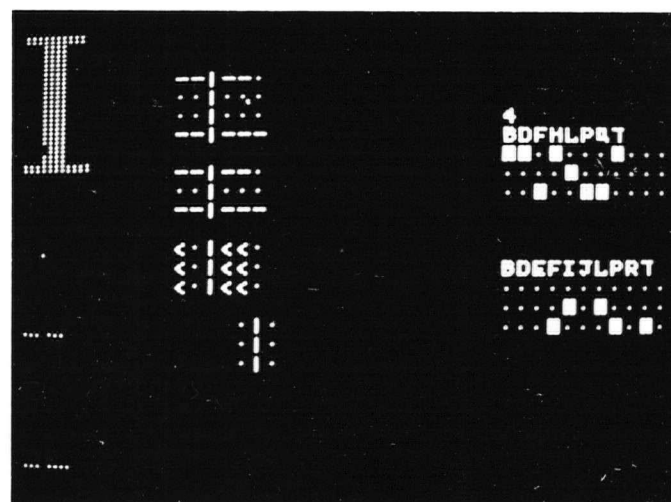
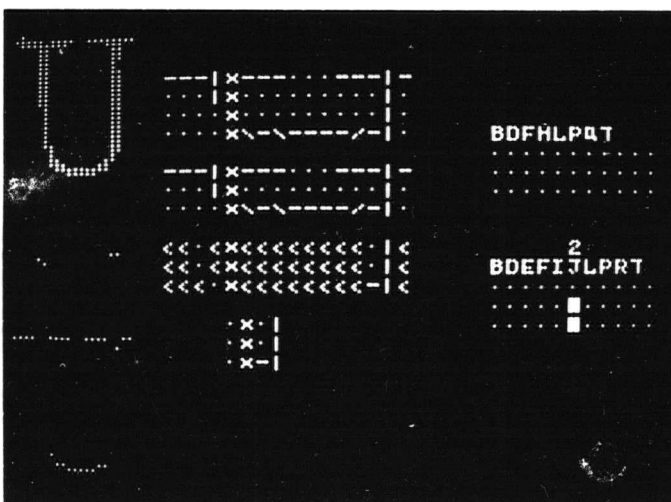
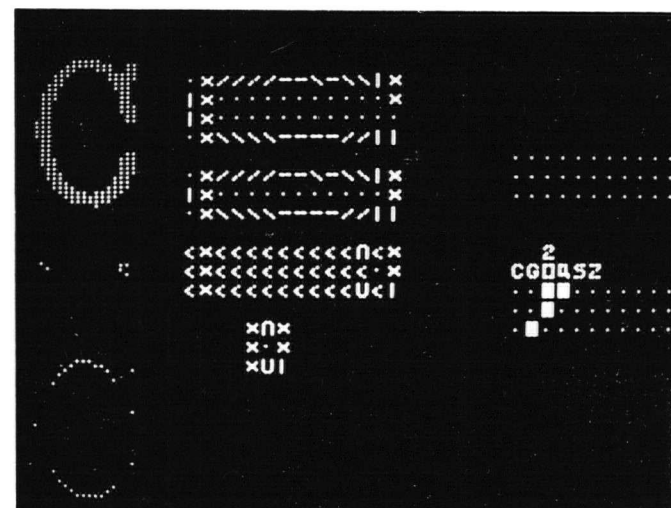
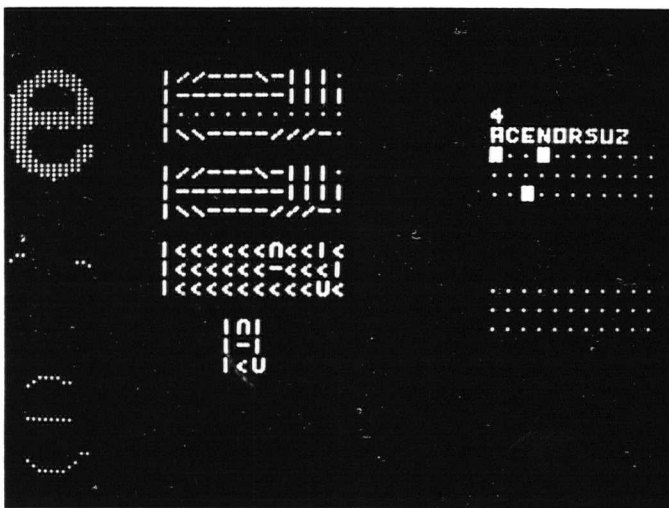
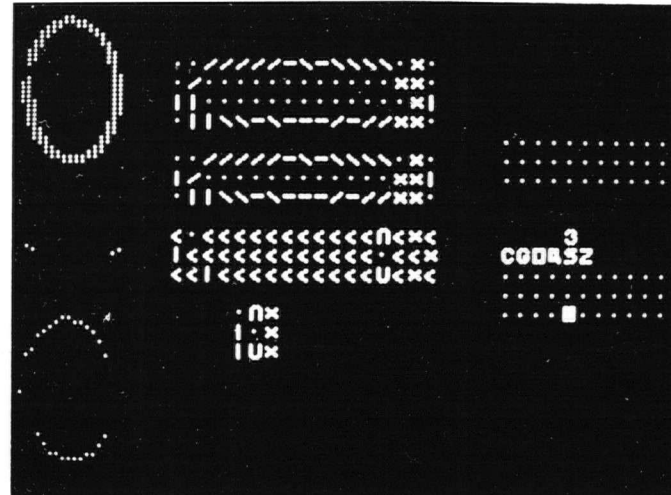
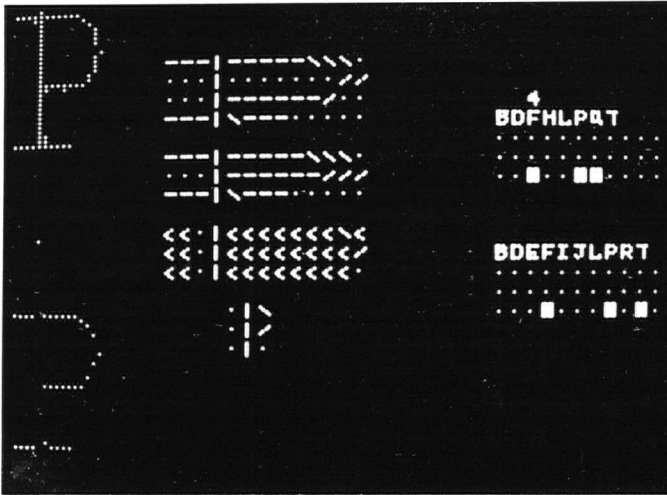
Top Right: A broken closure on the right side resulted in misclassification of this string as a minor vertical.

Centre Left: The closure on the right side was missing because the convergence criterion was not satisfied. Specifically, the last string in the lower branch ended before the string performing the actual closure. A more comprehensive test for a closure must replace the rigid, and in this case indiscriminant, criteria presently used.

Centre Right: The upper right corner was assumed to be the point of divergence for two branches. A check for branch continuity would have detected this error.

Bottom Left: The long string on the character's left side failed to satisfy the criterion for a full vertical. Any string other than a full vertical or a very short string was tested for closure. In this case, when the full vertical was missed, the string satisfied the closure criterion. Improper subgroup assignment occurred as a consequence of the lost vertical.

Bottom Right: The feature matrix was ambiguous for this letter because the deletion of the serifs resulted in a loss of features. By referring back to the character's structure before serif deletion, the ambiguity can be resolved. For this character, the system first searched through all the code-words. When no unique entries were found, a search was initiated for the character with the most feature level matches. When two or more equally likely possibilities existed, the programme chose the first letter in the series. For this example, with multiple choices, the character 'b' was selected.



APPENDIX D

Examples of System Flexibility

Top Left: High lighting intensity produced a break at the bottom of the character. Because the damage was localized to only one level, identification was still possible with the code words from the two remaining feature levels.

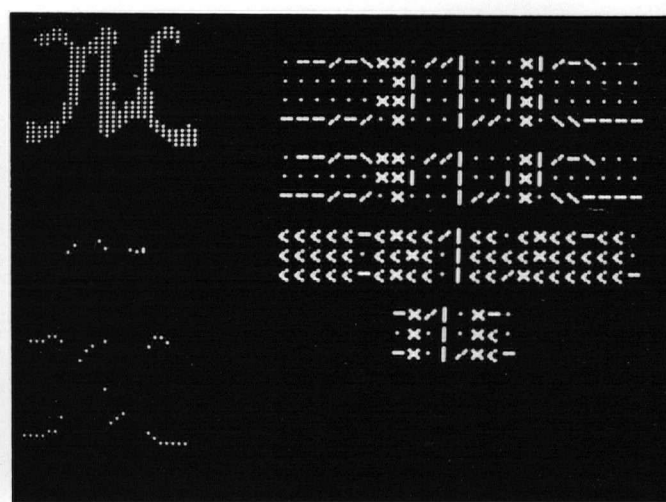
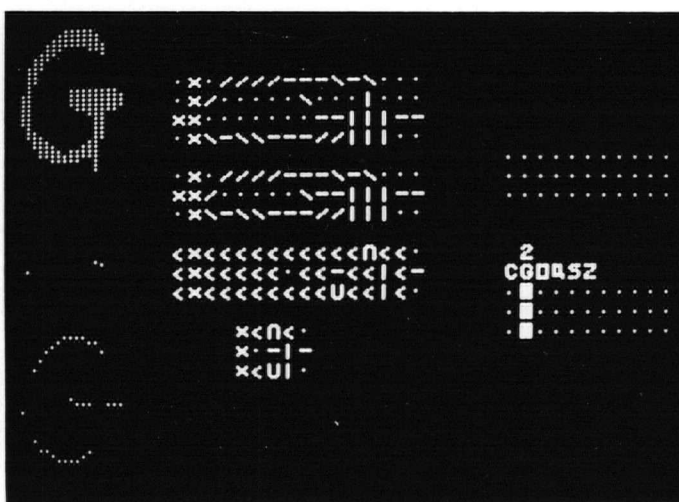
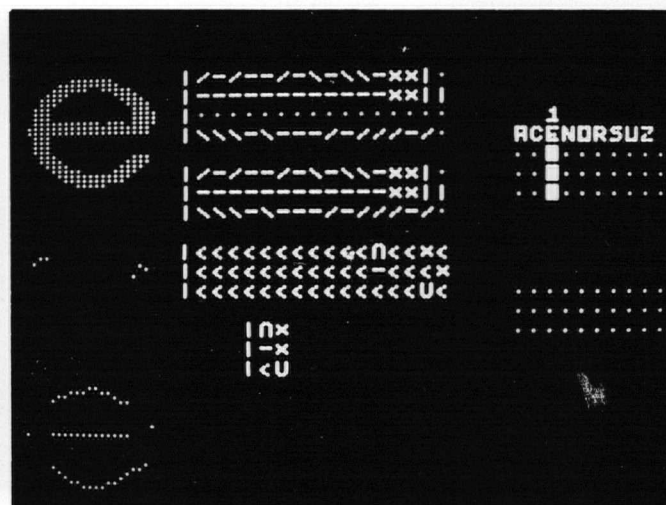
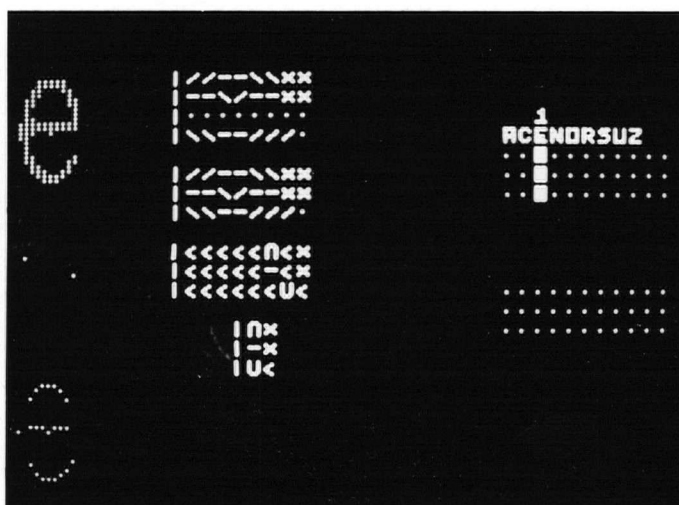
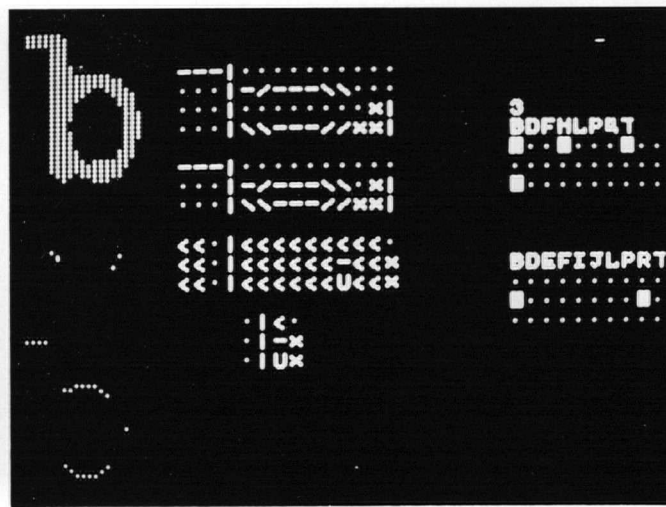
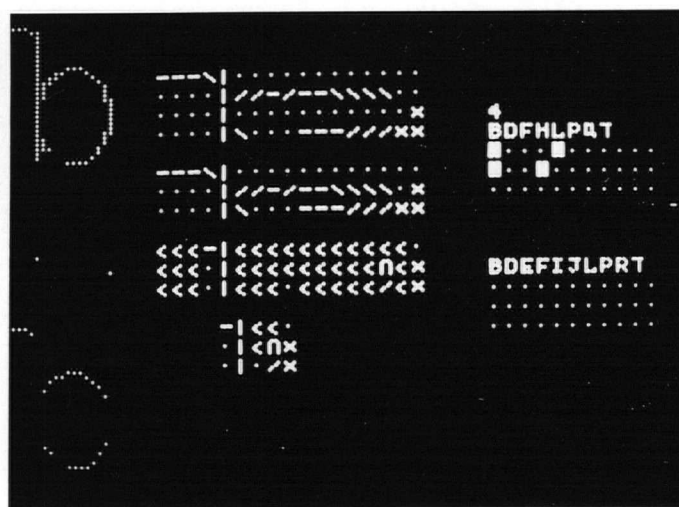
Top Right: Low lighting intensity produced a blotted image. The second level in the feature matrix, although still plausible, was not characteristic of this letter under normal conditions.

Centre Left: Correct classification resulted when this letter was compressed by 1/3.

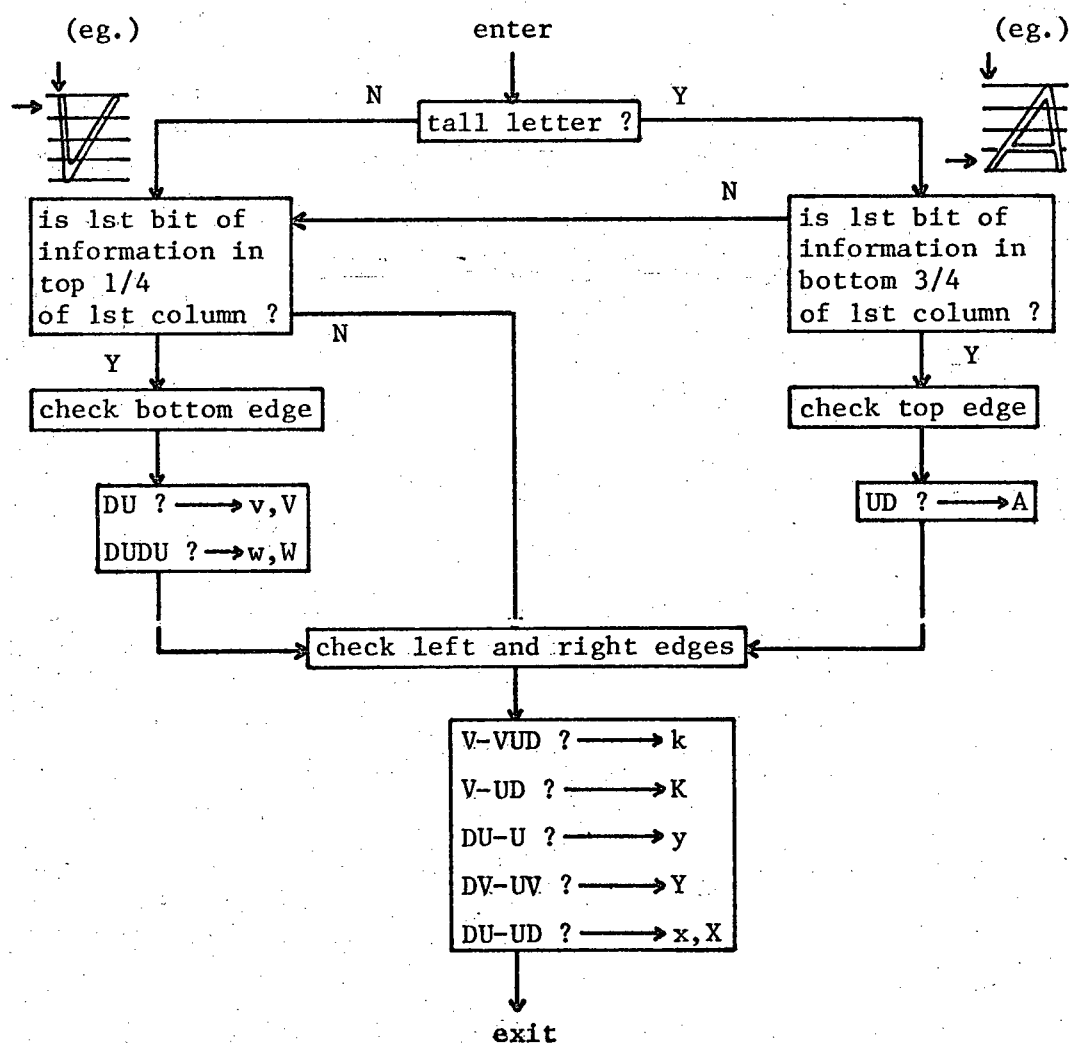
Centre Right: Elongation by 1/3 of the same character produced similar results as above.

Bottom Left: This carefully handprinted character was correctly identified.


Bottom Right: Good feature simplification was obtained on this handwritten character from the Ukrainian alphabet.



APPENDIX E

Basic Flowchart for the Diagonal Checking Routine

Notation:

vertical  up down OR V-UD