AN ADAPTIVE RECOGNITION SYSTEM

FOR MACHINE PRINTED CHARACTERS USING MICROCOMPUTERS

Wai Yip Chan

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements

for the degree of Master of Computer Science at

Concordia University

Montreal, Quebec, Canada

January 1979

ABSTRACT

AN ADAPTIVE RECOGNITION SYSTEM FOR MACHINE PRINTED CHARACTERS USING MICROCOMPUTERS

Wai Yip Chan

The recognition of machine printed characters is considered in this thesis. An adaptive recognition system has been designed and developed to recognize machine printed characters using microcomputers. The same system can be used to recognize OCR character fonts as well as common English and French character fonts by acquiring an automatic font analysis prior to reading without human intervention.

A two stage classifier using two sets of features, crossing number code (CNC) and grid point representation (GPR), has been developed and implemented on the microcomputer. The system was tested with more than 15,000 real character samples collected from the full character set of OCRA, OCRB, PICA10, ELITE12, and PRESTIGE CUBIC (French) fonts. Some encouraging recogniton rates have been observed --- 99.79% (OCRA), 99.59% (OCRB), 99.35% (PICA10), 97.10% (ELITE5) and 94.63% (PRESTIGE CUBIC).

Characteristic and behavior of the system have been studied through a program simulated on a large scale computer and a special development system for OCR has also been designed to assist OCR system software development. Font independent OCR data input and reading machine for the blind are some potential applications of our system.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my thesis supervisor for his able guidance, encouragement and fruitful suggestions and comments for this research.

I also wish to express my gratitude to J. Mulherin for his invaluable help in building the hardware and collecting the real character samples. My special thanks go to the staff of computer centre for their patience and cooperations in running the computer jobs for this research.

This study is supported by the Department of Education of Quebec.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

AN OVERVIEW OF OPTICAL CHARACTER RECOGNITION

This chapter describes the basic concepts and applications of optical character recognition (OCR) and its trend in the future. It stresses the impact of the latest technologies on the design of an OCR system and points out the advantages and disadvantages of an adaptive over conventional OCR system. With these basic concepts, the scope of this thesis is presented at the end of the chapter.

1.1 OPTICAL CHARACTER RECOGNITION

The idea of making use of a machine to recognize printed materials probably started in 1929 [1]. At first glance, the problem seems to be very simple and straight forward; however, as the research went on, a lot of factors that are not important to human beings turn out to be quite critical to machine in the recognition process [2].

Nearly all the early research work done in this area employed analog signal processing techniques. As a common way to gather analog signal out of an image, optical and electrical devices are usually used in measuring light intensity reflected from the image and thus the term optical character recognition (OCR) is generally known.

Since the introduction of large scale general purpose digital computers, most of the research in OCR has switched from analog to digital signal processing which is proven to be a more flexible and powerful approach.

As far as the interest of the researchers is concerned, character recognition can be divided into hand written and machine printed character recognition.

Recognition of hand written characters is in general more challenging than the recognition of machine printed characters. Besides the noise due to quality of paper, ink, pen and pressure applied, the boundless variation of the shape of hand written character is the major problem encountered. The "variability" problem of the hand written character is caused by diverse factors like mood, purpose and environment of writing [2].

According to the way of writing, work in the field of hand written character recognition is classified as cursive script recognition and hand printed block letter recognition (also known as hand printed character recognition).

The main problem associated with cursive script character recognition is the identification of individual letters within words. Although much research has been done in this area, yet the result is still preliminary and many researchers have adopted alternative approaches, like word identification and real time recognition that analyze the

pen motion during writing [3,4,5]. However, the result is not that satisfactory when compared to hand printed character recognition.

Hand printed character recognition is more encouraging when speaking of recognition rates. Performance in character recognition is usually measured by the "rejection rate" and the "substitution rate". The latter is a measure of the actual errors. The former is what the system considers unrecognizable. Some systems with a substitution error rate less than 5 percent are reported in many researches [6]. A considerable amount of this kind of research is confined to numerals only. This can be easily understood by considering the widespread use of numerals in daily life. Usually, topological feature of one form or another is used in this case. Tou and Gonzalez [7] have developed one such system using multistage classifier. A comprehensive survey on print and script recognition can be found in references [8,9,10,11].

On the other hand, machine printed character recognition is generally classified into different groups based on the ability to distinguish shapes. In general, there are four types of systems[12].

1) Single font system -- it is dedicated to a single printed or typewritten font.
2) Multiple font system -- it can read a few selected groups of printed or typewritten fonts. Usually, only

one particular font is active at a time.

3) Multi-font system -- it has the capability to read a variety of fonts that can be intermixed on a given page.

4) Omni-font system -- it trains itself to read any font by analyzing a given character set prior to reading.

Many commercially available systems may even allow a limited number of hand printed special symbols and numerials to be recognized in addition to the typewritten characters. Most of the systems recognize 20 to a few thousand characters per second with a substitution error rate of 0.1 percent or less and a rejection rate between 3 to 5 percent [12,14]. The cost of a typical commercial viable OCR system ranges from a few thousand to a few hundred thousand dollars depending on the performance of the system [12].

There are many applications of OCR in daily life, such as bank checque processing, utility bills, credit card charges, sale and inventory documents processing, label and address readers, journal tape reading, mail sorting, reading for the blind and data entry to computer. In an era in which computer speed is given in terms of nano-seconds, computer input is still measured in terms of seconds. The bottle neck of data entry to computer becomes obvious when the need of rapid processing of large volume of data is increasing. OCR has been considered as the most promising technique to solve the problem. Since a majority of the raw data are already in machine printed form, it would be ideal

to have a system that can accept machine printed materials as direct input data. A detailed analysis of using OCR versus conventional keypunching data entry is given in reference [12].

1.2 DEVELOPMENT OF STYLIZED FONTS AND STANDARDS IN OCR

The non-standardization of character type fonts is the major problem in machine printed character recognition. As stated in reference [13], there are roughly two thousand print and typewritten character fonts currently in use. In spite of the problem arising from variations in size and style, one would expect that better and less expensive OCR system could be designed and fabricated by incorporating some standards on the type fonts in which characters are printed.

An early pioneer in this work was Farrington [12], who developed two "self-check" fonts which provide at least two distinguishing features per character. In 1966 the American National Standards Institute (ANSI), under the sponsorship of the Business Equipment Manufacturers Association, completed a thorough study of OCR requirements and adopted a standard character set, known as OCR-A character set [19]. This set contains upper case and lower case alphabets, numerals and special symbols. During the development phase of ANSI's OCR project, the International Standardization Organization (ISO), adopted its own "style B" standard character which is known as OCR-B character set [20].

Although OCR-B set is aesthetically better than OCR-A set, it is claimed that OCR-A set is better suited for OCR equipment.

Almost all commercial OCR units recognize only a proper subset of OCR-A or OCR-B character sets [12,14,21] shown in fig. 1.1.

1.3 ELEMENTS OF AN OCR SYSTEM

All existing optical readers consist of six basic elements -- transportation unit, scanner, recognition unit, controller, output stacker and data output unit as functionally shown in fig. 1.2.

The transportation unit and the output stacker hahdle the flow of the documents, the scanner digitizes the document into binarized forms suitable for the recognition process, the recognition unit carries out the actual classification process that recognizes the information on the input document and the data output unit outputs this to the user assigned destination. The flow of the whole process is under the supervision of the controller.

The level of capability of an OCR system is governed by the performance of the recognition unit. In other words, the recognition unit is the heart of an OCR system. Basically the recognition unit can be functionally partitioned into three modules --- the preprocessing module,

OCR-A

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

1 2 3 4 5 6 7 8 9 0 $ % & * + - = " ' : ; / . , ?

OCR-B

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

1 2 3 4 5 6 7 8 9 0 $ % & * + - = " ' : ; / . , ?

Fig. 1.1 OCR CHARACTER SETS

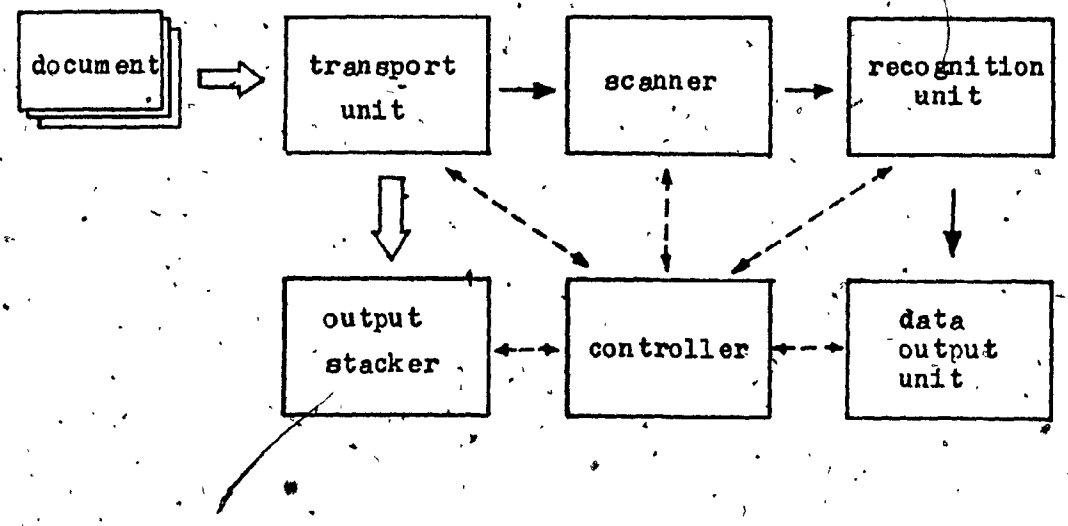


Fig. 1.2 A TYPICAL OCR SYSTEM

the feature extraction module and the classification module.

The function of a preprocessing module is to make the binarized pattern from the scanner more suitable for recognition. Noise elimination, normalization and justification are some of the common routines of a preprocessing module. Usually, information from the raw image of a character is scattered and subjected to distortion. To extract a concise and unique representation of the given pattern in the presence of distortion is the main task of the extraction module. Based on the selected features, an unknown pattern will be classified or rejected according to some pre-stored decision rules. All these descisions are made inside the classification modules. More details of the design of our recognition unit is given in chapter 2.

1.4 LATEST TECHNOLOGIES AND ITS IMPACT ON OCR DESIGN

The rapid development in semiconductor technology has lowered the price and increased the performance of most semiconductor products. Many electronic devices which were prohibitive to an OCR system designer before for their sky high prices are now available at a reasonable price. The slow mechanical movement during a scanning process can now be reduced to a minimum by using a page width photo-diode matrix as a scanning device. The inexpensive shift registers and fast access memories eliminate the need of

rescanning of the same image several times [15]. Among the very many latest LSI products, single chip microprocessors have added a new dimension to the digital system design and numerous applications.

A microprocessor is a general purpose processing unit with the capability of handling arithmetic and logical control over digital signals and is now commercially available with a price as low as a few dollars [16]. As a result of the competition among the semiconductor manufacturers, microprocessors are still experiencing a drop in price and an increase in performance. This trend has turned the microprocessor into a high potential candidate for inexpensive and moderately complicated control systems, such as a portable OCR machine [17,18].

In general, a microprocessor is much slower than a large scale digital machine. A typical instruction cycle is about 1 to 5 microseconds depending on the size of instruction repertoire. Despite its slow speed, one can design and develop OCR systems with acceptable speed and performance by incorporating several microprocessors working together [17]. By employing inexpensive microprocessor based OCR systems as data entry systems, large scale computers can be released to handle more complicated accurate calculations and manipulations of data. More and and more applications of the microprocessors in the OCR system design is expected in the near future. More details of our microprocessor

applications in this area is given in chapter 4.

1.5 AN ADAPTIVE OCR SYSTEM AND THE OBJECTIVE OF THE THESIS

Nearly all the commercially available OCR machines can recognize only one or two specific type fonts. The recognition logic is generally hard wired or burnt inside read only memory (ROM) to recognize only one or two particular type fonts. It is almost impossible for such an OCR machine to be adapted to recognize new type fonts or additional symbols without a great drop in its recognition rate. This rigidity has prevented OCR from being widely adopted in the data processing business. Although standards in OCR have been studied and developed, it does take time to descipline people to adopt new standards. It would be ideal to have an OCR system that can be adapted to recognize any new type font or extended to accept additional symbols with little or no modification on hardware and software of the OCR system. For the time being, SCANDATA 2250 and GRAFIX I [12,22] are the only systems that can "acquire", if necessary, fonts immediately before reading the data and are therfore close to what we proposed above. However, there are still a lot of imperfections. Further research in this field is necessary.

In order to distinguish this type of OCR system from the conventional one, we refer to it as an "adaptive" OCR system. One of the problems faced by an adaptive OCR system

is the variation in type sizes as well as type styles. The area occupied by a character needs to be defined in order to determine its shape, which is difficult to extract when type sizes vary. This problem is encountered also by those OCR readers that must deal with proportional spacing in which case the area containing the character is not predictable. Besides this, if the size of the character set of an OCR system is allowed to be adjusted dynamically, the internal data structure must also be dynamic. This would greatly complicate the system software owing to possible changes in the internal memory management. Again, the features employed in an adaptive OCR system should be more general in nature such that it can be used to describe any type fonts without loss of generality.

The main objective of this thesis is to develop an adaptive system for the recognition of machine printed characters using microcomputers and to study the characteristics and behavior of such a system. Furthermore, the flexibility and ease of using a microcomputer based development system to design an OCR system is also reported.

1.6 OUTLINE OF THE THESIS

Recognition methods for machine printed characters and the influence of microcomputer on how to design an OCR system is briefly given in chapter II. Along with this the actual design of an adaptive recognition system for machine printed

characters has been described. Chapter III presents how the designed system is simulated on a large scale computer. It also contains a detailed study of the characteristics of the present recognition system against the size of the trainning set, type fonts and different feature sets. Chapter IV contains a brief survey of microcomputer systems and the design of a large scale computer supported microcomputer development system. The microcomputer based software development and data structure for the proposed OCR system are described in chapter V. The final chapter summarizes the achievement and contribution of this research with some suggestions for further work in this area.

CHAPTER II

DESIGN OF AN ADAPTIVE RECOGNITION SYSTEM

This chapter reviews some basic concepts and methodologies in character recognition and the possibility of adopting the existing methods to a microcomputer based OCR system. Finally, the actual design of the proposed adaptive recognition unit for machine printed character is presented.

2.1 THE RECOGNITION PROCESS

The two stages of character recognition, deriving the decision rule and using it, can be performed concurrently or sequentially [24]. In a sequential procedure, as shown in fig. 2.1a, all the labeled character samples are collected and the best decision rule based on those samples is derived. That decision rule is used without change to classify the unlabeled samples. On the other hand, as shown in fig. 2.1b, the decision rule is modified as it is used. In this case, a sample is presented and classified; and an error detector or a teacher indicates whether the classification is correct and the decision rule is left unchanged or modified as appropriate. We refer to this latter type of recognition process as an "adaptive" recognition process.

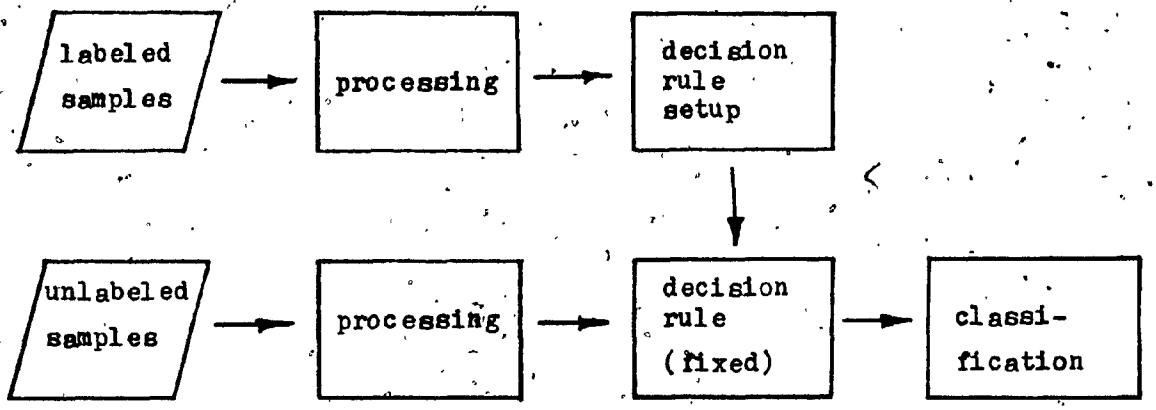


Fig. 2.1a LEARNING BEFORE RECOGNITION

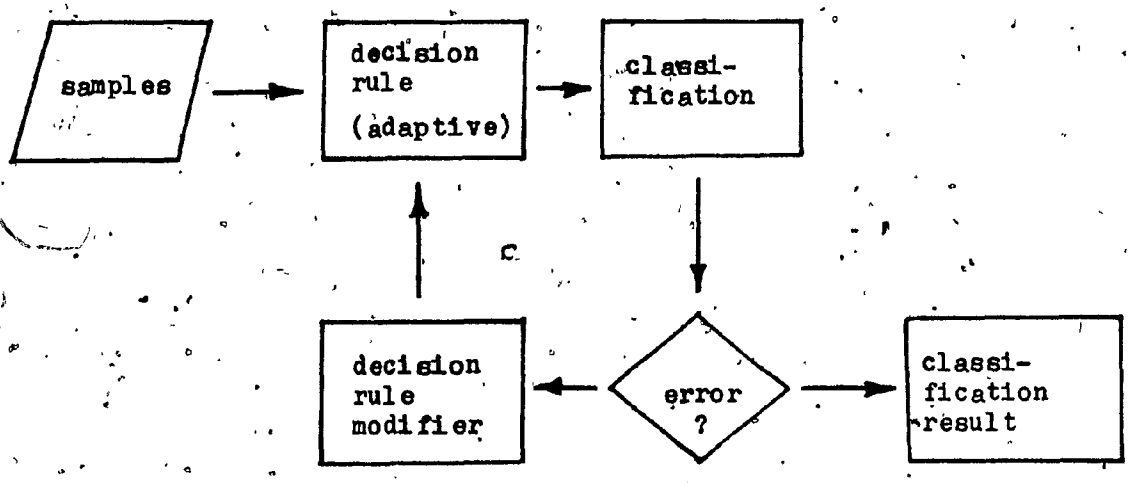


Fig. 2.1b LEARNING AND RECOGNITION CONCURRENTLY

The necessary processes in deriving the decision rule can be represented by the block diagram shown in fig. 2.2. The original raw data in the physical space enters the pattern space through a digitization process. The object of preprocessing and feature extraction is to reduce the variations between different patterns which belong to the same pattern class while at the same time stress the distinctions between different classes. Thus a finite and usually smaller set of descriptors of the pattern is obtained in the reduced pattern space. Finally, we must use a decision rule to classify an unlabeled sample in the reduced pattern space. Various kinds of problems exist at each of these stages and no general solution has ever been found [23]. The design of the recognition unit depends completely on the overall philosophy of the recognition process and the characteristics of the hardware with which the system is implemented. The hardware aspect is disregarded by many researchers who are engaged in OCR research, but as we shall see in the following sections, several design parameters, which dominate the performance of the recognition, are governed by the hardware limitations.

2.1.1 SOURCES OF NOISE AND PREPROCESSING

There are two major sources of noise commonly encountered in character recognition systems. One is the inherent noise associated with the document itself. Noise might be introduced into the documents at different stages. During

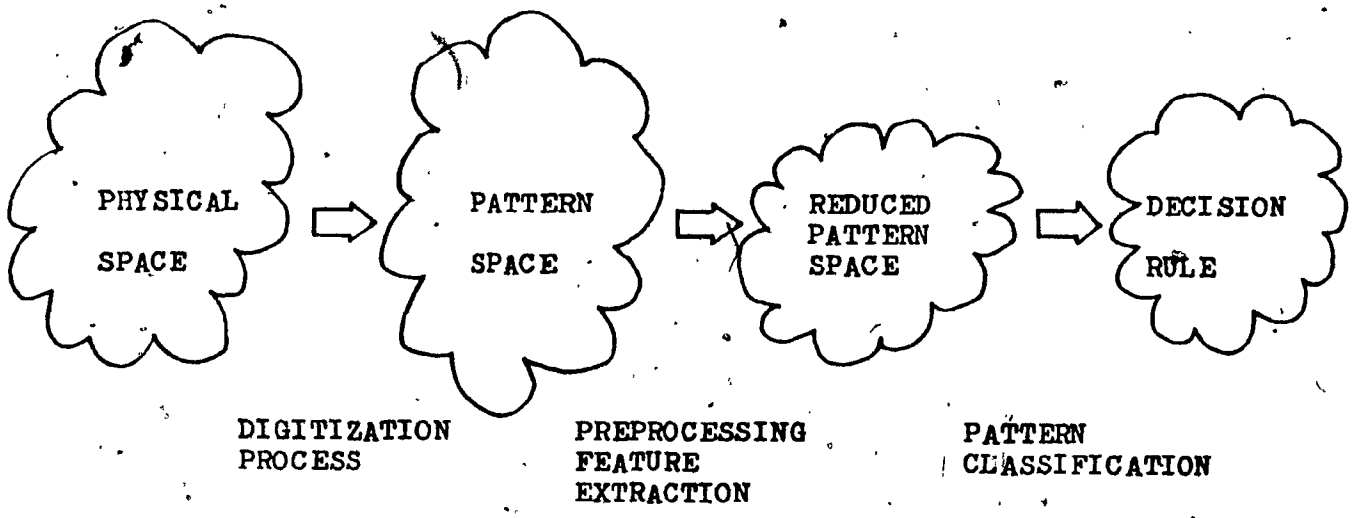


Fig. 2.2 STAGES IN DERIVATION OF DECISION RULE

the preparation of the document, noise might be generated by

(a) wear and tear of the type face,

(b) quality of the ribbon used,

(c) thickness and reflectance of the paper used, and

(d) contact between the type ball and the paper.

Some examples of the above printing defects are shown in fig. 2.3.

Additional noise might be introduced to the document during transportation and storage due to smudging, folding, shrinking and expansion of the paper.

The other source of noise comes from the transportation and scanning unit of the OCR system, such as

(a) improper position and feeding of the document,

(b) the condition of illumination,

(c) sensitivity of the scanning device,

(d) print contrast ratio and paper thickness and so on.

The most common noise observed to occur in a digitized pattern is the smearing around the edge of the character image and discontinuity of the character strokes.

A practical OCR system is capable of eliminating the noise mentioned above. Usually, it is done by a preprocessor. Besides the elimination of noise, preprocessing usually covers various kinds of video enhancement procedures, such as size normalization, stroke

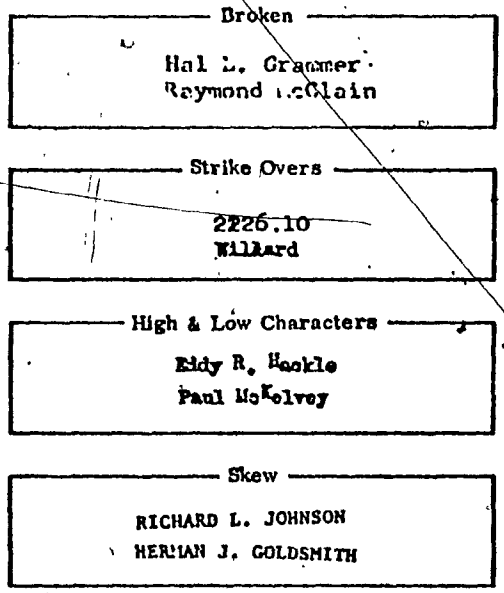


Fig. 2.3 EXAMPLES OF PRINTING DEFECTS *

* This figure is taken from reference[23].

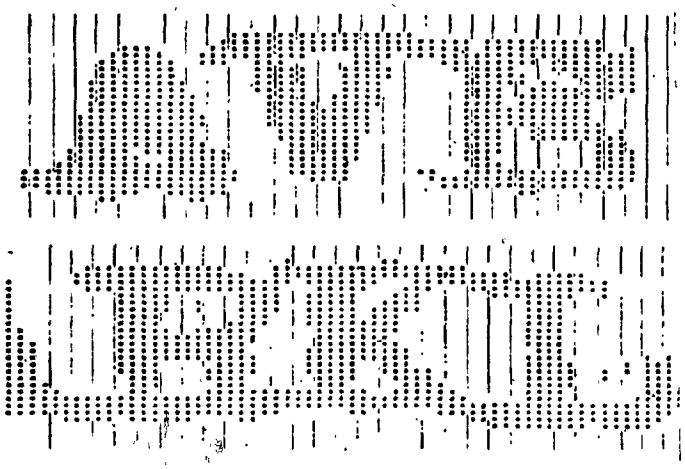
width standardization (skeletonization), character separation (segmentation), character location (registration), removing tilt (de-skewing) and reduction of pattern size (consolidation), (fig. 2.4). Some common noise elimination and video enhancement algorithms can be found in references [25,26,27,38].

2.1.2 DIMENSION REDUCTION AND FEATURE EXTRACTION

The pattern space obtained directly from the scanning unit tends to have very high dimensions. For economical reasons, a high dimensional pattern space is usually transformed into a lower dimensional space before recognition takes place. This kind of video transformation is usually known as feature extraction. There is no fixed method to perform this kind of transformation. One might use one's knowledge of the problem at hand to extract pertinent information. In the character recognition problem, features of a reduced space could be the number of grid points covered by the character, the number of crossings of a horizontal line, geometrical properties of that character, and so on. However, as mentioned before, a good feature extraction algorithm should minimize variations between different patterns within the same pattern class and stress the distinctions between different pattern classes. Moreover, the condition of the input pattern affects the choice of preprocessing and feature extraction algorithms as well as the performance of the classification scheme used. A

STROKE WIDTH STANDARDIZATION
(required for hand printed data)

SIZE NORMALIZATION
(required for hand printed data)



CHARACTER SEGMENTATION
(required for machine printed data)

Fig. 2.4 REQUIREMENT FOR VIDEO ENHANCEMENT *

* This figure is taken from reference [23].

detailed description and discussion of the most common feature extraction techniques can be found in references [24,28,29,30].

2.1.3 DECISION RULE AND PATTERN CLASSIFICATION

Pattern classification is a procedure used to classify a point in the reduced pattern space corresponding to an unlabeled sample by a decision rule.

Mathematically speaking, pattern classification is the same as the partitioning of the reduced pattern space into disjoint regions such that each region is associated with only one class. Thus a point in the reduced pattern space can be classified according to the region to which it belongs.

For supervised learning, all the labeled samples are grouped together according to their labels to form classes in the reduced pattern space. Different classification schemes can be used to describe the hyperplane that separates adjacent classes.

One of the decision schemes classifies a point as a member of the class to which its nearest neighbor belongs. If the membership is decided by a majority vote of the K nearest neighbors, it is known as K-nearest neighbor decision rule. Suppose we have a "perfect" example for a particular class, and every member of that class is

considered as a distortion of that prototype. There exists a very simple decision rule such that a point is assigned to a class according to the minimum distance of the point and the prototype of each class. In fact, this is equivalent to the K-nearest neighbor decision rule with one sample of each class. Example of this classification scheme is template matching where an unknown pattern is classified according to pre-stored prototypes known as templates.

A hierarchical approach can also be applied to make a decision. One can always define subsets by breaking the space into subregions and breaking the subregions into further smaller subregions and so on. Thus the reduced pattern space can be partitioned into a hierarchical structure. Tou and Gonzalez's [7] multi-stage classifier is an example of this type.

Usually classifiers are based on distance measurements of one form or another. Some mathematical descriptions of the most common types of distance measurements are covered in references [24,28,29,30,31].

2.2 THE PROPOSED OCR SYSTEM

Before we go into the design details of the proposed OCR system, some characteristics of the equipment are presented in the following. For this project, a hand held scanner, a microcomputer system and an interface between the scanner

and the microcomputer are available.

The scanner we used is a 54-cell photodiode array digitizer. Its vertical resolution is approximately 4.5 mm/54 cells and the horizontal resolution depends on how often we read in the sampled signals. A simplified diagram, as shown in fig. 2.5, shows how this scanner operates.

The microcomputer system we have is a MPS-885 [32] microcomputer which is an INTEL 8080A microprocessor based small system. The standard configuration of the MPS-885 consists of an 8080A microprocessor, 2K bytes of read only memory (ROM), 4K bytes of random access memory (RAM) and standard TTY I/O ports. The 8080 series microprocessor is an 8 bit machine with 8 bit and 16 bit arithmetic capability but no direct multiplication and division instruction. A typical 8 bit addition takes 8 microseconds.

The scanner and the microcomputer are connected in parallel. The memory mapped input output method is used to control and access the sampled signal from the scanner.

The vertical scanning range (4.5 mm) of our hand held scanner is sufficient to digitize most of the commercially available type fonts. In order to achieve optimal usage of the memory, the dimensions of the digitized pattern are normalized to be multiples of 8-bits (fig. 2.6) such that it can be stored in bytes.

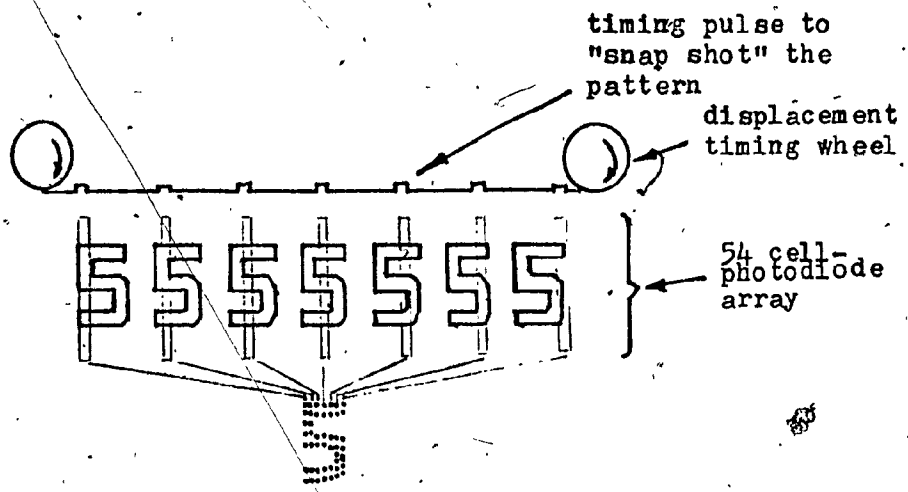


Fig. 2.5 A SIMPLIFIED DIAGRAM OF
THE OPERATION OF THE SCANNER

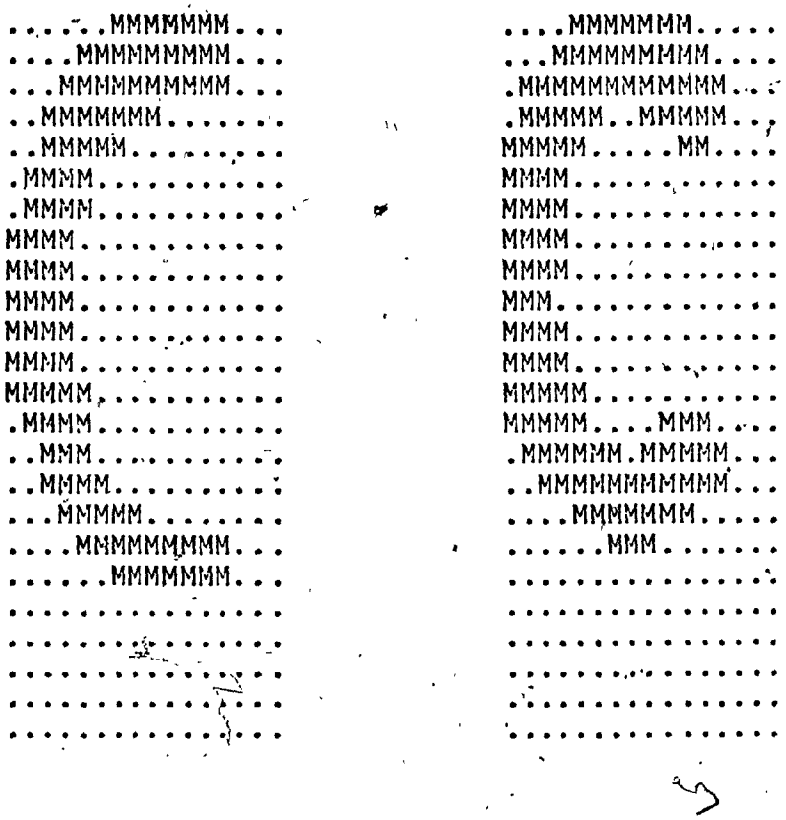


Fig. 2.6 SAMPLES OF DIGITIZED PATTERNS

A block diagram of the proposed adaptive OCR system is shown in fig. 2.7. Each component of the system is described below.

2.2.1 PREPROCESSOR

The functions of the preprocessor falls into two categories, namely

(a) scanner function -- character registration and consolidation.

(b) video enhancement -- noise elimination and justification.

When a document is being scanned, an OCR system should be able to locate a single character and digitize it into a preformatted binary pattern.

In our proposed system, the scanner transmits 54 bits to the microcomputer at a time, corresponding to a column in the digitized pattern. For character registration, we have a small program that takes care of this. The steps to follow depends on the status of the current column:

[STEP 1] If the current column is a non-blank column, skip until a blank one occurs.

[STEP 2] If the current column is a blank column, skip until a non-blank one occurs.

[STEP 3] Read in 16 consecutive columns and place them in the working buffer.

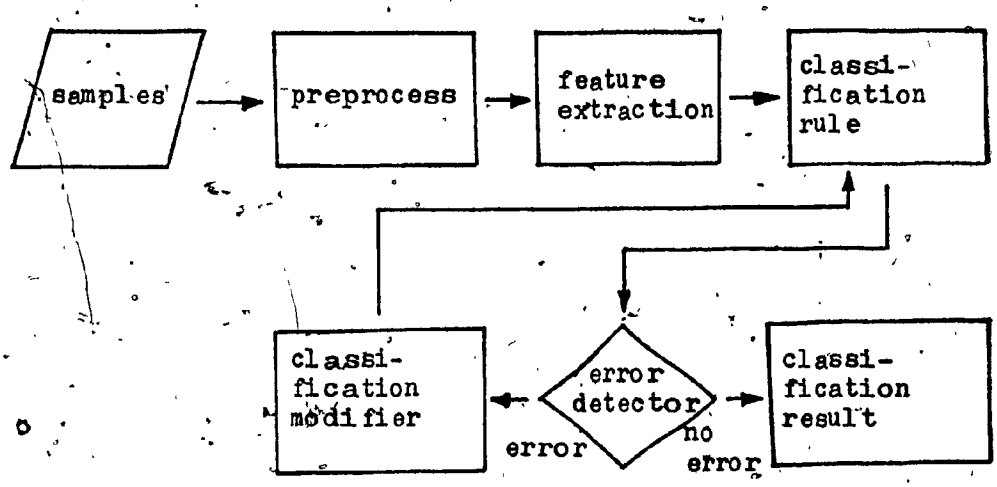


Fig. 2.7 A BLOCK DIAGRAM OF
THE PROPOSED ADAPTIVE OCR SYSTEM

As we can see, if random noise exists in the background of the document, the above algorithm will not work correctly. In order to overcome this problem, we redefine the conventional meaning of a blank column. Since the 54 bits received from the scanner are arranged consecutively and are available in 7 bytes of memory locations, we perform a logical "OR" operation with these 7 bytes and count the number of non "0" bits of the result. If the number of non "0" bits is greater than 2, the column just received is considered as a non-blank column; else it is a blank column. With this new definition, the above algorithm works as expected and the pattern obtained is already left justified. However, the size of the resulting pattern is still a little bit too big to be packed and stored economically. In order to reduce the size of the pattern further, the following algorithm is proposed.

[STEP 1] Get next 2 rows of the pattern and "AND" them together.

[STEP 2] If the number of non "0" bits of the result is less than 2, go to step 1.

[STEP 3] Get 2 rows of pattern and "AND" them together.

[STEP 4] Transmit the result to the pattern buffer.

[STEP 5] Have all 24 rows been transmitted? If false, repeat step 3.

With the above algorithm the final pattern obtained from the scanner is top left justified measuring 24 by 16 bits. Most of the salt and pepper noise is eliminated as a by-

product of the logical "AND" operation (fig.2.8).

The design of the above two algorithms has the advantages of combining the scanner and enhancement functions together. Thus redundance of retracing the same pattern several times is avoided.

2.2.2 FEATURE EXTRACTOR

Before we describe how the feature extractor works, we have to determine what type of feature is going to be used. Since we are designing an adaptive system that can be trained to recognize any type font, we must have some kind of representation that is general enough to do so. Fourier descriptor is one of the ideal descriptors for this purpose; however, it requires real number multiplication and division which are not suitable for microcomputer systems. As we stated above, there is no fixed method to select features and the selection is usually intuitive and empirical.

We start our design of the feature extractor by first studying the possibility of adopting those features well developed by other people. With the limitation of microprocessor and the adaptive nature of our proposed system in mind, we select a couple of simple and general descriptors as our preliminary features.

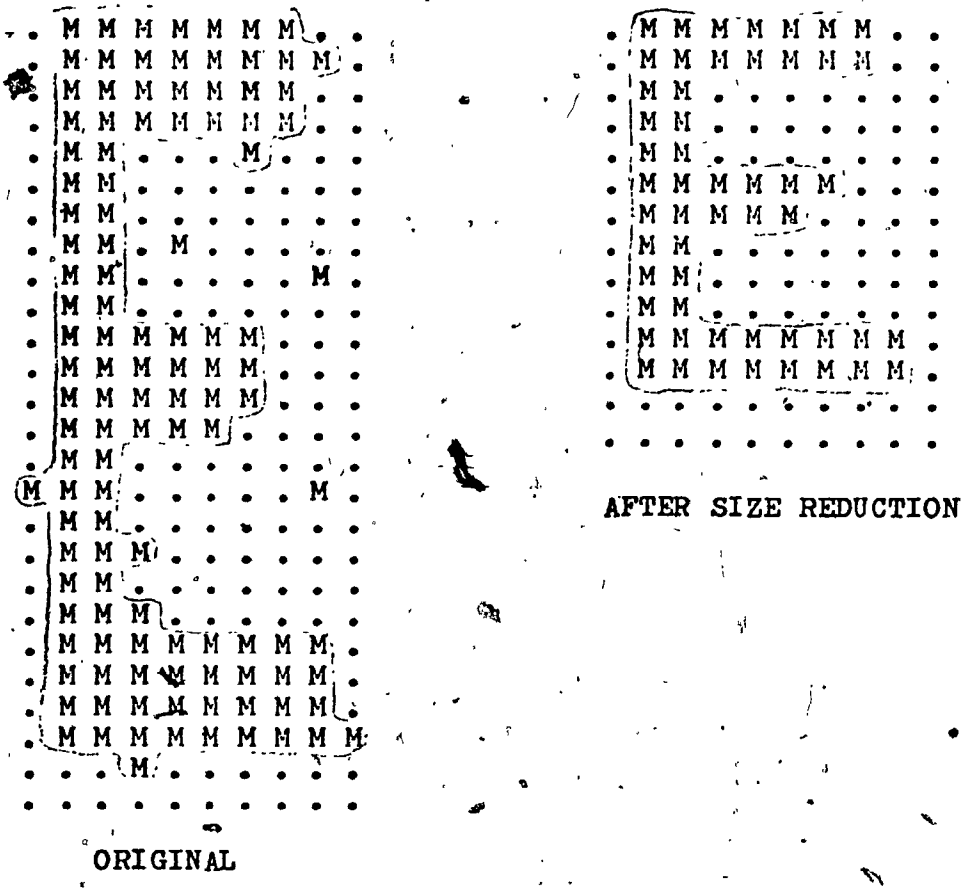


Fig. 2.8 PATTERN SIZE REDUCTION

(The background noises are eliminated as a by-product of the pattern size reduction algorithm.)

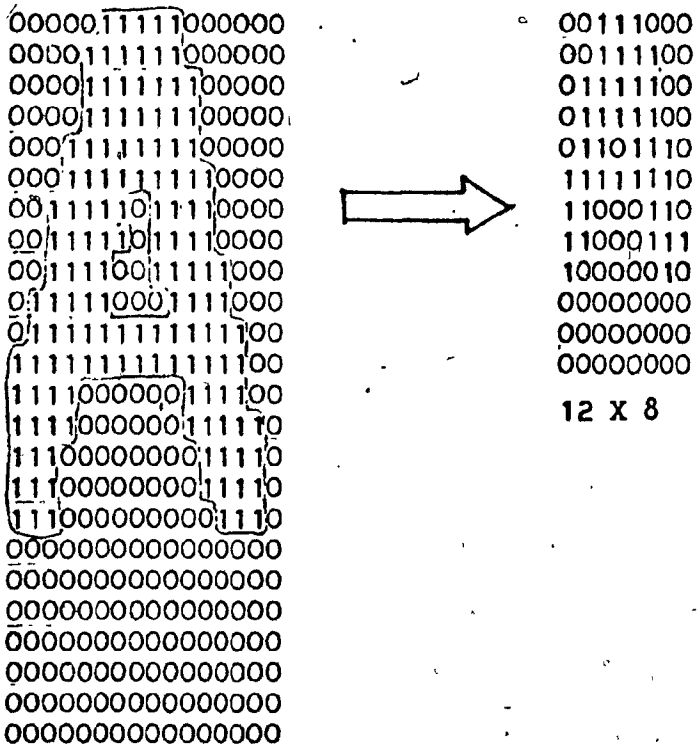
2.2.2.1 GRID POINT REPRESENTATION (GPR)

In this method, the pattern of interest is divided into small grids and each grid point will be assigned a value of "1" or "0" depending on whether the grid point is covered by the character or not. Usually a grid point is said to be covered by the character if more than half of the points within the grid is black (fig. 2.9). However, the size of the grid and the threshold value used to determine whether the grid point is covered by the character are indeterministic parameters. In general, the larger the grid is, the less sensitive it is to noise, but some information about the pattern may be lost. Some analysis of the performance of the GPR against the grid size will be presented in chapter 3.

The merit of this GPR representation is that the feature can be extracted easily. Furthermore, it is insensitive to random noise and is general enough to describe any type font.

2.2.2.2 CROSSING NUMBER REPRESENTATION [28,33,34,35]

A crossing number is defined as the number of groups of black points encountered in the direction of scanning. We can obtain different groups of crossing numbers at different scanning directions. For simplicity and ease of



2 X 2 grid transformation

Fig. 2.9 EXAMPLE OF GRID TRANSFORMATION

plementation on a microcomputer, we chose only the vertical and horizontal ones as shown in fig. 2.10. Unlike the GPR, the crossing number is a more explicit topological descriptor. From the distribution of crossing numbers, it is very easy to group together those patterns which share similar topological structure. For classification of a small number of classes, such as patterns of numerals, it is possible to use the crossing number distribution alone to identify all the classes. Kwon and Lai have developed such an experimental recognition system to identify handprinted numerals [34]. In our case, the number of classes to be classified is dynamic and might be as high as 90 for the full character set, e.g. ANSI OCR character set. We found the crossing number is not adequate to discriminate all the classes in the above case. Some of the confusion pairs are shown in fig.2.11.

Despite the mentioned drawback, crossing number representation can be easily implemented on the microcomputer. An analysis of the performance of this crosssing number is included in chapter 3.

2.2.2.3 QUASI-TOPOLOGICAL CODE (QTC)

This method is originated from Nadler [36]. He defined a set of descriptors indicating potential topological structures such as (,),∩,∪,⊂,⊃,⌐,⌙,|,—,/,\ (fig.2.1). He has also demonstrated how one can obtain these four sets of QTCs simultaneously without retracing the same pattern

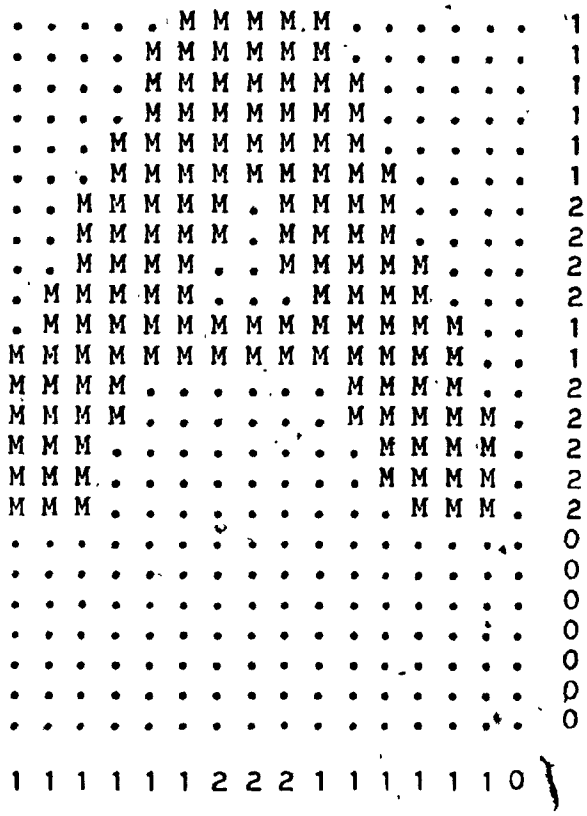


Fig. 2.10 EXAMPLE OF CROSSING NUMBER CODES

```
.....MMM. 1
....MMM.. 1
....MM... 1
...MMM... 1
...MM.... 1
..MMM.... 1
.MMM..... 1
MMM...... 1
MMM...... 1
MM....... 1
......... 0
......... 0
111111110
```

```
...MM.... 1
...MM.... 1
...MM.... 1
...MM.... 1
MMMMMMMM. 1
MMMMMMMM. 1
...MM.... 1
...MM.... 1
...MM.... 1
...MM.... 1
......... 0
......... 0
111111110
```

```
MM...... 1
MM...... 1
MM...... 1
MM...... 1
MM...... 1
MMMMMM... 1
MMMMMMM.. 1
MM...MM.. 2
MM...MM.. 2
MMMMMMM.. 1
MMMMMM... 1
......... 0
112221100
```

```
.....MM.. 1
.....MM.. 1
.....MM.. 1
.....MM.. 1
.....MM.. 1
.MMMMMM.. 1
MMMMMMM.. 1
MM...MM.. 2
MM...MM.. 2
MMMMMMM.. 1
.MMMMMM.. 1
......... 0
112221100
```

Fig. 2.11 SOME CONFUSION PAIRS IN CNC CODING

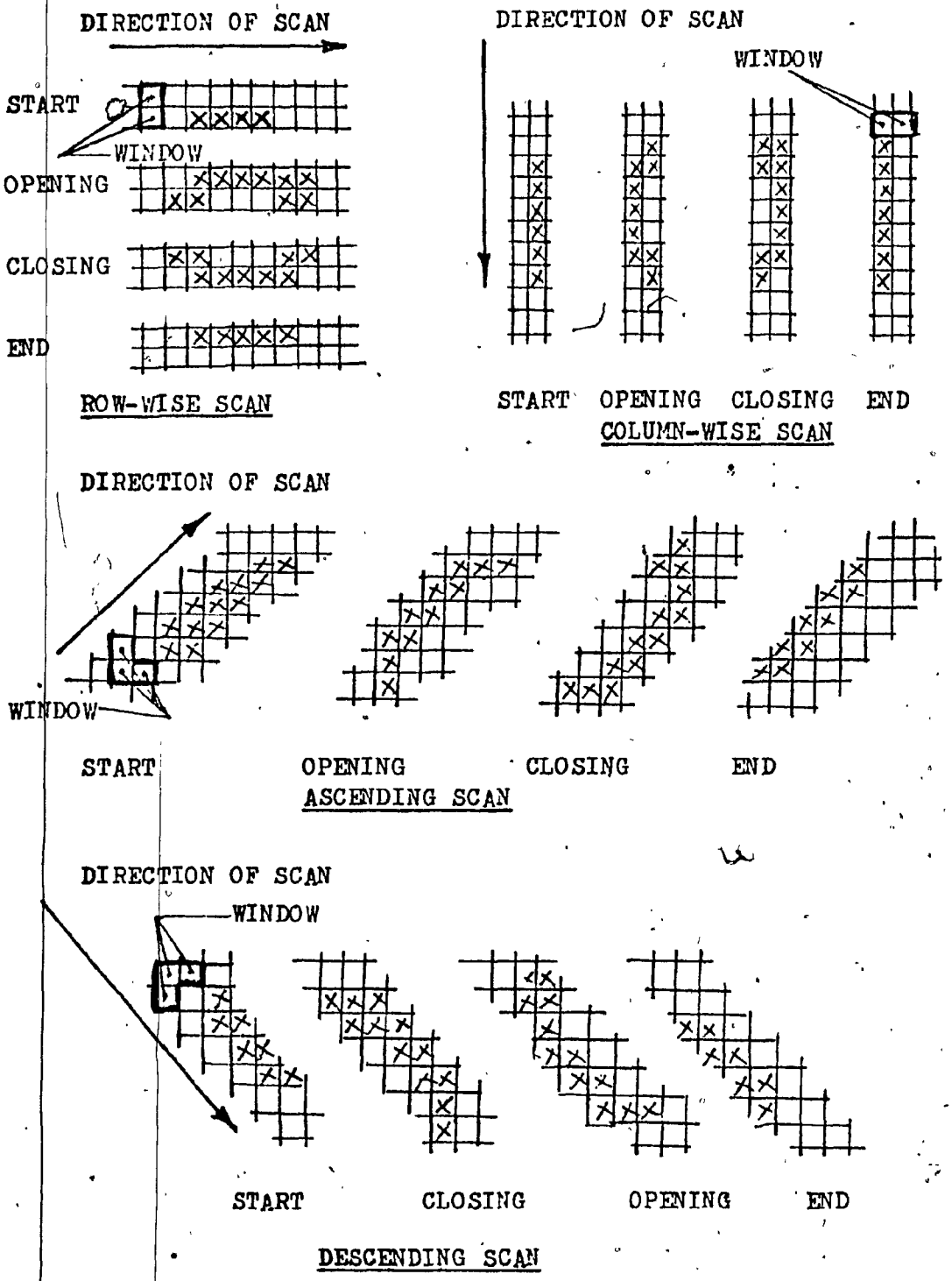


Fig. 2.12 QUASI-TOPOLOGICAL CODES (QTC)

again. As claimed in his later paper [37], the QTC can be used in omnifont or font independent recognitions. This is the main reason that attracted us to take a closer look at his method. However, as our analysis (presented in chapter 3) shows, it has the same drawback as those we found in the crossing number representation. Although there is no need to retrace the same pattern in order to obtain the four sets of QTCs, it does take three to four times longer to compute them than to count the crossings.

2.2.2.4 THE FINAL FEATURE SET

Among these three methods, the GPR seems to be the easiest one and powerful enough to distinquish even the full ANSI OCR character set by choosing a suitable grid size. However, when using the GPR, there are some disadvantages during the classification process. Since the topological information of the GPR is not explicit, it is very hard to construct a hierarchical classifier based on topological information in order to minimize the classification time. We must go through the long process to do an exhaustive search for all the pre-stored templates and the number of comparisons goes up linearly with the size of the character set to be recognized. On the other hand, the second and third methods do have the merit that is missing from the GPR. An intuitive solution to this is to combine these two or even three methods together to take the advantages of

them. As mentioned above, the crossing number representation and the QTC are as good as each other, but the crosssing number takes less time to compute. Hence we chose a mixture of GPR and crossing number representation as our final feature set.

Instead of using the crossing number directly, a modified crossing number code (CNC) is developed to serve as a topological group descriptor in the hierarchical classification scheme. It is noticed that almost all the English alphabets, numerals and special symbols we come across in our daily life have less than or equal to three crossings, so we can concentrate ourselves on the distributions of crossing number less than or equal to three. In order to make the code less sensitive to noise, a discrete distribution representation is used. It is summarized in the following.

BINARY CODE	NO. OF CROSSINGS
00	less than 3 (absent)
01	between 3 and 8 (low)
10	between 9 and 12 (medium)
11	over 12 (high)

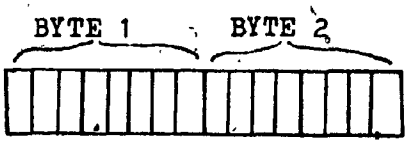
Since the pattern we are working with measures 24 by 16, the above definition can be applied to both the horizontal and vertical CNCs. There are four CNC distributions in the horizontal and vertical direction and each distribution can

be represented by 2 bits according to the above definition. With a fixed entry arrangement we can code the CNC in 16 bits as shown in fig. 2.13.

For the GPR a 2 by 2 grid is chosen as a result of the analysis shown in chapter 3. Thus a 24 by 16 pattern will be transformed into a 12-byte bit pattern. Appropriate microcomputer programs have been developed to extract the CNC and GPR codes. More details about the microcomputer program development are included in chapter 5.

2.2.3 CLASSIFIER AND ERROR DETECTOR

As mentioned in the previous section, a two-stage classifier is proposed to minimize the classification time. The two sets of feature, CNC and GPR, received from the feature extractor, are used at two different stages. The CNC is used as a key to represent a group of GPRs with similar topological structures. Thus when an unlabeled pattern is going to be classified, it first goes through a table lookup to see whether its CNC is matched with any of the pre-stored CNC groups. In other words, this is the first stage classification. If a match is found, its GPR is used in the second stage classification. In the second stage classification, Hamming distance (i.e. number of elements different from the templates) is used as a measure of match. The Hamming distances between the GPR of the unlabeled pattern and all the pre-stored GPRs in that CNC



H_i : distribution of horizontal crossing equals i

V_i : distribution of vertical crossing equals i

Fig. 2.13 FIXED ENTRY CNC (CROSSING NUMBER CODE) FORMAT

group are calculated. The pair with minimum Hamming distance is said to be matched. If this minimum is smaller than some pre-set threshold Tc, the corresponding class identity is output as the classification result; otherwise an exhaustive search for all the pre-stored GPRs will be performed and again the pair with minimum Hamming distance is said to be matched. This minimum is compared to an error detecting threshold Te. If it is greater than Te, the classification modifier will be invoked to modify the existing decision rule; else the corresponding class identity is output as the classification result. In the case the CNC of the unlabeled pattern is not matched with any of the pre-stored groups, it will go through the same procedure as if its minimum exceeds the first threshold Tc. The value of Tc and Te are determined empirically.

2.2.4 CLASSIFICATION MODIFIER

The classification modifier in our proposed system is a table setup and modification program. When this program is called with three parameters, namely, CNC, GPR and the class identifier, the CNC group table and the GPR table will be modified and all the necessary linkages for the classification are updated such that these tables are ready to be used for subsequent classifications.

Because of the dynamic nature of these tables, problems of memory usage and sharing arise. We solved this memory

management problem by a program which allocates a block of free memory during the modification of these tables. Memory overflow will be signaled to the user when the entire free space has been occupied.

2.3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

Our proposed OCR system is designed to run in three different modes, namely: learning, adaptive recognition and fixed recognition mode.

In the learning mode, one can train the system to recognize new type fonts or modify the existing classification logic to accept additional symbols. In this case, a set of training samples is usually required. The GPR template of a given class is generated by overlapping all the samples in the same class. Those points with more than 50% occurrence are used to construct the GPR (fig 2.14). The setup of the CNC group table is based on individual training sample. More information about the data structure of these tables is given in chapter 5. This learning mode is usually selected prior to the reading of a new type font.

In the adaptive recognition mode, if an error is detected by the error detector, the system will ask the operator to make a decision either to drop the unrecognized pattern or to modify the classification logic. Only the CNC

9	91	100	74	9	0	0	0	0	0	0	0	0	0	0	0
70	100	100	100	17	0	0	0	0	0	0	0	0	0	0	0
91	100	100	100	22	0	0	0	0	0	0	0	0	0	0	0
96	100	100	100	30	0	0	0	0	0	0	0	0	0	0	0
96	100	100	100	74	9	4	22	30	22	17	13	9	0	0	0
100	100	100	100	100	96	91	96	91	87	83	87	96	48	0	0
100	100	100	100	100	100	100	100	100	100	100	100	100	100	65	4
100	100	100	100	100	100	100	100	78	78	91	100	100	100	91	9
100	100	100	100	100	91	57	30	17	17	30	87	100	100	100	26
100	100	100	100	87	26	0	0	0	0	0	61	100	100	100	35
100	100	100	100	35	0	0	0	0	0	0	52	100	100	100	39
100	100	100	96	22	0	0	0	0	0	0	52	100	100	100	39
100	100	100	83	17	0	0	0	0	0	0	57	100	100	100	48
100	100	100	78	13	0	0	0	0	0	0	65	100	100	100	57
100	100	100	78	13	0	0	0	0	0	0	61	100	100	100	48
100	100	100	65	4	0	0	0	0	0	0	61	100	100	96	52
100	100	100	52	13	0	0	0	0	0	0	52	100	100	96	30
87	100	87	22	0	0	0	0	0	0	0	17	96	100	57	9
4	22	0	0	0	0	0	0	0	0	0	0	0	13	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A 50% TEMPLATE OF LOWER CASE "h" (OCRA)

Fig. 2.14 GENERATION OF TEMPLATE

group table will be modified in this mode. However, in the case of memory overflow, it will switch to the fixed recognition mode automatically.

In, the fixed recognition mode, error will still be signaled to the operator, but no modification of the system is performed.

CHAPTER III

SOME CHARACTERISTIC STUDIES OF THE PROPOSED OCR SYSTEM

In this chapter, some characteristic studies of the proposed OCR system are given. It starts with data acquisition and simulation of the proposed recognition system on a large scale computer. Then, the performance of several proposed feature sets are evaluated. Finally, some behavioral studies of the recognition system against different sizes of training set and type fonts are analyzed.

3.1 DATA ACQUISITION AND SIMULATION OF THE PROPOSED DESIGN

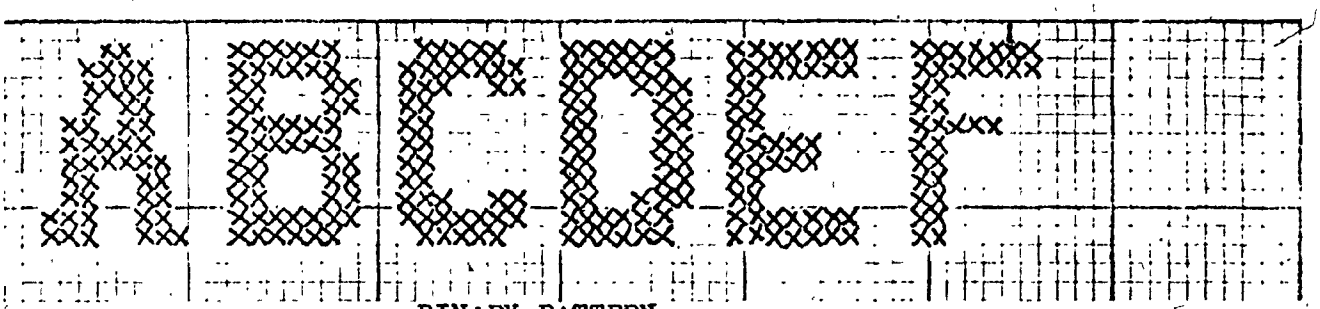
As a standard system development procedure, it is essential to prove the proposed system to be logically correct before the actual system development takes place. The simplest way is to carry out a checking on paper. Normally, paper checking is accomplished at flowchart level to verify the overall correctness of the design. However, it will not permit any evaluation of performance. Moreover, it is long and tedious. In order to evaluate performance, automation must be used.

Functionally simulating the proposed design on a large scale computer provides a solution to the above problem.

Testing and evaluating of such a simulation program requires non-trivial data. There are two possible ways to obtain such data. The first and straight forward way is to create a set of prototype characters, and using a random noise generator to impose random noise on the prototype and thus obtain a set of distorted character images which more or less resemble the data from real life and hopefully can be used in the training and testing of the simulation program. Alternatively, one can collect the data from real life. The former method is easy and efficient; however, when compared to those obtained from real life, one may observe that many real life conditions can never be reproduced by a random noise generator.

In our studies, the CDC (Control Data Corp.) CYBER 172/2 computer system was used to carry out the simulation and evaluation. Real life data were used for the analysis.

For correction of the type written character images, an ECRM 5000 [50] series autoreader was used to digitize different fonts of type written characters. The digitized images were transmitted to the CYBER computer through communication lines. Special purpose programs have been developed to handle the digitization and communication processes. In order to reduce the data storage as well as communication time, a data compression technique was employed. First, the straight binary pattern was represented in terms of run-length codes which give the



BINARY PATTERN

EQUIVALENT RUN-LENGTH CODE REPRESENTATION

70,
5,2,5,6,4,5,3,6,3,7,3,14,
4,4,4,6,3,7,2,6,3,7,3,14,
4,4,4,2,3,2,2,3,2,2,2,2,2,3,2,2,8,2,19,
4,1,1,2,4,2,3,2,2,2,7,2,3,2,2,2,8,2,19,
3,2,1,2,4,6,3,2,7,2,3,2,2,8,16,
3,5,4,6,3,2,7,2,3,2,2,5,5,5,16,
3,6,3,2,3,2,2,2,7,2,3,2,2,5,5,2,19,
3,2,2,2,3,2,3,2,2,7,2,3,2,2,2,8,2,19,
2,3,2,2,3,7,2,7,2,6,3,7,3,2,19,
2,3,2,3,2,6,4,5,3,6,3,7,3,2,19,
70,
70,
70,

Fig. 3.1 RUN-LENGTH CODING OF BINARIZED PATTERNS

number of alternative sequences of contiguous white points orblack points encountered per scan line (fig.3.1). For the full resolution of the ECRM autoreader, there are 2064 bits per scan line. Therefore, it needs 12 bits to represent each run-length code. However, over 90% of the run-length codes have very small values in our particular case. Expanded block coding scheme was used to achieve a higher compression ratio. A 6-bit code is used instead of 12-bit code. Since a run-length code can never be equal to zero, we use the zero as an expanded block indicator. Whenever a zero is encountered, the following 2 "6-bit codes" will form an expanded 12-bit code. The overall compaction of this coding algorithm is approximately 40% of the original and is easy to decode on the receiving end as opposed to most compression coding algorithms.

The simulation of the proposed recognition system is accomplished by writing programs in high level language and running them in a simulated mode on a large scale computer. For instance, the CYBER 172/2 computer system was used throughout the simulation and performance evaluation.

A high level programming language was used instead of lower level languages, (e.g. assembly language) because they are usually machine independent, easy to use and comprehend. These advantages allow the software designer to concentrate on the solution of the problem rather than worrying about the data transfer and control at machine

level.

Several special and general purpose high level programming languages are widely adopted, such as FORTRAN, PL/1, ALGOL, and the most recent one PASCAL [51]. Although PL/1, ALGOL and PASCAL are structured programming languages suitable for modulized software development, FORTRAN was chosen to accomplish our simulation because it is the most common programming language and the machine code produced by a FORTAN compiler is usually more efficient than those from other high level languages. Again, with good programming descipline, one can still write structured program in FORTRAN.

The simulation program of the proposed OCR system was partitioned into modules according to their functions as stated in the previous chapter. Apart from this, several performance evaluation programs have also been developed to carry out the analysis of the system performance. Details of these evaluations can be found in the following sections.

3.2 DISTANCE MEASUREMENT

In order to determine the effectiveness of a given set of features, one can evaluate it either through using it in a working recognition system or defining some measurements to evaluate it. Generally speaking, the first method involves huge amount of computation time so as to obtain reliable

statistical results. The latter approach is more systematic, less time consuming and hopefully more reliable than the former approach. For the above reasons, the second approach was adopted in our studies.

As mentioned before, the goodness of a feature set depends on its ability to minimize variations between different patterns within the same pattern class and to stress the distinctions between different pattern classes. Accordingly, measures of these two properties are essential for the evaluation of any feature set. Several distance measurements exist for measuring similarity and dissimilarity between binarized patterns [24,28,29,31]. In our analysis, the normalized dissimilarity coefficient is chosen as a measure of distance between patterns. The value of this coefficient ranges from 0.0 to 1.0 indicating the degree of dissimilarity of two patterns, from totally identical to completely different.

3.2.1 INTRA-CLASS DISTANCE

In our proposed OCR system, there is always a perfect prototype (template) to identify each class. An average distance from all the members in the class to this prototype will contribute a measure of the ability of the given feature set to remove intra-class variations.

Suppose there is a pattern class $\{A_{i,j} \mid j=1,2,\ldots,k\}$ with k members and a perfect prototype Ci, the intra-class distance can be calculated as

$$d_i = \frac{1}{k} \sum_{i=1}^{k} \frac{|A_{i,j} \oplus C_i|}{|A_{i,j} \cup C_i|}$$

Where $\oplus$ is the logical "exclusive or" operator,

$\cup$ is the logical "or" operator, and

$|x|$ denotes the number of non zero entries in x.

While the average intra-class distance of a m-class problem is given by

$$\bar{d} = \frac{1}{m} \sum_{i=1}^{m} d_i$$

$$= \frac{1}{mk} \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{|A_{i,j} \oplus C_i|}{|A_{i,j} \cup C_i|}$$

It is obvious that the smaller the average intra-class distance, the better the ability of the feature to minimize intra-class variations.

3.2.2 INTER - CLASS DISTANCE

On the other hand, the distances among all the prototypes will constitute another means of measure of the ability of the given feature set to detect inter-class distinctions.

Let Ci be the prototype of class i, for a m-class problem. The inter-class distance among class i and the remaining m-1 classes is given by

$$D_i = \frac{1}{(m-1)} \sum_{j=1}^{m} \frac{|C_i \oplus C_j|}{|C_i \cup C_j|}$$

And the average inter-class distance is given by

$$\bar{D} = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{(m-1)} \sum_{j=1}^{m} \frac{|C_i \oplus C_j|}{|C_i \cup C_j|}$$

$$= \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{|C_i \oplus C_j|}{|C_i \cup C_j|}$$

Since Ci and Cj is commutative, the above expression is in fact counting the distance between the same pair twice. It can be further reduced to

$$\bar{D} = \frac{2}{m(m-1)} \sum_{i=1}^{m} \sum_{j=1}^{i-1} \frac{|C_i \oplus C_j|}{|C_i \cup C_j|}$$

Apparently, the higher the average inter-class distance is, the better the ability of the given feature set is to detect inter-class distinctions.

3.3 FEATURE EVALUATION

The proposed features were analyzed by the mentioned

distance measuremet over a set of OCRA character patterns which is collected from real samples. There are 77 characters in the OCRA character set and 50 samples of each character were collected and used in this analysis. The result of the analyses of each feature set is reported in the following paragraphs.

3.3.1 DISTANCE ANALYSIS OF QTC FEATURE SET

Two graphs (fig. 3.2 and fig. 3.3) were obtained for the evaluation of the QTC feature set. Fig 3.2 shows the distribution of the inter-class distances. The maximum was found to be 29.80% at a distance equal to 1.0. In other words, 29.80% of the classes are completely different from one another. However, 0.44% of the classes have inter-class distance equal to zero and it meant that the QTC feature is not powerful enough to distinguish all the pattern classes. While the average inter-class distance is 0.87 which is higher than those of CNC'S and GPR'S (refer to the next two sections). The intra-class distance distribution can be found in fig. 3.3. Although the maximum of the distribution is located at a distance equal to zero, it does fluctuate when the distance goes up and the average intra-class distance is equal to 0.43 which is relatively high compared with those of CNC's and GPR's. The fluctuation as well as the large average intra-class distance indicate the fact that QTC feature is very sensitive to small changes of

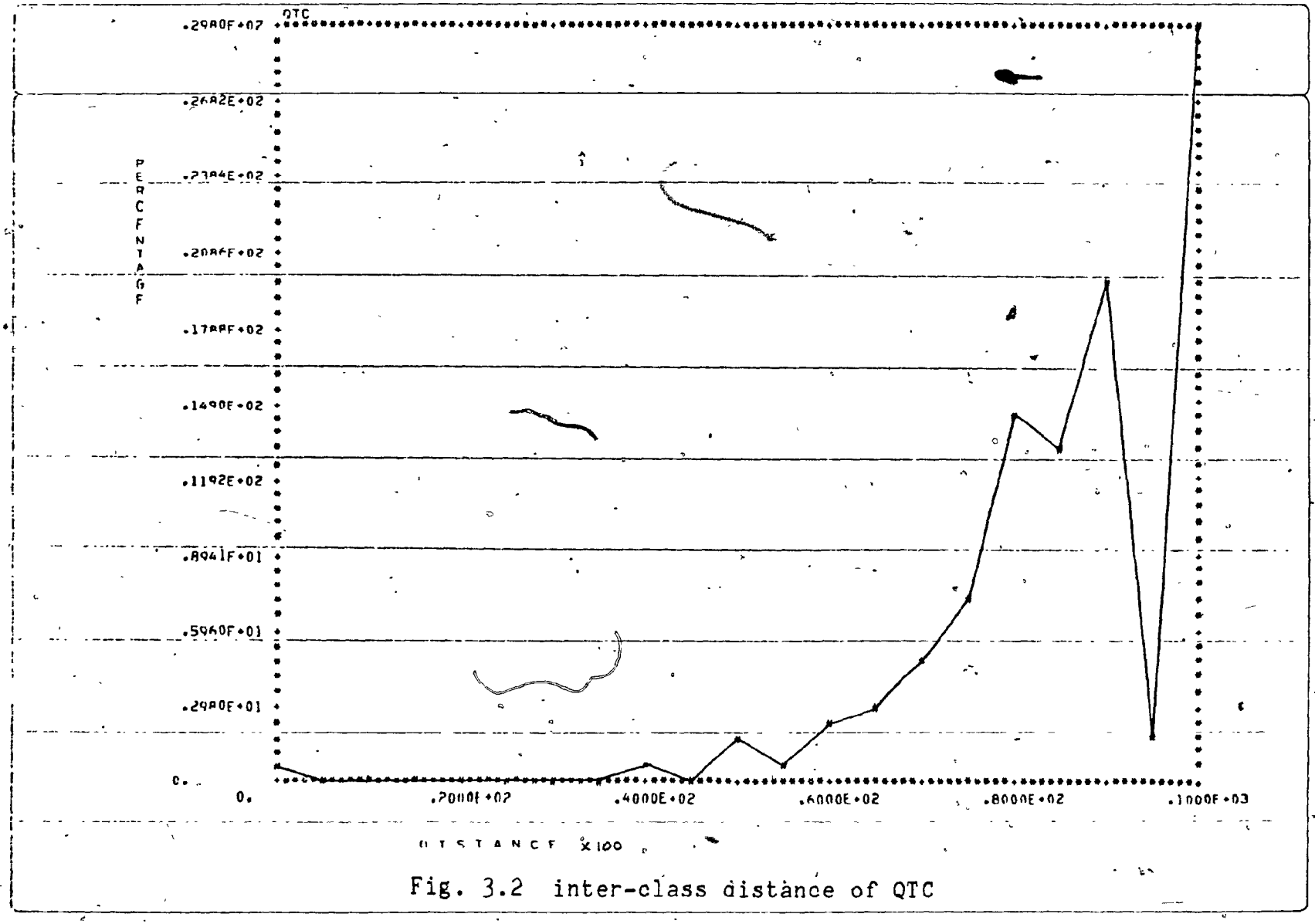


Fig. 3.2 inter-class distance of QTC

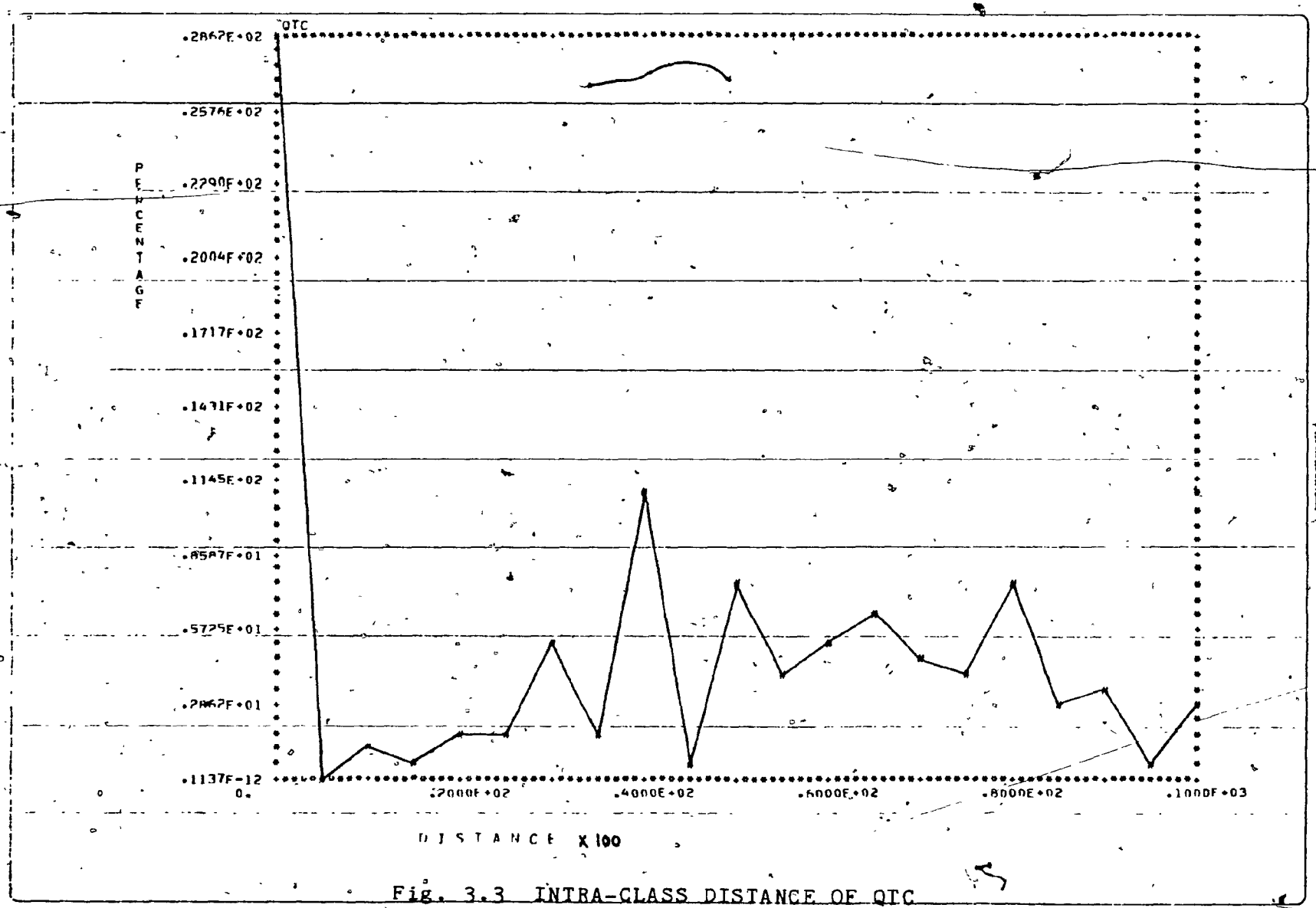


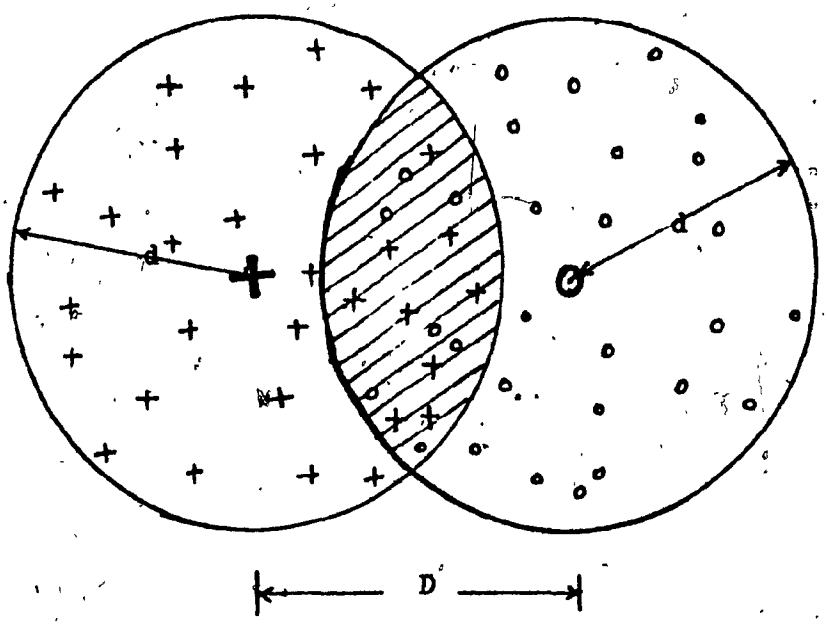
Fig. 3.3 INTRA-CLASS DISTANCE OF QTC

a pattern.

If two classes have inter-class distance less than twice their own average intra-class distance, ambiguity may occur. This situation is illustrated in fig. 3.4. A measure of "discrepancy" of a given feature of a m-class classification problem is defined as the percentage of the classes that has inter-class distance less than twice the average intra-class distance. The "discrepancy" of the QTC is 38.38%.

3.3.2 DISTANCE ANALYSIS OF CNC FEATURE SET

Similar distance analyses have been applied to the CNC feature. Fig. 3.5 and fig. 3.6 provide the inter-class distance distribution and intra-class distance distribution correspondingly. The maximum inter-class distance distribution is 34.21% located at a distance equal to 1.0 and 0.07% of classes have zero inter-class distance. The average inter-class distance is equal to 0.83. The maximum of the intra-class distance distribution is at zero distance and a 0.22 average was found. The shape of the distribution is more steady. It decreases as the distance goes up except at a distance of 1.0. This means the CNC'S is a more reliable intra-class discriptor. The discrepancy of CNC feature is 3.38% which is about ten times lower than that of QTC's.



D : inter-class distance between class + and class •.
d : average intra-class distance .
+: perfect prototype of class +.
O: perfect prototype of class •.
+: samples in class +.
•: samples in class •.

Those samples inside the shaded area will create conflict during the classification process.

Fig. 3.4 EXAMPLES OF "DISCREPANCY"

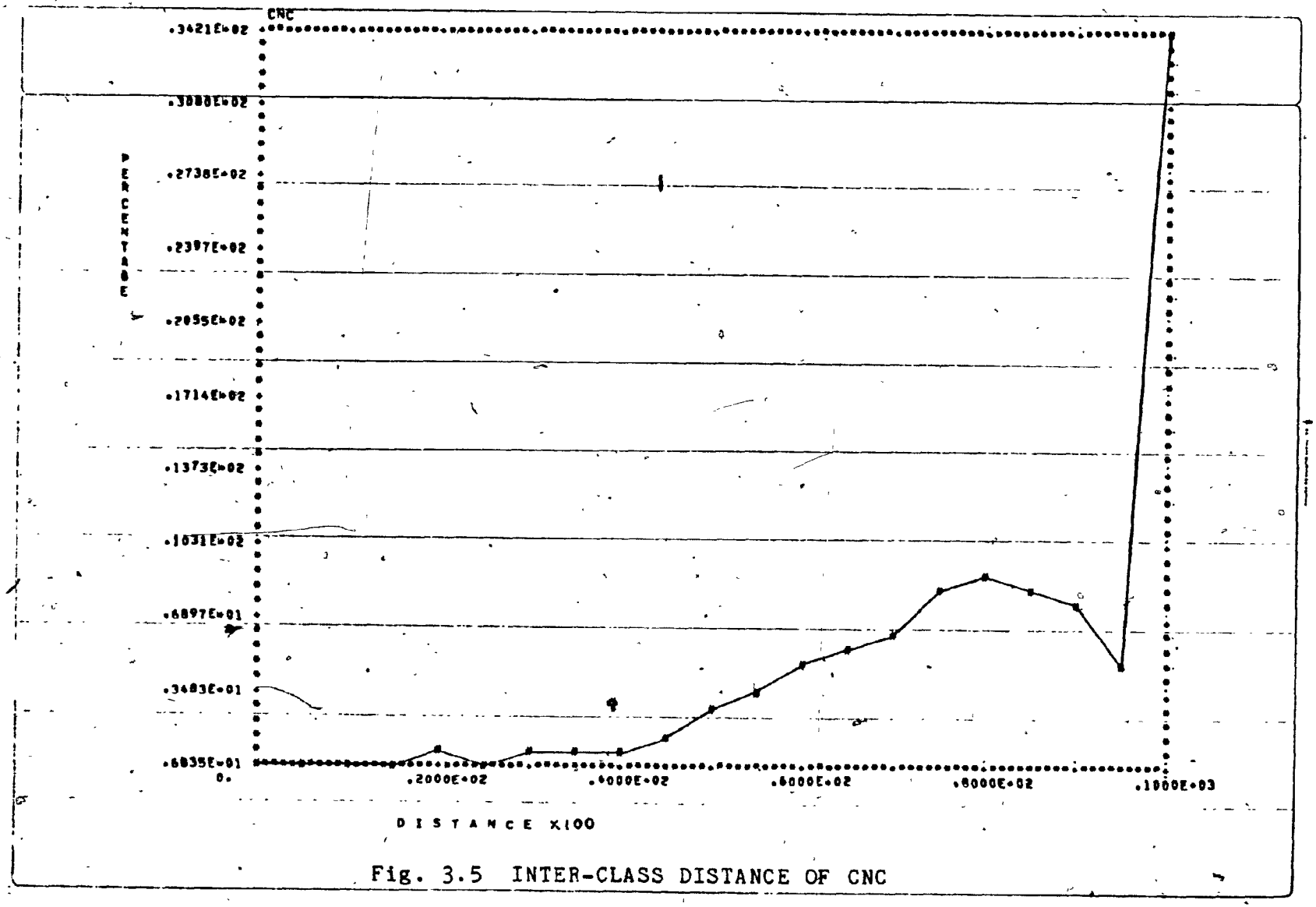


Fig. 3.5 INTER-CLASS DISTANCE OF CNC

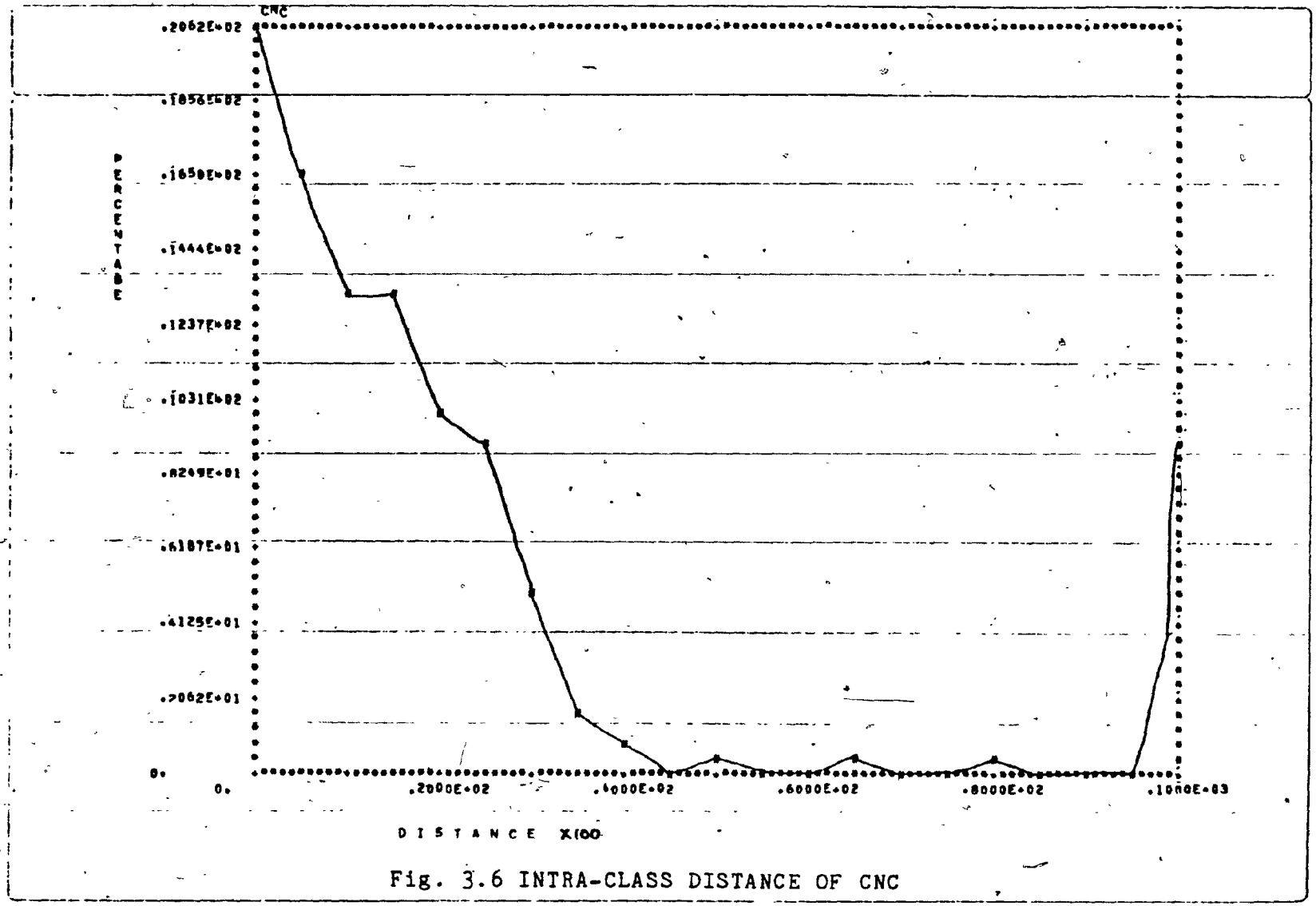


Fig. 3.6 INTRA-CLASS DISTANCE OF CNC

3.3.3 DISTANCE ANALYSIS OF GPR FEATURE SET

As opposed to QTC and CNC, GPR can be extracted using different grid sizes. For the present analysis, 3 grid sizes were studied, namely, 2 by 2, 3 by 3 and 4 by 4. The graphs showing the inter-class and intra-class distance distribution of each GPR are given in fig. 3.7-12. Table 3.1 compares the average inter-class and intra-class distances and the discrepancies among GPRs', QTC's and CNC's. It is noticed that the smaller the grid size is, the higher the inter- and intra- class distances are. As mentioned before, it would be ideal to have a high inter-class distance and a low intra-class distance --- corresponding to the ability of stressing distinctions among different pattern classes and reducing variations of patterns in the same pattern class. All average inter- and intra- class distances of these GPRs are less than those of CNC's and QTC's. However, the drop in the average intra-class distance is much faster than the inter-class distances. At the same time, the "discrepancies" of these GPRs are much smaller than those of CNC's and QTC's and are very close to zero. The distribution of the intra-class distances more or less decreases monotonically with distance, indicating that GPR is less sensitive to small variation of the pattern in the same pattern class. In short, the distance analysis shows GPR is superior to the

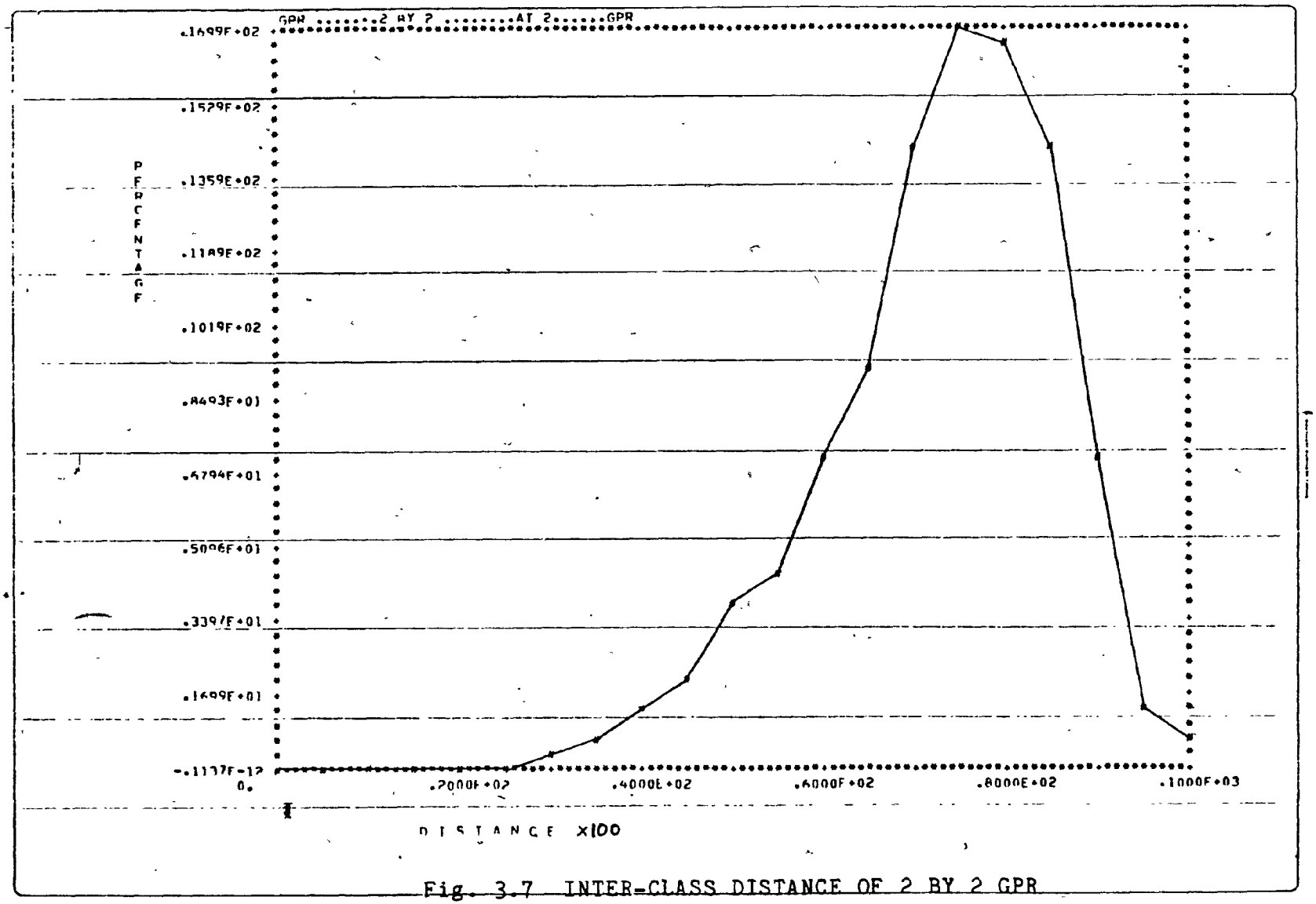


Fig. 3.7 INTER-CLASS DISTANCE OF 2 BY 2 GPR

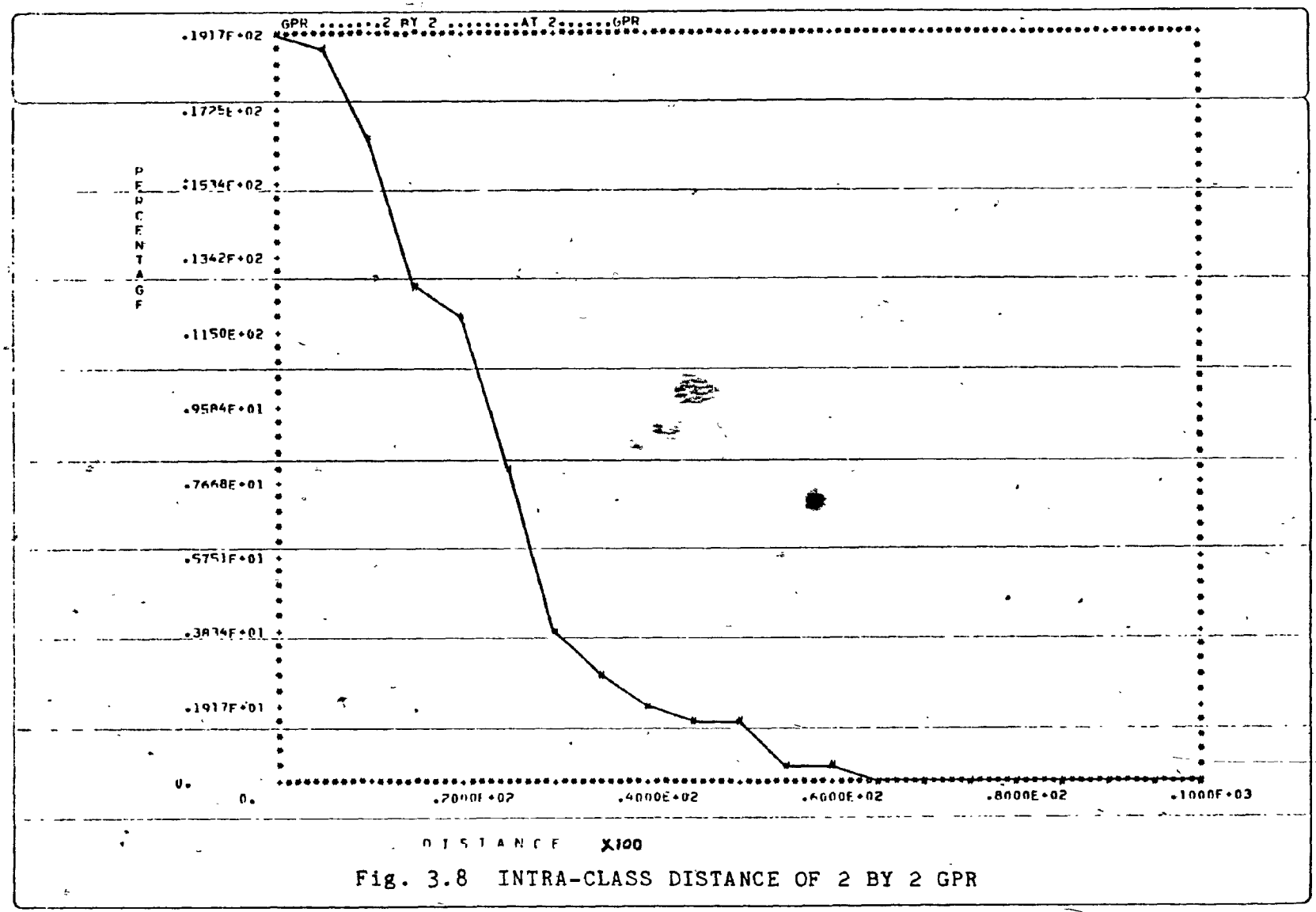


Fig. 3.8 INTRA-CLASS DISTANCE OF 2 BY 2 GPR

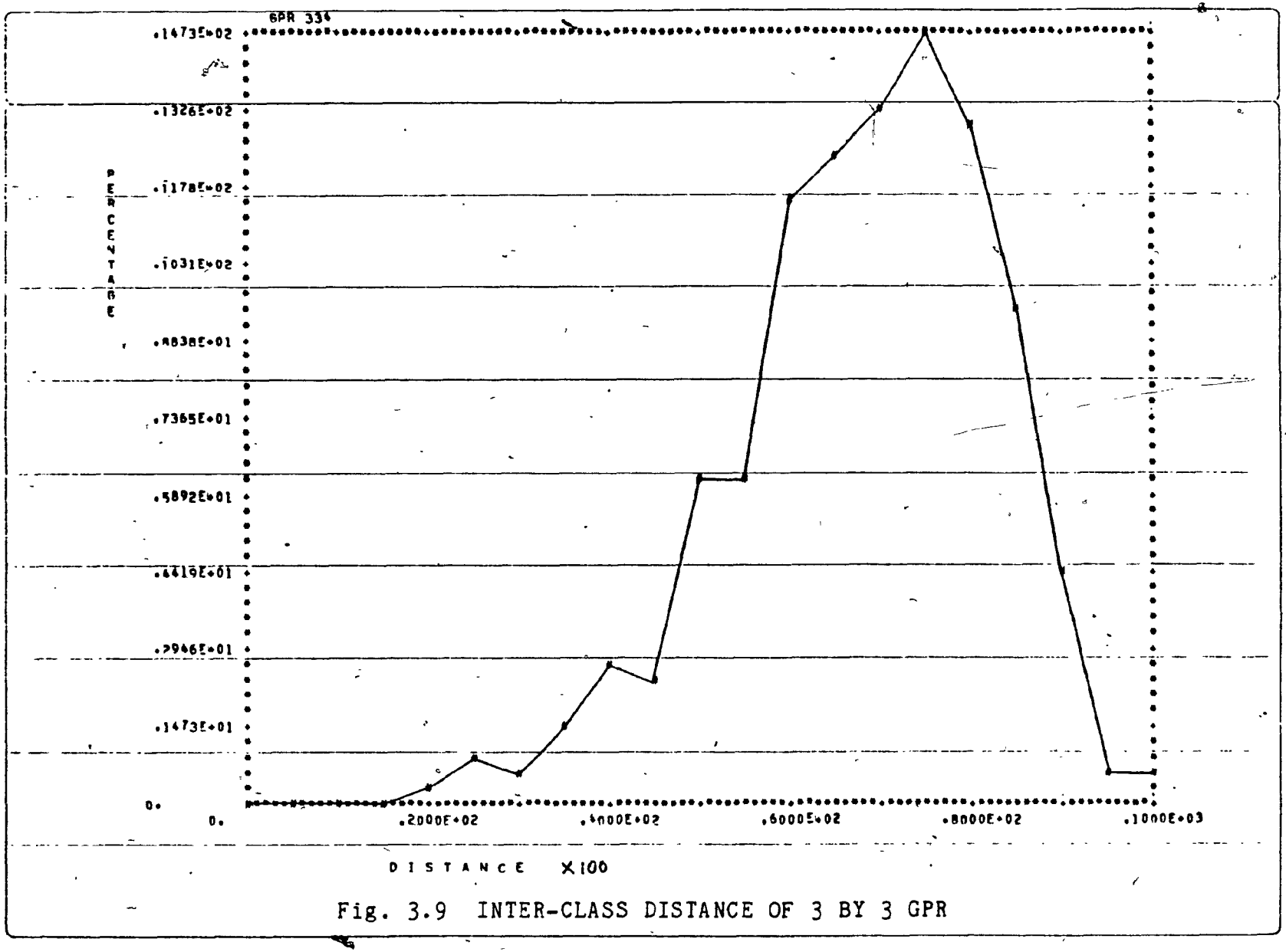


Fig. 3.9 INTER-CLASS DISTANCE OF 3 BY 3 GPR

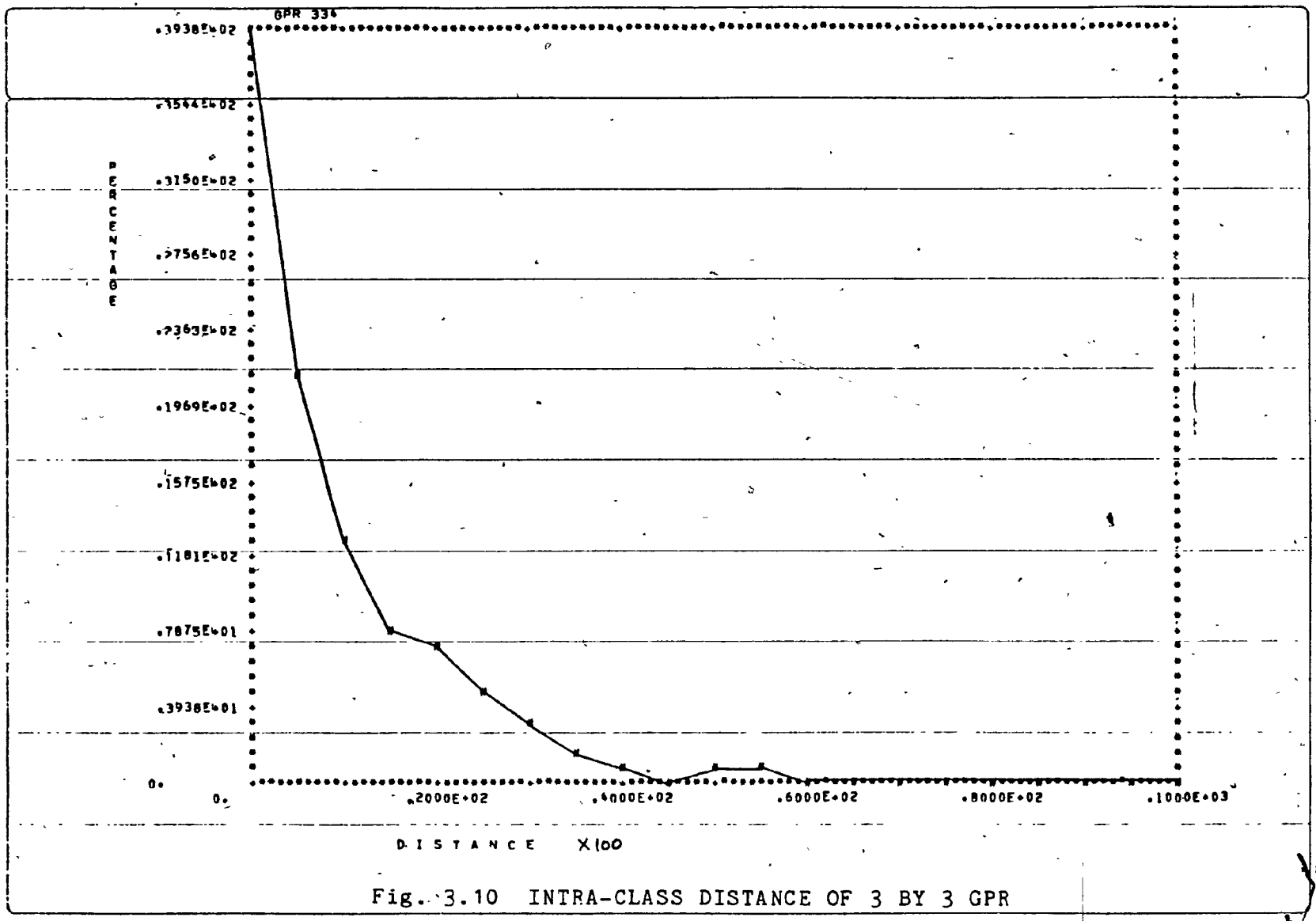


Fig. 3.10 INTRA-CLASS DISTANCE OF 3 BY 3 GPR

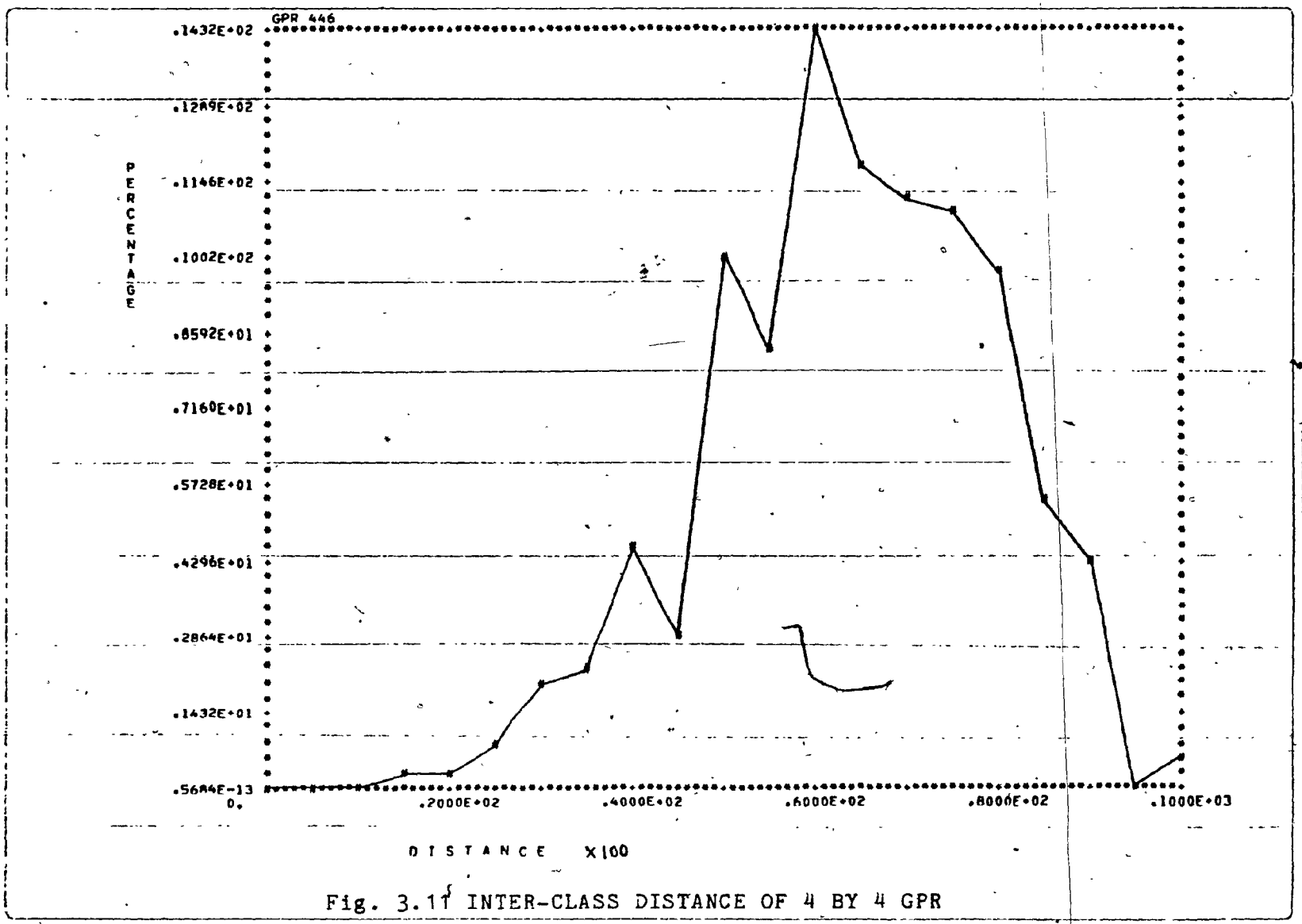


Fig. 3.11 INTER-CLASS DISTANCE OF 4 BY 4 GPR

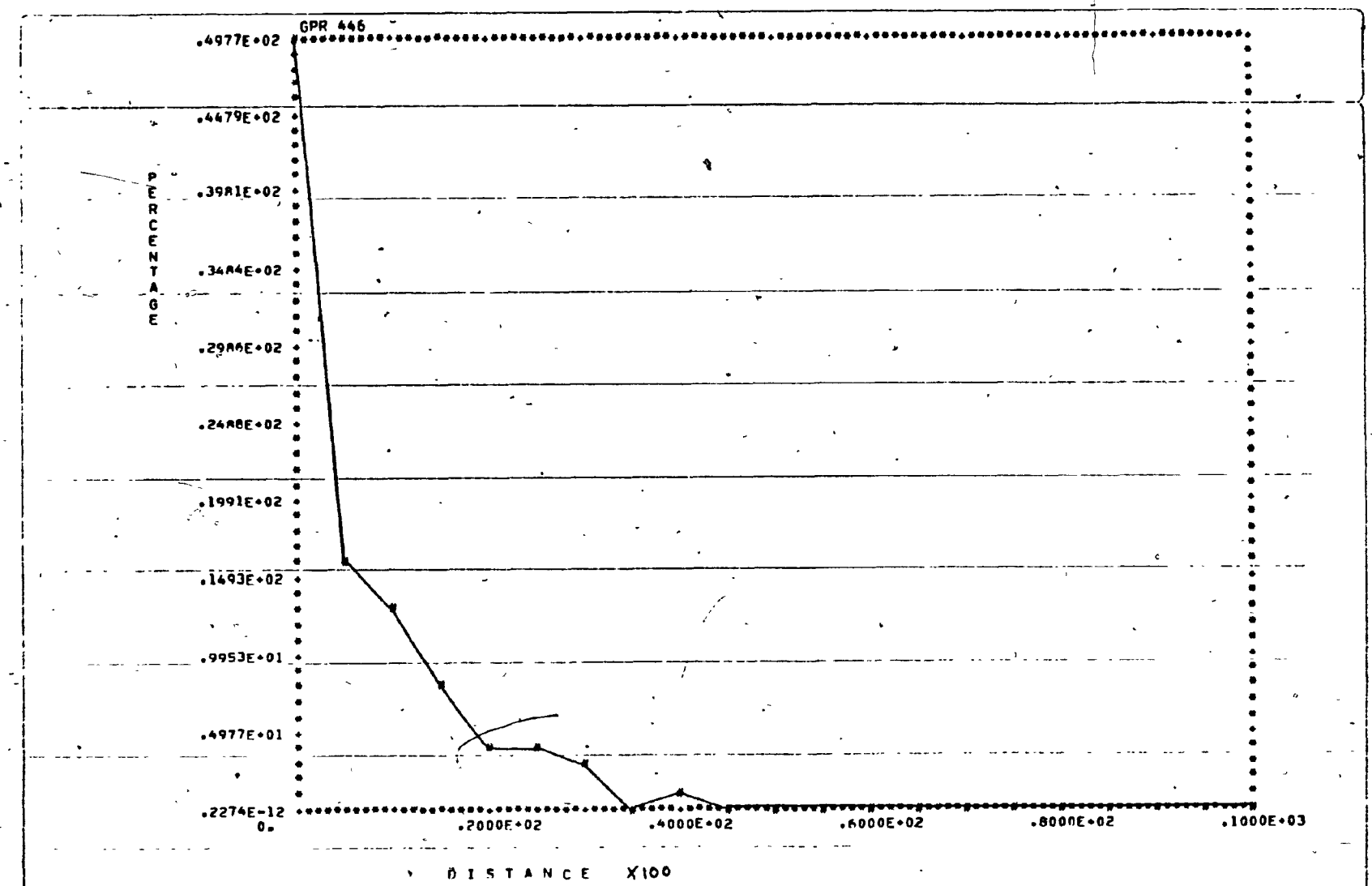


Fig. 3.12 INTRA-CLASS DISTANCE OF 4 BY 4 GPR

TYPE OF FEATURE	D	d	DISCREPANCY
QTC	0.87	0.43	38.38%
CNC	0.83	0.22	3.38%
2X2 GPR	0.75	0.16	0.72%
3X3 GPR	0.70	0.10	0.37%
4X4 GPR	0.65	0.08	0.41%

D : average inter-class distance

d : average intra-class distance

TABLE 3.1 A COMPARISON OF INTER- AND INTRA- CLASS DISTANCES AND DISCREPANCY

other two types of features.

3.4 BEHAVIORAL STUDY OF THE TRAINING SET SIZE

In our proposed recognition system, the CNC's are used as a key to represent a group of GPRs with similar topological features. It is obvious that the larger the training set is, the more reliable CNC group we can get. In order to study the effect of the size of the training set against the recognition result, several testing runs of the simulation program were carried out using training sets of different sizes. In this study, two character sets were used, namely, OCRA and OCRB. 50 and 44 samples of each character were collected from OCRA and OCRB correspondingly. The OCRA data was split into 2 equal parts --- one for training and one for testing. The OCRB data was also divided into two portions --- one contains 25 patterns of each character and the other contains 19.

In this behavioral study, the size of the training set was first set to 5 patterns per character and was gradually increased to 25 in steps of 5. The changes in the recognition rates for the OCRA and OCRB character set are summarized in fig. 3.13 and table 3.2. The result confirms our original prediction. In the testing of the OCRB character set, the recognition rate becomes stationary when the training set size exceeds 15 patterns per character. While for the OCRA, it becomes stationary when it reaches 21

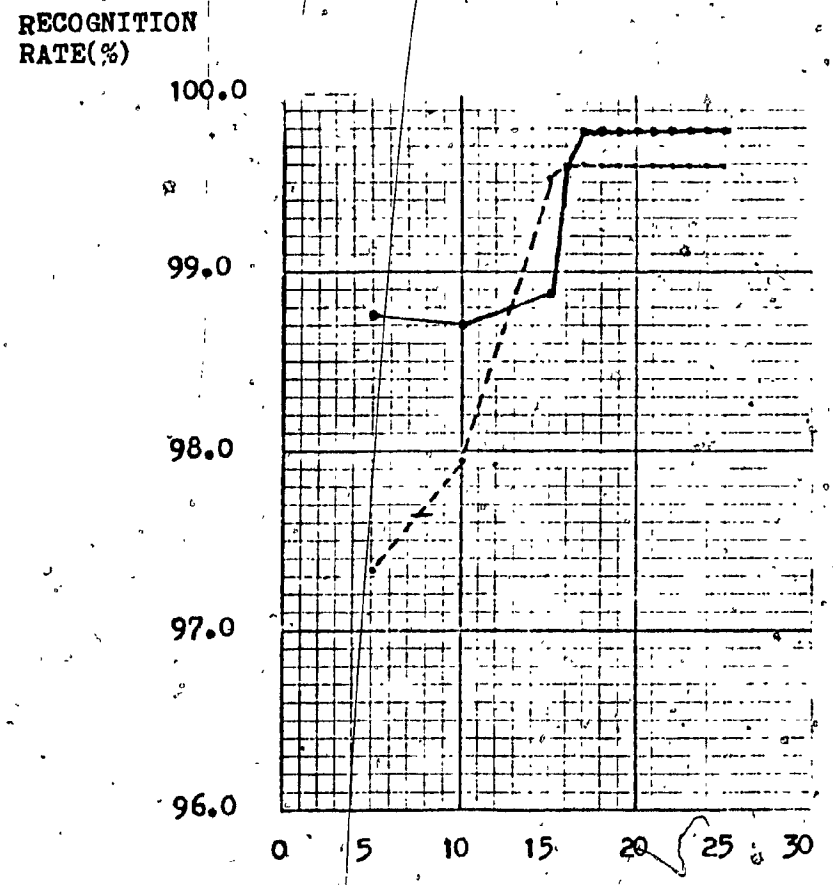


Fig. 3.13 BEHAVIORAL CURVE OF THE SIZE OF THE TRAINING SET

OCRA:-

TRAINING SET SIZE	RECOGNITION RATE(%)
5	98.75%
10	98.70%
15	98.86%
20	99.58%
21	99.79%
22	99.79%
23	99.79%
24	99.79%
25	99.79%

OCRB:-

TRAINING SET SIZE	RECOGNITION RATE(%)
5	97.33%
10	97.95%
15	99.52%
16	99.59%
17	99.59%
18	99.59%
19	99.59%
20	99.59%
21	99.59%
22	99.59%
23	99.59%
24	99.59%
25	99.59%

TABLE 3.2 RECOGNITION RATE AGAINST TRAINING SET SIZE

patterns per character. It can be concluded that although the size of the training set affects the recognition result, yet this effect will saturate when the size of training set reaches a certain threshold.

It is also noticed that when the system is used to recognize the same training set, much higher recognition rate can be obtained. After the system has been trained on 25 samples of each character of the OCRA font, the same samples could be recognized at a recognition rate as high as 99.95%. Similarily, a recognition rate of 99.94% was observed for the OCRB font.

CHAPTER IV

MICROPROCESSOR APPLICATIONS AND SYSTEM DEVELOPMENT

The intent of this chapter is to present an overview of microprocessor based system design. It begins with a short review of the microprocessor and its applications. Then the procedure for developing a microprocessor based system and the problems encountered are discussed. The concept of using a general purpose development system to help the design of such a microprocessor based system is then presented. Finally, a development system specially designed for developing microprocessor based OCR system is introduced at the end of this chapter.

4.1 MICROPROCESSORS, MICROCOMPUTERS AND THEIR APPLICATIONS

A microprocessor is a large scale integration (LSI) electronic component which implements most of the functions of a traditional central processing unit (CPU) of a computer on a single chip [40,41,42,43] and was first commercially introduced to the public in 1971 by INTEL Corp.

A microcomputer is a computer which uses an LSI microprocessor as its CPU. Usually, a minimal microcomputer system can be realized by adding memory, I/O and control circuitry to the microprocessors mentioned above (Fig.4.1).

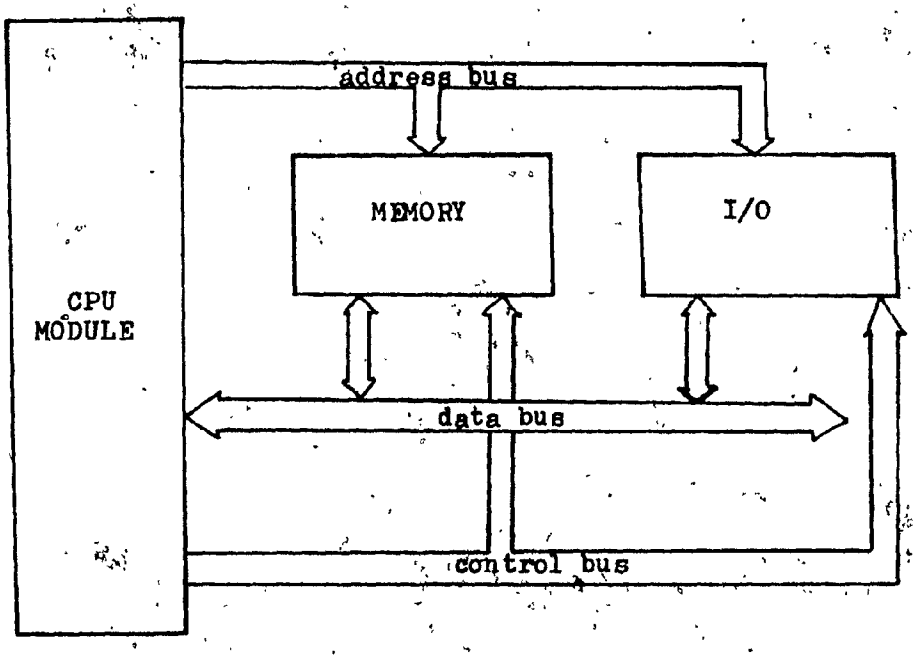


Fig. 4.1 BLOCK DIAGRAM OF A TYPICAL MICROCOMPUTER

The rapid progress of LSI technology now even allows a complete computer to be fabricated on a single LSI chip, such as the INTEL 8048 [39] microcomputer.

Microprocessor does not only replace a substantial number of traditional electronic circuits, but also be used in nearly every area involving programs and automatic control. The growth of microprocessor applications can be easily understood by considering the advantages of the microprocessor over traditional hard wired logic.

The very small number of components required to build a microprocessor system results in miniaturization, reliability, inexpensive and low power dissipation and consumption. This very special feature of microprocessor, allows some conventional huge system to appear in a portable form, like the portable OCR machine for the blind. Again, the flexibility of a microprocessor simplifies the design and reduces system development time, yet allows later modification and expansion of the system to be done easily. Since the main logic control of a microprocessor based system comes from the software program rather than the hard wired logic, the hardware required to build such a system can be standardized and more reliable and cheaper hardware modules will be expected.

A standard microprocessor system can be programmed for a variety of tasks without any hardware change, or may simply

require the substitution of memory chips containing the new program. In addition, new functions can be invented and implemented at a later date without substantial hardware modification.

Because of the above advantages of a microprocessor based system, the list of its possible application is almost unlimited. T.V. games, CRT controller, floppy disk controller, radar control, urban traffic control, televsion fine tuning, small business and personal computer system, microprocessor controlled spark ignition system and portable OCR machine are just a few of its very many known applications. And no doubt, our proposed "adaptive OCR system by microcomputers" will contribute an extra item in the list of microprocessor applications.

4.2 MICROPROCESSOR SYSTEM DEVELOPMENT

Development of a microprocessor based system is similar in most respects to the design of a traditional hard wired logic system. Both approaches include the fundamental steps as shown in Fig. 4.2. It consists of system specification, system flowcharts, hardware design and system test and debugging. However, the microprocessor based design adds another dimension to it. As indicated in Fig. 4.3, the designer now has the option of using software instead of hard wires; therefore one should decide whether each task is best done using the conventional approach or the software

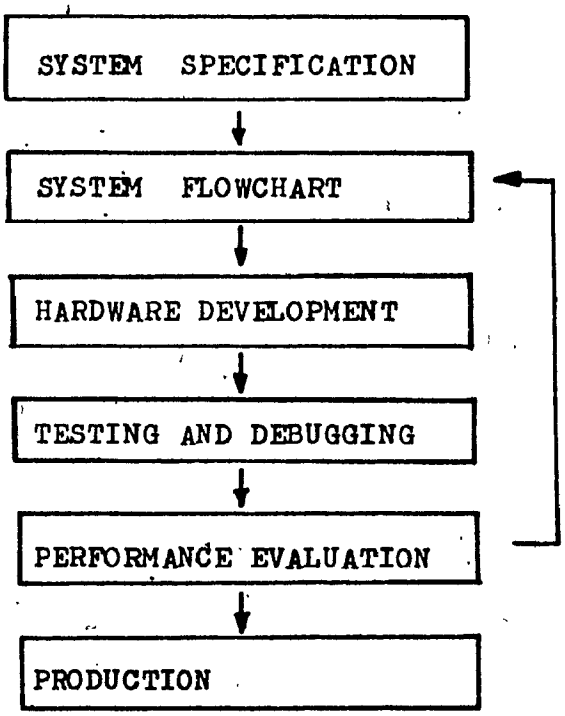


Fig. 4.2 CONVENTIONAL SYSTEM DESIGN CYCLE

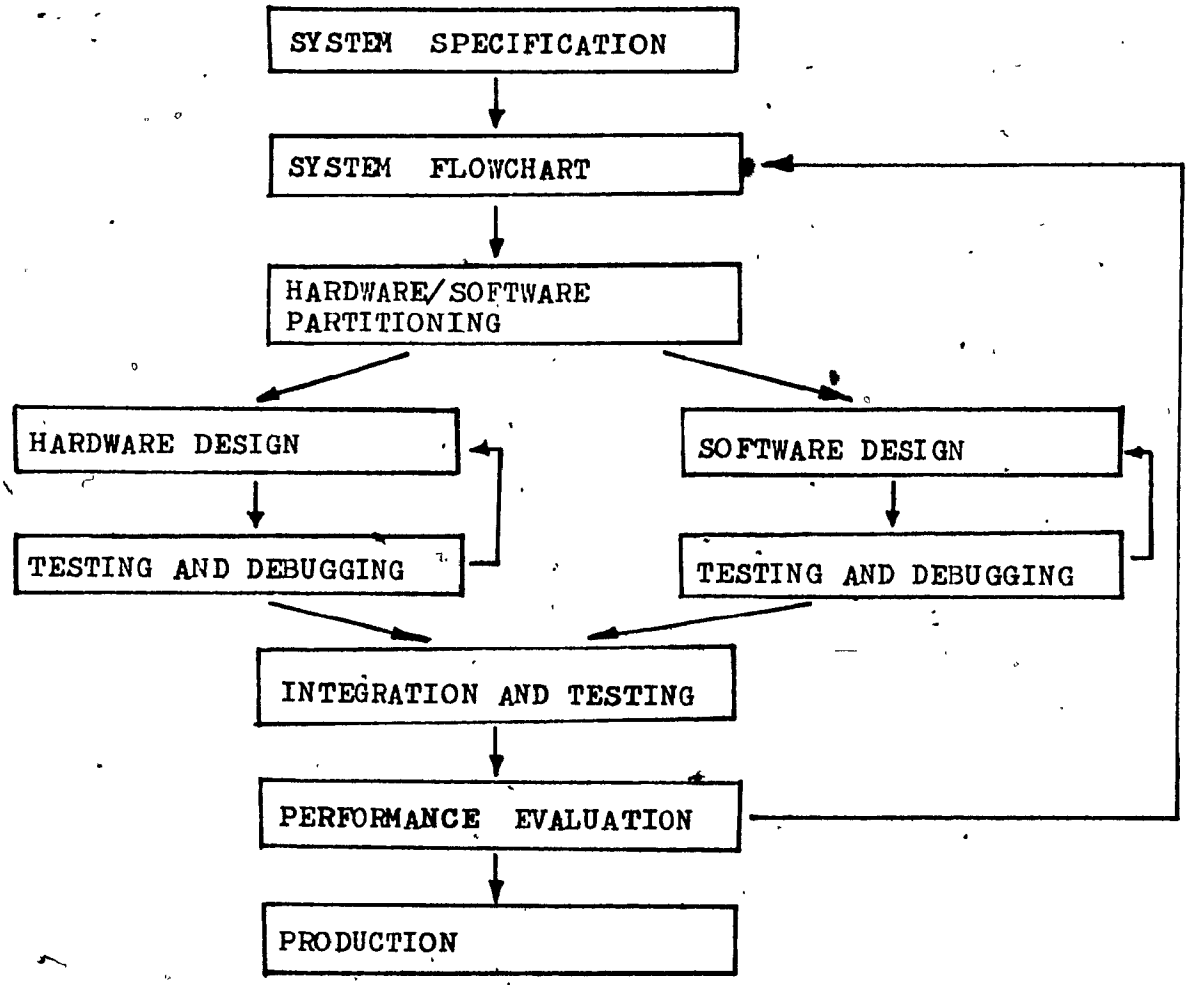


Fig. 4.3 MICROPROCESSOR BASED SYSTEM DESIGN CYCLE

approach.

Hardware and software tradeoffs provide a key to develop the most cost effective system. As we can see from Fig. 4.3, the design and testing of both hardware and software can be carried out simultaneously. This is a major difference, when comparing a microprocessor based system development to a conventional system development. As mentioned above, the hardware modules can be standardized in a microprocessor based system. Thus the design and development of the hardware is usually simple when it involves a "standard microcomputer system" . It may be more complex when it involves unusual interfaces.

Typically, the most significant task of developing such a microprocessor based system is the software design (i.e. program development). Usually, programming requirement of a microprocessor based application can be classified according to their degree of complexity.

(a) RELATIVE SMALL PROGRAM

Programs with just a few hundred or less instructions belong to this class and can be done in machine language by hand coding the instruction into machine codes. This method is usually very slow, inefficient and not reliable. It is only good for short programs that do not require frequent changes.

(b) LARGE PROGRAM

Programs which have a few hundred to a thousand instructions belong to this class and are usually programmed in symbolic assembly codes. These codes are then translated into machine codes by a special program called an assembler. This method is much faster than the hand coding procedure. However, the programmer must be familiar with the data transfer at the register and "internal data bus" levels. Programs written in assembly level are generally very efficient in terms of program execution time, because it allows direct manipulation of the register and data in machine level. It is the most frequently used language for any application that requires efficiency.

(c) COMPLEX PROGRAM

Programs that exceed a thousand instructions and require special interface and data structures are usually classified under this category. Because of the complexity of the program, high level languages, like PL/1, FORTRAN and BASIC, are generally used to overcome difficulties in programming. When using a high level programming language, the designer can concentrate on the design of the complex logic flow of the target task rather than the machine level data transfer. A program written in a high level language will finally be translated into machine codes automatically by special programs called a compiler. Since the translation is

done automatically, the code produced by the compiler is usually two to five times more bulky than that from a skillful assembly language programmer. PL/M and BASIC are the most popular high level languages for microcomputers. The former is a PL/1 like programming language which was originally introduced by INTEL Corp. [44]. The latter is usually a small subset of the BASIC programming language which interpretes the high level programming statements through some functionally equivalent machine code subroutines. Because of the interpreting process, the result of each high level program instruction can be displayed in an interactive fashion. It is good for debugging, but is relatively slow in nature for its interpreting process.

Except for a very small program that can be easily hand coded into machine codes, usually some program development aids are required for developing software for the microprocessor based system. In general, assembler, high level language compiler, interpreter, text editor, file management system, loader, simulator and debugger are required for developing complex softwares. According to where these software development tools reside, they are usually classified into

(1) resident software -- that the mentioned development aids reside in the same machine as the target system.

(2) cross software -- that the mentioned development aids

reside in a machine other than the target machine.

Usually, a more powerful machine is used in the cross software group, like PDP 11, IBM/370 and etc. For using cross software, one can have the benefits such as, speed in processing, powerful and sophisticated software facilities and peripherals. The only drawback is that we cannot run our program in the real environment. However, a simulator is usually available on the cross software to allow the user to simulate the target machine. The simulation is usually slow and some real time conditions cannot be reproduced on the simulator and thus may result in error. On the other hand, the resident software allows designer to develop his or her software on the the target machine and test it in the real environment such that some design errors may be detected at an earlier stage and thus save time in system development. However, because of the limited ability of the microcomputer system, the software development aids themselves are slow in nature and not very powerful.

4.3 DEVELOPMENT SYSTEM

A development system is a microcomputer system equipped with all the facilities required for convenient system development. Physically, it looks like a traditional minicomputer. From the software stand point, it should be equipped with all the required software development aids, such as text editor, assembler, debugger and other

supporting programs. In addition, it is highly desirable to have high level language compiler or interpreter like PL/M [44], FORT80[45] and BASIC. Unfortunately, these compilers require large amount of memory (24K for PL/M, 12K for FORT80) and are seldom implemented on the development system itself. Commercial development systems, like EXORCIZER [46] (Motorola Corp.) and MDS80 (INTEL Corp.) [48] are typical.

4.4 A SPECIAL DEVELOPMENT SYSTEM FOR OCR

As we can see, the cross software has several advantages over the resident software on a development system. For example, the conditional macro assembler is an important facility but it is seldom found on a development system, cross compiler for high level languages provides the system designer a convenient way to program complex task and is usually only available on a powerful in house machine where the cross software resided. Besides these, the powerful and efficient file management and text editing packages available on the in house machine greatly facilitate the system designer in the preparation and modification of programs.

In view of the above facts, it would be ideal to have a development system that can take the merits from both conventional development system and the cross software. We furnish this idea by developing a time-sharing computer supported development system which allows a designer to have

a choice to use resident as well as cross software.

The proposed development system is specially designed for developing microprocessor based OCR systems. The configuration of the system is shown in Fig. 4.4. The system consists of an 8080A microcomputer system, a system console (CRT terminal), an optical scanner, a paper tape punch/reader and a high speed modem port. This system provides standard system commands to load, modify, execute and save a user program either via the system console, paper tape punch/reader or the time-sharing computer system through the modem port. The time sharing computer we are working with is a CDC CYBER 172/2 which works as an extended memory device and high level language compiler to the microcomputer system. We have the advantages of using conditional macro assembler, high level language compiler and powerful file management and text editing facilities on the CYBER computer system. On the microcomputer system, a resident assembler, text editor, an integer BASIC interpreter, tracing and debugging program are supported. The system also provides standard means to control an optical scanner which contains an array of photo-diode. Therefore, this development system can be used to develop general microprocessor based system as well as special microcomputer based systems for OCR.

This development system was actually implemented and used to develop our proposed "adaptive OCR" system and is

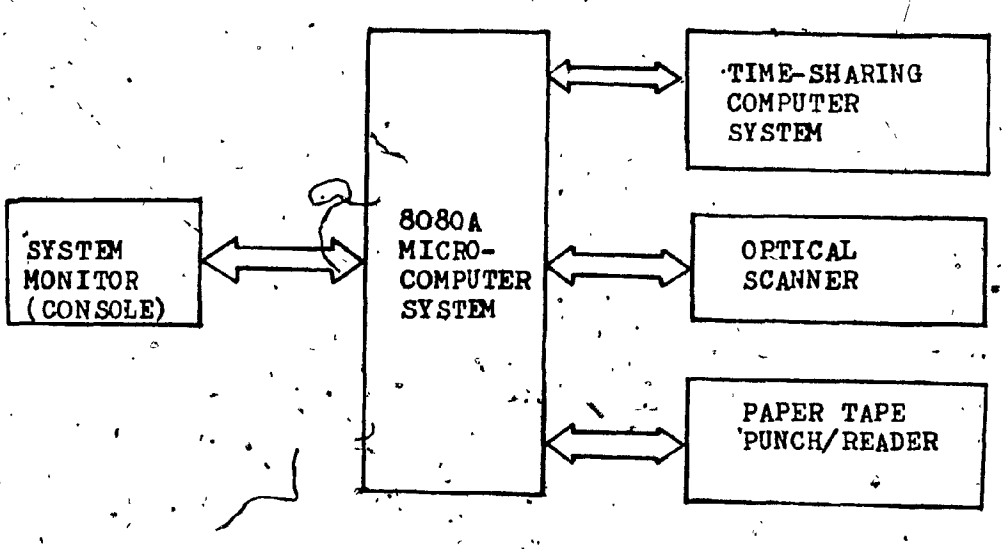


Fig. 4.4 A BLOCK DIAGRAM OF A TIME-SHARING COMPUTER SUPPORTED DEVELOPMENT SYSTEM

proven to be flexible, easy to use and efficient. Further information of the system can be found in the system reference manual written by the author [47].

CHAPTER V

DATA STRUCTURE AND MICROCOMPUTER BASED SOFTWARE DEVELOPMENT

The data structure required to implement the proposed adaptive OCR system is discussed in the first section. Then the microcomputer based software development is described in a modulized top-down approach.

5.1 DATA STRUCTURE

The proposed OCR system consists of three system tables --- the symbol table, the GPR table and the CNC table. The symbol table contains the identity of each pattern class corresponding to the GPRs in the GPR table. The CNC table serves as a key to a group of GPRs sharing similar topological structures. It is used in the first stage classification. Each entry of the CNC table consists of a CNC identity and a group of pointers pointing to those GPRs which have the same CNC as the identity of the entry.

To accomplish the adaptive nature of the proposed OCR system, special data structure has been employed to furnish the dynamic property of these system tables. Although the number of entries of each table may change during the learning and adaptive recognition mode of the system, a

record with fixed length can still be used in the symbol table as well as the GPR table because their entry size is usually determined before learning and adaptive recognition and is static throughout the rest of the process. On the other hand, the size of the entry of the CNC table changes along with the updating of the table for new patterns can always be adopted by the system and the corresponding CNC entry will be modified accordingly. The size of the entry of the CNC table varies from 1 to 10 or more depending on the type of the patterns being classified. Several dynamic data structures [52,53] can be adopted to implement the above table structure, such as linked list and detached key representation.

In order to implement such a dynamic table economically on a microcomputer, a variable length record, as shown in fig. 5.1, was proposed to represent each CNC entry. It consists of a record header and a record body. The record header contains the information of the identity of the entry and the length of the record body. In the record body, a list of index keys which can be used to find the entry addresses to the GPR table and symbol table was stored. There are two reasons that the index key was used instead of the entry addresses to those tables. First of all, an address takes 16 bits (2 bytes) in the INTEL 8080 microcomputer and two addresses must be stored for each pattern which is a very inefficient way of using the memory.

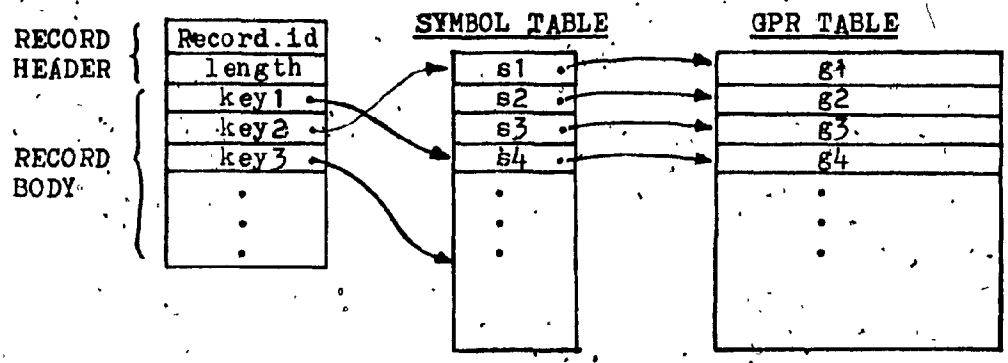


Fig. 5.1 A VARIABLE LENGTH CNC RECORD

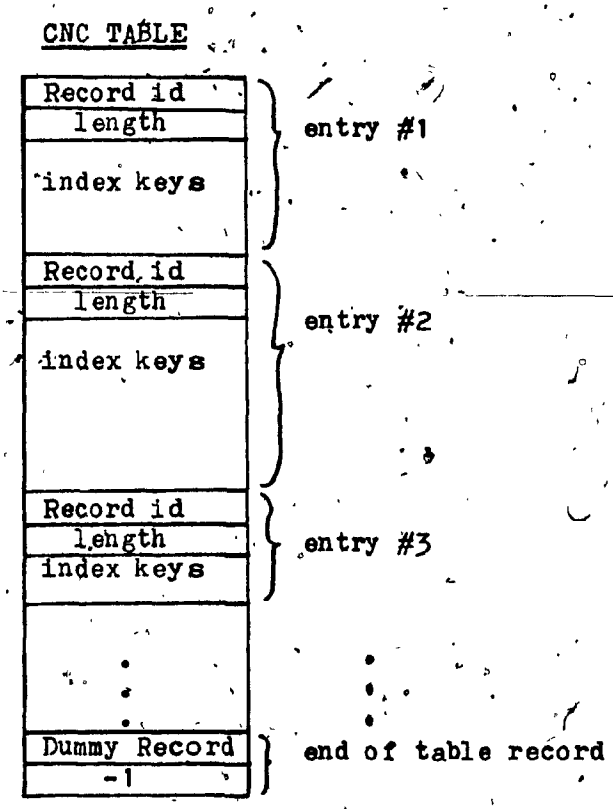


Fig. 5.2 STRUCTURE OF THE CNC TABLE

Secondly, the GPR table and the symbol table as mentioned before is one one correspondent to each other. A single index key will be sufficient to reference both of them. In our case, the ordinal number of the pattern class identity in the symbol table was used as an index key in the proposed CNC record body. There exist very simple formulae to calculate the entry addresses to those tables based on this index key.

$$A_{i,j} = BA_j + K_i * S_j$$

Where $A_{i,j}$ is the entry address to table j from index i,

BA_j is the base address of the table j,

K_i represents the ith index in the record body, and

S_j is the size of entry of table j.

The CNC table can be realized by a sequence of such CNC records and the end of table is denoted by a dummy record with record length equal to -1 (fig. 5.2).

There are two kinds of modification of the CNC table, namely opening a new CNC entry and expanding an existing entry. Special programs have been written to take care of these tasks. Normally, it is done by moving the table from the location of interest until the end of table down by the number of locations required for the modification and

expansion. Default table limits have been set inside the system. If all the default memory reserved for those tables has been used up, a memory overflow message will be signaled to the operator when further modification and expansion are requested. This default table limits can be altered for different applications.

Using the mentioned data structure, the following algorithm has been proposed to accomplish the searching of the CNC table.

[1] Compare the current CNC identity to the unknown. If matched, go to [3].

[2] Get current record length. Increment it by one. If it equals zero, go to [4]; else add it to the current address and the resultant address becomes that of the next record. Go to [1].

[3] Signal matched and stop.

[4] Signal end of table and stop (i.e. no match).

The advantage of the above searching algorithm is that no explicit information of the size of the CNC table is required during the searching for an end of table record has been added to the very last entry of the table. Thus the CNC table is allowed to be modified and no extra linkages are required to be updated before it can be used in the subsequent recognition process.

5.2 MICROCOMPUTER BASED SOFTWARE DEVELOPMENT

The naive concept of an adaptive system, as introduced in chapter one, has already been simulated and analyzed on a large scale computer as described in chapter three. However, the actual implementation mainly relies on the microcomputer based software development. In order to develop the software in an efficient and systematic way, modulized top-down approach was chosen.

A modulized top-down approach of system software development [49] has several advantages for developing, maintaining and expanding complex system software. With the modulization, a complex software task can always be broken down into smaller and manageable program modules. For example, in our case the proposed "adaptive OCR system" can be treated as a single piece of system software task. It can also be partitioned into four main modules, namely preprocessing, feature extracting, classifying and modifying the existing classification rules. And these modules can be further broken down into smaller program modules and so on. This functional partition distributes the complexity of the original task over several smaller tasks and thus a reduction in complexity is generally expected. This partition process can be accomplished with a task chart. A task chart resembles a tree structure. The high level task corresponds to the root and smaller tasks corresponds to the leaves. Those tasks described in the leaves level are not

necessary to occur concurrently. There is no explicit time relation between these leaves. In fact they only simply represent the required tasks. A simplified task chart of our proposed adaptive OCR system is given in fig. 5.3. As an example, the character image registration and justification in the preprocessing stage are two different tasks. They can be done sequentially or concurrently. It is subjected to the choice of the software designer and the limitation of the hardware.

Once a task chart is obtained, a complex system software task is ready to be partitioned into program modules in a hierarchical structure. In the top-down approach, high level program modules development is always completed before the lower level ones. This allows the system integration and testing at different levels. Whenever a lower level module is completed, it is added to the system and tested. Modification and expansion of the system will only involve addition or modification of some of the program modules because each module is already partitioned according to different functions. This approach is especially good for team projects as the task has already been partitioned.

With the proposed data structure and the task chart described in fig.5.3, corresponding microcomputer based program modules were developed and tested under the guide line of the top-down approach. The special OCR development system (section 4.4) has been used throughout the software

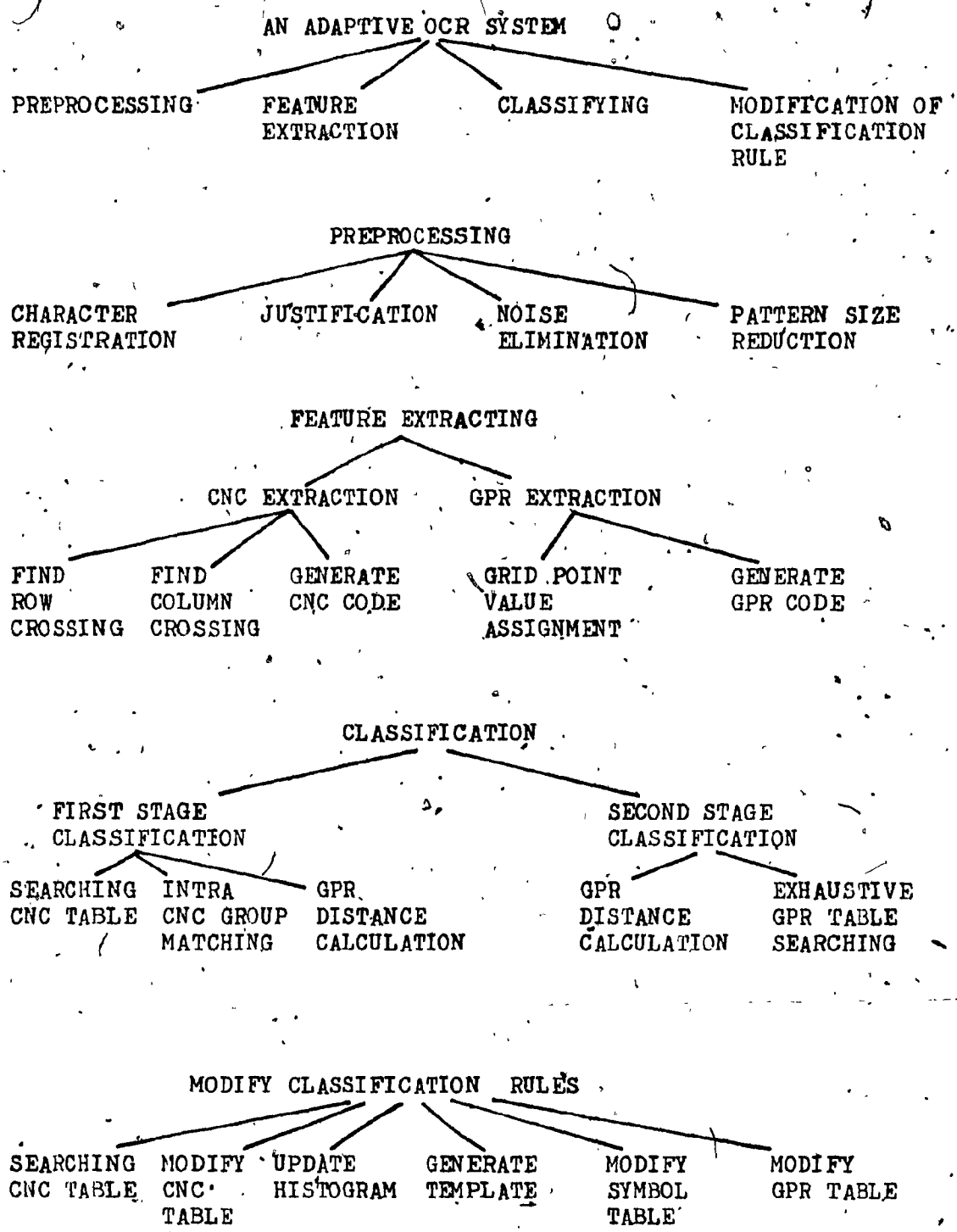


Fig. 5.3 A TASK CHART OF THE PROPOSED OCR SYSTEM

development process. Each program module was developed in INTEL 8080 assembly language [54,55] so as to achieve better efficiency. Some statistics of these program modules are listed in table 5.1. It is noticed that the average size of our program module is about 45 bytes as opposed to the overall software counting 1110 bytes excluding the data required. One will easily figure out how the modulized top-down approach assists the development of large and complicated software.

The developed microcomputer based software has been implemented on the microcomputer system and results similar to those obtained from the simulation program have been observed. Details of the result can be found in chapter 3 as well as in the final chapter.

MODULE	SIZE(BYTES)
ADAPTIVE OCR SYSTEM	1110
main line	19
A) PREPROCESSING	181
mainline	98
character registration and left justification	32
size reduction and top justification	51
(noise elimination is a by-product of above module)	
B) FEATURE EXTRACTION	291
mainline	24
a) CNC EXTRACTION	174
mainline	70
CNC code generation	35
row crossing finder	35
column crossing finder	54
b) GPR EXTRACTION	97
mainline	22
grid point value assignment	49
GPR code generation	26
C) CLASSIFICATION AND ERROR DETECTING	215
mainline	73
a) first stage classification	86
CNC group searching	37
intra CNC group matching	49
b) second stage classification	56
mainline	15
GPR distance calculation	41
D) CLASSIFICATION MODIFIER	404
mainline	130
table initialization	16
histogram initialization	22
update histogram	32
template generation	37
update symbol table	39
modification of GPR table	30
CNC group searching	37
open a new CNC entry	23
extend a CNC entry	38

TABLE 5.1 STATISTICS OF MODULES OF THE ADAPTIVE OCR SYSTEM

CHAPTER VI

CONCLUSIONS

The main results of this research are first summarized. Possible applications and suggestions for further work in this area are then given.

6.1 CONCLUSIONS

In this research an adaptive OCR system for the recognition of machine printed characters has been designed and was successfully implemented on a microcomputer system. A simulation program of the mentioned OCR system has also been developed on a large scale computer system so as to test the performance of the system. In order to study the adaptive nature of the system, OCR fonts (OCRA and OCRB) as well as common English (PICA and ELITE) and French (PRESTIGE CUBIC) fonts have been employed in the performance evaluation. A summary of the data base used in this research is listed in table 6.1.

In the testing of the adaptive nature of the system, the same simulation program was used to recognize the 5 mentioned type fonts by acquiring an automatic font analysis by the system just before the recognition. Very

SIZE (samples)	FONT	UPPER CASE (26 char.)	LOWER CASE (26 char.)	NUMERAL (10 char.)	SYMBOLS (15 symbols)
50	OCRA	yes	yes	yes	yes
44	OCRB	yes	yes	yes	yes
50	PICA10	yes	yes	yes	no
50	ELITE12	yes	yes	yes	no
50	PRESTIGE CUBIC	yes	yes	yes	yes (5 French symbols)

FONT	TOTAL NUMBER OF SAMPLES
OCRA	3850 samples
OCRB	3008 samples
PICA10	3100 samples
ELITE12	3100 samples
PRESTIGE CUBIC	3350 samples

TABLE 6.1 SUMMARY OF DESCRIPTION OF THE CHARACTER SAMPLES

satisfactory results have been observed. OCRA (99.79%) and OCRB (99.59%) have shown superiority over those non OCR fonts --- PICA (99.35%), ELITE (97.10%) and PRESTIGE CUBIC (94.63%). Confusion tables of these fonts can be found in the appendix.

The high performance of these OCR fonts confirms the necessity of the standardization of type fonts. Although the performances of those non-OCR fonts is less satisfactory, the result is still quite encouraging by considering that the recognition is done by an automatic font analysis of the system and no human intervention has ever been involved. The low performance of ELITE and PRESTIGE CUBIC can be accepted if we take a closer look at these font styles (fig. 6.1) because fewer distinguishable features are included in these fonts to differentiate characters with high resemblance in their topological structures. In fact, there is no distinction between the upper case "O" and the numeral "0", lower case "l" and numeral "1" in the PRESTIGE CUBIC font. This fact accounts for part of the reasons why a poor performance was observed for this type font. For example, there is no ambiguity of the template of the upper case "G" and "O" in ELITE (fig. 6.2) to human beings; however, if we take a measure of these two patterns in terms of our GPR feature, a Hamming distance of 4 was observed under a 2 by 2 grid. This small difference between the two templates might create a large

PICA 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

1 2 3 4 5 6 7 8 9 0

ELITE 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

1 2 3 4 5 6 7 8 9 0

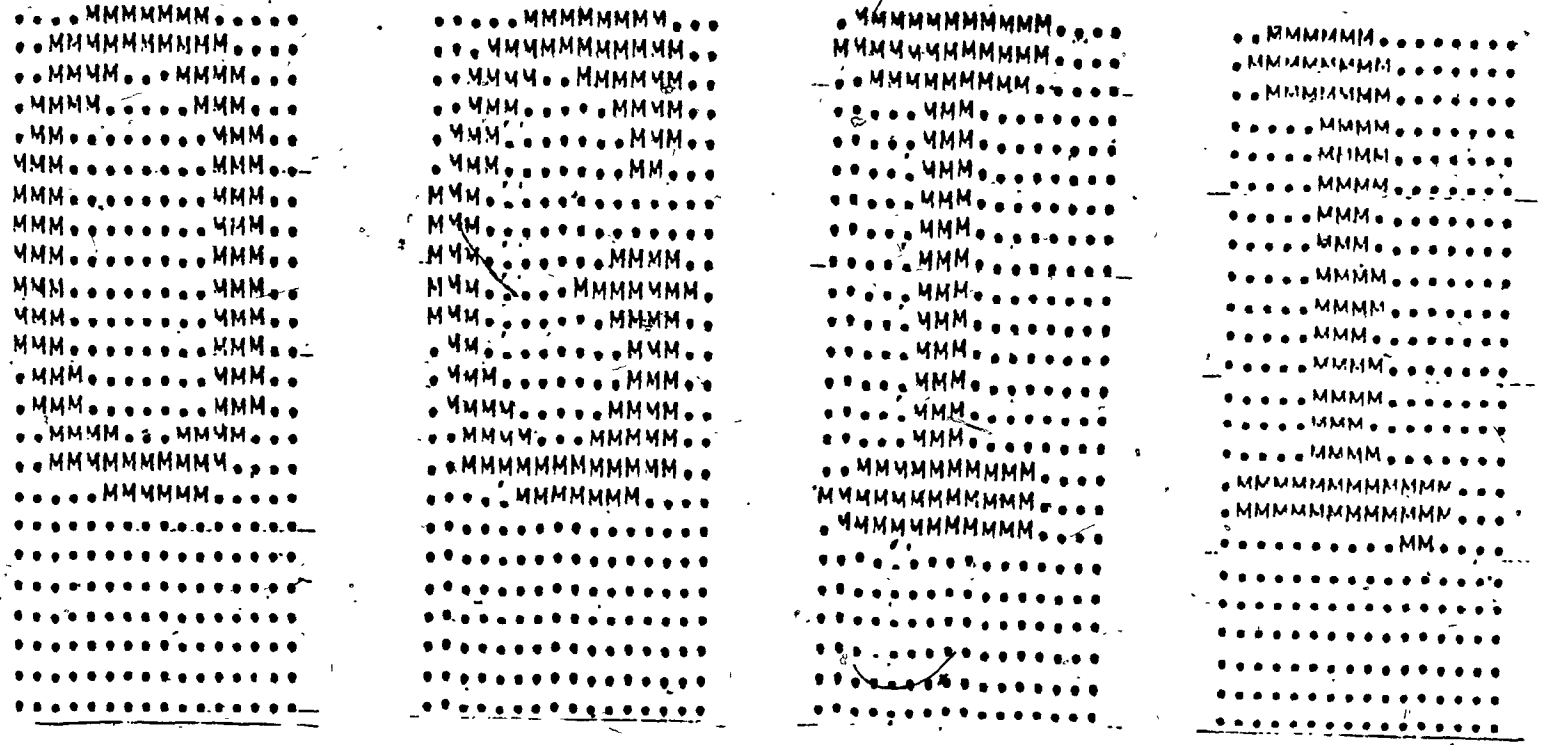
PRESTIGE CUBIC

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

1 2 3 4 5 6 7 8 9 0 à ç é è ù

Fig. 6.1 NON-OCR FONTS



TEMPLATE OF UPPER CASE "O"

TEMPLATE OF UPPER CASE "G"

TEMPLATE OF UPPER CASE "I"

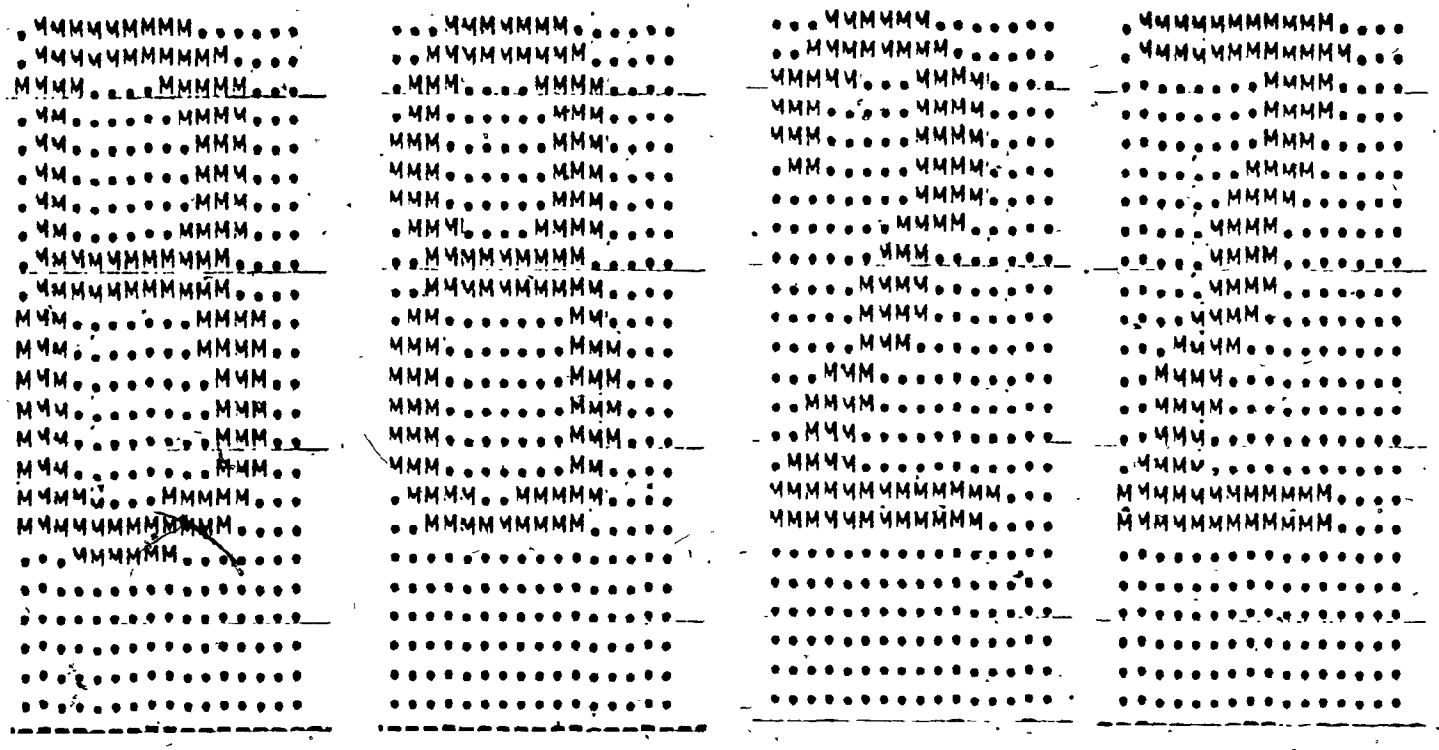
TEMPLATE OF LOWER CASE "l"

Fig. 6.2 SOME CONFUSION PAIRS OF NON-DCR FONTS (ELITE12)

ambiguity during the recognition process. The same situation has been found between upper case "I" and lower case "l" (fig. 6.2). Similar confusion pairs have been observed in PRESTIGE CUBIC as shown in fig. 6.3. Samples with typical defects and samples misrecognized are given in fig.6.4 and 6.5 correspondingly.

In order to test the flexibility of this system, the OCRA and PICA fonts have been chosen to test the system performance using different sizes of character set. At first, only the upper case alphabets of both fonts were tested, then the lower case alphabets and finally the numerals. For OCRA, 15 special symbols have also been used to test the system. Equally good performance has been observed over different sizes of character set. 100% recognition rate was observed in the case of numerals and special symbols. The result of the analysis is given in table 6.2.

The recognition speed of our system has been observed to be about 6.5 characters per second on a single processor microcomputer system under the testing of a 36 character OCRA set (upper case alphabets and numerals). The recognition speed might change when different fonts and sizes of character set were being recognized. According to the simulation, the average recognition speed stayed more or less the same for all the evaluation runs ranging from 35 to 45 characters per second on the CYBER 172/2 computer system.



TEMPLATE OF UPPER CASE "B"

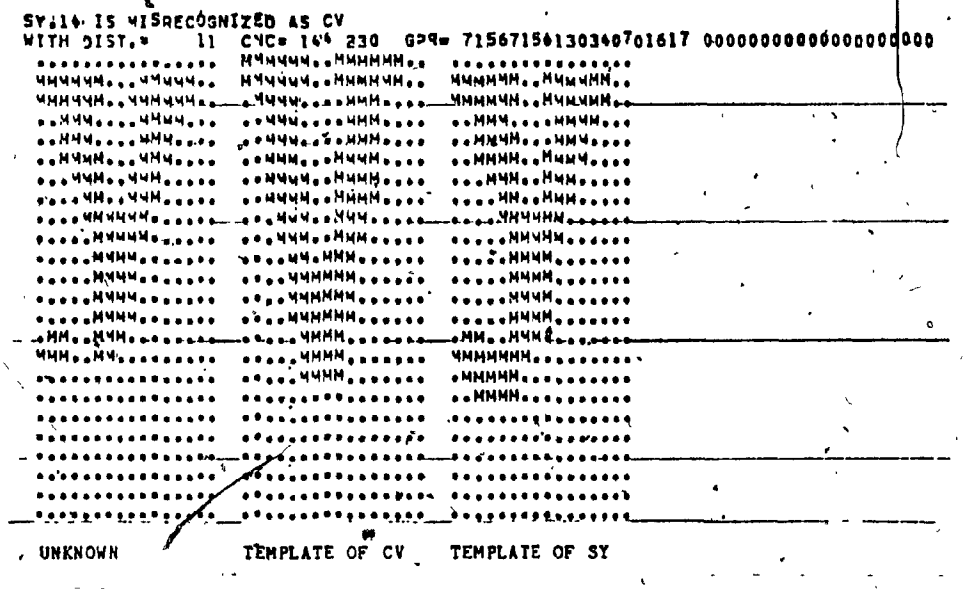
TEMPLATE NUMERAL "8"

TEMPLATE OF NUMERAL "2"

TEMPLATE OF UPPER CASE "Z"

Fig. 6.3 SOME CONFUSION PAIRS OF NON-OCR FONTS (PRESTIGE CUBIC)

Fig. 6.4 TYPICAL DEFECTS OF CHARACTER SAMPLES



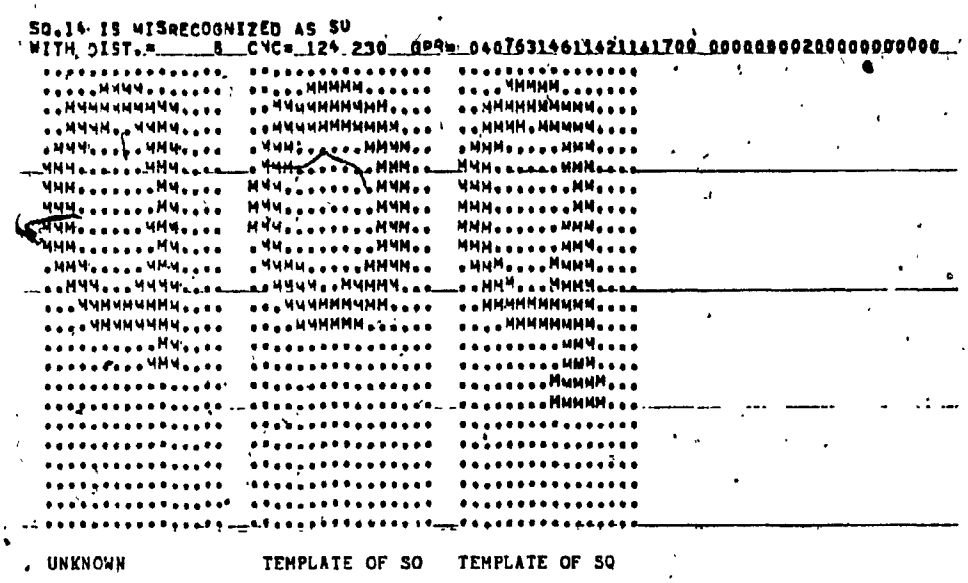


Fig.6.5 EXAMPLES OF MISRECOGNIZED SAMPLES

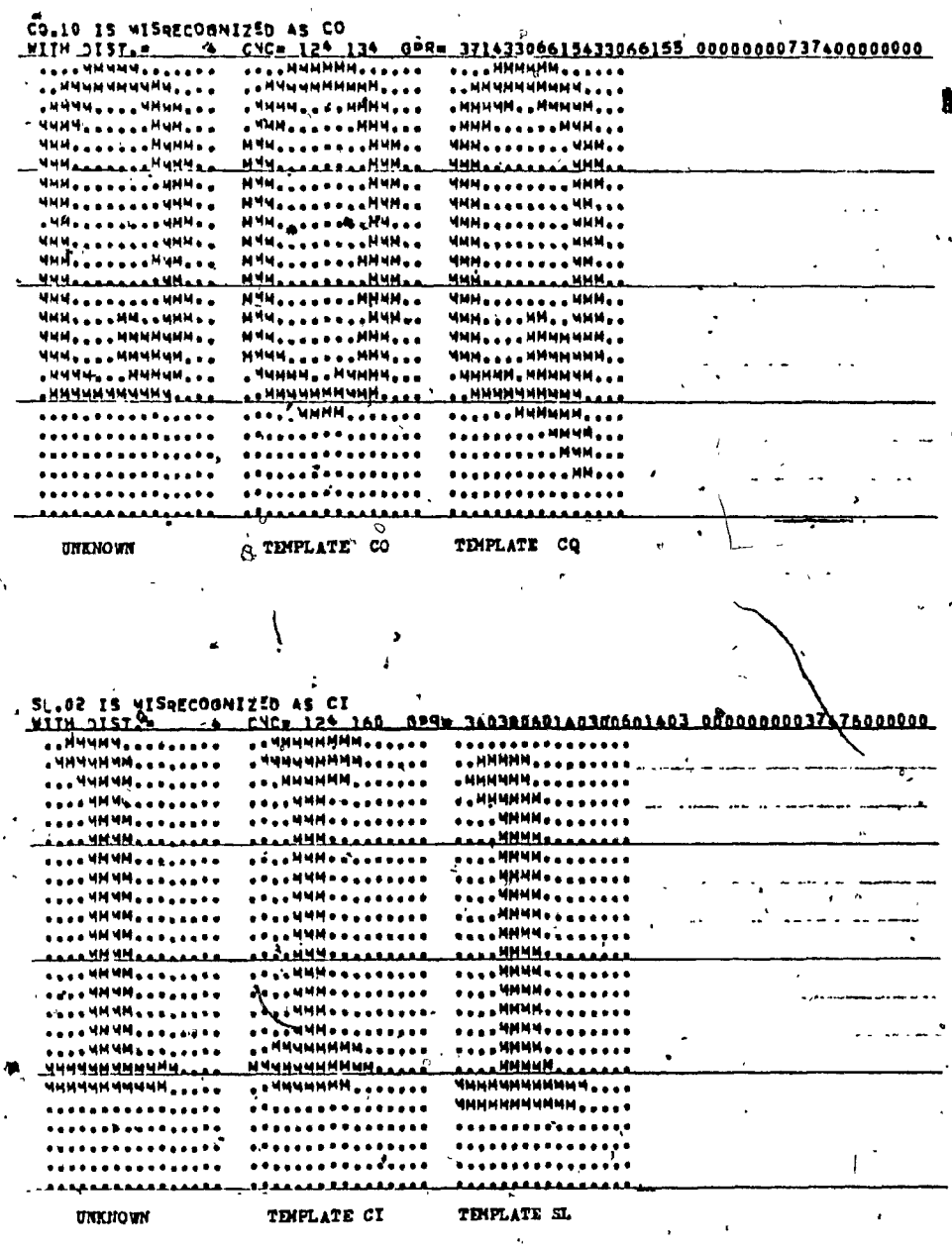
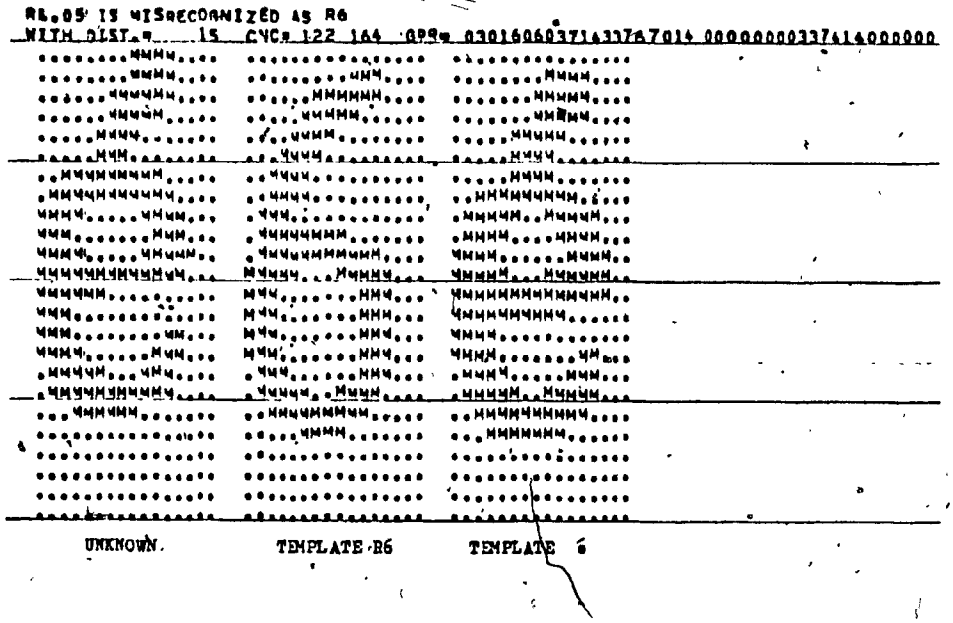


Fig. 6.5 EXAMPLES OF MISRECOGNIZED SAMPLES(cont.)



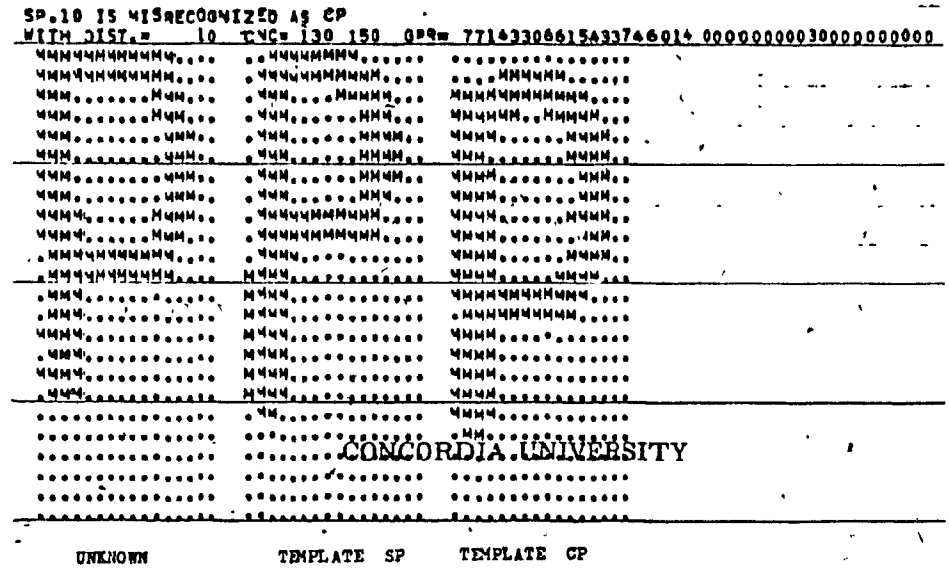


Fig. 6.5 EXAMPLES OF MISRECOGNIZED SAMPLES(cont.)

FONT	CHARACTER SET	RECOGNITION RATE
OCRA	UPPER CASE	99.54%
OCRA	LOWER CASE	99.85%
OCRA	UPPER AND LOWER CASE	99.61%
OCRA	NUMERALS	100.00%
OCRA	SPECIAL SYMBOLS	100.00%
OCRA	FULL SET	99.79%
PICA10	UPPER CASE	99.69%
PICA10	LOWER CASE	99.08%
PICA10	UPPER AND LOWER CASE	99.31%
PICA10	NUMERALS	100.00%
PICA10	FULL SET	99.35%

TABLE 6.2 RECOGNITION RATE AGAINST CHARACTER SET SIZE

This stability of the recognition speed can be accounted by the two stage classification scheme which alters the linear proportion of the recognition speed against the size of the character set. Better recognition speed on the microcomputer can be achieved by incorporating several microprocessors to funotion together.

Our recognition algorithm may not produce a recognition rate as high as that claimed by some OCR manufacturers. Since the objective of this research is to develop an adaptive system that can be implemented on a microcomputer system, the features we have chosen are simple and more general in nature as opposed to those used by OCR manufacturers. Also, commercial OCR readers are generally dedicated to one or two specific type fonts with a smaller character set than those we have tested. Because of the above arguments, the slightly lower recognition rates observed in our studies can be justified.

Despite the small drawback from the non-OCR fonts, the overall result of the performance evaluation is encouraging. The ability of recognizing different type fonts based on an automatic font analysis prior to the recognition without human intervention suggests a new approach in the OCR research.

In conjunction with the adaptive OCR system design, several tools have been proposed and were found useful in

OCR system development. A distance analysis of feature sets (section 3.2) has been designed which provides a systematic way to analyze and measure the effectiveness of a given set of features. This analysis can be easily extended to a tool for the construction of optimal feature sets. A special time sharing computer supported development system for OCR (section 4.4) has been developed to assist the software design of the microcomputer based OCR systems and was found to be very flexible and powerful in developing OCR software. Furthermore, it can also be used as a general purpose microcomputer based system development tool.

6.2 SUGGESTIONS FOR FURTHER WORK IN THIS AREA

One possible application of our system is a reading machine for the blind because the conventional reading machine is usually dedicated to read a particular type font and thus limits its usefulness. With the adaptive capability of our system, one can adapt the system to recognize different type fonts. Although, recognition rate may drop a little bit if non-OCR fonts are being recognized, this error rate may be tolerated by the blind people because small errors can be easily corrected by human beings using the context of the information being recognized.

Inexpensive voice output unit [56] can be used as the output of the system to provide a "spelled speech unit." If more intelligent speech synthesis algorithm is added to the

system, a "talking machine" can be realized.

In certain real time applications where the speed of our system may be found to be inadequate. Some frequently used and time consuming task can be replaced by hardware to achieve better system performance. For example, the inner most loop of the GPR distance calculation between the unknown and different templates can be replaced by hardware comparitor and adder. An alternative way of using multi-processors to achieve higher speed is also possible.

6.3 CONTRIBUTIONS

The main contribution of this research is the development of an adaptive OCR system that can be adapted to recognize different type fonts by an automatic font analysis algorithm of the system prior to the reading. This approach provides a new outlook for the future OCR research. This development has also proven the flexibility of the design of a microcomputer based system.

A special software development system for OCR has been developed to assist microcomputer based OCR system design. This system can be used as a laboratory tool for studying OCR algorithms in a real environment and it can also be used to study inter-computer communication processes between microcomputers and a large time sharing computer system.

REFERENCES

1. Tauschek,G. "Reading Machine" U.S. Pat. 2026329 Dec 1935 (appl. May 1929).
2. Suen,C.Y., "Advances in Optical Character Recognition", Canadian Computer Conference, Edmonton, Canada, 263 -268, May 1978.
3. Teitelman,W., "Real Time Recognition of Hand-drawn Characters", Proc. Fall Joint Computer Conf., 559 - 575, 1964.
4. Miller,G.M., "On-line Recognition of Hand Generated Symbols", Proc. Fall Joint Computer Conf., 399 - 412, 1969.
5. Berthod,M. and Maroy,J.P.,"Morphological Features and Sequential Information in Real-time Handprinting Recognition", Proc. 2nd Int. Joint Conf. on Pattern Recognition, 358 - 363, Aug 1974.
6. Caskey,C.L. and Coates,Jr.C.L., "Machine Recognition of Handprinted Characters", Proc. 1st Int. Joint Conf. on Pattern Recognition, 41 - 49, Oct 1973.
7. Tou,J.T. and Gonzalez,R.C., "Automatic Recognition and Multi-level Decision", International Journal of Computer and Information Science, vol.1 no.1, 43 - 65, 1972.
8. Harmon,L.D., "Automatic Recognition of Print and Script", Proc. IEEE, vol. 60 no. 10, 1165 - 1172, Oct 1972.

9. Nagy,G.,"State of Art in Pattern Recognition", Proc. IEEE vol. 56, 836 -862, May 1968.

10. Kanal,L., "Patterns in Pattern Recognition: 1968 - 1974", IEEE Trans. Information Theory, vol. II - 20, no. 6, 697 - 722, Nov 1974.

11. Suen,C.Y. "Advances in Optical Character Recognition", Proc. Canadian Computer Conference, May 1978.

12. "Guide to OCR and Mark Sense Readers", Auerbach, 1974.

13. Suen,C.Y., "State of the Art Report --- Optical Character Recognition", Canadian Datasystems, May 1974.

14. "Character Recognition 1971", by British Computer Society, 1971.

15. Holt,A.W., "The Impact of New Hardware on OCR Designs", Pattern Recognition, Pergamon Press 1976. Vol. 8, 99 - 105.

16. Zaks,R., "Microprocessors from Chips to Systems", SYBEX Inc. 1977.

17. Venkatesh,K., "A Microprocessor Based Character Recoghition System", Master thesis, Concordia Univ. Canada, 1977.

18. "Keytronic M9 Recognition Module", product description, Keytronic Corp. OCR devision, Spokane, Washington, USA, 1976.

19. "Style A Character Set for Optical Character Recoghition", Canadian Standards Association, 1975.

20. "Standard ECMA-11 for the Alphanumeric Character Set OCR-B for Optical Recognition", ECMA., Oct 1971.

21. "Optical Character Recognition and the Years Ahead", The Business Press, 1969.

22. Griffith,A.K., "The GRAFIX I System and its Application to Optical Character Recognition", Proc. 3rd Int. Joint Conference on Pattern Recognition, 650 - 652, Nov 1976.

23. Balm,G.J., "An Introduction to Optical Character Reader Considerations", Pattern Recognition, vol. 2, 151 - 166, Pergamon Press, 1970.

24. Meisel,S.M., "Computer Oriented Approaches to Pattern Recognition", Academic Press, 1972.

25. Dineen,G.P., "Programming Pattern Recognition", Proc. West Joint Computer Conference, page 94, 1955.

26. Unger,S.H., "Pattern Detection and Recognition", Proc. IRE, 1737 - 1752, 1959.

27. Doyle,W., "Recognition of Sloppy Handprinted Characters", Proc. West Joint Computer Conference, vol. 17, 133 - 142, May 1960.

28. Ullmann,J.R., "Pattern Recognition Techniques", Crane, Russak & Co. Inc., New York, 1973.

29. Tou,J.T. and Gonzalez,R.C., "Pattern Recognition Principles", Addision Wesley Publishing Co., 1974.

30. Fu,K.S., "Syntactic Methods in Pattern Recognition", Academic Press, 1974.

31. Andrews,H.C., "Introduction to Mathematical Techniques in Pattern Recognition", Wiley Interscience, 1972.

32. "MPS885, System Manual", Pro-Log Corp., 1975.

33. Weeks,R.W., "Rotating Raster Character Recognition System", AIEE Transaction, 80, pt.I, Communication and Electronics, 353 - 359, Sept 1961.

34. Rohland,W.S. (IBM), "Character Reader", US Pat. 2919426, 1959.

35. Kwon,S.K. and Lai,D.C., "Recognition Experiments with Handprinted Numerals", IEEE Joint Workshop on Pattern Recognition and Artifical Intelligence, 74 - 83, 1976.

36. Nadler,M., "Sequential-local Picture Operators", Proc. 2nd Int. Joint Conference on Pattern Recognition, 131 - 135, Aug 1974.

37. Nadler,M., "Structural Codes for Omnifont and Hand Written Characters", Proc. 3rd Int. Joint conference on Pattern Recognition, 135 - 139, Nov 1976.

38. Gudesen,A., "Quantitative Analysis of Preprocessing Techniques for the Recognition of Hand-printed Characters", Pattern Recognition, vol. 8, 219 - 227, 1976.

39. "8048 Microcomputer System User's Manual", INTEL Corp., 1976.

40. Klingman,E.E., "Microprocessor System Design", Prentice-Hall, Inc., 1977.

41. Hilburn,J.L. and Julich,P.M., "Microcomputers / Microprocessors, Hardware, Software, and Applications", Prentice-Hall, Inc., 1976.

42. Soucek,B., "Microprocessors and Microcomputers", Wiley-Interscience, 1976.

43. Saks,R., "Microprocessor", SYBEX, 1977.

44. "PL/M-80 Programming Manual", INTEL Corp., 1976 -1977.

45. "FORT/80 language manuals", by Unified Technologies Inc., 4800 Dundas St. W., Suite 2 Islington, Ontario Canada.

46. "M6800 Microprocessor Applications manual", Motorola Semiconductor Products Inc., 1975.

47. Chan,W.Y., "OCR Development System -- User's Manual", Internal Report, Dept. Of Computer Science, Concordia University, Montreal Canada, 1978.

48. "MDS80 Development system's manual", Intel Corp., 1977.

49. Huges,J.K. and Michtom,J.I., "A Structured Approach to Programming", Prentice-Hall Inc., 1977.

50. "ECRM 5000 Series Autoreader Users Guide", ECRM, Inc., june 1974.

51. Jensen,K. and Wirth,N., "Pascal User Manual and Report", Springer-Verlag, 1975.

52. Page,E.S. and Wilson,L.B.,"Information Representation and Manipulation in a Computer", Cambridge University Press, 1973.

53. Horowitz,E. and Sahni,S., "Fundamentals of Data Structures", Computer Science Press, Inc., 1976.

54. "MAC80 8080 Macro Assembler Reference Manual", INTEL Corp., 1975.

55. "TSC 8080 assembler reference manual", TSC Inc., 1976.

56. "Model-1000 Speech Synthesizer", AI Cybernetic systems, Sept 1976.

APPENDIX A

CONFUSION TABLES

OCRA:-

STIMULUS	RESPONSE	STIMULUS	RESPONSE	STIMULUS	RESPONSE
A(25)	A(25)	a(25)	a(25)	1(25)	1(25)
B(25)	B(25)	b(25)	b(25)	2(25)	2(25)
C(25)	C(25)	c(25)	c(25)	3(25)	3(25)
D(25)	D(25)	d(25)	d(25)	4(25)	4(25)
E(25)	C(1),E(24)	e(25)	e(25)	5(25)	5(25)
F(25)	F(25)	f(25)	f(25)	6(25)	6(25)
G(25)	G(25)	g(25)	g(25)	7(25)	7(25)
H(25)	H(25)	h(25)	h(25)	8(25)	8(25)
I(25)	I(23),T(1),Z(1)	i(25)	i(25)	9(25)	9(25)
J(25)	J(25)	j(25)	j(25)	0(25)	0(25)
K(25)	K(25)	k(25)	k(25)	$(25)	$(25)
L(25)	L(25)	l(25)	l(25)	%(25)	%(25)
M(25)	M(25)	m(25)	m(25)	&(25)	&(25)
N(25)	N(25)	n(25)	n(25)	*(25)	*(25)
O(25)	O(25)	o(25)	o(25)	+(25)	+(25)
P(25)	P(25)	p(25)	P(1),p(24)	-(25)	-(25)
Q(25)	Q(25)	q(25)	q(25)	=(25)	=(25)
R(25)	R(25)	r(25)	r(25)	"(25)	"(25)
S(25)	S(25)	s(25)	s(25)	'(25)	'(25)
T(25)	T(25)	t(25)	t(25)	:(25)	:(25)
U(25)	U(25)	u(25)	u(25)	;(25)	;(25)
V(25)	V(25)	v(25)	v(25)	/(25)	/(25)
W(25)	W(25)	w(25)	w(25)	.(25)	.(25)
X(25)	X(25)	x(25)	x(25)	,(25)	,(25)
Y(25)	Y(25)	y(25)	y(25)	?(25)	?(25)
Z(25)	Z(25)	z(25)	z(25)		

NOTE

X(nn) indicates the occurrence of character "X" to be nn times.

CONFUSION TABLE

OCRB:-

STIMULUS	RESPONSE	STIMULUS	RESPONSE	STIMULUS	RESPONSE
A(19)	A(19)	a(19)	a(19)	1(19)	1(19)
B(19)	B(19)	b(19)	b(19)	2(19)	2(19)
C(19)	C(19)	c(19)	c(19)	3(19)	3(19)
D(19)	D(19)	d(19)	d(19)	4(19)	4(19)
E(19)	E(19)	e(19)	e(19)	5(19)	5(19)
F(19)	F(19)	f(19)	f(19)	6(19)	6(19)
G(19)	G(19)	g(19)	g(19)	7(19)	7(19)
H(19)	H(19)	h(19)	h(19)	8(19)	8(19)
I(19)	I(17),;(2)	i(19)	i(19)	9(19)	9(19)
J(19)	J(19)	j(19)	j(19)	0(19)	0(19)
K(19)	K(19)	k(19)	k(19)	$(19)	$(19)
L(19)	L(19)	l(19)	l(19)	%(19)	%(19)
M(19)	M(19)	m(19)	m(19)	&(19)	&(19)
N(19)	H(2),N(17)	n(19)	n(19)	*(19)	*(19)
O(19)	O(19)	o(19)	o(19)	+(19)	+(19)
P(19)	P(19)	p(19)	p(19)	-(19)	-(19)
Q(19)	Q(19)	q(19)	q(19)	=(19)	=(19)
R(19)	R(19)	r(19)	r(19)	"(19)	"(19)
S(19)	S(19)	s(19)	s(19)	'(19)	'(19)
T(19)	T(19)	t(19)	t(19)	:(19)	:(19)
U(19)	U(19)	u(19)	u(19)	;(19)	I(2),;(17)
V(19)	V(19)	v(19)	v(19)	/(19)	/(19)
W(19)	W(19)	w(19)	w(19)	.(19)	.(19)
X(19)	X(19)	x(19)	x(19)	,(19)	,(19)
Y(19)	Y(19)	y(19)	y(19)	?(19)	?(19)
Z(19)	Z(19)	z(19)	z(19)		

CONFUSION TABLE

PICA10:-

STIMULUS	RESPONSE	STIMULUS	RESPONSE	STIMULUS	RESPONSE
A(25)	A(25)	a(25)	a(25)	1(25)	1(25)
B(25)	B(25)	b(25)	b(25)	2(25)	2(25)
C(25)	C(23),G(2)	c(25)	c(25)	3(25)	3(25)
D(25)	D(25)	d(25)	d(25)	4(25)	s(1),4(24)
E(25)	E(25)	e(25)	e(25)	5(25)	5(25)
F(25)	F(25)	f(25)	f(25)	6(25)	6(25)
G(25)	G(25)	g(25)	g(25)	7(25)	7(25)
H(25)	H(25)	h(25)	b(1),h(24)	8(25)	8(25)
I(25)	I(25)	i(25)	i(25)	9(25)	9(25)
J(25)	J(25)	j(25)	j(25)	0(25)	0(25)
K(25)	K(25)	k(25)	k(25)		
L(25)	L(25)	l(25)	l(25)		
M(25)	M(25)	m(25)	m(25)		
N(25)	N(25)	n(25)	n(23),o(2)		
O(25)	O(25)	o(25)	o(25)		
P(25)	P(25)	p(25)	p(25)		
Q(25)	Q(25)	q(25)	q(25)		
R(25)	R(25)	r(25)	r(25)		
S(25)	S(25)	s(25)	s(25)		
T(25)	T(25)	t(25)	o(1),t(21),z(3)		
U(25)	U(25)	u(25)	u(25)		
V(25)	V(25)	v(25)	v(25)		
W(25)	W(25)	w(25)	w(25)		
X(25)	X(25)	x(25)	x(25)		
Y(25)	Y(25)	y(25)	y(25)		
Z(25)	Z(25)	z(25)	z(25)		

CONFUSION TABLE

ELITE12:-

STIMULUS	RESPONSE	STIMULUS	RESPONSE	STIMULUS	RESPONSE
A(25)	A(25)	a(25)	a(24),e(1)	1(25)	1(5),t(1),1(19)
B(25)	B(25)	b(25)	b(25)	2(25)	2(25)
C(25)	C(25)	c(25)	c(25)	3(25)	3(25)
D(25)	D(24),O(1)	d(25)	d(25)	4(25)	4(25)
E(25)	E(25)	e(25)	a(1),b(1),e(23)	5(25)	5(25)
F(25)	F(25)	f(25)	f(25)	6(25)	6(25)
G(25)	G(25)	g(25)	g(24),q(1)	7(25)	7(25)
H(25)	H(24),8(1)	h(25)	h(25)	8(25)	8(25)
I(25)	I(25)	i(25)	i(23),l(2)	9(25)	q(1),9(24)
J(25)	J(25)	j(25)	j(25)	O(25)	G(1),O(6),O(18)
K(25)	K(25)	k(25)	k(25)		
L(25)	L(25)	l(25)	l(21),1(4)		
M(25)	M(25)	m(25)	m(25)		
N(25)	N(22),R(3)	n(25)	n(25)		
O(25)	G(3),U(1),O(21)	o(25)	o(25)		
P(25)	F(1),P(24)	p(25)	p(25)		
Q(25)	O(1),Q(24)	q(25)	o(1),q(24)		
R(25)	R(25)	r(25)	r(25)		
S(25)	S(25)	s(25)	a(1),s(23),z(1)		
T(25)	T(24),1(1)	t(25)	t(25)		
U(25)	J(1),U(24)	u(25)	u(25)		
V(25)	V(24),v(1)	v(25)	v(25)		
W(25)	W(25)	w(25)	w(25)		
X(25)	X(25)	x(25)	x(25)		
Y(25)	V(1),Y(24)	y(25)	y(25)		
Z(25)	Z(25)	z(25)	a(1),z(24)		

CONFUSION TABLE

PRESTIGE CUBIC:-

STIMULUS	RESPONSE
A(25)	A(25)
B(25)	B(25)
C(25)	C(25)
D(25)	D(23),O(2)
E(25)	E(25)
F(25)	F(25)
G(25)	G(25)
H(25)	H(25)
I(25)	I(19),t(6)
J(25)	J(25)
K(25)	K(25)
L(25)	L(25)
M(25)	M(25)
N(25)	M(1),N(24)
O(25)	D(7),G(2),O(12),O(4)
P(25)	P(25)
Q(25)	D(1),O(3),Q(21)
R(25)	R(25)
S(25)	S(25)
T(25)	T(25)
U(25)	U(25)
V(25)	V(25)
W(25)	W(25)
X(25)	X(25)
Y(25)	Y(25)
Z(25)	Z(25)
1(25)	1(24),1(1)
2(25)	Z(2),2(23)
3(25)	3(25)
4(25)	4(25)
5(25)	5(25)
6(25)	6(25)
7(25)	7(25)
8(25)	B(4),8(21)
9(25)	9(25)
O(25)	D(3),O(5),O(17)

STIMULUS	RESPONSE
a(25)	a(25)
b(25)	b(25)
c(25)	c(25)
d(25)	d(25)
e(25)	a(1),e(22),u(2)
f(25)	f(25)
g(25)	g(25)
h(25)	h(25)
i(25)	I(3),i(18),l(3),t(1)
j(25)	j(25)
k(25)	k(25)
l(25)	I(7),l(16),t(2)
m(25)	m(25)
n(25)	n(25)
o(25)	o(25)
p(25)	P(3),p(22)
q(25)	q(25)
r(25)	r(25)
s(25)	s(24),w(1)
t(25)	l(1),t(24)
u(25)	u(25)
v(25)	v(25)
w(25)	w(25)
x(25)	x(25)
y(25)	y(25)
z(25)	z(25)
à(25)	à(25)
ç(25)	ç(25)
é(25)	é(23),6(2)
è(25)	è(25)
ù(25)	ù(25)

APPENDIX B

CNC AND GPR TABLES GENERATED FROM 25 SAMPLES OF EACH MODEL OF UPPER-, LOWER- CASE AND NUMERIC OF THE ELITE12 TYPE FONT.

CNC TABLE:-

```
144 150     3 R4 SY CA
124 150     8 R4 SY SQ SP SK SD SB CA
144 130     4 SY SH CV CA
124 130    10 R0 SQ SP SK SH SD SB CR CD CA
140 150     2 R4 CA
025 130     3 CK CD CB
025 230     1 CD
125 230     2 SN CB
122 130     1 CB
125 130     4 SP CK CE CB
121 230     1 CB
121 130     1 CB
110 144     2 CZ CC
030 144     2 CY CC
110 150     3 CX CQ CC
130 150     5 SK SB CX CQ CC
111 144     2 CZ CC
030 134     2 CO CD
024 230     1 CD
124 134     5 R0 CU CR CK CD
130 134     5 R0 CR CO CK CD
024 134     1 CD
105 150     6 R9 R6 SG CZ CS CE
106 150     2 CS CE
105 144     3 CZ CS CE
125 150     5 R9 R6 SQ SP CE
105 130     3 R8 SG CE
106 130     1 CE
125 144     2 CP CF
121 244     1 CF
125 244     3 SS SE CF
121 144     1 CF
011 130     3 CQ CK CG
010 130     3 CX CQ CG
031 130     1 CG
111 130     3 CQ CK CG
030 130     3 SK CX CG
130 130     7 R0 SK CX CR CQ CK CG
105 134     3 R8 CK CG
031 134     1 CG
111 134     1 CG
110 130     3 CX CQ CG
025 134     2 CK CH
041 114     1 CH
125 114     1 CH
121 114     1 CH
025 114     1 CH
021 114     1 CH
121 134     2 R8 CH
021 134     1 CH
130 160     1 CI
121 160     7 R7 ST SL SJ SL SF CI
104 160     2 SJ CJ
110 160     1 CI
144 161     4 R7 SF CL CJ
124 164     5 R7 ST SJ SF CJ
```

```
144 144    1 CJ
140 164    3 R7 CL CJ
124 144    5 ST SQ CY CP CJ
125 134    3 R8 CU CK
040 164    1 CL
045 111    1 CM
044 115    2 CW CM
044 111    2 CW CM
044 114    2 CW CM
024 111    1 CN
124 115    1 CN
121 115    1 CN
124 215    1 CN
125 115    1 CN
124 114    1 CN
125 111    1 CN
124 111    1 CN
024 115    1 CN
024 114    1 CN
144 115    1 CN
030 150    2 CX CQ
110 134    1 CQ
010 134    1 CQ
130 230    3 SX SO CR
124 230    4 SO SN SK CR
005 144    1 CS
005 150    2 SG CS
106 144    1 CS
124 161    1 CT
024 161    1 CT
144 161    1 CT
144 134    1 CU
044 130    3 SH CW CV
140 130    1 CV
044 131    1 CW
144 111    1 CW
044 134    1 CW
041 131    1 CW
040 131    1 CW
010 150    1 CX
130 144    2 SP CY
024 144    1 CY
111 150    1 CZ
121 224    1 SA
125 224    2 SE SA
130 224    2 SX SC
130 244    1 SC
110 244    1 SC
110 224    3 SZ SX S
124 250    4 SY SP SI SD
105 224    3 SZ SS SE
105 244    2 SS SE
106 244    1 SE
141 164    1 SF
121 164    2 R9 SF
141 160    1 SF
```

```
121 160     2 SI SF
106 250     1 SG
106 230     1 SG
006 230     1 SG
006 130     1 SG
006 150     1 SG
005 130     1 SG
124 260     2 SL SI
125 260     1 SI
125 160     2 SJ SI
105 260     1 SI
124 064     1 SJ
104 164     1 SJ
124 060     1 SJ
105 160     3 R5 R3 SJ
030 230     1 SK
144 160     1 SL
024 222     1 SM
044 202     1 SM
030 222     1 SM
024 206     1 SM
044 222     1 SM
024 202     1 SM
044 206     1 SM
044 205     1 SM
144 210     3 SW SU SN
125 210     1 SN
124 210     1 SN
144 230     1 SN
144 224     2 SV SR
144 244     1 SR
140 244     1 SR
105 230     1 SS
140 230     2 SV SU
140 210     1 SU
144 214     1 SU
140 214     1 SU
140 224     1 SV
044 210     1 SW
044 224     1 SW
044 230     1 SW
040 210     1 SW
060 210     1 SW
060 230     1 SW
040 230     1 SW
111 224     1 SZ
111 230     1 SZ
111 244     1 SZ
220 160     1 R1
106 150     3 R5 R3 R2
105 164     3 R5 R3 R2
026 164     1 R2
106 164     2 R5 R2
121 150     2 R9 R6
101 150     1 R6
```

GPR TABLE:-

CA	1607417036066174&316	0000000073140000000
CB	7704310637072463157	0000000060000000000
CC	1746730660146300&147	0000000070300000000
CD	7606614631463146315&7	0000000061600000000
CE	7754310636074110215&7	0000000073740000000
CF	7736315236074150301&7	0000000001400000000
CG	1746730660147706614&7	0000000070300000000
CH	7346310633471063146	0000000033560000000
CI	7703406014030601407	0000000063740000000
CJ	0740601403006154631&7	0000000061700000000
CK	7344613034070130230&6	0000000073560000000
CL	7016014030060140314&6	0000000073780000000
CM	6176354777777533607&6	0000000017430000000
CN	7346316635457136235&6	0000000071440000000
CO	1706330661543306614&7	0000000060700000000
CP	7706314631476140301&6	0000000001400000000
CQ	1706230661543306774&7	0000000070360140000
CR	7706314637076154331&6	0000000071000000000
CS	3714314436006006615&7	0000000070300000000
CT	7757737606034070060&3	0000000060740000000
CU	7354330661443106214&7	0000000060700000000
CV	7346615433066070160&3	0000000040200000000
CW	7174350627477176374&6	0000000070440000000
CX	7346607416034070331&6	0000000073560000000
CY	7346615416034070160&7	0000000061740000000
CZ	7914623016030140315&7	0000000071740000000
SA	1417611437146316374&0	0000000000000000000
SB	4014036077143306215&6	0000000021740000000
SC	1407631663140156170&0	0000000000000000000
SD	0300603437146314631&4	0000000061760000000
SE	1407630677540156170&0	0000000000000000000
SF	0703714676060140300&6	0000000037000000000
SG	1747751433074176617&4	0000000157600000000
SH	6014011037063106214&6	0000000033560000000
SI	1403000074030060140&3	0000000037400000000
SJ	0201601017006014030&0	0000000601463074000
SK	6016014223456170360&6	0000000635600000000
SL	3017006014030060140&3	0000000037400000000
SM	1557773366755733774&0	0000000000000000000
SN	0617614621443106734&0	0000000000000000000
SO	0607630661543156170&0	0000000000000000000
SP	0617614621443154370&4	0000000003400000000
SQ	1407631461146114370&0	0000000060160000000
SR	0317714630060140700&0	0000000000000000000
SS	1417635437103354770&0	0000000000000000000
ST	2006036076060140310&6	0000000060700000000
SU	4314630461147316354&0	0000000000000000000
SV	6354615432074070040&0	0000000000000000000
SW	6054350637477156310&0	0000000000000000000
SX	7346707016034154734&0	0000000000000000000

SY	735671541303407016l7	0000000001400000000
SZ	37176330140623767700	0000000000000000000
R1	3417026014030060l403	0000000004000000000
R2	160763060141616060l6	0000000063740030000
R3	36146014030440140100	0000000201476000000
R4	30064154330663147700	0000000060140200000
R5	77140300761460060300	0000000063700000000
R6	17060100741763146114	0000000021741400000
R7	77546210060340601403	0000000006000000000
R8	36046314330763046154	0000000031741600000
R9	36166314631461740300	0000000060303000000
R0	16076314615433066106	0000000060700000000