

ACKNOWLEDGEMENTS

My deepest sincere thanks go to my advisor, Dr. C.Y. Suen, without whose advice, this thesis would not have existed. His guidance, suggestions, careful reading of drafts and helpful comments have been an invaluable aid throughout both the research and the writing of this thesis.

I am very grateful to Mr. R. Shinghal for the preparation of frequency diagrams and probability distribution tables used in this thesis. Special thanks go to the operators in the computer centre of Concordia University for their patience and cooperation to run numerous computer jobs for this thesis.

The formatting of my typescript was done by using the 'TYPESET' -- a text formatting program implemented in the computer system of the university.

This study is supported by a research grant from the National Research Council of Canada.

TABLE OF CONTENTS

	Page
SIGNATURE PAGE.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES.....	vii
LIST OF TABLES.....	viii
CHAPTER	
1. INTRODUCTION	1
1.1 Background in Character Recognition.....	1
1.2 Constraints Imposed on Writing.....	3
1.3 Data Arrangement.....	7
1.4 Scope of the Thesis.....	12
2. SELECTION OF STANDARD CHARACTERS	14
2.1 Introduction.....	14
2.2 Dispersion Factor.....	15
2.3 Representation of Dispersion Factors.....	19
2.4 Selection of Characters.....	21
2.5 Results.....	25
3. TEST OF PERFORMANCE OF THE CHOSEN CHARACTERS	29
3.1 Introduction.....	29
3.2 N-tuple Feature Extraction Method.....	30

CHAPTER	Page
3.3 The 'characteristic loci' Algorithm.....	31
3.4 Recognition Scheme.....	33
3.5 Results and Comparison.....	36
4. DEVELOPMENT OF A NEW FEATURE EXTRACTION ALGORITHM	42
4.1 Introduction.....	42
4.2 Description of Features.....	43
4.3 Feature Detection.....	45
4.4 Recognition with the New Feature Set.....	49
5. CONCLUSION AND SUGGESTIONS FOR FURTHER STUDY	57
5.1 Conclusion.....	57
5.2 Comments and Suggestions for Further Study.....	58
REFERENCES.....	62
APPENDIX: Aspects related to Computation.....	68

LIST OF FIGURES

Figure		Page
1-1	Guidance schemes for constrained writing	5
1-2	Alphanumeric character models used in this study	8-9
1-3	Digitized character samples	11
2-1	The frequency diagram of model 124 ('2')	17
2-2	The probability distribution table of model 124 and the associated templates	18
2-3	The D-line graph of character models of '2'	20
2-4 (a)	The chosen character set for right-handers	23
2-4 (b)	The chosen character set for left-handers	23
2-4 (c)	The final chosen character set	24
2-4 (d)	The ANSI character set	24
2-5	The D-line graph of models 'u' and 'w'	27
3-1	Illustration of an 5-tuple and its state	32
3-2	Examples of coding in 'characteristic loci' algorithm	32
4-1	The new feature set	44
4-2	Illustration of a bar feature	47
4-3	Illustration of a diagonal feature	47
4-4	Samples of misrecognized character	53
4-5	Samples of recognized character	54

LIST OF TABLES

Table		Page
3-1	Confusion tables from N-tuple (n=2) method.	39
3-2	Confuseion tables from N-tuple (n 3) method	40
3-3	Confusion tables from 'characteristic loci' algorithm	41
4-1	Confusion tables from the new system	51
4-2	Feature distribution of numerals	56

CHAPTER 1

INTRODUCTION

1.1 Background in Character Recognition

Pattern recognition plays an important role in the application of computer science. Character recognition is one of the most interesting and popular subjects developed in this field. Research work was carried out by Dineen as early as in 1955 [1].

'Character' in this text is restricted to mean the English alphabet and numerals only. Two kinds of characters are going to be considered: machine printed and handprinted. Recognition of machine-printed characters is less complicated because of invariance. A high degree of reliability makes these recognition machines commercially viable. Some machines in the market can read more than one font of type-written characters, such as the IBM 1975 optical page reader[2], GRAFIX I system[3].

The difficulties of the recognition of handwritten characters are obvious due to its boundless variability in writing (we ignore other factors[4,5], such as writing instruments, paper quality etc.). Handwritten characters can also be divided into two groups: i) cursive script, and ii) block capital letter (handprint).

Because of its utmost difficulties, automatic recognition of cursive script is not appealing to researchers. Although there have been some attempts to simplify and solve the problem, it is still in its preliminary experimental stage. Another barrier to this problem is that the weight between the usefulness and the cost of developing such a recognition system is somewhat out of balance. A detailed review in these works is discussed in [6,7].

Most of the research work in character recognition is concerned with the recognition of handprinted alphanumeric characters because of its comparatively low cost and less complexity. A considerable amount of research is confined to numerals only because of its wide applications, such as letter sorting [8].

A general handprinted character recognition system consists of three parts: preprocessing, feature extraction and decision. The preprocessing unit is used to 'clean' input characters so that they are in the most suitable condition for recognition, for example, normalization and segmentation. Information from the raw image of a character is scattered and there is a lot of redundancy. Extracting all unique and essential information in a concise way is the function of the feature extraction unit. M.D. Levine described this field in detail [9]. The decision algorithm is the heart of a recognition system. Different approaches

have been used to attempt complete discrimination of character classes with information obtained from feature extraction. Some remarkable results with recognition rate over 95 % have been reported, such as the deterministic system developed by Caskey and Coates[11], Tou and Gonzalez's multi-level recognition system[10]. However, a reliable perfect recognition system has not been established yet.

Another branch in this field is real-time character recognition [12,13]. Recognizing characters in an on-line system can provide valuable information at the time of input, such as the speed, direction and sequence of the strokes[14]. It may be too slow to recognize handprinted documents in this way, but it has definite advantage in such situations as recognition of signatures, identification of graphic symbols, etc..

1.2 Constraints Imposed on Writing

Imposing constraints on the writing format of handprinted characters can limit the variability of a character in writing and therefore simplify the recognition problem. Obviously, the more constraints imposed on the characters, the easier for the recognition process, but on the other hand, there is less freedom of writing. It will go back to the situation of machine-printed characters when

the freedom of writing is lost.

To reduce the problems arising from size or spacing variation of writing, handprinted characters are usually written within preprinted guidelines[23], or just using coding sheet as guidance[15]. Boxes of various sizes are also used[5]. This kind of constraints has no actual effect on the characters themselves. It is very helpful for unifying input characters.

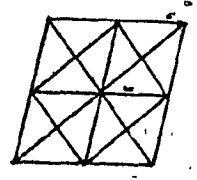
High constraints can be imposed by introducing preprinted guidance scheme as the writing aid. Various schemes have been designed to minimize the variability of writing of a character. Dimond[16] proposed a guidance scheme of 2 dots (Fig.1-1(a)). The Postal Giro Service in the Netherlands also tested this scheme together with some others[17]. Holt[18] introduced a vertical red line (Fig.1-1(b)) as the writing guidance for his handprint reader. One of the most constrained guidance schemes was developed by Lin and Scully[19]. They used a 20-feature guidance to form a highly constrained handprinted font(fig.1-1(c)). The recognition rate of this system is as high as 99.4%. There are some other special schemes designed and tested by Suen[20] and Apsey[21]. Some of these schemes are listed in Fig.1-1(d). There is no doubt that high recognition rate would be obtained by using highly constrained guidance for writing, but the consequent constrained handprinted font is usually unrealistic. Apsey[21] has shown that the time

: 1 2 3 4 5 6 7 8 9 0

(a)

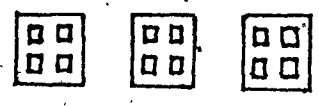
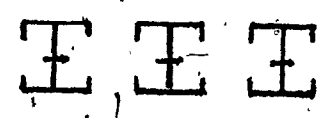
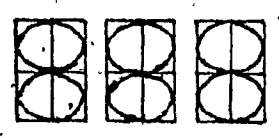
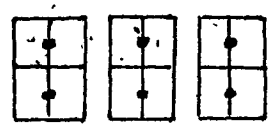
1 2 3 4 5 6 7 8 9 0

(b)



1 2 3 4 5 6 7 8 9 0

(c)



(d)

Fig.1-1 Guidance schemes for constrained writing

needed to write this kind of constrained font, as a data entry medium is much longer than other data entry methods.

Another way of imposing constraints on writing is to fix the format of writing of a character without using any special guidance scheme. For example, Kuhl[22] placed topological constraints on the character writing for character classification. In order to have a better set of disjointed characters, Munson[15], created his database by printing the numeral '0' with a diagonal slash, letter 'z' with a midline slash, crossbars on letter 'i', no serifs in numeral '1'. This kind of constraints, which is regarded as a low constraint, reduces the variation of writing by using writing models which are usually chosen from characters in common use. Selection of character model becomes an important concern in this method. The most intensive research in this matter is undertaken by the American National Standards Institute. A standard set of characters for handprinting has been developed[23,24]. The set of alphanumeric characters is listed in Fig.2-4(d).

Choosing character model depends on many variables. The place where the characters are used is a major factor. Different people may have different style of writing, or different character font. For example, researchers in Japan have their own set of character models which is adapted from the ANSI character set for their convenience[25], Canadian Standard Association is developing its own standard set of

characters[26]. The structure of a character is another important factor to be considered, because it determines the writing performance of the character, such as speed, degree of confusion with other characters, etc.. Selection also depends heavily on what sort of recognition scheme is being used for recognition, or the state-of-the-art of the OCR. Different recognition schemes may require different features on the character models.

Besides constraint purpose, the expanding computer society also urgently needs a standard character set for man-to-machine and man-to-man (such as programmer to key-punch operator) communications. In this thesis, a new approach for character selection will be attempted.

1.3 Data Arrangement

The database used in this study was prepared by Suen[5,27]. It comprises 168 different alphanumeric handprinted character models selected from a detailed examination of more than 30 writing systems used in North America (including all ANSI characters). The geometric configuration and stroke sequence of these models are shown in fig.1-2. Note that there are some models which have the same geometric structure but formed by different stroke sequences, e.g. characters E(23) to E(30). Different sequence of stroke(s) may cause different speed and degree.

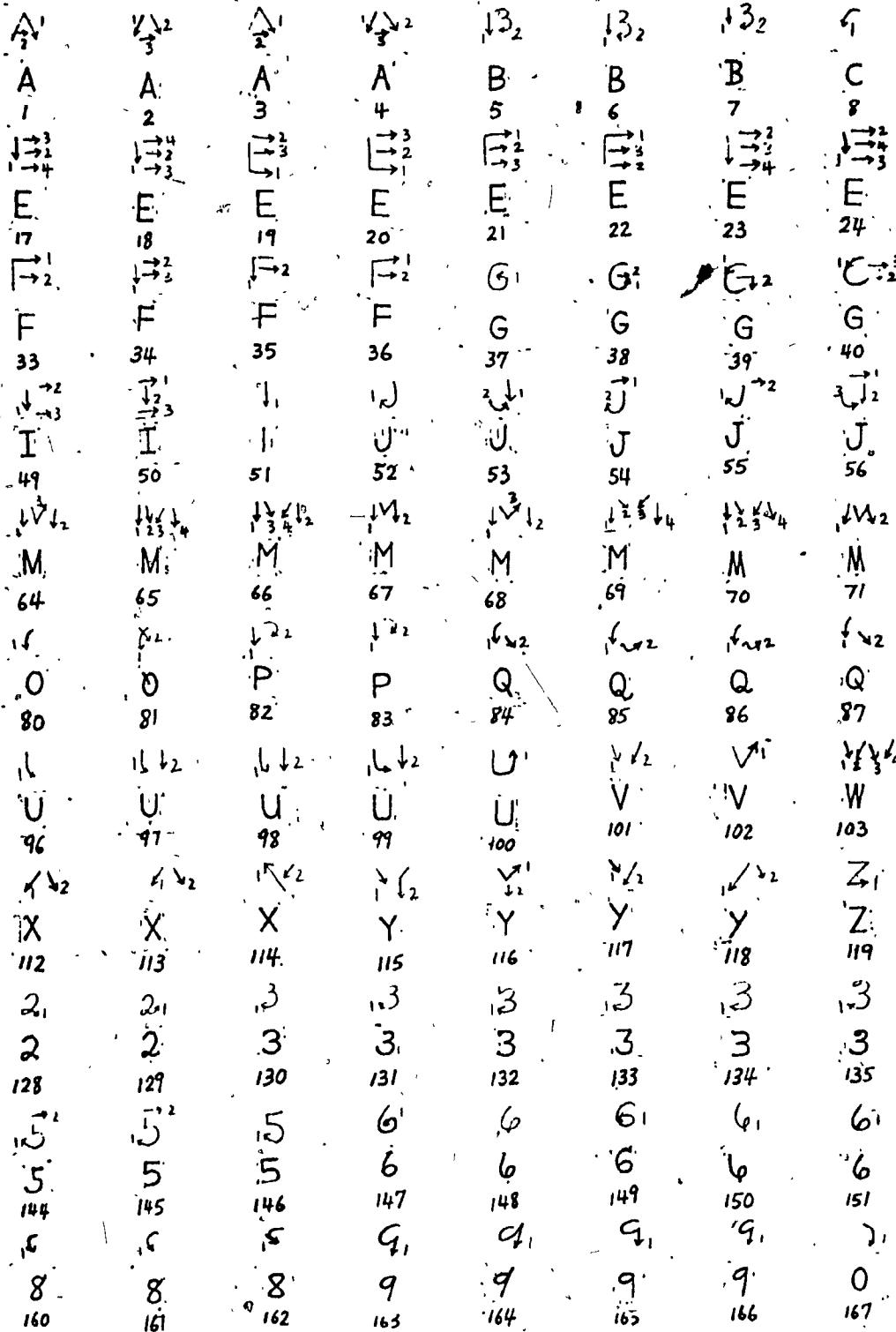


Fig.1-2 The alphanumeric character models used in this study (Part. 1)

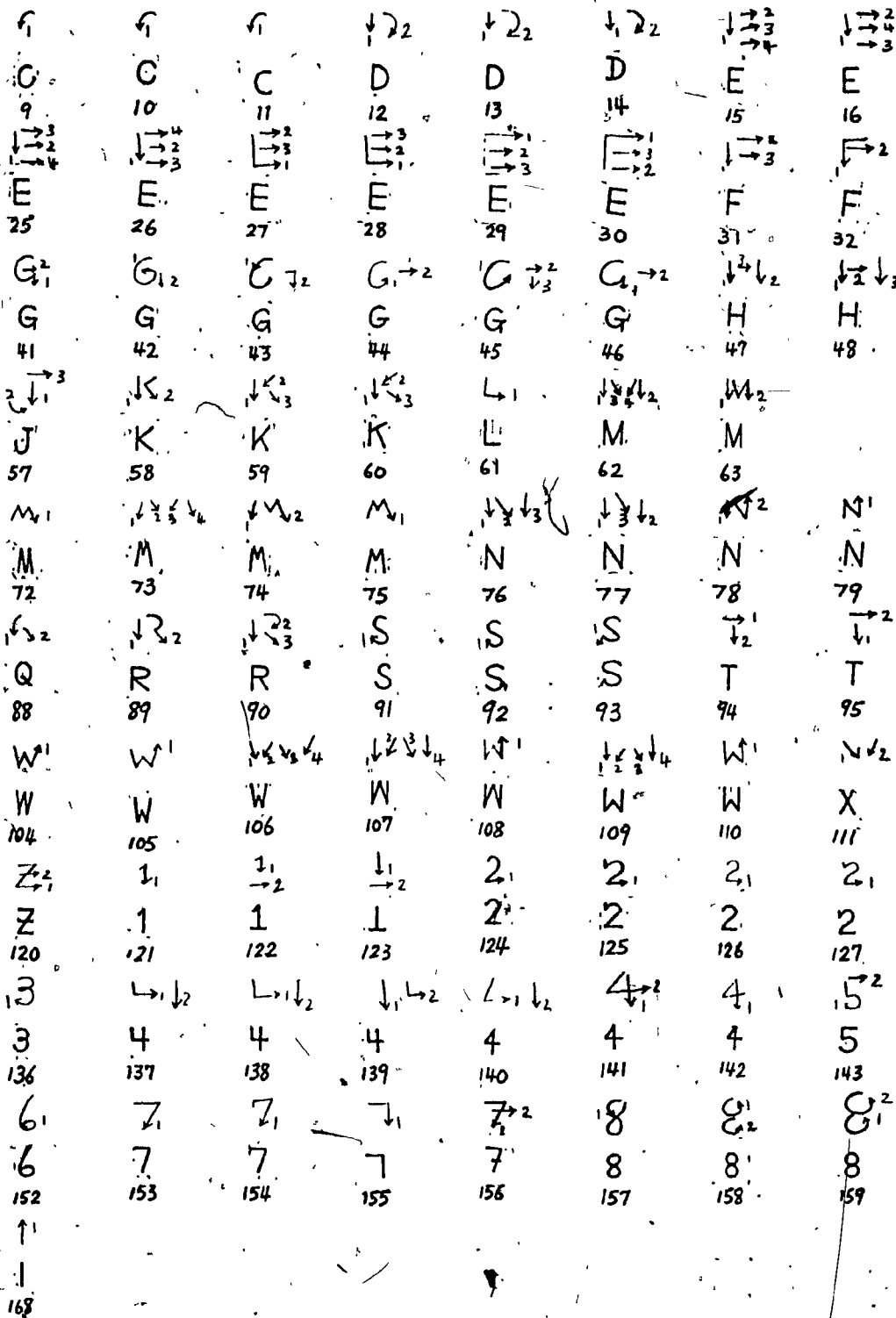


Fig.1-2 The alphanumeric character models used in this study (Part 2)

of difficulty of writing, thus affecting the writing quality of a character.

Each character model consists of 600 samples written by 30 authors, 15 of them are left-handed writers, the other 15 are right-handed. Before the samples were written, brief instructions of writing were given and subjects were asked to write as close as possible to the models in the shortest time. All samples were written in a rectangular box of size 0.16"x0.24". Altogether, 100800 character samples were used in this study.

The 600 samples of each character model are divided into two groups (300 each): the left-handed group and the right-handed group. In the procedure to select the standard character set, each group is treated independently so that one can study the difference, if any, between left-handed samples and right-handed samples when a character model is written. When the recognition system is tested (see chapters 3 and 4), all 600 samples of the same model are used. They will now be divided into two groups and will be so arranged that each group contains 300 samples, 10 from each author, i.e. 150 left-handed samples and 150 right-handed samples. These two groups will be used as testing data set and training data set alternatively.

Character samples are digitized in binary value (0 or 1) and stored in matrix form with the dimension of 64x32.

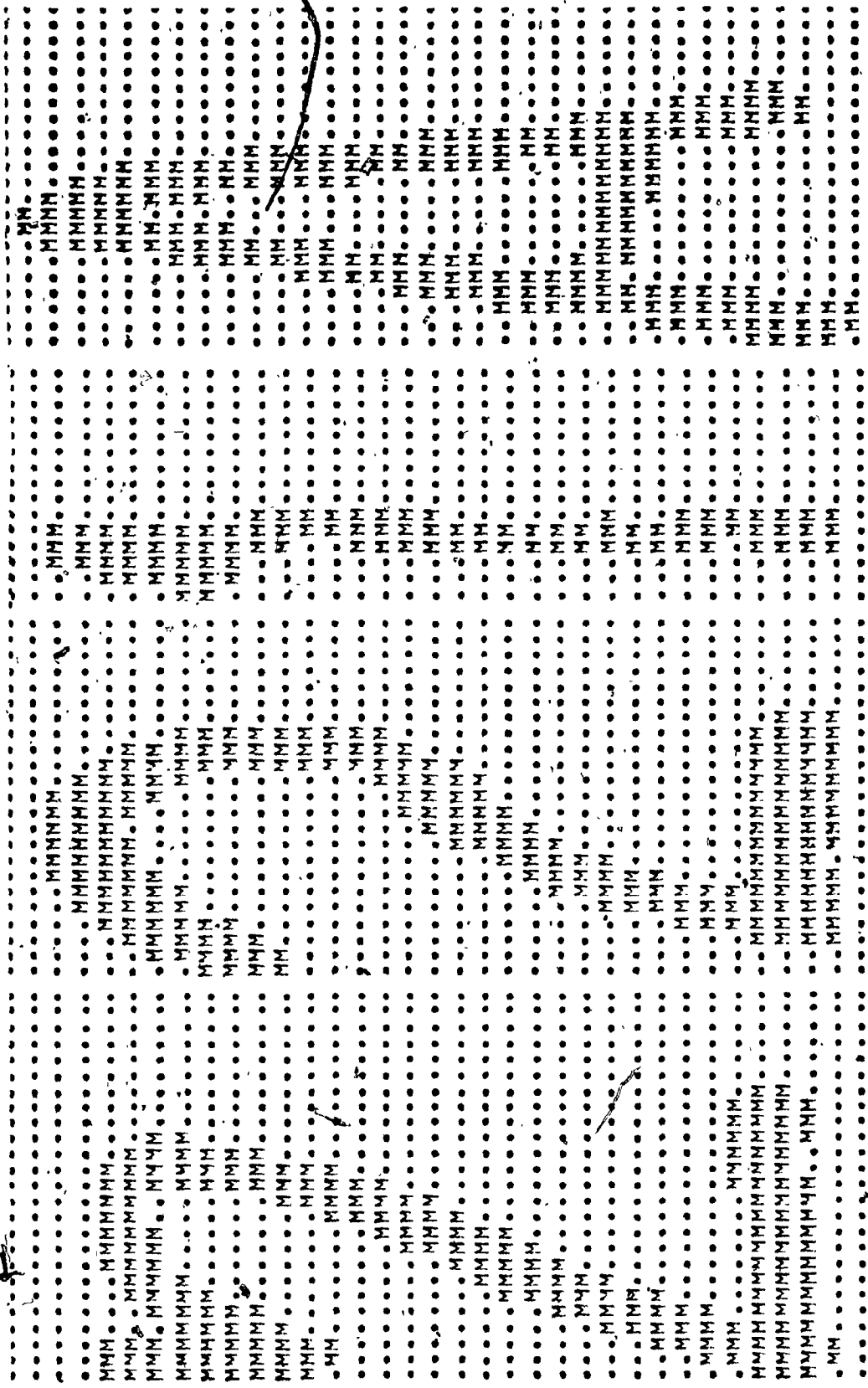


Fig. 1-3 Digitized character samples

Noise is removed by preprocessing programs. The top and left shift normalizations make the character image appear on the top left of the matrix. Fig.1-3 shows some character samples. Character 'M' is placed in the entry which is black (with value 1) and character '.' is placed in the entry which is white (with value 0). 'M', '1' and 'black' thus have the same meaning with respect to matrix entries and so are '.', '0' and 'white'.

1.4 Scope of the Thesis

In chapter 1, the general background of character recognition was mentioned. The problem of imposing constraints on writing also has been explained. The arrangement of the data used in this study has been described in detail and the description of work done in each of the following chapters is presented.

In chapter 2 of this thesis, an optimized set of alphanumeric characters will be selected based on the effect of geometric structure and stroke sequence of a character for best recognition purpose. The general characteristic of the chosen character set will be discussed after the selection.

Tests of the chosen set of characters will be described in chapter 3. The performance on this chosen set will be compared with another set of characters which represents the

worst set chosen from the same selection criteria. Finally a promising recognition system is established to recognize the data of the chosen characters. The new feature set and its extraction method are described in chapter 4 and the results of recognition of this system will be analysed.

Chapter 5 will give the conclusion, some comments and suggestions for further study.

CHAPTER 2

SELECTION OF STANDARD CHARACTERS

2.1 Introduction

Recognition of handprinted character is greatly affected by the quality of data. Obviously, data with high quality will produce a higher recognition rate. The intrinsic properties of a character, such as its geometric structure, stroke sequence, substantially influence the writing quality of that character. For example, writing character '2' by model 2 with the stroke sequence 2 is much simpler and easier to follow than by model 2 with the stroke sequence 2. Therefore character data of model 2 have better quality than character data of model 2 regardless of all other factors (This will be proved by the probability distribution table and recognition rate in this chapter and the next chapter). Knoll's experiment [28] showed that when different sets of handprinted character data were tested by the same algorithms of feature extraction and decision rule, different recognition rates were obtained.

The above consideration gives the idea that well-chosen characters will simplify the recognition problem. Restricting handprinting to a set of chosen standard characters is a constraint imposed on the handwriting.

However, if this set of characters is used for education and becomes a conventional style of writing, it will no longer be a constraint.

2.2. Dispersion Factor

If the shape of a given character has little change whenever it is written, this character is said to have a high quality of reproducibility. One way to investigate the reproducibility of a given character is to examine the shape deviation of that character when it is written by different individuals. A character with less deviation makes good sense in terms of recognition of that character, especially for automatic recognition, because writing this kind of character reduces unexpected situations that violate the recognition algorithm. This is the basic concept used in this study for the selection of standard characters. A quantitative measure called 'dispersion factor' (defined later) is used to evaluate the reproducibility of a character model.

All samples from left-handed (right-handed) writers of the same model are superimposed to create a frequency diagram[29]. This diagram gives the frequency of occurrence of each entry in the resultant matrix representing the character model after superimposition. The probability of occurrence at each entry is also calculated to form a

probability distribution table. In this table, probabilities are expressed in terms of percentage. Templates are constructed at 5 % intervals, e.g. 5%, 10%, 15%, ..., etc.. As an example, Fig.2-1 shows the frequency diagram of the model 124(2') from right-handed data. Fig.2-2 is the probability distribution table of the diagram and the associated templates at 30% and 65%.

Dispersion factor of a template gives the shortest distance between the template and non-zero points outside the template, it is calculated by the following formula[30]:

$$D = \frac{\sum_{j=1}^{N_0} F_j \left\{ [R(x_j - x_t)]^2 + (y_j - y_t)^2 \right\}^{1/2}}{\sum_{k=1}^{N_i} F_k}$$

where

F_j = Frequency of occurrence at point j outside the template.

F_k = Frequency of occurrence at point k inside the template.

N_0 = Total number of non-zero points outside the template.

N_i = Total number of non-zero points inside the template.

R = Ratio of horizontal sampling interval to vertical sampling interval.

x_j = Abscissa of point j outside the template.

- y_j = Ordinate of point j outside the template.
- x_t = Abscissa of boundary point closest to the point j .
- y_t = Ordinate of boundary point closest to the point j .

corresponds to the shortest distance between the template and the point j outside the template.

$$\left\{ [x_j - x_t]^2 + [y_j - y_t]^2 \right\}^{1/2}$$

2.3 Representation of Dispersion Factors

The minimum percentage for template construction is set to 30 because this is the percentage that a character model generally comes out with good shape from the distribution table. Information obtained below this percentage is of little significant value. The maximum percentage differs from one character model to another and is set to equal to the last percentage at which the character model is still connected. (i.e. Retaining the original shape of the model.)

A sequence of dispersion factor values (called D values) is obtained by calculating the dispersion factor for each existing template of a distribution table. These D values are plotted in graph vs percentage of templates as in Fig.2-3. All points of the same sequence are connected by smooth segments to form a line of dispersion factors (D-line for simplicity). This D-line shows the tendency of the D values of the character model from which the probability

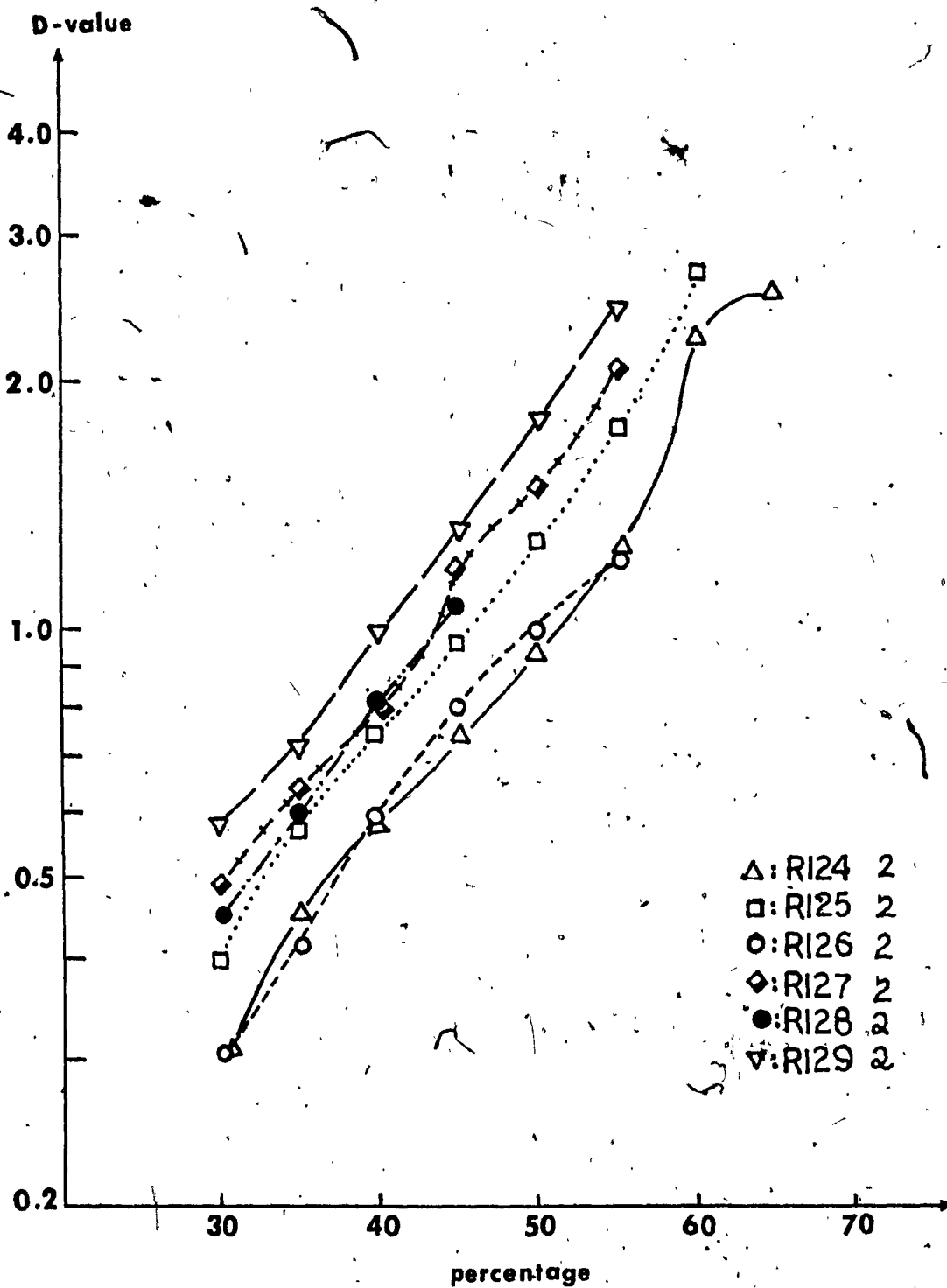


Fig.2-3 The D-line graph of character models of '2'

distribution table is established. It will be used as the main tool for character model selection.

There are two D-lines for each character model: one is obtained from left-handed data while the other is from right-handed. In order to make a comparison, all D-lines of models of the same character class are drawn in the same graph, also one graph for the left-handed, one for the right-handed. Fig.2-3 shows the D-lines of all models of character 2 for the right-handed.

2.4 Selection of Standard Characters

The selection criterion is to choose the most compact and reproducible model as the standard character. This can be visualized from the following facts:

- 1) From the definition of dispersion factor, one can see that the less the D values, the less the difference between the samples, thus the higher the reproducibility of the character. Therefore the character which has the lowest D-line in the D-line graph is chosen.
- 2) If two D-lines are not distinct, points in higher percentage will have higher priority since they have a better representation of the quality of the character.
- 3) The length of a D-line indicates the compactness of a character model. It is another quality to be considered when two D-lines are compared. The longer the D-line,

the better the character.

Since there are two D-line graphs for each model, two results can be obtained, one representing the selection for the right-handed persons, the other for the left-handed. They may not be the same (see results on the next section)! It is impractical if a standard set of alphanumeric characters is designed for right-handers while another one for left-handers. Further considerations will also be made to solve the problem.

Based on the statistics[31], about 10% of the people in North America are left-handed and 90% are right-handed. These factors are used as weights to calculate the modified D values (D') by the following formula:

$$D' = D_L / 10 + 9D_R / 10 \dots\dots\dots(2.1)$$

where D_L is D value of left-handed.

D_R is D value of right-handed.

D' combines D_L and D_R with weights giving a new D value. It is obvious that D' is right hand biased. With these D' values, new D-line graphs are drawn for the character models which have results slightly different from the left-handers and right-handers (Decisions for models with same results in both right-hander and left-hander do not change under the new consideration). Unique results will be obtained by applying the same selection criterion as

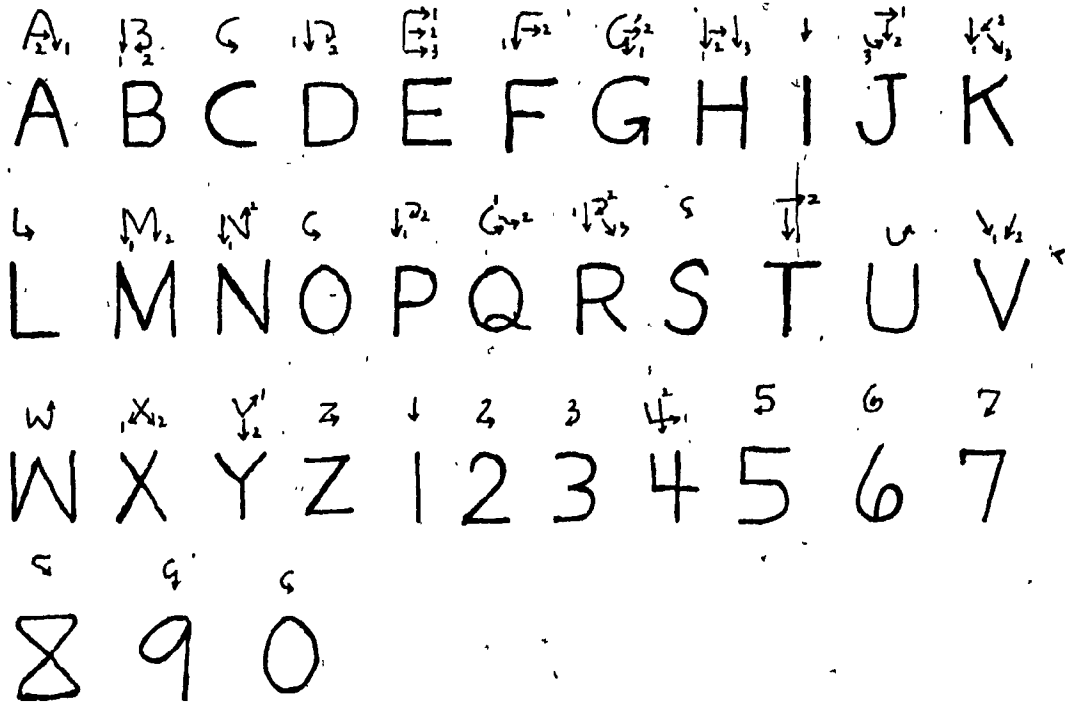


Fig. 2-4(a) The chosen character set for right-handers

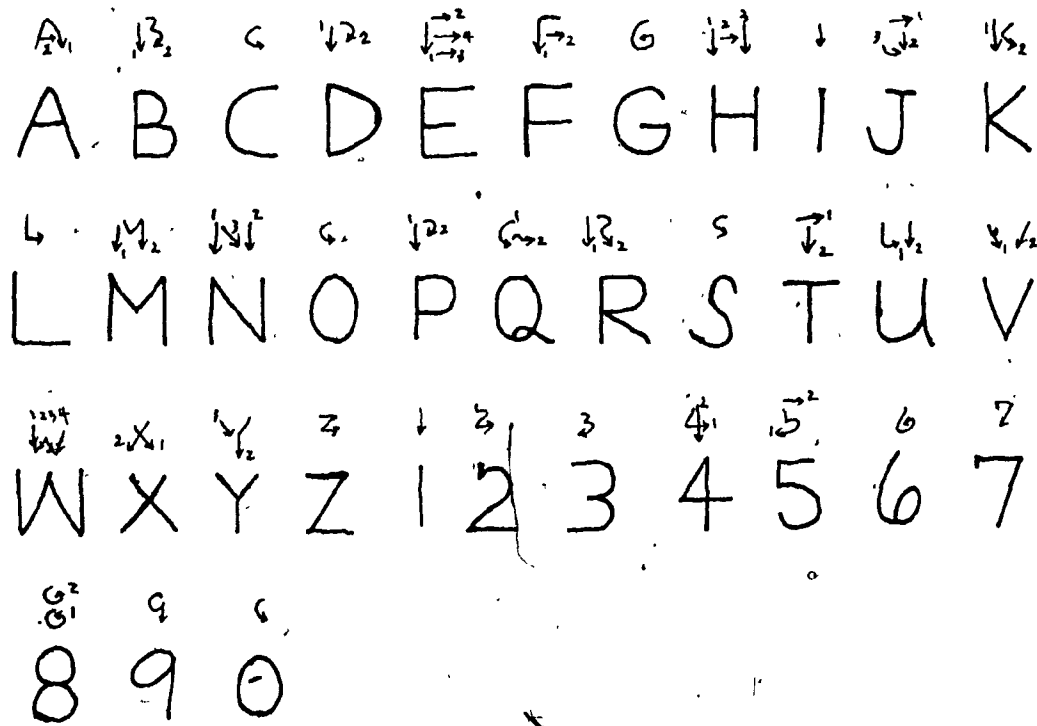


Fig. 2-4(b) The chosen character set for left-handers

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 1 2 3 4 5 6 7
 8 9 0

Fig.2-4(c) The final chosen character set

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 0 1 2 3 4 5 6
 7 8 9

Fig.2-4(d) The ANSI character set

before.

2.5 Results

The chosen sets of alphanumeric characters, as well as their stroke sequences, are listed in Fig.2-4. Fig.2-4(a) shows the chosen character set for the right-handed persons, Fig.2-4(b) shows the character set for the left-handed. The final optimum set of characters after using D' values is shown in Fig.2-4(c).

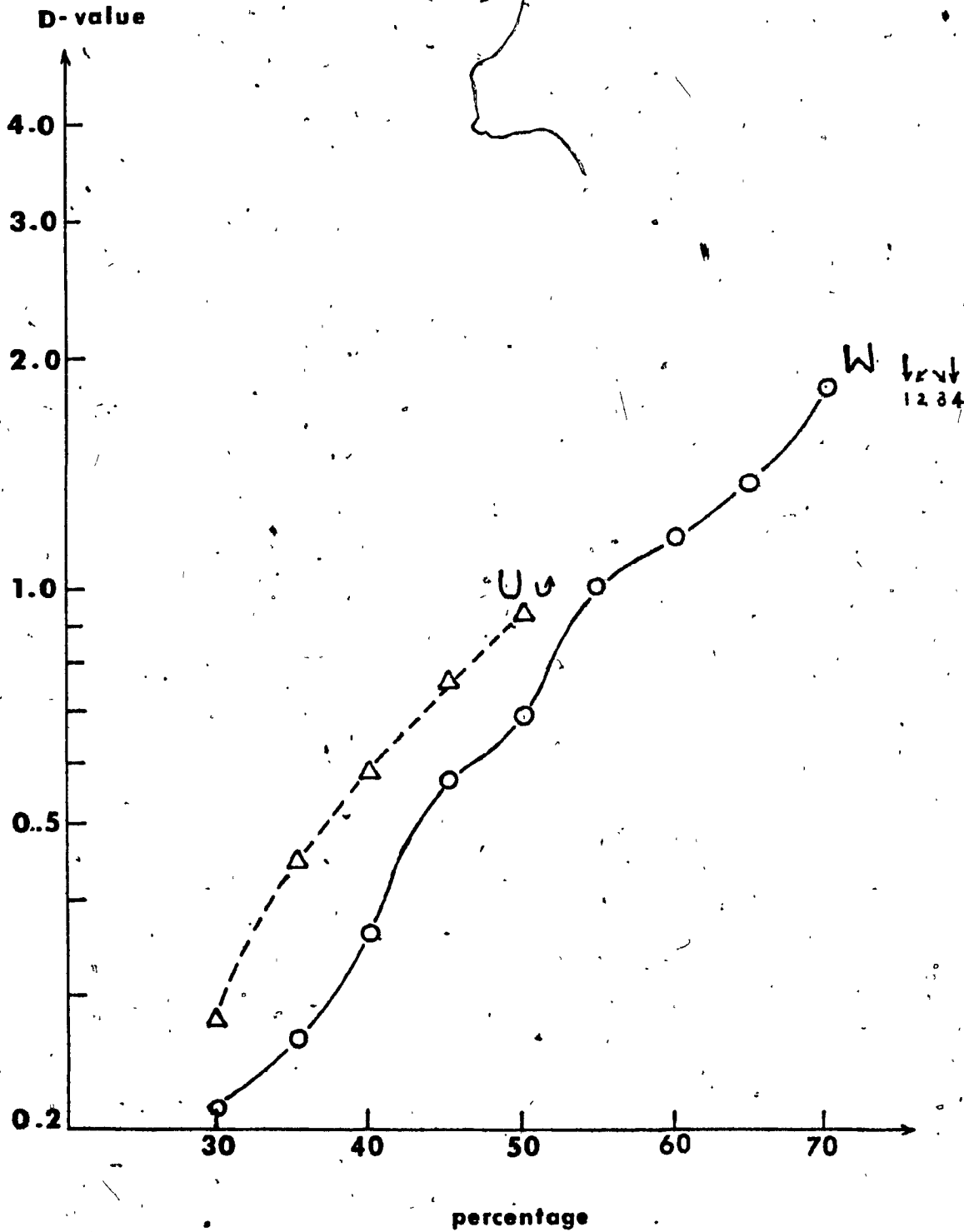
Out of 36 characters, only three characters ('n', 't', and '4') were chosen from left-handed character set because of the heavily right-hand-biased function. The best choices of alphabetic characters 'o' and 'i' were abandoned since the same character models were chosen for numerals '0' and '1'. The second choices of characters 'o' and 'i' were the ones listed in the figure. It is interesting to compare the final result with the ANSI characters which are shown in Fig.2-4(d). 13 characters out of 36 are greatly different between the chosen character set and the ANSI character set. They are: B, C, D, G, K, M, R, S, W, Z, 4, 6 and 7.

The general configuration of these sets of characters is the simplicity of geometric structures. For example, model B is chosen instead of model \mathcal{B} , model U rather than model \mathcal{U} . This doesn't imply that the simplest model in a character class is chosen as the best character. For

example, the model G is chosen instead of G . Adding unusual features to a character to achieve a greater distinction among characters is a good way for better recognition, however, it may reduce the reproducibility of the character.

Several facts are observed from this study:

- 1) The frequency of occurrence of a stroke at each matrix entry in the frequency diagram decreases from left to right (or from top to bottom) if it is drawn from left to right (from top to bottom). Strokes drawn in other directions have the same phenomena. This is because the beginning of a stroke is usually more prominent.
- 2) Left-handed samples are less deviated and get closer to the model. This can be observed by comparing the D values and the D -line graphs produced by left-handers and the right-handers. This fact may be explained as follows: since all character models are right-handed biased, there are some constraints on the mechanical movement of the left hand when it is used for writing these models. It is these constraints which make left hand writing more rigid and pattern following.
- 3) It is not necessarily true that a simple pattern will yield a low dispersion factor. For example in Fig.2-5, character 'w' has 4 strokes while character 'u' has only one, but the D -line of 'w' is lower than that of 'u'.



- Fig.2-5 The D-line graph of models 'u' and 'w'

This suggests that straight lines should be used whenever possible as long as the general appearance of the character is preserved. This suggestion is further confirmed by the fact that the character model 'S' has the highest D values while character model 'l' has the lowest and longest D-line.

- 4) D-values of two character models which have the same geometric shape but different stroke sequences are not the same. Sometimes there exists a big gap between these values. This means that besides the shape of a model, the stroke sequence also affects the dispersion factors of that model. For example, the character model 140('4') with stroke sequence 4₁ is chosen as the best model for character '4'. However, the model 142('4') with stroke sequence 4₂ is one of the worst models for character '4' (see the recognition results from the confusion tables in chapters 3 and 4).

CHAPTER 3

TEST OF PERFORMANCE OF THE CHOSEN CHARACTERS

3.1 Introduction

To evaluate the performance of the chosen character set, several recognition algorithms have been implemented. For simplicity and testing purposes, only numerals are subjected to tests. The result can reflect the validity of the dispersion factor theory. Two outputs are needed from the test to reflect the performance of the character set: recognition rate and confusion table. The recognition rate indicates the recognition potential of the character set and the confusion table points out the weaknesses. Another set of numeric models, which is the worst under the same selection criterion, is chosen and tested by the same recognition algorithms for comparison with the best set. This reference set (the worst) of characters is shown below:

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

Most recognition systems are feature-oriented. For certain types of characters, special features are designed and the decision algorithm depends heavily on them. In the

present case, two sets of different characters are to be tested, bias on any characters due to the feature extraction process could produce unfair judgement on the results. Thus, only feature-independent recognition systems have been chosen to perform the tests.

The n-tuple method[32] is used for feature extraction. The decision algorithm is based on the minimum distance classifier[38]. In order to make the conclusion of the test more positive and to ensure that it is not a biased result, another feature extraction technique, the characteristic loci[35], with the same decision rule is applied to test the data.

3.2 N-tuple Feature Extraction Method

N-tuple method was developed by W.W. Bledsoe and I. Browning[32,33,34]. It has been chosen because it is the least biased feature extraction algorithm.

Suppose the character pattern is composed of $m \times n$ elements. N elements are randomly chosen to form an n-tuple and thus there are m disjoint n-tuples which cover all elements of the whole pattern. A state of the n-tuple is defined as the set of values of those elements that compose the n-tuple. Each state is treated as an independent feature of the character to be extracted. Since pattern elements are in binary values, 0 or 1, so each n-tuple has

2^n states and totally $m \times 2^n$ states in the whole pattern. For example, in Fig.3-1, the 5 pattern elements marked by circles constitute a 5-tuple and its state is listed below the pattern.

When the method is applied to this study, the character pattern grid is restricted to the size of 46×24 since all character samples are within this frame. N is set equal to 2 and 3. That is to say, at least 552 2-tuples or 368 3-tuples, thus 2208 or 2944 features could be extracted from each sample. Element pairing is done by a built-in random number generator in the CDC computer. 2 sets of random pairs are generated to investigate if different random pairs will affect the test results (discussed in section 3.5).

3.3 The 'characteristic loci' Algorithm

The 'characteristic loci' method is another feature extraction scheme to be used. Its features, devised by Glucksman[35] and modified by some researchers[28,36,37], are briefly described as follows:

A 4-digit code is associated with each white ('0') point in the character image matrix. Each of the digits contains the count of the number of line crossing from the point to one of the four perpendicular directions: left, up, right and down respectively. The count is restricted to a maximum of 2, any digit with number larger than 2 will be set equal

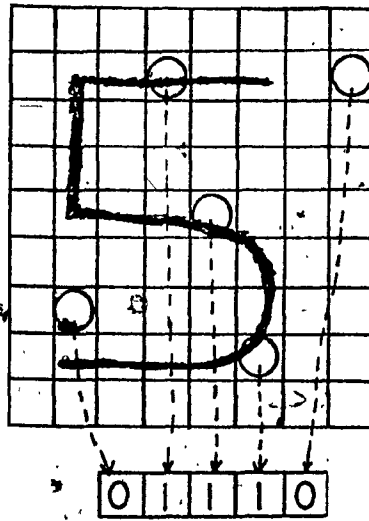
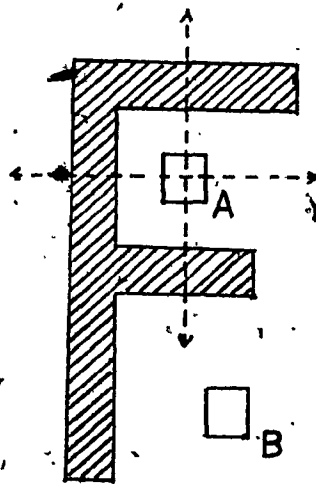


Fig.3-1 Illustration of an 5-tuple and its state



CODE(A) : 1101
 CODE(B) : 1200

Fig.3-2 Examples of coding in 'characteristic loci' algorithm

to 2. These 4 digits are then combined into a 4-digit ternary code. For example, in Fig.3-2, the code 1101 is associated with point A while point B has its code equal to 1200. With this 4-digit code, a feature space of $3^4 = 81$ dimensions can be defined. Since any codes with 3 zero digits are associated with points outside the frame (see the next chapter) of the character image, they are not used in the construction of the feature vector. That is to say, only 72 codes are actually used as features. The magnitude of a feature is defined as the total number of white points matching the code associated with that feature.

A serious problem arises when character model '1' is present. None of the features in the above feature vector can be extracted when this character is perfectly written since all codes associated with this character have at least 3 zeros. An extra feature must be created to deal with it. Glucksman[35] used the total number of black points in the character matrix as this extra feature. Because of this, a total of 73 features are actually used in this algorithm. All features are divided by the total number of points inside the frame of the character image for size normalization purpose.

3.4 Recognition Scheme

Perfect recognition is not the main goal of this test. However, a general-purpose, character-independent, easily-implemented and highly efficient recognition algorithm is needed to test the performance of character sets. The minimum distance classifier [38,39] meets this requirement.

In an n -dimensional vector space, if R points are given:

$$P_i (p_{i1}, \dots, p_{in}) \quad i=1, \dots, R$$

The Euclidean distance between an arbitrary point $X (x_1, \dots, x_n)$ and P_i is given by:

$$d = \sqrt{(X - P_i)^2}$$

$$= \sqrt{\sum_{j=1}^n (x_j - p_{ij})^2} \dots \dots \dots (3.1)$$

Let R be the number of classes in the character data, N be the number of features of the feature vector, X be the feature vector of the input character sample. If P_i represents the ideal feature vector of character class i , $i=1, \dots, R$. X would represent the feature vector of an unknown character sample. Minimum Euclidean classifier will assign X to the class i if the distance between X and P_i is the shortest among all other i 's, i.e. assigning X to the class for which equation (3.1) is minimal. Equation (3.1) can be expressed in an alternative form as:

$$\begin{aligned}
 d^2 &= \sum_{j=1}^n (x_j - P_{ij})^2 \\
 &= \sum_{j=1}^n x_j^2 - 2 \sum_{j=1}^n x_j P_{ij} + \sum_{j=1}^n P_{ij}^2
 \end{aligned}$$

$\sum_{j=1}^n x_j^2$ has no effect on minimization since it is independent of character classes. Instead of getting minimum value from the expression:

$$-2 \sum_{j=1}^n x_j P_{ij} + \sum_{j=1}^n P_{ij}^2$$

it is equivalent to getting maximum value from the expression:

$$\sum_{j=1}^n x_j P_{ij} - \frac{1}{2} \sum_{j=1}^n P_{ij}^2$$

If the weight of a discriminate function is set as

$$w_{ij} = P_{ij}$$

we have

$$\sum_{j=1}^n x_j w_{ij} - \frac{1}{2} \sum_{j=1}^n w_{ij}^2$$

where w_{ij} is the weight of the j^{th} feature in class i .

A set of discriminate functions is defined from this expression as:

$$g_i(x) = \sum_{j=1}^n w_{ij} x_j - \frac{1}{2} \sum_{j=1}^n w_{ij}^2 \quad i=1, \dots, R$$

The value of w_{ij} is established by the probability of occurrence of feature j in class i and can be obtained by calculating the average value of this feature through a training set of examples.

The input sample X will be assigned to the class i_0 if for all $i, i=1, \dots, R$ and $i \neq i_0$

$$g_{i_0}(X) > g_i(X)$$

Note that no threshold value is set to the above decision function because:

- 1) Threshold values may be different in two different character sets for the same reject rate.
- 2) Recognition rates of two different recognition algorithms should be compared with reject rate equal to zero.
- 3) Confusion matrices of the two sets of characters are needed for further analysis.

3.5 Results and Comparison

As described in section 1.3, character samples are stored in two data sets. One of the data sets is used for training and getting the weights in the discriminate functions while the other is recognized on the basis of the

resultant functions, and then vice versa.

The performance of the chosen character should be shown from the recognition rate of the whole character set rather than from the recognition rates of individual characters. It is because the recognition decision depends on the whole set of discriminate functions of the character set. The recognition rate of a particular character, which is high in the reference set, is not necessarily high in the chosen character set. The reverse is also true.

By applying the n-tuple algorithm, the recognition rate of the chosen character set was 97.68% while the reference character set was 95.02%. The results do not change much as n equals to 3, they are 97.65% and 94.78% respectively. When the characteristic loci algorithm was used, the recognition rates of the two character sets were 99.15% and 98.64% respectively. These results indicate that the chosen character set does have better potential to be correctly recognized (regardless of the recognition system used). The overall high percentage of recognition achieved in the whole character data is due to the high resolution of character images, as well as the good quality (except reproducibility) of the data.

Confusion tables are shown in Tables 3-1, 3-2, 3-3. Character '1' in characteristic loci algorithm gives the highest mis-recognition rate. This is due to the nature of

the feature vector of this algorithm' (as mentioned in section 3.3). In the confusion table of the n-tuple method, characters '8', '7' and '2' produce the greatest confusion. When the second set of random pairs is applied to the n-tuple method, the result has little change. It means the random pairs are quite insensitive to the recognition process.

	0	1	2	3	4	5	6	7	8	9
0	584		2				6		7	1
1		578	5	3	4	3	7			
2	6		570	8		10	5			1
3	2	1	3	547		19	24		4	
4		4		3	580	1	5		2	5
5			4			573	4		14	1
6	4	5	2			1	574		7	7
7		1		8		1		579	1	10
8	1		2	29		6	8		551	3
9	13			6	13			2	15	551

(b)

	0	1	2	3	4	5	6	7	8	9
0	583		2		1	1	11			2
1		597					3			
2	5		563					3	29	
3			1	592		1		2	4	
4					598				1	1
5	1	1		4		585			9	
6	5	1					594			
7			18	2		1		576	1	2
8	1		15			4			579	1
9	1				3	1		3		592

(a)

Table 3-2 Confusion tables from N-tuple (n=3) method
 (a) the chosen character set (b) the reference character set

	0	1	2	3	4	5	6	7	8	9
0	596									4
1	586	9			4			1		
2	3	596	1			1			2	
3		19	562	2					17	
4		1			597					2
5			2	3	585	7			2	1
6						3	597			
7								598		2
8									600	
9					2					598

(b)

	0	1	2	3	4	5	6	7	8	9
0	592	1					1		6	
1	565		12				14	7	2	
2		600								
3			599	1						
4				600						
5					600					
6						600				
7		1						593		
8									600	
9										600

(a)

Table 3-3 Confusion tables from 'characteristic loci' method

(a) the chosen character set (b) the reference character set

CHAPTER 4

DEVELOPMENT OF A NEW FEATURE EXTRACTION ALGORITHM

4.1 Introduction

In the previous chapter, 2 feature extraction algorithms were applied to the chosen numeric character set. Although good results have been achieved, they are by no means the most suitable feature extraction algorithms for the character set. Characteristic loci algorithm is not good for character '1' and the n-tuple method has low recognition rate. Some extra techniques and programming have to be done to get rid of these problems. For example, in Michael's thesis[37], 3 more contour features were added to distinguish the character '1' from others in the loci's algorithm. This sort of manipulation work will complicate the system. Computer utilization is another major factor to be considered. The characteristic loci algorithm takes a large amount of computer time to establish a feature vector. The n-tuple method requires comparatively a huge memory space to store all states (see the comparison in section 4.4).

It is desirable to have a simpler and faster feature extraction algorithm that can handle the chosen character set. A set of simple features is designed in an attempt to

solve this problem.

4.2 Description of Features

In the new feature set, there are 45 features in total and most of them are simple geometric line structures. Their shapes and relative positions in the rectangular frame of a character are drawn in Fig. 4-1 and the arrow on some of the features indicates the direction of scanning for that feature when it is examined for its presence. Each feature has a number assigned to it for easy management and identification purposes. These features are divided into 5 groups according to their functions: risers, bars, diagonals, windows and crossings.

1) RISERS

number :6, from 1-6

function: detecting vertical strokes.

examples: B, 5

2) BARS

number :6, from 7-12

function: detecting horizontal strokes.

examples: 4_r, J^f

3) DIAGONALS

number: 12, from 13-24.

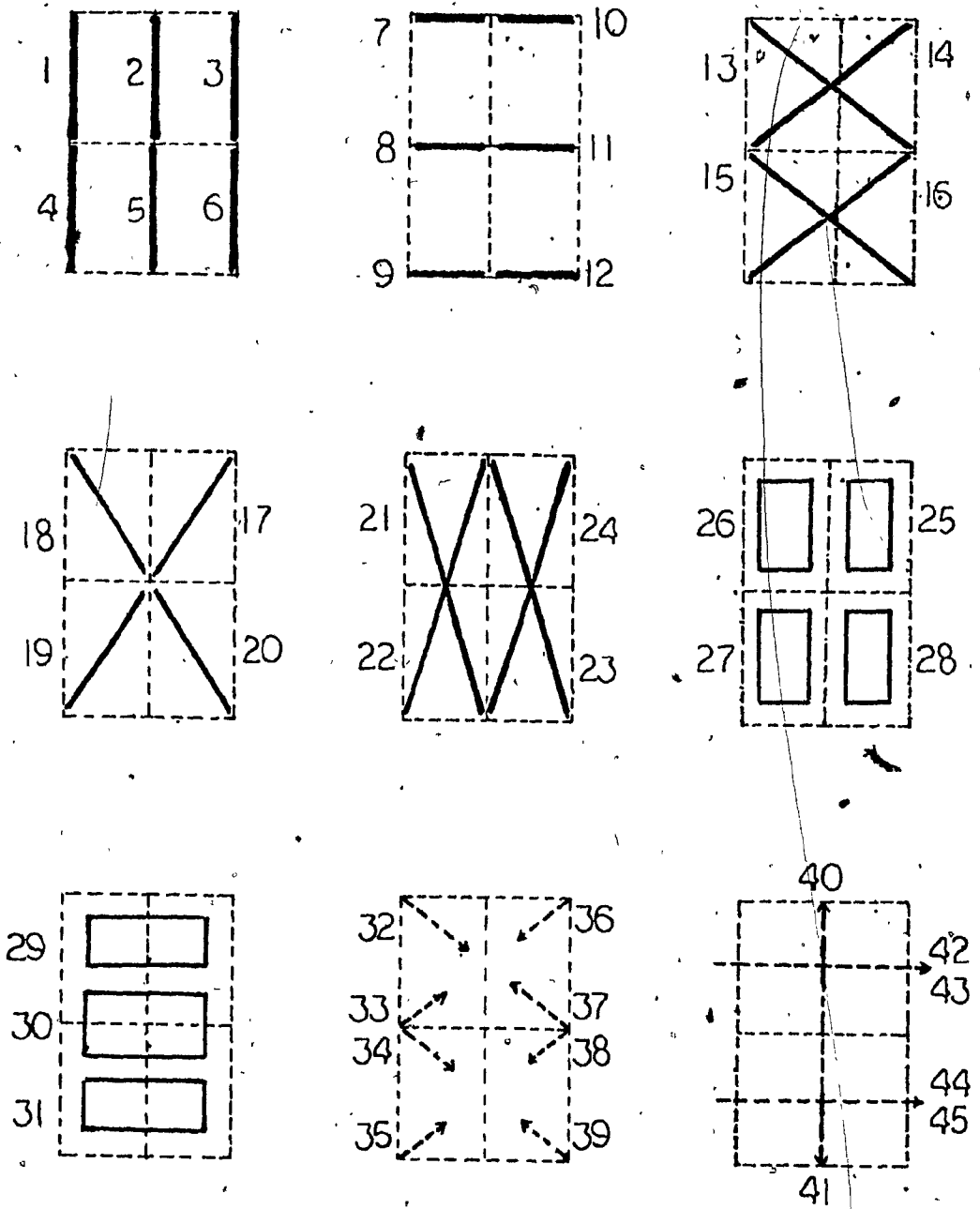
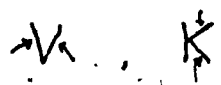


Fig.4-1 The new feature set

function: detecting inclined strokes.

examples: 

4) WINDOWS

number : 7, from 25-31.

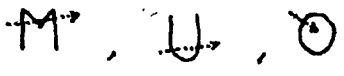
function: detecting loops and bays.

examples: 

5) CROSSINGS

number : 14, from 32-45.

function: collecting crossing information.

examples: 

4.3 Feature Detection

In this character data set, the width of a line drawing usually occupies 3 rows or 3 columns; this width is taken as the nominal width of a stroke. The actual size of a character image is smaller than the dimension of the matrix storing it. Before any feature is to be detected, a routine called WHERE will scan through the matrix to find out the frame of the character image. The output of this routine provides the boundaries (left, right, top and bottom) as well as the centre of the character image in terms of column or row of the matrix.—Different detection techniques apply to different features:

1) BARS

Three parameters are used: starting column(SC) and row(SR), terminating column(TC). 3 consecutive rows, starting from SR to SR+2, are checked in parallel with logical operation OR. If any one of the rows of the same column is black('M'), then that column is black. A feature is said to be present if the ratio of number of 'M' to bar length (TC - SC + 1) is greater than a threshold value. As shown in Fig.4-2, a bar is present under this decision rule. The threshold is arbitrarily set at 85%.

2) RISERS

The same detecting technique as above is applied for examining risers. In this case, the given parameters are starting row, starting column and terminating row.

3) DIAGONALS

Three kinds of diagonals are detected. Given the slope of a feature, the following analytical geometric equation is used to find all points, along this slope:

$$y = mx + b$$

where m is the slope of the diagonal, b is an initial constant, x,y are the coordinates of a point (column number or row number). In addition to the slope parameter, there are 4 other parameters which control

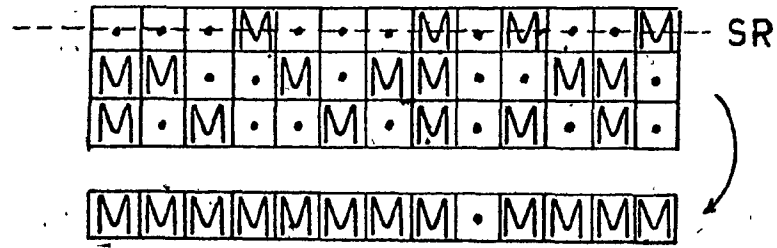


Fig.4-2. Illustration of a bar feature

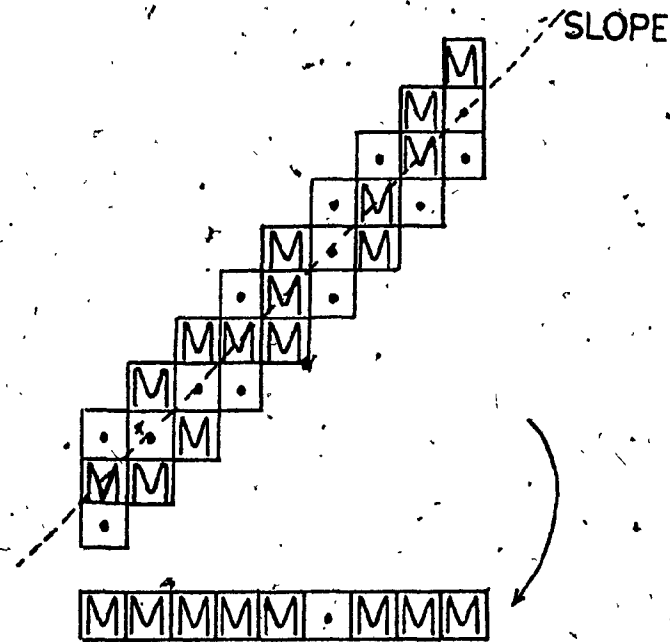


Fig.4-3 Illustration of a diagonal feature

the initial constant and domain of points of the diagonal feature. A point along the slope is 'black' if the point or its upper neighbour (-1 row) or its lower neighbour (-1 row) is 'black'. The feature is present if over 90% of the points along the slope is 'black'. This threshold value is also arbitrarily chosen. Fig. 4-3 gives an example of a diagonal feature.

4) WINDOWS

A window feature has 4 parameters to fix the size of window. All windows are rectangular in shape. A window is present if the number of black points inside the window is less than a threshold. Window with width longer than length (number 29, 30, 31) has a threshold equal to twice the window length. Other windows (number 25-28) with length longer than width has a threshold set to the window length.

5) CROSSINGS

Corner crossings (number 32-39) detect a line drawing (straight or cursive) passing by one of the eight corners. A feature is present if, along the bisector of a corner, a sequence of points with format

$$\begin{array}{cccc}
 0 & 01 & 10 & 0 \\
 n_1 & n_2 & n_3 & n_1 > t_1, n_2 > t_2, t_3 > t_3
 \end{array}$$

appears, where t_1, t_2, t_3 are thresholds set to 2.

Boundary crossings (number 40-41) detect crossings along the top and bottom of the central column of the frame.

A feature is present if a sequence of contiguous points with format 0011 appears.

Line crossings (number 42-45) count the number of line crossings along the central row of the upper half and along the central row of the lower half of the character frame. This count is restricted to have a value of 1 or 2. Any count larger than 2 is set equal to 2. The counter adds one when a continued sequence (greater than 2) of isolated (by white points) black points is hit.

Note that no direct curve detection is designed in this feature set. However, most of the curve drawings in a character pass by the corners. Corner crossings will detect their existence.

4.4 Recognition with the New Feature Set

The same database is used again to test this new feature extraction system. 12 diagonal features are taken out since a screening program shows that they have little effect on the numerals. In order to have a comparison with previous methods, the decision scheme remains the same as before, i.e. using minimum distance classifier.

The results of this experiment show good response to the initial design of this feature set. The feature pool used in this system (only 33 features) is much smaller than the ones in characteristic loci (73 features) and in n-tuple

(2208 or 2944 features). The time required to recognize a character is also greatly reduced, 12 characters are recognized in one second of CPU time (including feature extraction and recognition process). This recognition speed is 4.4 times faster than the loci method and 4.8 times faster than the n-tuple method with $n=2$. The average time for feature extraction in the new system is about 0.07 second per character (0.34 second in the loci algorithm and 0.08 second in the n-tuple method with $n=2$).

The recognition rate of the chosen character set is 99.35% which is also higher than the two previous methods. The confusion table is shown in Table 4-1. No character manifests poor behaviour against the system. Characters '2' and '5' give the highest mis-recognition scores—9 out of 600 while characters '1', '6' and '9' have full score. The recognition rate of the reference character set is 96.17%. The characters '3' and '2' have the lowest recognition rates. This indicates that the curve detection algorithm in this system is not very accurate in some cases.

Two kinds of samples usually lead to mis-recognition:

- 1) The sample of a given character class written in a way different from the chosen model of that class. Because of difference in shape, the features associated with the sample are quite different from the features of a normal character sample of that character class. When the discriminate function of the given class is applied to

0	591																			
1		577	6		3															
2	29	1	555																	
3	24																			
4		5																		
5		2	1	18																
6																				
7																				
8	6																			
9																				

(b)

0	598																			
1		600																		
2																				
3																				
4																				
5																				
6																				
7																				
8	1																			
9																				

(a)

Table 4-1 Confusion tables from the new system (a) the chosen character set (b) the reference character set

the sample, a low value will be obtained because of mismatch of the feature vector. The sample will then be assigned to other class with higher discrimination value. For example, in Fig.4-4(b), although the character '5' appears to have the good quality of a '5' to the human being, it is mis-classified as '7' since it is written in a shape completely different from the standard model of '5'. The samples in this kind can be mis-recognized as any other characters since the appearances of these samples are not similar to any character classes in terms of the standard models.

- 2) The sample of a given class that is out of the toleration level of the system although it is written according to the model of that class. This shape deviation also changes the feature vector of the sample from normal, but not as much as in the 1st case. The resultant discriminate function of the given character class will have a value lower than the average. Mis-recognition will occur when the value is lower than one of the other discriminate functions. In this case, the samples are usually mis-recognized as one of the confused characters of the actual character class. In Fig.4-4(a), the character '4' is mis-classified as '9', the second choice being '4'. This mis-recognition is due to the incomplete horizontal stroke that violates the left-right proportion of the character. Note that the difference in weight between the first choice and

.....MM.....
.....MMMM.....
.....MMMMMM.....
.....MMMMMM.....
.....MMM.MMM.....
.....MMM.MMM.....
.....MMM.MM.....
.....MMM.MMM.....
.....MMM.MMM.....
.....MMM.MMM.....
.....MMM.MMM.....
.....MM.MMM.....
.....MMM.MMM.....
.....MMM.MMM.....
MMM.....MMM.....
MMMMMMMMMMMMMMMM.....
MMMMMMMMMMMMMMMM.....
MMMMMMMMMMMMMMMM.....
.....MMMMMMMM.....
.....MM.....
.....MMM.....
.....MMM.....
.....MM.....
.....MMM.....
.....MM.....
.....MMM.....
.....MMM.....
.....MMM.....
.....MMM.....
.....MMM.....

(a)

.....MMMMMMMMMMMM.....
.....MMMMMMMMMMMMMMMM.....
.....MMMMMMMMMMMMMMMM.....
MMMMMMMM.....
MMM.....
MMM.....
MMM.....
MMM.....MMMM.....
MMM.....MMMM.....
MM.....MMMMMMMM.....
MM.....MMMMMMMMMMMM.....
MM.....MMMM.....MMM.....
MMM.....MMM.....MMM.....
MMMMMMMM.....MMM.....
MMMMMMMM.....MMM.....
MMMM.....MMM.....
MMMM.....MMM.....
MMMM.....MMM.....
MMMM.....MMM.....
MMMM.....MMM.....
MMMM.....MMM.....
MMMM.....MMMM.....
MMMM.....MMMM.....
MMMM.....MMMM.....
MMMM.....MMMM.....

(b)

Fig.4-4 Samples of misrecognized character

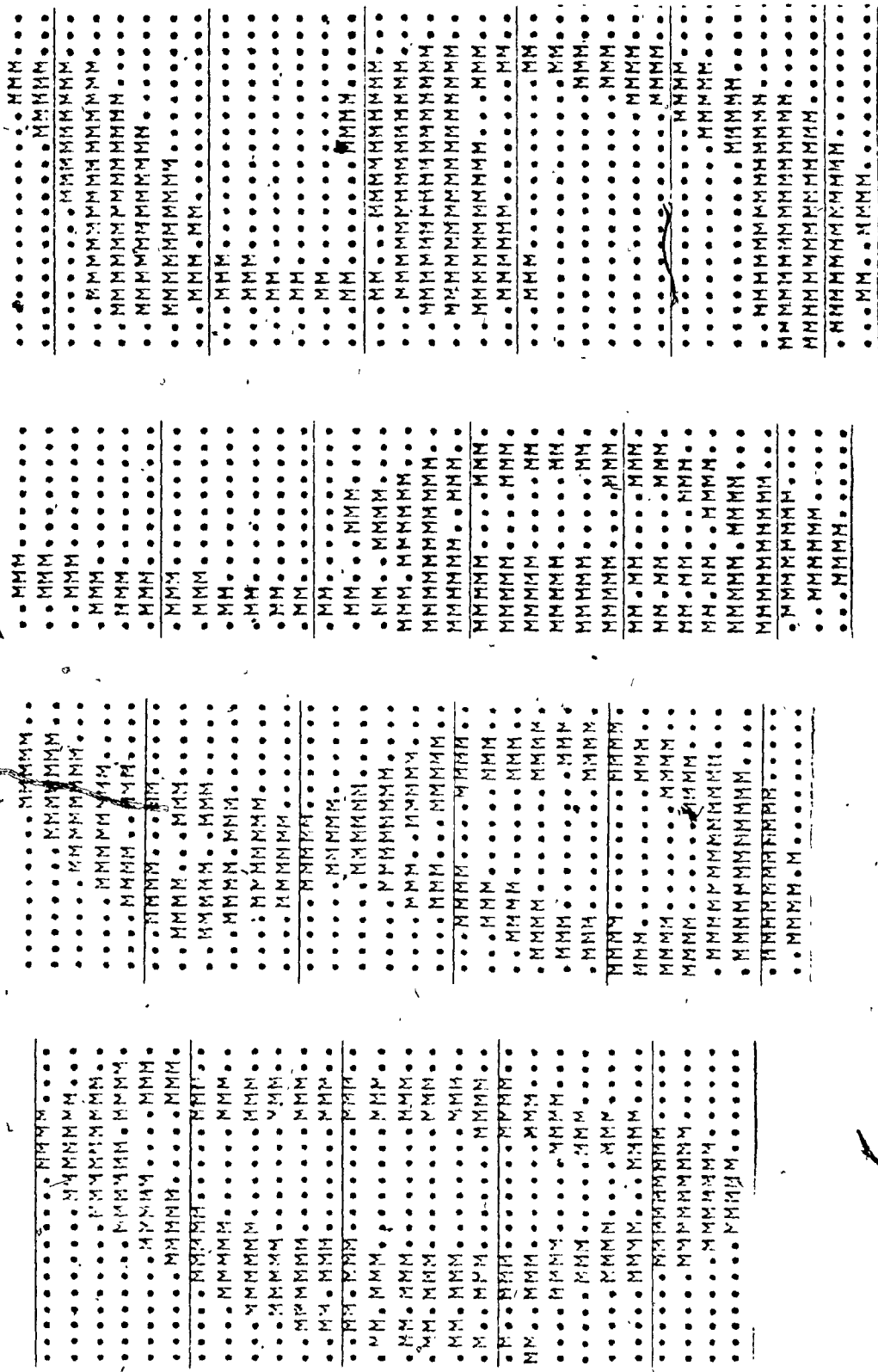


Fig. 4-5 Samples of recognized character

the second choice for character '4' is 3.58 on the average while in this example the difference is only 0.55. Suggestions in section 5.2 will minimize this kind of errors. Some recognized samples are shown in Fig.4-5.

From the results of statistics (one of these results is shown in Table4-2), the most reliable features are the line and boundary crossings. Because of the untrained threshold values, some features are not in their best states. Choosing the most suitable threshold values from statistical results will definitely improve the performance..

TABLE 4-2

FEATURE DISTRIBUTION OF NUMERALS-

	0	1	2	3	4	5	6	7	8	9
TLR	0.056667	0.986667	0.000000	0.000000	0.000000	0.853333	0.016667	0.006667	0.056667	0.066667
TMR	0.000000	0.000000	0.003333	0.000000	0.976667	0.000000	0.033333	0.000000	0.036667	0.000000
TNR	0.250000	0.000000	0.450000	0.243333	0.003333	0.000000	0.000000	0.250000	0.033333	0.853333
BLR	0.326667	0.983333	0.000000	0.000000	0.000000	0.000000	0.673333	0.000000	0.056667	0.000000
BLR	0.000000	0.000000	0.040000	0.000000	0.973333	0.000000	0.766667	0.350000	0.006667	0.000000
BRR	0.120000	0.000000	0.000000	0.303333	0.000000	0.226667	0.143333	0.000000	0.080000	0.960000
TLB	0.016667	0.000000	0.260000	0.000000	0.000000	0.886667	0.000000	0.000000	0.770000	0.120000
MLB	0.236667	0.003333	0.000000	0.980000	0.000000	0.636667	0.043333	0.000000	0.056667	0.833333
TRB	0.176667	0.000000	0.750000	0.516667	0.003333	0.986667	0.000000	0.993333	0.840000	0.830000
MRB	0.000000	0.000000	0.013333	0.473333	0.910000	0.586667	0.806667	0.000000	0.034667	0.410000
BRB	0.733333	0.000000	0.976667	0.000000	0.000000	0.263333	0.093333	0.000000	0.880000	0.000000
TLD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
TRD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
BLD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
BRD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
D1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
D2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
D3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
D4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
LLD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
LRD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
RLD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
RRD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
W1	0.180000	1.000000	0.520000	0.210000	0.776667	0.676667	1.000000	0.073333	0.030000	0.076667
W2	0.036667	0.726667	0.406667	0.820000	0.000000	0.273333	0.063333	0.596667	0.056667	0.033333
W3	0.230000	0.696667	0.006667	0.823333	0.000000	0.730000	0.360000	0.943333	0.100000	0.033333
W4	0.033333	1.000000	0.680000	0.373333	0.716667	0.246667	0.210000	0.276667	0.203333	0.500000
W5	0.140000	0.906667	0.333333	0.920000	0.000000	0.830000	0.466667	0.760000	0.776667	0.500000
W6	0.000000	1.000000	0.406667	0.000000	0.000000	0.023333	0.693333	0.646667	0.000000	0.056667
W7	0.776667	0.996667	0.433333	0.970000	0.020000	0.933333	0.050000	0.053333	0.970000	1.000000
C1	0.876667	0.933333	0.946667	0.140000	0.090000	0.433333	0.576667	0.196667	0.740000	0.946667
C2	0.000000	0.050000	0.003333	0.843333	0.016667	0.073333	0.080000	0.006667	0.893333	0.310000
C3	0.000000	0.056667	0.333333	0.026667	0.006667	0.046667	0.040000	0.006667	0.930000	0.000000
C4	0.960000	1.000000	0.043333	0.176667	0.003333	0.403333	0.416667	0.010000	0.623333	0.000000
C5	0.836667	0.000000	0.656667	0.926667	0.010000	0.036667	0.000000	0.090000	0.280000	0.926667
C6	0.033333	0.000000	0.766667	0.720000	0.050000	0.366667	0.000000	0.973333	0.630000	0.126667
C7	0.000000	0.000000	0.000000	0.416667	0.020000	0.233333	0.066667	0.193333	0.806667	0.150000
C8	0.916667	0.000000	0.170000	0.913333	0.016667	0.900000	0.526667	0.113333	0.570000	0.130000
XT	0.996667	0.000000	1.000000	1.000000	0.173333	1.000000	0.816667	1.000000	0.990000	1.000000
XB	0.000000	0.000000	0.996667	1.000000	0.103333	1.000000	0.760000	0.810000	1.000000	0.003333
XU1	0.010000	0.000000	0.193333	1.000000	0.223333	0.983333	1.000000	0.126667	0.000000	0.000000
XU2	0.990000	0.000000	0.806667	0.000000	0.776667	0.016667	0.000000	0.873333	1.000000	1.000000
XL1	0.010000	1.000000	0.996667	1.000000	1.000000	0.963333	0.000000	1.000000	0.000000	1.000000
XL2	0.990000	0.000000	0.003333	0.000000	0.000000	0.036667	1.000000	0.000000	1.000000	0.000000

CHAPTER 5

CONCLUSION AND SUGGESTIONS FOR FURTHER STUDY

5.1 Conclusion

As mentioned in chapter one, most researchers concentrate in the fields of feature extraction and decision theory, few have paid particular attention to the quality of handprinted data. This thesis confirms the validity of a method of selection of handprinted alphanumeric characters based on "Dispersion Factor" theory which reflects the reproducible nature of a character.

All character models of a character class can be put in order of choice by means of dispersion factor theory. However, in certain cases, the first choice and the second choice are so close that they have a little difference, such as some of the E models which have the same shape but different stroke sequences. This difference may become insignificant when other major factors are taken into consideration. All chosen characters have simple shapes. Besides better recognition potential, which has been shown in chapters 3 and 4, these characters obviously have the advantages of easier writing because of their simplicity.

Different behaviour observed between left-handed and right-handed persons in writing is another fruitful result

obtained in this study. Conflicts appear in the results of character selection due to different writing habit and ability between right-hander and left-hander. An accommodation has to be made between them. The equation (2.1) used in Section 2.4 is proposed as a trial only, not necessarily the best solution.

The algorithms used in chapter 3 have been chosen to minimize the possibility of bias on the character models. However, the results show that they still bring some bias to certain models, such as the '1' in the 'characteristic loci' algorithm. These biased characters have little influence to the overall judgement on the performance of the optimum character set.

Although the new feature extraction system developed in chapter 4 is in its primary stage, the result is very encouraging. The recognition rate obtained is as high as 99.35% and a speed of 12 characters per second is attained. It is felt that further tuning of this system would improve these figures. Since the binary feature vector and the minimum distance classifier can easily be implemented as hardware circuit, an improvement in speed is quite feasible.

5.2 Comments and Suggestions for Further Study

The study in this thesis is quite successful, but further developments should turn this study into a reliable

and powerful recognition system:

- 1) The test data used in this thesis were numerals only, the tests should be extended to the whole alphanumeric data set.
- 2) The features designed in this thesis are simple and easy to detect. It tolerates variation up to 3 columns or 3 rows. The major weakness of this feature set is that no special treatment has been implemented for curve detection. Cursive line drawing is detected indirectly by means of corner crossings (see Section 4.3). Implementation of a simple curve detector will increase the power of the feature set.
- 3) Most of the thresholds used in feature detection were set by best guess. A systematic selection of thresholds from training results will make the detection of feature more accurate.
- 4) It seems that 33 features for recognition of numerals (or totally 45 features for alphanumeric characters) may be too many. Redundant features tend to confuse the weights of the decision functions since not only the features themselves, but also the combinations of features play an important role in the decision functions. Selecting the essential features from the primary feature pool should increase the speed and efficiency of the system.
- 5) Minimum distance classifier is only a basic and general

decision rule. Additional modification may greatly enhance the recognition rate:

a) Establishing second stage of recognition[10,11,40]:

The minimum distance classifier is used as the first recognition rule in a two-stage recognition system. Most of the typical samples will be correctly recognized in this first stage. A threshold can be set to the decision functions in this stage so that any rejected unknown sample will go into the second stage for detailed examination. "Similar" characters which bring high confusions in the primary recognition phase can be grouped together. A sample rejected from the first stage will fall into one of these groups. In each group, features which favour a certain character will be assigned the appropriate "confidence" marks when they are present in that character. The relative importance of a feature to the character decides the weight of the mark. Final decision is in favour of the heaviest marked character in the group. In addition to the original features, special new features may be created (in special purpose) for discriminating some confused characters in the second stage.

b) Weight modification[41]:

Adjusting the weights in the discriminate functions so that they have the greatest separation power to character classes. One way to do it is by applying

linear programming technique to derive a set of weights based on the set of ideal feature vectors of the character models.

c) Combination of a) and b).

Results obtained in this study were entirely based on the reproducibility of a character. The chosen alphanumeric character set may not be the final optimum character set, it should be compared with the results obtained by other considerations, such as writing speed, distance measurements among characters[27], to make the best compromise. It is hoped that the works done in this study could give some valuable ideas to those who are interested in this area of research.

REFERENCES

1. Dineen G.P., 'Programming Pattern Recognition', Proceedings of the Western Joint Computer Conference, 94-100, 1955.
2. Hennis R.B., 'The IBM 1975 Optical Page Reader, Part I: System Design', IBM Journal of Research and Development, Vol.12, No.5, 346-353, Sept. 1968.
3. Griffith A.K., 'The GRAFIX I System and Its Application To Optical Character Recognition', Proceedings of the 3rd International Joint Conference on Pattern Recognition, 650-652, Nov. 1976.
4. Suen C.Y., 'Factors Affecting the Recognition of Handprinted Characters', Proceedings of the International Conference on Cybernetics and Society, 174-175, Nov. 1973.
5. Suen C.Y., 'Human Factors in Character Recognition', Proceedings of the International Conference on Cybernetics and Society, 253-258, Oct. 1974.
6. Harmon L.D., 'Automatic Recognition of Print and Script', Proceedings of the IEEE, Vol.60, No.10, 1165-1176, Oct. 1972.
7. Lindgren N., 'Machine Recognition of Human Language, Part III-Cursive Script Recognition', IEEE Spectrum, Vol.2, 104-116, May 1965.

8. Genchi H., Mori K., Watanabe S. and Katsuragi S., 'Recognition of Handwritten Numeral Characters for Automatic Letter Sorting', Proceedings of the IEEE, Vol.56, No.8, 1292-1301, Aug. 1968.
9. Levine M.D., 'Feature Extraction: A Survey', Proceedings of the IEEE, Vol.57, No.8, 1391-1407, Aug. 1969.
10. Tou J.T. and Gonzalez R.C., 'Automatic Recognition of Handwritten Characters Via Feature Extraction and Multi-level Decision', International Journal of Computer and Information Science, Vol.1, No.1, 43-65, 1972.
11. Caskey C.L. and Coates, Jr. C.L., 'Machine Recognition of Handprinted Characters', Proceedings of the 1st International Joint Conference on Pattern Recognition, 41-49, Oct. 1973.
12. Teitelman W., 'Real Time Recognition of Hand-drawn Characters', Proceedings of the Fall Joint Computer Conference, 559-575, 1964.
13. Miller G.M., 'On-line Recognition of Hand-Generated Symbols' Proceedings of the Fall Joint Computer Conference, 399-412, 1969.
14. Berthod M. and Maroy J.p., 'Morphological Features and Sequential Information in Real-Time Handprinting Recognition', Proceedings of the 2nd International Joint Conference on Pattern Recognition, 358-363, Aug. 1974.
15. Munson J.H., 'The Recognition of Hand-printed Text', Pattern Recognition, edited by L.N. Kanal, Thompson Book Co., 115-139, 1968.

16. Dimond T.L., 'Devices for Reading Handwritten Characters', Proceedings of the Eastern Joint Computer Conference, 232-237, 1957.
17. Spanjersberg A.A., 'Experiments with Automatic Input of Handwritten Numerical Data into a Large Administrative System', Proceedings of the International Conference on Cybernetics and Society, 476-478, Nov. 1976.
18. Holt A.W., 'Algorithm for a Low Cost Hand Print Reader', Computer Design, 13, 85-89, Feb. 1974.
19. Lin W.C. and Scully T.L., 'Computer Identification of Constrained Handprinted Characters with a High Recognition Rate', IEEE Trans. On Systems, Man and Cybernetics, SMC-4, 497-504, Nov. 1974.
20. Suen C.Y., Private Communication.
21. Apsey R.S. 'Human Factors of Constrained Handprint for OCR', Proceedings of the international Conference on Cybernetics and Society, 466-470, Nov. 1976.
22. Kuhl F., 'Classification and Recognition of Handprinted Characters', IEEE International Convention Record, Vol.II, Part 4, 75-93, March 1963.
23. American National Standards Institute, 'Presentation of Alphanumeric Characters for Information Processing', Communications of ACM, Vol.12, No.12, 696-698, Dec. 1969.
24. American National Standards Institute, 'Proposed American National Standard Character Set for Hand-Printing', Document No. X3A1/72-60, 1972.

25. Mori S., Mori T., Yamamoto K., Yamada H. and Saito T.,
'Recognition of Handprinted Characters', Proceedings of
the 2nd International Joint Conference on Pattern
Recognition, 233-237, AUG. 1974.
26. Suen C.Y., 'Optical Character Recognition --the State
of the Art Report', Canadian Datasystems, Vol.6, 40-44,
May 1974.
27. Suen C.Y., Shiau C., Shinghal R. and Kwan C.C.,
'Reliable Recognition of Handprint Data', Proceedings of
the Joint Workshop on Pattern Recognition and Artificial
intelligence, 98-102, June 1976.
28. Knoll A.L., 'Experiments with "Characteristic Loci" for
Recognition of Handprinted Characters', IEEE Trans, On
Computers, 366-372, April 1969.
29. Spooner M.G. and Ahlgren R.C., 'An Experimental
Evaluation of an Incremental Scanning and Recognition
Technique for Alphanumeric Character Recognition',
Proceedings of the International Conference on
Information Processing, UNESCO, Paris, 481-499, 1959.
30. Suen C.Y., Shinghal R. and Kwan C.C., 'Dispersion
Factor: A Quantitative Measurement of the Quality of
Handprinted Characters', Accepted for publication,
Proceedings of the International Conference on
Cybernetics and Society, Sept. 1977.
31. Burns P.C., 'Improving Handwriting Instruction in
Elementary Schools', 2nd edition, 1968, Burgess
Publishing Company.

32. Bledsoe, W.W. and Browning I., 'Pattern Recognition and Reading by Machine', Proceedings of the Eastern Joint Computer Conference, 301-316, 1959.
33. Bledsoe, W.W., 'Further Results on N-tuple Pattern Recognition Method', IRE Trans. On Electronic Computers, EC-10, 96-97, March 1961.
34. Bledsoe W.W. and Bisson C.L., 'Improved Memory Matrices for the N-tuple Pattern Recognition Method', IRE Trans. On Electronic Computers, EC-11, 414-415, June 1962.
35. Glucksman H.A., 'Classification of a Mixed-Font Alphabets by Characteristic Loci', 1967 Digest of the 1st ann. IEEE Computer Conference, 463-479, Sept. 1967.
36. Spanjersberg A.A., 'Combinations of Different Systems for the Recognition of Handwritten Digits', Proceedings of the 2nd International Joint Conference on Pattern Recognition, 208-209, Aug. 1974.
37. Michael M.T., 'Feature Evaluation Criteria for Pattern Recognition', Ph.D. Dissertation, Case Western Reserve University, June 1972.
38. Nilsson N.J., 'Learning Machines', McGraw-Hill, 1965.
39. Ullmann J.R., 'Pattern Recognition Techniques', Butterworths and Co. 1973.
40. Backer E., 'Two-Step Discrimination of Handwritten Numerals', Advances in Cybernetics and System, Vol. 1, J. Rose(ed.), Gordon and Breach Science Publishes, 373-388, 1974.

41. Tou J.T. and Gonzalez R.C., 'Pattern Recognition Principles', Addison-Wesley Publishing Corp. Inc., 1974.