1969

An algorithm for the recognition of handprinted alphameric characters

Ronald Emil Haglund
Iowa State University

69-20,645

HAGLUND, Ronald Emil, 1933-
AN ALGORITHM FOR THE RECOGNITION OF
HANDPRINTED ALPHAMERIC CHARACTERS.

Iowa State University, Ph.D., 1969
Engineering, electrical

University Microfilms, Inc., Ann Arbor, Michigan

AN ALGORITHM FOR THE RECOGNITION OF HANDPRINTED ALPHAMERIC CHARACTERS

by

Ronald Emil Haglund

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major Subject: Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

Head of Major Department

Signature was redacted for privacy.

Dean of Graduate College

Iowa State University
Ames, Iowa

1969

TABLE OF CONTENTS

INTRODUCTION

Great interest has been shown over the past few years in the problems of character and pattern recognition. This interest stems from the desire to apply information systems to new classes of problems and from the need for new input-output systems for central processors.

With the advent of faster, larger, and more sophisticated central processors, it has become necessary to seek new methods of man-machine communication and to eliminate inefficiences in present techniques. An optical character recognition system seeks a solution to this problem by eliminating input forms such as punched cards or punched paper tape used only for man-machine communication.

A character recognition system must, on the basis of stored information, separate and identify an unknown character from the set of all possible characters for which the system was designed. In order to perform this separation and ultimate identification, the system needs information concerning the characteristics of the anticipated inputs and a set of rules for operating on those characteristics.

The set of characteristics needed by the system for the recognition process must be extracted from a sample set representing those characters that it is expected to identify. Thus the sample set should include samples from all possible input devices. In the case of multi-font systems designed to recognize machine printed characters of variable style, it is necessary to insure that all type fonts are represented and that a sample of the printing characteristics of each printing device be included. For the recognition of handprinted characters the designer

needs a sample which represents the idiosyncracies found in characters produced by the human system.

If a system is designed to recognize only one type font, then only that font need be included in the sample space. Likewise if a system were to recognize the printing of one person, then only that person need be included in the sample set.

The set of rules or the algorithms that a recognition system uses in the recognition process may take many forms (38). In general a multi-stage process is used where there is great variation in the character set. Handprinted characters due to their many variations require a multi-stage system.

The recognition algorithm operates on the unknown character with a series of transformations to improve the vector separation distance from all possible nearest neighbors in character identification space and to extract the information needed for recognition. The algorithm must then use that information to make a decision as to the identity of the character.

One characteristic of most recognition schemes is that they are performed on a bit-by-bit basis. This type of task when performed on a general purpose machine becomes prohibitively expensive and thus suggests the desirability of a special purpose processor. A processor designed to perform most efficiently those operations needed for character recognition. Such a processor has been proposed (53).

This small processor is capable of efficiently performing the necessary bit-by-bit manipulations so common to recognition systems. The processor contains only limited arithmetic capability, thus a recognition

scheme to be used with this processor would have to make extensive use of boolean manipulations.

Character recognition systems are designed to recognize a specific set of characters. Some recently introduced commercial systems (2, 43) will read handprinted numbers plus a few alphabetic characters. One announced system will read a set of 40 characters. The larger character set provides a system with a more general utility.

It is characteristic of the above mentioned commercial systems that they are not designed to accept inputs from an untutored user. Each requires some degree of training for those who are producing input documents for the system. While training may ultimately become a standard requirement for handprinted character recognition systems, it would be more desirable to have a system which accepts characters produced by an untutored user.

Problem Definition

This investigation is part of a project which seeks to develop a recognition system for handprinted alphameric characters. The system includes both a processor and a recognition algorithm. While the project is research oriented and ultimately seeks to investigate many recognition algorithms, the motivation for the system came from a possible application as an input device for short student programs at a university computation center.

An economic justification for character recognition systems can be provided (44) based usually on the number of documents that have to be read. There is also an economic justification for using a character

recognition system in a university computation center, but it is based more on the amount of equipment used to prepare student programs. The computation center incurs considerable expense providing keypunch equipment and services to be used by the students.

Zingg (53) has designed a small special purpose processor as a research tool for investigating character recognition algorithms. It is the fundamental purpose of this investigation to develop a recognition scheme to be used with this processor. The recognition algorithm will make extensive use of the bit manipulation capability of the processor and avoid, as much as possible, the use of the processors limited and relatively inefficient arithmetic capability. This algorithm is to produce a recognition rate of 100 characters per second.

The recognition system is to be individualized for each user. Samples of an individual users handprinted characters are to be collected and used to recognize only that users printing.

Individualizing a system for each user is not the approach that is commonly taken for character recognition systems. Systems must of necessity consider a broader applicability. An individualized system need not be considered of a narrow applicability if the process of adapting to each user can be done quickly and with a minimum of cost.

Since the more general system is to recognize characters produced by a large population, it is necessary to collect a large number of samples. This sample population must be large enough to be statistically significant.

The approach to the general system with its requirements of a large sample population was ruled out due to cost. All trial runs and

algorithm design work would have to be done on the 360/65 computer because the recognition processor would not be complete. Earlier experience with the large system and the cost of bit-by-bit manipulation of character arrays indicated that this approach to the problem would be quite expensive. It was doubtful that local support could be obtained. This approach to the problem would have to await the completion of the processor.

In addition, cognizance was taken of the fact that an individualized system approach would also require a study of the time-varying characteristics of an individuals printing. However this type of study is also very expensive and would have to await processor completion.

This investigation also seeks the design of a data acquisition system to act as an input device to the recognition processor. The input system is to provide sample sets of handprinted characters to be used for testing the recognition algorithm on a larger computer. Design of the input system will also include design of the input document and selection of a suitable writing instrument.

LITERATURE REVIEW

The general problem of pattern recognition has received the attention of many investigators over the last decade. While many have chosen to attack the problem in its full generality, many more have chosen to investigate a special case of pattern recognition, character recognition. In the latter, the inputs or unknowns to the recognition system become a known subset of patterns, a character set, instead of the more generalized patterns that may be considered for a pattern recognition system.

In spite of the possible differences in pattern sets, it is quite common to test proposed pattern recognition systems on an alphameric character set or portion thereof.

The problem of character recognition is a multi-step one. The complexity of the character set and the environment in which it is produced are the determining factors with respect to the number of steps in the process. If the system is to recognize only single font characters from printing devices with controlled characteristics, a single stage system using template matching works well. Should the inputs be hand-printed characters with the usual wide range of characteristics, a multi-stage system is required.

The input element of a recognition system uses a scanning apparatus to convert the character image into one suitable for processing. The character is then subjected to a processing scheme designed to provide a more uniform or invariant form of the image. The information needed for recognition is then extracted from the character and finally subjected to a set of decision criteria.

The literature for both pattern and character recognition has grown to the extent that a review of the field has become a major undertaking and produces an extensive bibliography. Fortunately, there are several good sources of review material, both past and present (13, 30, 38, 46, 48).

Approaches to this Problem

There have been many significant approaches to the problem of character recognition, all with varying degrees of success. One characteristic is common to many of these approaches. The method of solution is quite often determined by a very specific task. That is, a system is to be designed to accept a specified character set produced by input devices with a specified range of print variations. In those approaches which tend to be more generalized, the solution may also be limited by the availability of input data or the limited capabilities of the input devices and processing equipment.

The common approaches allow recognition techniques to be grouped on the basis of what information is extracted for recognition, how it is extracted, and what decision criteria are used to identify the character.

Character properties used for recognition

Many properties have been extracted from characters for recognition schemes. One approach, commonly referred to as feature analysis, looks for the geometric or topological characteristics of the character. Bomba (5) uses line segments while Doyle (15) uses shape characteristics. Grimsdale *et al*. (21) use strokes, lines, and curves. Ungar (50) has used cavities and holes. Munson (36) uses edge detection masks.

Many investigators have used curve tracing or variations thereof to obtain the topological characteristics from a character (20, 26, 31, 45, 51). This method is attractive because of its relative insensitivity to size and positional variations. This advantage is also present in the use of moments (1).

The advent of real-time computing systems plus new input devices such as the Rand tablet have produced much interest in on-line character recognition (9, 22, 47). These systems use the stroke sequence information to recognize the character. This also is a form of curve tracing.

The previously mentioned approaches are characterized by an a priori definition of the features. Other investigators have used generated sets of characteristics. Bledsoe and Browning (4) use randomly-generated 2-tuples. Others have considered similarly generated N-tuples (29, 34). Bonner (6) seeks an automated approach to descriptor generation as does Lewis (33). Lewis also attaches a measure of "goodness" to the selected characteristic. Uhr and Vossler (49) would have the system generate, evaluate, and adjust its own parameters. Nagy and Shelton (39) have the system produce a feature vector for an unknown character on the basis of a coarse description of the possible classes. Casey and Nagy (10) use heuristic and iterative algorithms to partition the data.

Features extracted from handprinted characters produced by an untutored population exhibit great variations. As one approach to this problem some investigators (12, 20, 43) have employed training for potential users of a system. Dimond (14) and Minneman (35) have used constraints on the input document to normalize a character and reduce

its variations.

Some investigators have looked at the problem using an untutored population and made use of context to improve handprinted recognition systems (17, 36).

Decision mechanisms

In order to identify the unknown character it is necessary to apply a decision criteria to the information extracted from the character. Bomba (5) and Ungar (50) have used decision trees. Chow (11), Duda and Fossum (16), and Highleyman (25) have used decision functions, both linear and piecewise linear. Cross-correlation was used by Minneman (35) and Highleyman (24).

While most of the systems perform their testing in parallel, Fu *et al.* (19) suggest an approach using sequential testing with a decision after each test. This would minimize the amount of testing.

Adaptive Recognition Schemes

It is also possible to categorize some of the above as adaptive recognition systems (4, 16). Bonner (7) and Roberts (42) have proposed adaptive schemes. These systems modify the decision criteria on the basis of experience.

Available Operating Systems

Commercial character recognition systems have grown in number and sophistication. Some of the early systems have been described by Fischer *et al.* (18), Leimer (32), and Rabinow (40). Currently available systems have been described and categorized by many

authors (2, 8, 23, 27, 28, 41, 43, 52).

Systems are being designed to accept an ever widening set of input patterns. Andrews (3) and Hennis (23) discuss the design of a large omnifont system.

DESIGN OF DATA COLLECTION EQUIPMENT

Certain design criteria made at the initiation of the character recognition project will be reflected here in the design of the reading system. The first design decision deals with the size of the binary array used to represent the character.

Early investigators in the field used very large bit arrays for the character images. At that time no one had attempted to define a minimum array size. These large arrays were probably picked more for their redundancy than their adequacy. The large arrays where usually determined by existing or available equipment. The equipment limitations usually appear in every research effort outside of the commercial area.

It was felt that the amount of redundancy present in a large array was not needed for a successful recognition scheme and that the large bit arrays would represent an unduly heavy processing load to the system. Also it was decided to accept the higher quantization noise of a smaller array.

Thus it was decided to use a 32 bit high by 24 bit wide array as the character image. This size would represent a reasonable compromise between the high quantization noise associated with a very small array and between the large amount of processing associated with a large array. Also this size could be reduced at a later date much more easily than it could be increased.

One investigator has recently suggested that a 20 x 20 bit array represents the minimum and a 30 x 30 bit array the optimum for hand-printed character recognition (37).

With regard to the input document design it was decided to provide a simple form of common size which placed a minimum of constraint on the user. The writing instrument used to prepare the input document was to be of a common type.

Page Reader

As the first step in this investigation it was necessary to design and construct a page reader. While there was an obvious need for the reader as an input device to the recognition processor, there was a more urgent need to provide a source of raw data which could be used in designing and testing the recognition algorithm. The raw character samples that were generally available from other workers were tested and found to have severe quantization noise. Attempts to enlarge the data to approximate the image size required for this investigation indicated extensive preprocessing time would be required with little promise of being able to reduce the quantization noise. Munson et al. (37) have recently tested this data and their results illustrate the severe limitations that this data would place on any recognition system.

A page reader usually contains a very elegant system for picking a sheet of paper from a stack of papers and moving it past a reading station. Such a system is needed to make efficient use of a commercial page reader, but a significant portion of system cost goes into this paper handling capability. In a research oriented system there is no need for this magnitude of paper handling equipment.

The design for the paper handling and sensing portions of the page reader are shown in Figure 1. Because the reader is research rather

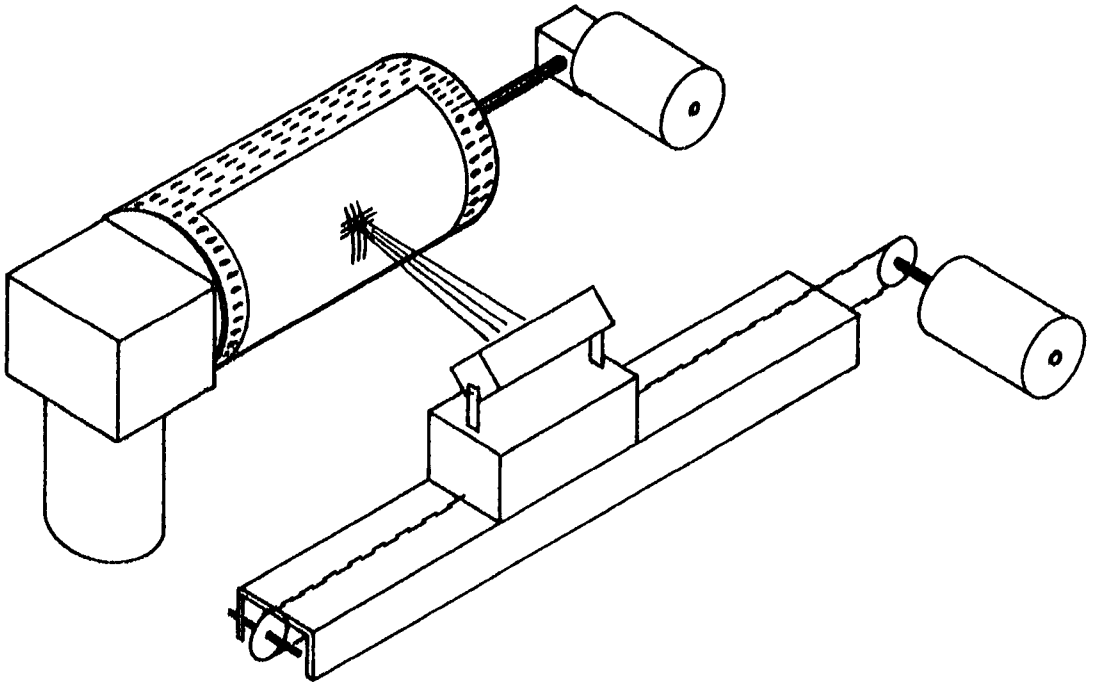


Figure 1. Page reader apparatus for scanning input documents

than production oriented, input documents are hand loaded. The input document is placed on a perforated drum and held in position by air pulled through holes in the surface of the drum. Beside the drum is the sensing head containing a light source, the optical system, and a strip of 32 silicon readout cells. The head is mounted on a track parallel to the drum and moves on that track under the control of a stepping motor.

Once the page to be read is in place on the drum, the reading head is placed in position at the top of the page. The drum then begins turning and a line of handprinted characters is viewed in succession by the optical system and projected on the strip of 32 readout cells. As the character image moves across the cells the reader control circuits gate the outputs of the 32 cell amplifiers into a storage buffer. This is done 24 times across the width of the character to provide a binary image of 32 x 24 bits.

Having completed the scanning of a full line of characters the reader senses the end of a line and causes the reading head to be stepped into position for reading the next line. The head movement takes place during the time the scanning system is viewing the backside of the drum. This positioning is complete by the time the drum is finishing its revolution of travel and the input document is again coming into view. The scanning operation continues down the page and stops when the control circuits sense end of line and last line to signal end of page.

As each 32 bit slice of the character is placed in the buffer it will then be transferred to the processor. In order to obtain data before the processor was completed, it was necessary to interface the reader to a

data recording device. A seven-track incremental magnetic tape unit was available and was selected for recording the character arrays.

A data selection matrix was added to the reader to transfer each 32 bit word to the tape unit four bits at a time. Even though the tape unit was operated in its high-speed write mode, it did not have a high enough data rate. This necessitated the recording of the character arrays in two passes. On the first revolution the upper half of each character was recorded and on the second revolution the bottom half.

Recording the characters in two separate passes introduced some noise on a few characters. This was due to the difference in synchronization times for the reader to become slaved to the recorder for each new line. The noise appeared as a shear noise and was evidenced by a one or two bit shift of the top half of the array with respect to the bottom half.

Input Document

The upper portion of the input document is shown in Figure 2. This is the prototype for a coding form to be used in a possible application of this project. The complete form is 8½ inches wide by 11 inches long.

The black vertical marks at the top of the page are timing marks used to signal valid reading areas. The bit amplifiers are gated on as the control circuits sense a black-to-white transition at the timing track. Reading then takes place until the control circuits sense a white-to-black transition indicating the end of a reading area. The writing space provided is 0.2 in. square. It was felt that this was sufficiently large for the prototype and that a detailed human factors study could be made at a later date to select the appropriate size.

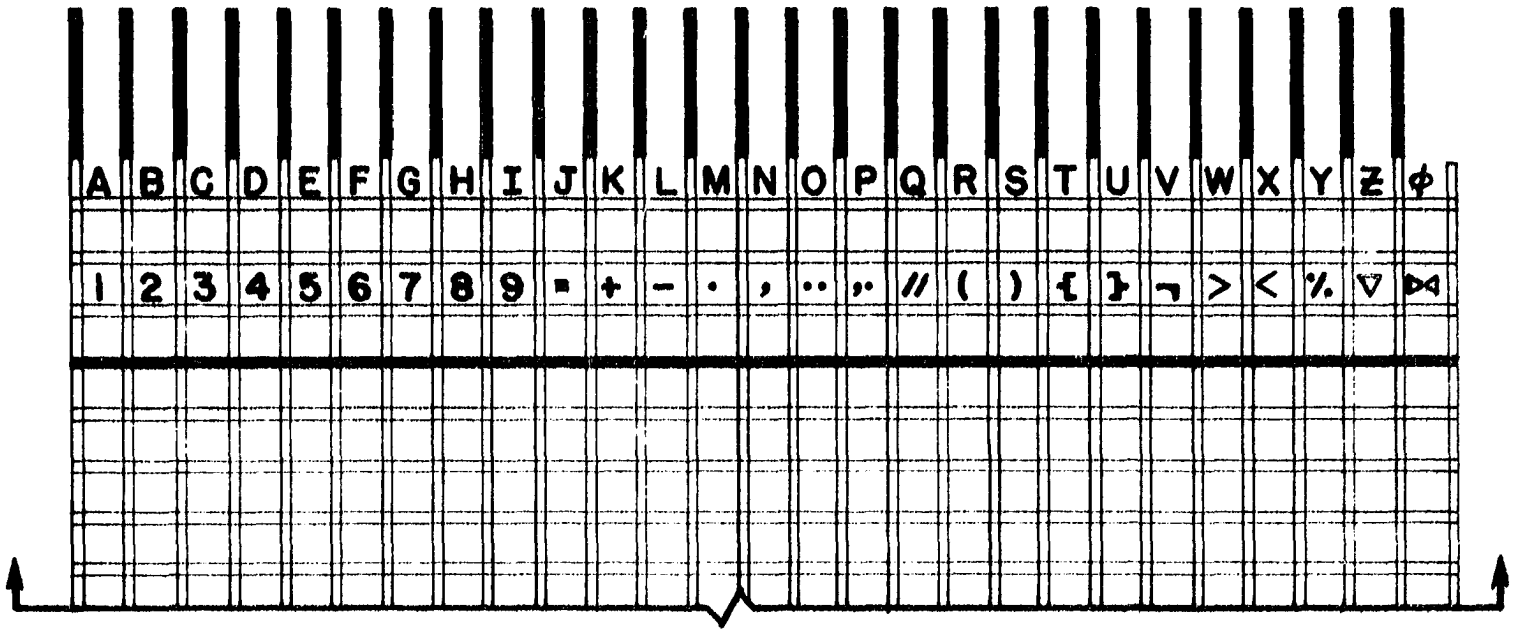


Figure 2. Input document used for data collection

The blank areas or guard zones surrounding the character spaces represent a modest restriction on the user and are necessary for a simple reader of this type. Larger commercial systems are capable of conducting a preliminary search scan to find the beginning of each line.

In addition to defining the writing space on the input document, the guard zones are useful in separating the characters. On systems which use no document control in the form of guard zones the system must be designed to pull apart contiguous characters.

The character set shown at the top of the form is also a prototype set from which a final set could be taken. Obviously the full alphameric set is required, it is the symbol set that remains undefined for the moment. For the purpose of this investigation only the alphameric set was considered.

In the blank spaces below each line of the character set in Figure 2 the user is to copy the character set. The blank areas remaining on the form would be used as an ordinary coding sheet.

Data Format

Samples of handprinted characters were collected using the previously described coding sheet and a very soft lead pencil. While the original design proposed a No. 2 lead, a preliminary test run showed a large variation in print density due to the amount of pressure applied to the writing instrument. The softer lead produced a very uniform print density, almost independent of applied pressure.

Sample character sets were collected from 25 people. Each person was asked to print two copies of the character set. Thus each person

printed 104 characters for a total of 2,600 characters. The first sample set from a user is to be used to provide the information for recognizing that users second set.

Those giving the samples were given the following list of instructions:

1. Print only in the white character space.
2. Do not use serifs.
3. Follow in general the style characteristics of the given character set. Open top 4's and curved or straight back 9's are permitted.
4. Try to be style consistent. That is, if you use an open top 4 on the first copy of the character set, do so on the second set.

It is possible to argue that the above instructions and the design requirement to seek an untutored population are not compatable. However, the slight amount of training involved here in the style restrictions and in the copying of a given character set does not require that this be considered a tutored population.

RECOGNITION ALGORITHM DESIGN

One major design philosophy strongly influenced the design of the algorithm. The philosophy that the amount of testing to which each character was subjected was to be held to some minimum. In order to accomplish this an attempt would be made to use a general group of tests on all characters that would steer the unknown character to a subgroup of tests designed just for a small group or cluster of characters. This design philosophy is particularly appropriate for a small system where it is always necessary to maximize the value of each operation. This approach lies between the pure sequential approach where a decision is made after each test and the parallel approach where all characters are subjected to a large group of tests before a decision is made.

Method of Approach

The method used in the design of this algorithm can be classified as a modified template matching technique. Template matching as originally defined refers to the technique used in early recognition systems where the unknown character was projected on a series of photographic masks or templates representing the anticipated inputs for which the system was designed. The recognition decision was based on the amount of light transmitted through the templates.

This technique is better understood as area correlation. That is, the unknown is compared or correlated with the resident system templates on an area-by-area basis. If the system has only one stage, the character identification is then made on the best correlation and is termed a best match decision.

For this investigation the templates are the binary array representations of the characters. The area-by-area comparison or correlation is a bit-by-bit comparison. Decisions are based on the total number of bit comparisons made in an array.

Template matching has been used very successfully in commercial systems which are designed to recognize a single type font. This type of system requires well controlled input conditions. These are usually single stage systems where the decision is made on a best match criteria.

Template matching is usually ruled out for handprinted characters because they do not represent well controlled inputs. However, the individualized system does offer considerable degree of control. This fact plus the use of the template only as a partitioning device, not as a single stage recognition scheme, offers the possibility of a successful system.

Structure of the Algorithm

A schematic representation of the recognition algorithm is shown in Figure 3. The unknown character to be classified enters the system from the scanning device as a 32 x 24 bit array. Prior to entering test stage 1 the character is operated on by a group of preprocessing routines. This is basically a group of smoothing techniques used to reduce the noise level of the array. The unknown character leaves this portion of the system and enters test stage 1 retaining its original 32 x 24 bit dimensions.

The purpose of test stage 1 is to subject the unknown to a series of tests and then on the basis of these tests, make a decision as to which

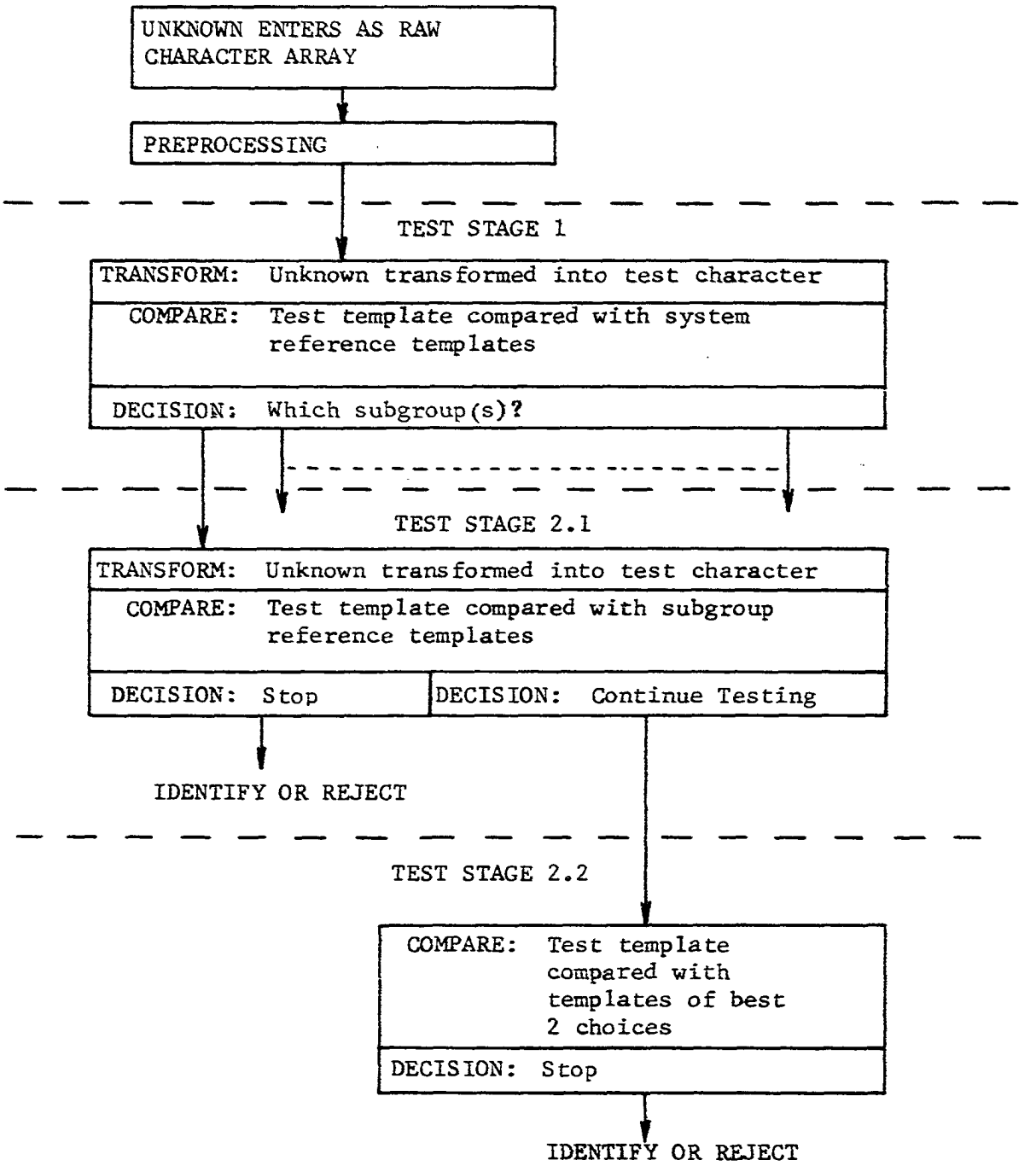


Figure 3. Functional diagram of recognition algorithm

subset or subsets of the character set the unknown belongs. This classifies the unknown with a small set of its nearest neighbors in character recognition space. Stage 1 accomplishes its task by means of a three-step sequence of operations: transformation, comparison, and decision.

Upon entering stage 1 the character is first transformed into a coarse representation on a small array. This becomes the test template. The small array provides a coarse representation of the shape characteristics of the larger array. For the purpose of this investigation both a 16 x 12 and an 8 x 8 bit array have been considered.

Having generated a template for the unknown character, the system now begins the comparison operation. The system has previously generated and stored a set of reference templates from a sample set given by a user. Both sample and unknown have been produced by the same person.

All comparisons for this scheme produce a Comparison Score which can be represented as:

$$CS_{ij} = \sum_{r} W_{ri} C_{rij}$$

Where i is the reference template, j the test or unknown template, and r the array on which they are compared. The weighting function W will take on the values 0 and 1. C is the boolean comparison function:

$$C = A \cdot B + \overline{A} \cdot \overline{B}$$

As used here this function becomes:

$$C_{rij} = (A_i \cdot B_j + \overline{A}_i \cdot \overline{B}_j).$$

The value of CS will range from 0 (no bits compare) to a maximum of 64 (all bits compare on 8 x 8 array) or 192 (all bits compare on 16 x 12

array).

The template representing the unknown is compared bit-by-bit with the set of reference templates. This comparison is performed over the entire template. After each template comparison is completed the system stores the CS value. After all comparisons have been completed a list of 36 CS values has been generated.

The unknown character now enters the decision portion of test stage 1. Here the system uses the comparison scores to determine which subgroups will be tested to identify the unknown. The CS list is searched for its maximum value. Having found CS_{max}, a new search begins for all values within a distance d of CS_{max}. Those comparisons that are greater than or equal to CS_{max}-d and their associated templates determine which subset of templates is to be used to identify the character. For each reference template there is a known subgroup of templates that are to be tested for the final identification.

After the subgroup identification has been made the character enters stage 2.1. Stage 2 will make the final identification of the characters. While stage 2 employs the basic sequence of stage 1; transform, compare, and decide, it has an additional feature. This stage may request additional levels of testing until the comparison scores are such as to permit a decision.

When the character to be identified enters one of the subgroups at test stage 2.1 another transformation and change in array size may take place. Again both 8 x 8 and 16 x 12 bit templates are considered in this design.

Following the transformation is another comparison step. The template representation of the unknown is now compared with a subset of reference templates. While the comparisons are still done on a bit-by-bit basis as in stage 1, this stage does not seek comparison over the entire template as stage 1 did. When the unknown is compared to a reference template at this level, additional templates are superimposed on the comparison. The additional templates determine on which portions of the array comparison will be checked.

The added templates are of two types. The first is called a difference template (DTEMP) and represents the differences between the members of a subset. Its purpose is to restrict comparison to those portions of the templates where differences exist in the reference subset. This template contains both true and false differences. True differences being those unique features which distinguish one character from another. False differences occur because of variations in line width, feature position, and feature size.

The second type of template is a selected area comparison (SAC) template. This template is constructed to reflect the true difference areas that would be expected from a noise free subset of characters. When used to separate P and R, for example, it would restrict comparison to the lower right quadrant of the template.

Stage 2 uses both DTEMP and SAC templates. They may be used separately or in combination. When used with DTEMP, SAC is effective in removing many of the false differences from the difference template.

The decision mechanism for stage 2 varies from subgroup to subgroup.

A subgroup with a small population will initially request only one level of comparison. The comparison values will be searched to find CS_{max} and the CS value closest to CS_{max}. If the distance between the first and second choice exceeds a preset value, the unknown is identified as the character associated with CS_{max}. If the distance does not exceed the preset value, a new level of testing is initiated. A difference template is formed using only the reference templates associated with the two highest CS values. New CS values are determined and a decision is made on a best match or CS_{max} basis.

A subgroup with a larger population will have two designed comparison levels. After the first level of comparison, the scores direct the unknown to a further subdivision of the subset. The unknown now encounters the second level of comparison. Following this a recognition decision is made on the same basis as that described for a small population subset.

The Algorithm in Detail

In the previous section the general operational structure of the algorithm was discussed. This section deals with the detailed operations and design of each section of the algorithm.

Preprocessing

Before a character image moves from the raw data stage into stage 1 of the recognition algorithm it is necessary to operate on the image with a set of preprocessing algorithms. The term "preprocessing" is defined differently among people working in the field of character recognition, but for this investigation it will be used in its most common definition.

It is a series of operations designed to remove noise added to the character during the transformation from paper to binary array and to add some degree of uniformity to the character image.

Noise added to the character during the sensing operation is quantization noise and takes the form of rough irregular character edges, gaps in character segments, and small holes in a character segment. Noise also appears in the image due to dirt and paper imperfections.

The preprocessing routines were chosen to best fit the characteristics of the input data. In addition they could be implemented easily on the recognition processor.

A typical raw character array is shown in Figure 4. The array size is 32 x 32 bits because of additional width added during the recording process. The array is later reduced to 32 x 24 bits.

Delete algorithm The raw character image is first operated on by a delete algorithm due to Ungar (50). This eliminates isolated bits in the character array and removes small projections along edge segments.

The algorithm considers each bit position in the array and its eight nearest neighbors. This nine bit array and the logical statement for the algorithm are shown in Figure 5. The algorithm states that the final value of $E(E_f)$ will be a "1" only if the initial value of $E(E_i)$ is a "1" and certain population requirements are satisfied for its nearest neighbors.

Line thinning Following the delete operation the image is operated on by a line thinning algorithm due to Bomba (5). The 7 x 7 bit operating area and the expression for the algorithm are shown in

```
--------------------------------
--------------------------------
--------------------------------
--------------------------------
--------------------------------
--------------------------------
-------------HHHHHHHH-----------
------------HHHHHHHHHH----------
----------HHHHHHHHHHHH----------
----------HHHH-----HHHH---------
---------HHHH------HHHH---------
--------HHHHH-----HHHHH---------
--------HHHH------HHHH----------
-------HHHH--------HH----------
-------HHHH----------------------
-------HHH-----------------------
------HHHH----------------------
------HHH-----------HHH---------
------HHH---HHHHHHHHHHH---------
-----HHHH---HHHHHHHHHHH---------
-----HHHH---HHHHHHHHHHH---------
-----HHHH------HHHHHHHH---------
------HHH------HHHHHHH----------
------HHHH-----HHHHHHH----------
-------HHH----HHHHHHH----------
-------HHHH----HHHHH-----------
--------HHHHH-HHHHHHH----------
--------HHHHHHHHHHHHH----------
---------HHHHHHHHHHHH----------
----------HHHH--HHH-----------
-------------H----HHH----------
-------------------HHH----------
```

Figure 4. Typical raw character array

A	B	C
D	E	F
G	H	J

$$E_f = E_i \cdot [(A+B+D)\cdot(F+H+J)+(B+C+F)\cdot(D+G+H)]$$

Figure 5. Subarray and logical statement for delete algorithm

Figure 6. The quantities A-H represent the number of bits counted in areas A-H and range in value from 0 to 3. The algorithm says that the final value of bit P (P_f) will be a "1" if the initial value of P(P_i) is a "1" and the difference in the number of bits between areas A and B and between C and D are both less than three or if the difference in the number of bits between areas E and F and between G and H are both less than three.

The thinning rule was modified from the original to fit the data for this investigation. Bomba used an area of 11 x 11 bits to thin his data which varied from four to ten bits wide. His difference criteria for the bit counts was that they be less than four bits.

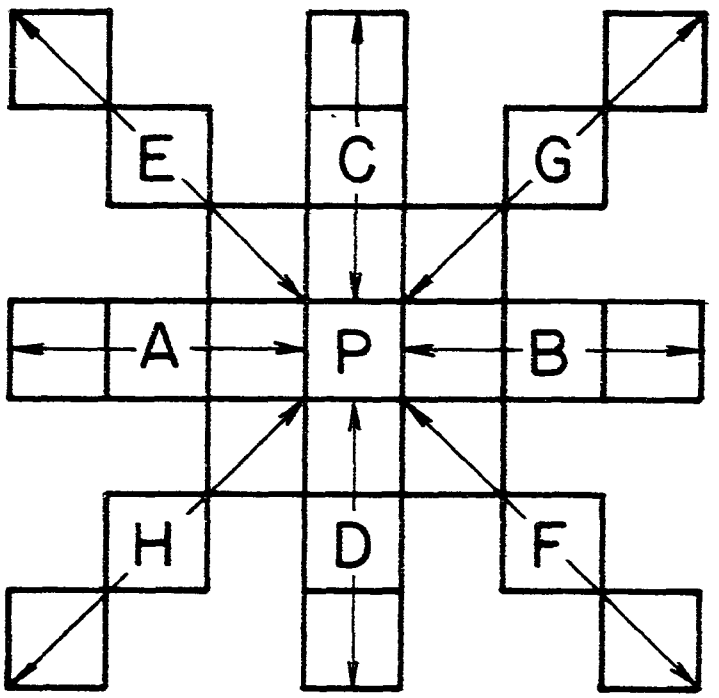
The characters used in this investigation had a line width of two to six bits. The algorithm thinned these to an average of three bits.

The thinning is done to reduce the possibility of two identical characters not comparing correctly due to differences in line thickness.

The raw character shown previously in Figure 4 is shown again in Figure 7 after having been operated on by the delete and line thin algorithms. The 38 x 38 bit array size is due to a border of zeros needed for operating the preprocessing algorithms.

Large noise delete As a preliminary step to the operations that follow, a bit count or census is taken for each row and column in the array. The row and column bit counts are then assembled into a row and column list. These lists are searched to determine the character boundaries.

Some of the image arrays contained large noise areas or blobs due to smudges or paper noise. This type of noise could not be eliminated



$$P_f = P_i \cdot [(|A-B|<3) \cdot (|C-D|<3) + (|E-F|<3) \cdot (|G-H|<3)]$$

Figure 6. Operating area and algorithm expression for line thinning

```
----------------------------------------
----------------------------------------
----------------------------------------
----------------------------------------
----------------------------------------
----------------------------------------
------------------HHHHH-----------------
----------------HHHHHHHHHH--------------
---------------HHHHHHHHHHHH-------------
---------------HHHH-----HHH------------
--------------HHHH------HHH------------
--------------HHHH------HHH------------
-------------HHHH-------HHH------------
-------------HHH---------H--------------
-------------HH-----------------------
------------HHH-----------------------
------------HH------------------------
-----------HHH------------------------
-----------HHH---HHH-----HH------------
-----------HHH---HHHHHHHHHH------------
-----------HHH---HHHHHHHHHH------------
-----------HHH-------HHHHHH------------
-----------HHH-------HHHHH-------------
------------HHH-----HHHHHH-------------
------------HHH------HHHHH-------------
-------------HHH-----HHHH--------------
-------------HHHHH-HHHHHH---------------
-------------HHHHHHHHHHHH---------------
--------------HHHHHHHH-HH--------------
------------------H----HHH-------------
----------------------HHH-------------
----------------------HH--------------
----------------------------------------
----------------------------------------
----------------------------------------
----------------------------------------
----------------------------------------
----------------------------------------
```

Figure 7. A character after the delete, line thin, and centering operations

using a simple delete algorithm.

The bit count lists are searched first to determine the edges of the character. Having established the edges, the search is continued to determine if any bit counts of zero exist within the character boundaries. If zero bit counts are encountered, the character edges are re-established at new boundaries so as to exclude the blob area. The character image within the new edge limits is then defined in a new array and the old image is removed from the system. This routine will remove an area of low population from a large populated area. The routine will not remove a blob that is internal to a character. While the occurrence of an internal blob is a real possibility, it has not occurred in the 2,600 character samples collected for this project.

Figure 8 shows a raw character with a large noise area. Figure 9 shows the same character after the character has been preprocessed and the noise area eliminated.

Magnification As the last step in the preprocessing routines, the character is centered in the array and magnified to approximate the full array size of 32 x 24 bits. Magnification produces a character image with reduced size and position variance.

Most of the character samples did not fill the full array. This routine is designed to stretch the characters until they fill the array. A routine is not needed to reduce the size of the samples because anything larger than the 32 x 24 bit size is cut off during the reading operation. The magnification is done only in the horizontal and vertical directions.

The flow chart for that portion of the routine used to expand the height of the character to approximately 32 bits is shown in Figure 10.

```
--------------------------------
--------------------------------
--------------------------------
-H-------------------------HHHHH
----------------------------HHHHH
----------------------------HHHHH
----------------------------HHHHH
-----------------------------HH-H
--------------------HHH--------H
--------------------HHH---------
-------------------HHHH---------
----------HH-------HHHH-------HH
----------HH------HHHHH---------
----------HHH-----HHHH----------
----------HHH----HHHH-----------
----------HHH--HHHH-------------
----------HH----HHHH-----------
----------HHH--HHHH-------------
----------HHHHHHHH--------------
----------HHHHHHH---------------
----------HHHHHH----------------
---------HHHHHH-----------------
---------HHHHHH-----------------
---------HHHHHH-----------------
---------HHHHHH-----------------
---------HHH-HHH----------------
---------HHH-HHHH---------------
---------HHH--HHHH--------------
---------HHH---HHHHH------------
---------HHH----HHHHH-----------
---------HH------HHHHH----------
-----------------HHHHH----------
```

Figure 8. Raw character array with large noise area

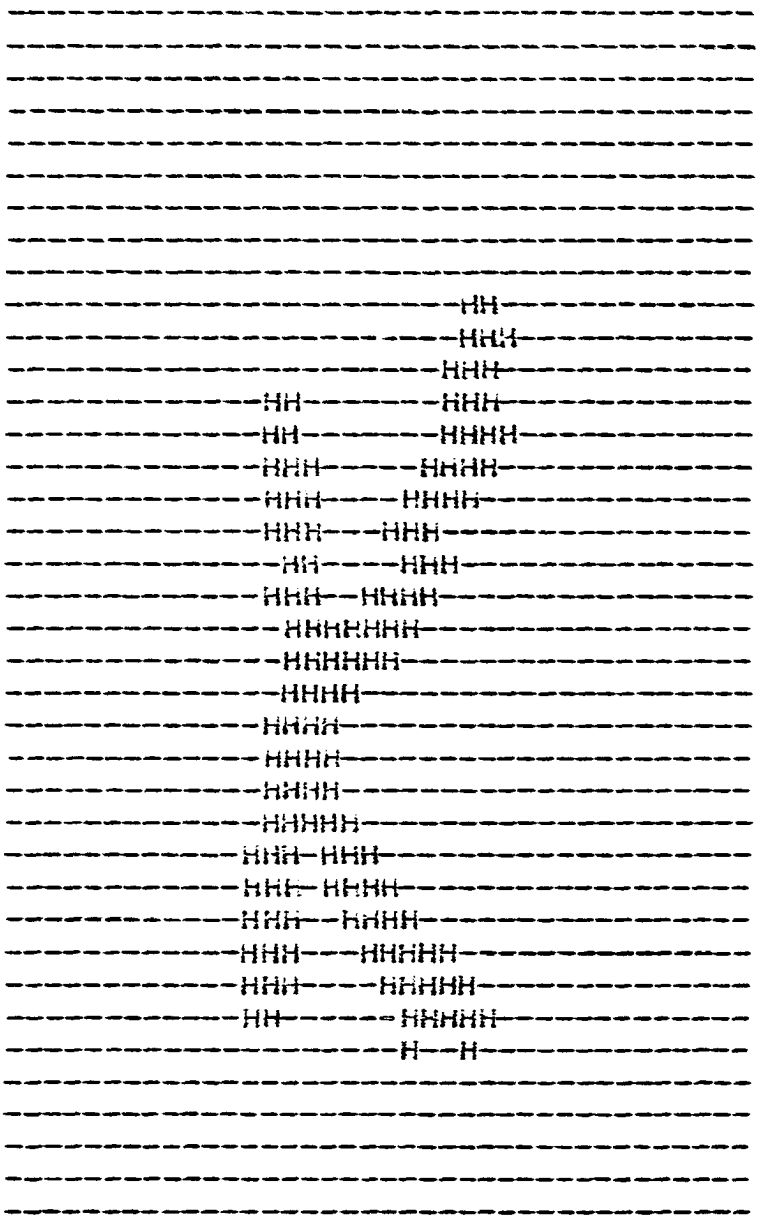


Figure 9. Preprocessed character with noise removed

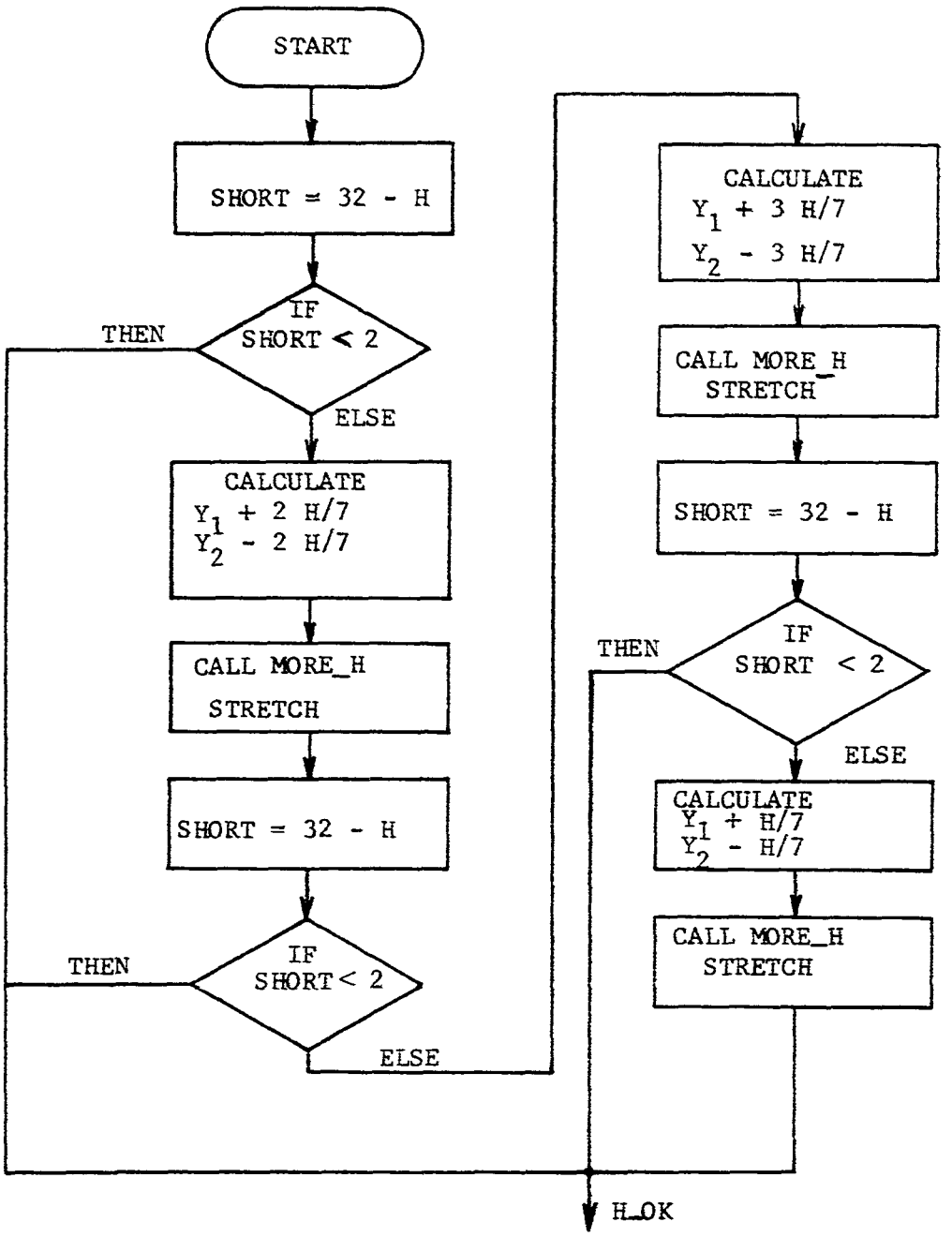


Figure 10. Flow chart for magnification routine. H = height, Y_1 = top edge of character, and Y_2 = bottom edge

A similar routine is used to increase the width of the character to approximately 24 bits.

The routine checks the height (H) of the character and if short, then stretches the character so that the height is increased by two bits. This is done by entering the array at selected rows and shifting the upper portion of the character up one row and the lower portion down one row. This increases the height by producing two copies of the rows at which the shifting began. The first shift moves the upper 2/7 of the character up one row and the lower 2/7 down one row.

At this point the operation stops if the height is correct. Note that if the height is 30 bits when the height check is made, the routine will provide a 2 bit increase to the full 32 bits. If the height is 31 bits when the check is made, no additional increase in the height will take place.

Should the character require additional height, the array is now entered at rows 3/7 from each end of the character and shifted. If the character is still too short, the array is re-entered at rows 1/7 from each end.

Once the size has been increased by 6 bits, the magnification routine ends. The size is checked again at this point and if the character is still too short or too narrow, the magnification process is re-initiated until the character fills the array.

Figure 11 shows the raw character of Figure 4 after magnification.

```
----------------------------------------
----------------------------------------
----------------------------------------
--------------------HHHHHHHHHH----------
-----------------HHHHHHHHHHHHHH---------
------------HHHHHHHHHHHHHHHHHHH---------
------------HHHHHHH---------HHH--------
-----------HHHHHHH----------HHH--------
-----------HHHHHHH----------HHH--------
-----------HHHHHHH----------HHH--------
----------HHHHHHH------------HHH--------
----------HHHHHH--------------H---------
----------HHHHHH--------------H---------
----------HHH---------------------------
---------HHHH----------------------------
---------HHH-----------------------------
--------HHHH----------------------------
--------HHHH----------------------------
--------HHHH-----HHHHH--------HH--------
--------HHHH-----HHHHHHHHHHHHHHH--------
--------HHHH-----HHHHHHHHHHHHHHH--------
--------HHHH-----HHHHHHHHHHHHHHH--------
--------HHHH-----------HHHHHHHHH--------
--------HHHH-----------HHHHHHHH---------
---------HHHH---------HHHHHHHHH---------
---------HHHH---------HHHHHHHHH---------
---------HHHH---------HHHHHHHH----------
----------HHHHHH-------HHHHHHH----------
----------HHHHHHHHH-HHHHHHHHHH----------
----------HHHHHHHHH-HHHHHHHHHH----------
----------HHHHHHHHHHHHHHHHHHHH----------
-----------HHHHHHHHHHHHHHH-HHH----------
------------------H-------HHHH---------
--------------------------HHHH---------
--------------------------HHH----------
----------------------------------------
----------------------------------------
----------------------------------------
```

Figure 11. Magnified character

Stage 1 - Partitioning the character set

It is to be recalled that stage 1 of the recognition algorithm is designed to partition the character set into small subsets. The purpose being to identify the unknown as a member of one or more of these subsets. Referring again to Figure 3, this task is accomplished by transforming the unknown into its template representation, comparing this template with the 35 reference templates, and using the comparison scores in conjunction with a separation distance criteria to establish subgroup membership. The 1 is tested differently from the rest of the characters. It can be recognized using a test for character width. A similar test for both height and width and total bit density would be helpful in recognizing small symbols.

Template generation The first step in stage 1 generates a set of 35 templates representing the alphameric character set. The templates are generated from the sample set provided by the user and will be used to identify that users handprinted characters. As each unknown provided by this user appears as an input to the system it also is transformed into a template.

The transformation applied to the preprocessed 32 x 24 bit array maps the array into both an 8 x 8 and 16 x 12 bit array. The 32 x 24 bit array is first divided into 64 subarrays. These subarrays are 4 bits high by 3 bits wide and as such have populations ranging from 0 to 12 bits. Each subarray is associated with a one bit area in the 8 x 8 array. The 4 x 3 bit array is mapped as a "1" in the 8 x 8 array if the population of the 4 x 3 bit array is at least 2 bits. Populations of 0 and 1 map as a "0".

For the 16 x 12 bit template the large array is divided into 192 2 x 2 bit subarrays. The 2 x 2 bit subarray is mapped into its corresponding bit position in the 16 x 12 array as a "1" if the 2 x 2 subarray contains at least 1 bit. Figure 12 shows examples of the 16 x 12 and 8 x 8 bit arrays.

Transforming the character to a smaller array offers a considerable reduction in processing time. Comparing two 32 x 24 bit arrays would involve 768 comparisons while there would be only 64 for the 8 x 8 array. Also the high information content of the larger array is not needed to group a character with its nearest neighbors in recognition space. Only the coarse shape characteristics of a character need be used at this level of testing.

The 64 bit vector representation of a character was most attractive for its potential speed advantage. Comparing two 64 bit vectors could be done in a few microseconds. The 192 bit (16 x 12) arrays were selected for comparison and for possible use in stage 2 if the 8 x 8 array did not retain sufficient detail for recognition.

The threshold population requirements on the 4 x 3 and 2 x 2 subarrays were determined by several test runs on the sample character sets. These tests will be discussed at the end of the following section in terms of their effects on subgroup population.

<u>Subgroup design</u> Figure 13 shows the character subgroups that have been designed on the basis of this investigation. The first character appearing in each column is the template which produces the subgroup. The results from using the 8 x 8 templates show 34 unique subgroups with an average population of 3.47 members. The 16 x 12 bit

```
----HHHHHHHH
--HHHHHHHHHH
--HHHH----HH
-HHHHH----HH
-HHH-------H
HHH---------
HH----------
HH--HHH----H
HH--HHHHHHHH
HH--HHHHHHHH
HHH----HHHHH
HHH----HHHHH
-HHHHHHHHHH-
-HHHHHHHHHH-
--HHHHHHHHHH
---------HHH
```

```
-HHHHHHH
-HHH---H
HHH----H
HH-HH--H
HH-HHHHH
HH--HHHH
HHHHHHHH
-HHHHHHH
```

Figure 12. Samples of 16 x 12 and 8 x 8 bit templates

8x8 SUBGROUPS

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

16x12 SUBGROUPS

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

Figure 13. Subgroup populations

templates produce 30 unique subgroups with an average population of 2.9 members. Subgroup populations range from 1 to 7 for the 8 x 8 templates and 1 to 6 for the 16 x 12 templates.

In order to discuss the design subgroups it is necessary to first discuss the selection of the distance value d used in conjunction with CS_{max}, the maximum comparison score, to determine subgroup membership. That is, all those nearest neighbors within a distance d of CS_{max} would become members of that particular subgroup.

Comparison operations using the 8 x 8 templates were performed on two sample sets and those characters that fell within a distance of 10 from the character with the highest score were selected as the preliminary subgroup members. These preliminary subgroups were then inspected to determine if a minimum value of d could be selected such that subgroup populations would be reasonably small. This minimum distance was also not to be so small as to prohibit the template that was generating the subgroup from including itself as a member of the group.

It is possible for a character to exclude itself from its own subgroups if the subgroup characters are normally very close in character space and if a value of d has been selected which does not permit sufficient variation between reference and test characters. The characters D and 0 are a good example of this. These characters normally differ only in the upper and lower lefthand corners. If the test D differs modestly from the reference D, it is quite possible to have the D-0 comparison produce a higher CS value than the D-D comparison. This coupled with a very small value of d could eliminate D from its own subgroup.

After inspecting the separation distances on the first two sample sets, a value of 5 was chosen for D as a working value. This value also proved to be appropriate for the tests performed on three additional sets.

In order to determine a minimum separation distance, values of d = 3 and 4 were used to partition the character sets. These values proved unsatisfactory however because several templates would not include themselves as members of their subgroups. Thus for the five sets tested, d = 5 proved to be a minimum value.

The five sample sets were again partitioned using the 16 x 12 templates and a value of d = 15 for the separation distance. The results of these operations indicated a value of d = 10 to be appropriate for this data on the 16 x 12 arrays. Note that while the comparison space of the 16 x 12 array is three times that of the 8 x 8 array, the separation distance has only increased by a factor of two. This is indicative of the tighter subgrouping on the larger array.

It is now appropriate to discuss the selection of a threshold population for the transformations discussed in the previous section. Two sample sets were partitioned using a population requirement of two and three bits. The subgroups were then compared and found to be approximately the same. There was, however, some shifting of characters between subgroups. The nearest neighbor distances for the templates with the three bit requirement tended to be smaller. This is to be expected because this template is not as coarse as the template transformed by using a two bit requirement and as such is not as tolerant of feature size and position noise.

No further attempt was made to optimize the threshold population used in generating the templates. The value of 2 was selected as a working value because it produced larger nearest neighbor distance.

Partitioning the five sample sets provided the basic information needed to design the subgroups. First the subgroup populations were accumulated from the template comparison on the sample sets. For example, the B template subgroup became the total of all the characters selected by B from five sets. It was observed that a few subgroups on individual sample sets were very large and contained unusual populations. These populations were unusual in that they contained characters that normally would have a large separation distance in character space with respect to the template generating the subgroup. Upon checking the templates that had large subgroups it was found that these had been produced by extreme differences between the reference and test templates. Some differences were due to defective templates caused by failures in the scanning operation. The other differences were due to a user providing two radically different samples of the same character. The characters selected by these noisy templates were deleted from the subgroups.

The next step in the design process used a reciprocal membership rule which is perhaps best explained by example. If, for example, the Z template selects the 2 as a member of its subgroup, it is reasonable to expect that the 2 template would possibly select the Z as a member of its subgroup. The similarities between the two characters justify this inclusion even though the results of some of the sample comparisons did not always show the existence of the reciprocal membership.

At times it was necessary to violate this rule. The rule was disregarded in those cases where a template selected a sample because its shape had been degraded in the direction of the reference template and where the reference template could not be degraded reasonably to justify the reciprocity relationship. Note that T has been included in the 7 subgroup for the 16 x 12 templates but that the 7 has not been included in the T subgroups. By rotating the vertical element of the T, the T can be degraded into a 7. The degradation of a 7 into a T does not appear to be a reasonable consideration.

An evaluation of stage 1 as a single-stage recognition system, i.e. template matching with a best match decision, produces the results shown in Figure 14.

Stage 2 - Identifying the character

Stage 2 seeks to identify the character as one of the members of the subgroup or groups to which it has been directed by stage 1. Again the system uses the basic sequence of operations: transformation, comparison, and decision. However, the comparisons are now weighted through the use of additional templates. In the expression for CS:

$$CS_{ij} = \sum_{r} W_{ri} C_{rij},$$

the weighting function W_{ri} takes on the values of 0 and 1. The decision mechanism is also more complex. A subgroup test may request additional testing if the CS values for the first and second choice identification are too close.

Template design Characters within a subset have similar shape characteristics. These similarities are due to the natural similarity

ARRAY SIZE	8 x 8	16 x 12
SET NO.	% CORRECT	% CORRECT
1	82.8	71.4
2	65.7	68.5
3	60.0	68.5
4	80.0	80.0
5	48.6	57.2
AVERAGE	69.4	69.4

Figure 14. Recognition scores using stage 1 as best match recognition system

that exists in the form of the characters, their possible shape degradation during printing, and any changes introduced through transformations applied by the system. The character will now be tested for those detailed characteristics that separate it from its nearest neighbors.

The tests used in this stage seek comparison only on those areas of the templates that represent character differences within the subgroup. The character similarities within the subgroup provide no information with regards to separating the unknown from the other members of the set. On the contrary, inclusion of areas of similarity for the comparison process will often produce errors due to false correlations or lack of correlation. As an example consider a subgroup of F, P, and R. These characters differ from each other only along the upper-half right side and the lower right quadrant. Any differences present on the remainder of the array are due to noise caused by feature position, size, and density variations.

The first template used by the system for a subgroup test is the difference template DTEMP which is constructed from the set of subgroup reference templates. When reference is made to a specific DTEMP it will be represented for example as D^{F}_{PR}, the difference template of F with respect to P and R. The DTEMP array contains those areas where a subgroup member differs from at least one other member of the subgroup. DTEMP is the exclusive OR of the templates constituting a subgroup. In the case of D^{F}_{PR}, the array shows those areas of the reference template F that differ either from the P template or the R template and can be expressed as: $D^{F}_{PR} = F \oplus R \oplus P$.

Figure 15 is the difference template D^F_{PR} that one would expect from a set of well controlled or low noise samples. When used in the comparison of templates, this DTEMP seeks comparison only in the white areas. These areas are the windows through which the unknown is projected on the reference template for comparison.

Figures 16-18 show the FPR subgroup templates and their associated difference templates as 8 x 8 arrays. DTEMP is shown to the right of the three characters from which it is constructed. The "H" areas in DTEMP are the allowed comparison areas. Note the departure from the more idealized template shown in Figure 15. These additional differences in the templates are caused by position and density differences in the features that are common to all three characters; e.g., position differences in the left vertical member of each character or density differences in the same feature.

Note that only one DTEMP appears with each subgroup. If DTEMP is constructed by considering those areas where a character differs from at least one other character in a subgroup, there will be only one DTEMP for the subgroup. For F, P, and R then $D^F_{PR} = D^P_{FR} = D^R_{FP}$.

It is possible to construct DTEMP using other difference criteria. DTEMP could be generated by considering those areas where a character differed from at least two other members of the subgroup. This would have to be applied to subgroups containing more than three members. When applied to subgroup populations of 3, it is possible that not all the difference templates will exist, i.e. they will not contain significant comparison areas.

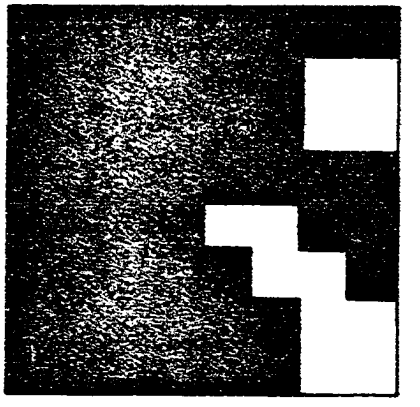


Figure 15. Example of low noise difference template; DTEMP = D^{F}_{PR}

SAMPLE SET 1

F	P	R	DTEMP
HHHHHHHH	HHHHHHH-	HHHHHHH-	-------H
HH------	HH---HHH	H----HHH	-H---HHH
HH------	HH-----H	H-----HH	-H----HH
HHHHHHH-	HHHHHHHH	HH--HHHH	--HH---H
HHHHHHHH	HHHHHHHH	HHHHHHH-	-------H
HH------	HHHHHH--	HH-HHHH-	--HHHHH-
HH------	HH------	H---HHHH	-H--HHHH
HH------	HH------	H-----HH	-H----HH

REFERENCE TEMPLATES

HHHHHHHH	HHHHHHH-	HHHHHHH-	-------H
HH------	H----HHH	H-----HH	-H---HHH
HH------	H------H	H-----HH	-H----HH
HHHH----	HH----HH	HH----HH	--HH--HH
HHHHHHH-	HHHHHHH-	HHHHHHH-	--------
HH------	HH------	HHHHHH--	--HHHH--
HH------	HHH-----	HH--HHH-	--H-HHH-
HH------	HHH-----	HH---HH-	--H--HH-

TEST TEMPLATES

SAMPLE SET 2

F	P	R	DTEMP
HHHHHHHH	-HHHHHHH	-HHHHHHH	H-------
HH------	-HH---HH	HHHH---H	H-HH--HH
H-------	-HH--HHH	HH----HH	HHH--HHH
HH--HH--	HHHHHHHH	HHHHHHHH	--HH--HH
HHHHHH--	HHHHH---	HHHHH---	-----H--
H-------	HH------	H--HH---	-H-HH---
HH------	H-------	H--HHH--	-H-HHH--
HH------	H-------	H---HH--	-H--HH--

REFERENCE TEMPLATES

HHHHHHHH	HHHHHHHH	HHHHHHHH	--------
HHHHH---	HHH---HH	HH-----H	--HHH-HH
HH----H-	HHHH-HHH	HH--HHHH	--HHHH-H
HHHHHHH-	HHHHHH--	-HHHHH--	H-----H-
HHHHH---	HH------	HHH-----	--HHH---
HH------	H-------	HHHH----	-HHH----
H-------	H-------	H-HHH---	--HHH---
H-------	H-------	H--HH---	---HH---

TEST TEMPLATES

Figure 16. FPR (8 x 8) templates and difference templates for sample sets 1 and 2 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SET 3

```
     F          P          R          DTEMP
-HHHHHHH   -HHHHHHH   HHHHHHHH   H-------
-HHHHHH-   -HHHHHHH   HH---HHH   H-HHH--H
-HHH----   HHH---HH   HHHHHHHH   H--HHHHH
-HH--HH-   HHH----H   HHHHHHHH   H--HHHHH
-HHH--H-   -HHHHHHH   HHHHHHH-   H---HH-H
HHHHHHH-   HHHHHHHH   HH--HHH-   --HH---H
HHHHHHH-   HH------   HH---HHH   --HHHHHH
H-------   HH------   HH-----H   -H-----H
REFERENCE TEMPLATES

--HHHHHH   HHHHHHH-   HHHHHHH-   HH-----H
--HHH---   HHHHHHHH   HHHHHHHH   HH---HHH
--HH----   -HHHHHHH   HHH-HHHH   HH-HHHHH
-HH-HH--   -HHHHHHH   HHH-HHHH   H--H--HH
-HHHHH--   -HHHHHHH   HHHHHHHH   H-----HH
HHH-HH--   -HHH----   HHHHHHH-   H--HHHH-
HH------   -HH-----   HHHHHHH-   H-HHHHH-
H-------   HHH-----   HH--HHHH   -HH-HHHH
TEST TEMPLATES
```

SAMPLE SET 4

```
     F          P          R          DTEMP
-HHHHHHH   HHHHHHH-   HHHHHHHH   H------H
HHHHH---   HHH--HHH   HHH---HH   ---HHHHH
HHH-----   HH-----H   HHH-HHHH   --H-HHHH
HHHHHHH-   HH---HHH   HHHHHH--   --HHH-HH
HHHHHHH-   H---HHHH   HHHHHH--   -HHH--HH
HH------   HHHHHHH-   H---HHH-   -HHHHHH-
HH------   HHHH----   HH---HH-   --HH-HH-
HH------   HH------   H-----H-   -H----H-
REFERENCE TEMPLATES

HHHHHHHH   --HHHHHH   HHHHHHH-   HH-----H
HHHHHH--   -HHH---H   HH--HHHH   H-HHHHHH
HH------   -HHH----   HH----HH   H-HH--HH
HHHHHHH-   -HHH--HH   HHH-HHHH   H--HHH-H
HHHHHHHH   -HHHHHHH   HHHHHHH-   H------H
HH-HHHH-   -HH-----   HH-HHH--   H-HHHHH-
HH------   -HH-----   HH--HHH-   H-H-HHH-
HH------   -HH-----   HH---HH-   H-H--HH-
TEST TEMPLATES
```

Figure 17. FPR (8 x 8) templates and difference templates for sample sets 3 and 4 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SET 5

```
F          P          R          DTEMP
HHHHHHHH   -HHHHHHH   HHHHHHH-   H------H
HHHHHH--   HHHHH-HH   HHHHH-HH   -----HHH
HHHHHHH-   HHH---HH   HHH--HHH   ---HHH-H
HHHHHHHH   HHH-HHHH   HHH-HHHH   ---H----
HHH-----   HHHHHHHH   HHHHHHH-   ---HHHHH
HHH-----   HH------   HHHHHHH-   --HHHHH-
HH------   HH------   HHHHHHH-   --HHHHH-
H-------   HH------   HH-HHHH-   -H-HHHH-
REFERENCE TEMPLATES

HHHHHHHH   HHHHHHHH   HHHHHHHH   --------
HHHH----   HHHH--HH   HHHH-HHH   -----HHH
HHH-----   HH----HH   HHH---HH   --H---HH
HHHHHHHH   HHHHHHHH   HH---HHH   --HHH---
HHHHHHHH   HHHHHHH-   HHHHHHH-   -------H
HH------   HHHHH---   HHHHHHH-   --HHHHH-
HH------   HH------   HHHHHHH-   --HHHHH-
HH------   HH------   HH---HH-   -----HH-
TEST TEMPLATES
```

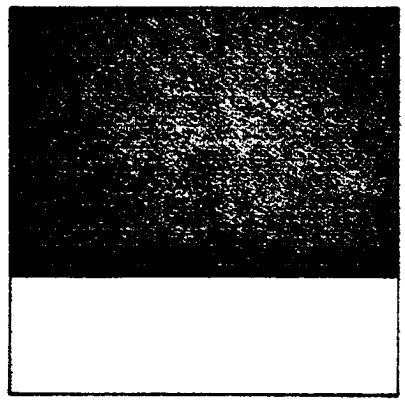
Figure 18. FPR (8 x 8) templates and difference templates for sample set 5 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

Consider again as an example the subgroup F, P, and R. D^F_{PR} is now to be constructed by considering the areas where F differs from both P and R or more generally, where does F differ from all other members of the set. D^F_{PR} can be constructed and will contain only a small area on the upper right edge of the array. If one attempts to generate D^P_{FR}, the array is entirely zero or contains 1's caused by noise. This template cannot be constructed because the character in question, P, is a fundamental component used in the construction of another character in the subgroup, viz. R. Alternatively, P has no differences with respect to both F and R.

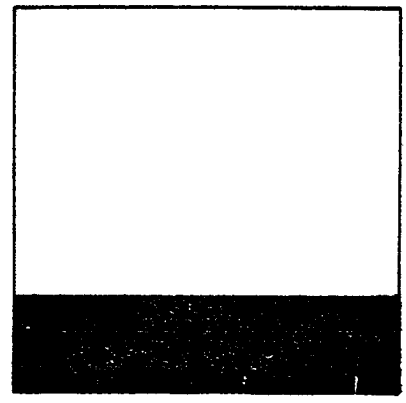
Figures 16-18 indicate the amount of noise that can exist in a difference template due to false differences between characters. In order to eliminate this noise, the selected area comparison template was added to the comparison scheme. Figure 19 shows some of the SAC templates that were tested for this investigation. The white areas are allowed comparison areas. The templates are shown with the characters that they are designed to operated on. The dark portions of these templates represent the areas of similarity for their respective characters; e.g. the SAC template for E and F would prohibit comparison on the upper two-thirds of the array where these two characters have similar shapes. These templates are not offered as the optimum templates for the indicated characters.

No attempt was made to determine the optimum SAC template for any subgroup. This would require exhaustive testing of more than the five data sets. The templates shown in Figure 19 are offered as examples of

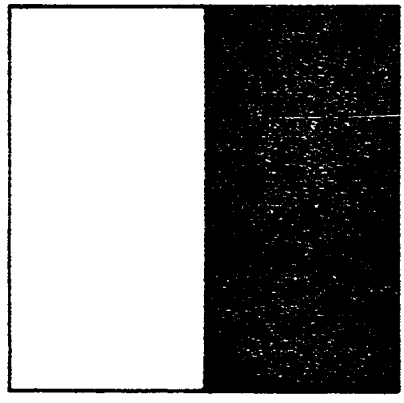
Figure 19. Sample SAC templates. White areas are comparison areas



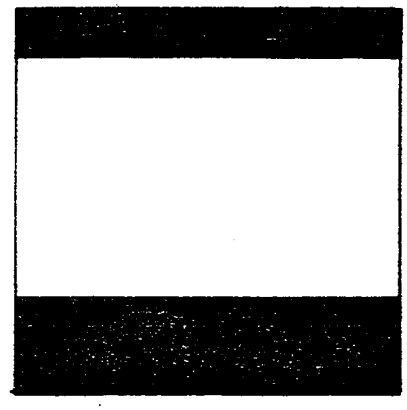
EF



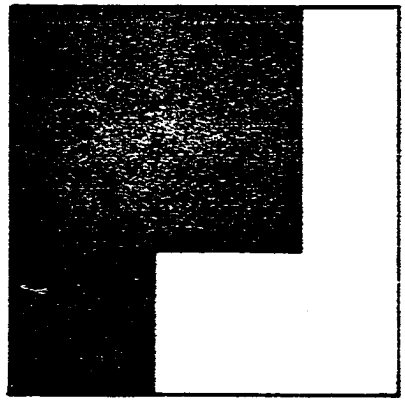
Z2



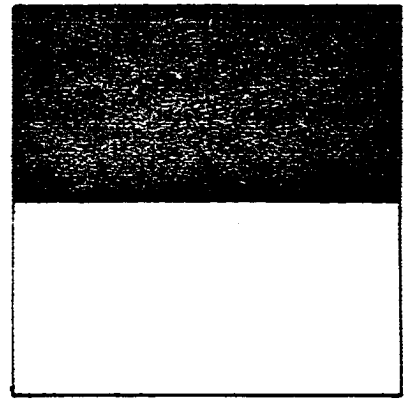
B8



HN



FPR



IJT

those SAC templates which helped to mask out the noise in the difference templates and thus increase the accuracy of the recognition algorithm.

Template tests A functional diagram for a typical subgroup test is shown in Figure 20. This is typical of the type of testing that would be performed within a four member subgroup such as CDGO or BS38. DTEMP1 represents the differences between characters 1 and 2 while DTEMP2 represents the differences between characters 3 and 4. Some of the subgroups may not use a SAC template in stage 2.1.

A four member subgroup would not have the ability to call for additional testing if its first and second recognition choices are too close. If the distance between these two choices was not sufficient, a reject would have to be indicated.

A five member group such as F, N, P, R, and W would have this additional testing capability. Stage 2.1 would seek to partition the subgroup into FPR and NW. If the FPR subdivision has been selected, stage 2.2 will seek to identify the unknown as either F, P, or R. Should the comparison values be too close at this point, the system will request additional testing of the two recognition possibilities. Should these two be P and R, stage 2.3 would construct a difference template for P and R and attempt recognition again. In the case of P and R, the system would also use a SAC template permitting comparison in the lower right quadrant.

The comparison operations performed at this stage of the system have the form shown in Figure 21a. Here the unknown character X has been directed by stage 1 to the FNPRW subgroup for identification. The unknown

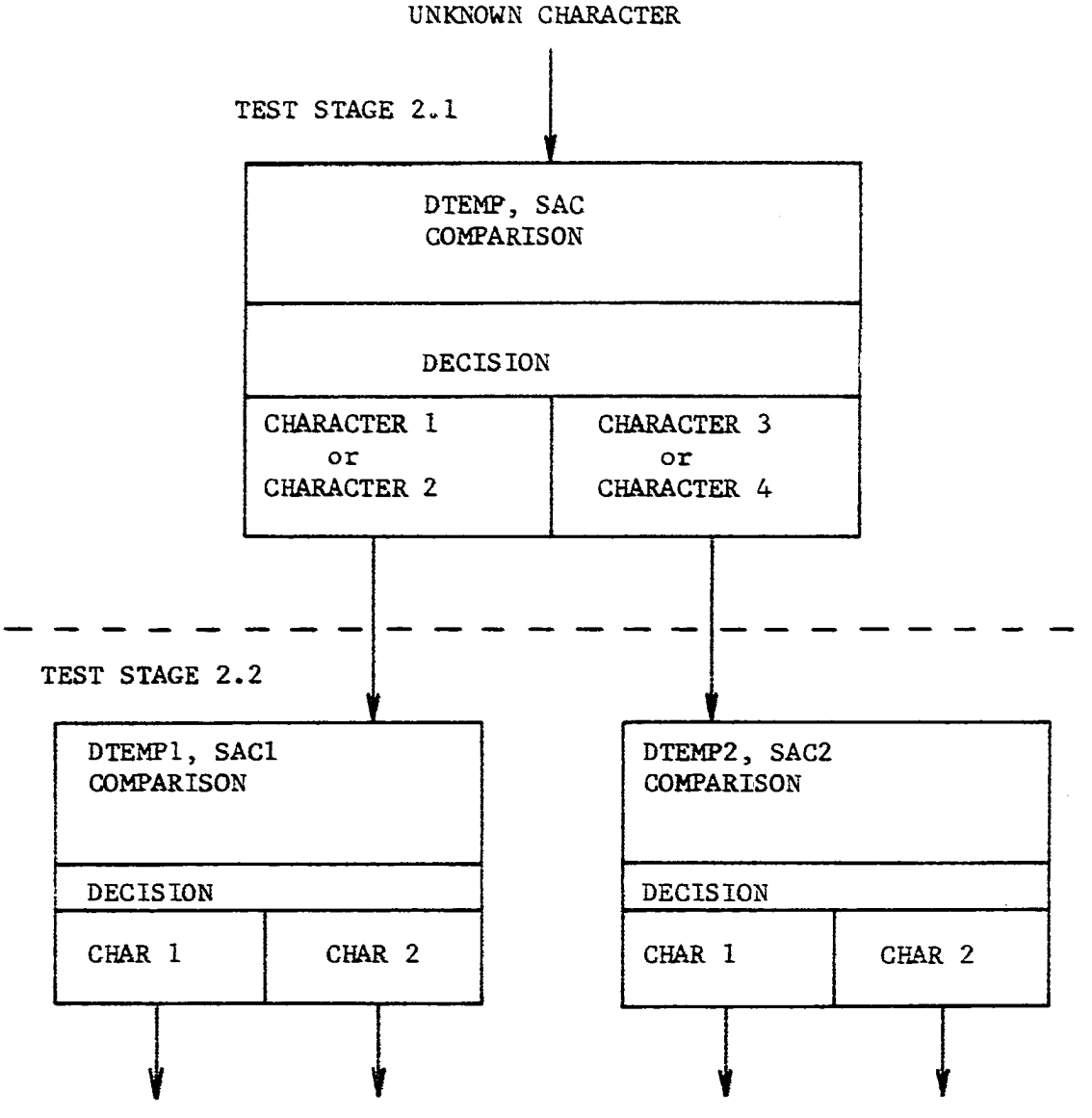


Figure 20. Stage 2 testing procedure

is to be compared to the reference templates F_i - W_i. The bracketed quantities represent the weighted comparison arrays which restrict comparison checking to specific areas, i.e. the comparison windows through which the unknown is projected onto the reference templates for comparison checking.

Figure 21b represents a modification of the approach described above. As the testing moved from three-to-five-member subgroups it was noted that the size of the comparison areas within DTEMP were approaching the full array size. This presented the possibility that the efficiency of a difference template may decrease if it represented the differences within a larger subgroup. The approach shown in Figure 21b limits DTEMP to a subdivision of three templates.

Now the unknown is compared to each reference template twice and the scores from each are summed to obtain the CS value. One additional operation must then be performed before the CS values can be compared. These values must be adjusted to a common base before comparison can take place. This was not necessary in Figure 21a because there the difference templates were identical. They were all generated from the expression:

$$DTEMP = F \oplus N \oplus P \oplus R \oplus W.$$

The divided difference templates were tested on the FNPRW subgroup. This technique did not reduce the number of recognition errors, but it did provide a significant increase in character separation distances.

Some test results are shown in Figure 22 for the subgroup CDGO. The underlined characters are recognition errors. The character and difference templates for these tests are shown in Figures 23-30.

$$X[D^{F}_{NPRW}, SAC]F_i$$

$$X[D^{N}_{FPRW}, SAC]N_i$$

$$X[D^{P}_{FNRW}, SAC]P_i$$

$$X[D^{R}_{FNPW}, SAC]R_i$$

$$X[D^{W}_{FNPR}, SAC]W_i$$

Figure 21a. Comparison operation for unknown X in FNPRW subgroup using one DTEMP

$$X[\frac{D^{F}_{NP}, SAC}{D^{F}_{RW}, SAC}]F_i$$

$$X[\frac{D^{N}_{FP}, SAC}{D^{N}_{RW}, SAC}]N_i$$

$$X[\frac{D^{P}_{FN}, SAC}{D^{P}_{RW}, SAC}]P_i$$

$$X[\frac{D^{R}_{FN}, SAC}{D^{R}_{PW}, SAC}]R_i$$

$$X[\frac{D^{W}_{FN}, SAC}{D^{W}_{PR}, SAC}]W_i$$

Figure 21b. Comparison operation for unknown X in FNPRW subgroup using multiple difference templates

	SET NO.	8 x 8 ARRAY				16 x 12 ARRAY			
		1 DTEMP				10 DTEMP			
		UNKNOWN				UNKNOWN			
		C	D	G	O	C	D	G	O
DECISION	1	C	D	CG	C	C	D	C	D
	2	C	D	G	O	C	D	G	O
	3	C	O	G	O	C	G	G	G
	4	C	D	G	O	C	D	G	D
	5	C	D	G	X	C	D	G	X

Figure 22a. Comparison of CDGO subgroup tests. Underlined characters are recognition errors. X = sample not tested

	SET NO.	8 x 8 ARRAY				16 x 12 ARRAY			
		1 DTEMP 2 DTEMP - 2 SAC				10 DTEMP 2 DTEMP - 2 SAC			
		UNKNOWN				UNKNOWN			
		C	D	G	O	C	D	G	O
DECISION	1	CG	DO	CG	CG	CG	DO	CG	DO
		C	D	C	C	C	D	C	D
	2	CG	DO	CG	DO	CG	DO	CG	DO
		C	D	G	O	C	D	G	O
	3	CG	DO	CG	DO	CG	CG	CG	CG
		C	O	G	O	C	G	C	G
	4	CG	DO	CG	DO	CG	DO	CG	DO
		C	D	G	O	C	D	G	O
	5	CG	DO	CG	X	CG	DO	CG	X
		C	D	G	X	C	D	G	X

Figure 22b. Comparison of multi-level CDGO subgroup tests. Underlined characters are recognition errors. X = sample not tested

Figure 23. CDGO (8 x 8) templates and difference templates for sample sets 1 and 2 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SET 1

C	D	G	O	DTEMP
-HHHHH--	HHHHHHH-	--HHHHH-	-HHHHHH-	HH----H-
HHH--HH-	HH--HHHH	-HH---HH	HHHH--HH	H-HHHH-H
H-------	HH----HH	HH-----H	HHH---HH	-HH---HH
H-------	HH----HH	H------H	HH----HH	-H----HH
H------H	HH----HH	H---HH--	HH----HH	-H--HHHH
H-----HH	HH----H-	H---HHH-	HH---HHH	-H--HH-H
HHH--HHH	HH-HHHH-	HH----H-	HHHHHHH-	--HHHH-H
-HHHHH--	HHHHH---	-HHHHHH-	-HHHH---	H----HH-

REFERENCE TEMPLATES

HHHHHHHH	HHHHHHH-	-HHHHHH-	-HHHHHH-	H------H
HHH---HH	H----HHH	HHH---HH	HHH--HHH	-HH--H--
HH------	H------H	HH-----H	HH----HH	-H----HH
H-------	HH-----H	H----H--	H-----HH	-H---HHH
H-------	HH-----H	HH-HHHHH	H-----HH	-H-HHHHH
H-------	HH----HH	HH----HH	H-----HH	-H----HH
HHH---HH	HH---HHH	HHHH--HH	HH---HHH	--HH-H--
HHHHHHHH	HHHHHH--	-HHHHHH-	HHHHHHH-	H-----HH

TEST TEMPLATES

SAMPLE SET 2

C	D	G	O	DTEMP
--HHHHH-	HHHHHHH-	-HHHHHHH	-HHHHHHH	HH-----H
-HHH--HH	HHH---HH	-HHH---H	-HHH--HH	H--H--H-
HH----HH	HH-----H	HHH----H	HH----HH	--H---H-
HH------	HH-----H	HH-HH--H	HH----HH	---HH-HH
HH------	HH----HH	HH-HHHHH	H-----HH	-H-HHHHH
HH------	HH----HH	HH--HHHH	H-----H-	-H--HHHH
HHH---HH	HH--HHH-	HHHHHHHH	HH--HHH-	--HHHH-H
-HHHHHH-	HHHHH---	-HHHHHHH	HHHHH---	H----HHH

REFERENCE TEMPLATES

--HHHHHH	HHHHHHH-	---HHHHH	---HHHH-	HHH----H
-HHH--HH	HHH---HH	-HHHHH-H	--HHHHH-	HH-HHHHH
HHH---HH	HHH----H	HHHH----	-HHH--HH	H--H--HH
HH------	HH-----H	HH--HHHH	-HH----H	H-H-HHHH
HH------	HH-----H	HH--HHHH	HH----HH	----HHHH
HH------	H------H	HH--HHHH	H-----H-	-H--HHHH
HH------	HH---HHH	HHHHHH-H	HH--HHH-	--HHHHHH
HHHHHH--	HHH-HH--	-------H	HHHHH---	HHHHHH-H

TEST TEMPLATES

Figure 24. CDGO (8 x 8) templates and difference templates for sample sets 3 and 4 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SET 3

C	D	G	O	DTEMP
--HHHHHH	HHHHHHHH	--HHHHHH	--HHHHH-	HH-----H
-HHHHH-H	--HH--HH	HHHHHH-H	-HHHHHH-	HH--HHHH
-HHH---H	--HH---H	HHH-----	HHH--HHH	HH-H-HHH
HHH----H	--HH---H	HH-HHHH-	HH----HH	HHHHHHHH
HH------	--H----H	HH--HHHH	HH-----H	HHH-HHHH
HH------	-HH---HH	HH--HHHH	HH----HH	H-H-HHHH
HHHH----	-HH--HHH	HHHHHHH-	HH--HHHH	H-HHHHHH
-HHHHHH-	-HHHHHH-	HHHHHHH-	HHHHHHH-	H-------

REFERENCE TEMPLATES

---HHHHH	HHHHH-H-	-HHHHHH-	HHHHHHH-	HHH--H-H
HHHHHH-H	HHHHHHHH	HHHHHHH-	HHHHHHH-	------HH
HHH----H	HH---HHH	HHHH----	HHHHH-HH	--HHHHHH
HH------	HH----HH	HHHHHHH-	HH----HH	--HHHHHH
HH------	HH---HHH	HHHHHHH-	HH---HHH	--HHHHHH
HH------	HH-HHHH-	HHHHHHHH	HH--HHH-	--HHHHHH
HHHH----	HHHHHH--	HHHHHHHH	HHHHHHH-	----HHHH
HHHHHHHH	HHH-----	HHHHHHH-	HHHHHH--	---HHHHH

TEST TEMPLATES

SAMPLE SET 4

C	D	G	O	DTEMP
-HHHHHH-	HHHHHH--	-HHHHHH-	-HHHHHHH	H-----HH
HHH--HHH	HH--HHH-	HHH-----	-HHH-HHH	H-HHHHHH
HH-----H	HH---HHH	HHH-H---	-HH----H	H-H-HHHH
HH------	HH-----H	H--HHHH-	HH----HH	-H-HHHHH
H-------	HH-----H	H--HHHHH	H-----HH	-H-HHHHH
HH------	HH----HH	HHH-H-HH	H-----HH	-HH-H-HH
HHH----H	HH--HHH-	HHH---HH	HHH--HH-	--H-HHHH
-HHHHHHH	HHHHHH--	-HHHHHHH	HHHHHH--	H-----HH

REFERENCE TEMPLATES

HHHHHHHH	HHHHHH--	-HHHH---	-HHHHHH-	H----HHH
HHHHH--H	HHHHHHH-	HHHH----	HHHHHHH-	----HHHH
HHH-----	HH---HH-	HHHHH---	HHH---HH	--HHHHHH
H-------	HH----HH	H--HHHH-	H-----H	-H-HHHHH
H-------	HH----HH	H--HHHHH	H-----HH	-H-HHHHH
HH------	H----HHH	HH-HH-HH	H-----HH	-H-HHHHH
HHHHH---	HHHHHHH-	HHHHHHHH	HHH-HHHH	---H-HHH
--HHHHHH	HHHHHH--	-HHHHHH-	HHHHHHH-	HH----HH

TEST TEMPLATES

Figure 25. CDGO (8 x 8) templates and difference templates for sample sets 5 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SETS

C	D	G	O	DTEMP
HHHHHHHH	HHHHHHHH	-HHHHH--	-HHHHHHH	H-----HH
HHHHHHHH	HHHHHHHH	-HHH-H--	HHHHHHHH	H---H-HH
HHH-----	HH---HHH	HHH-----	HHH---HH	--H--HHH
HHH-----	HH-----H	HH------	HH-----H	--H----H
H-------	HH-----H	HH---HHH	H------H	-H---HHH
H-------	HH-----H	HH--HHHH	HH-----H	-H--HHHH
HHH---HH	HH--HHHH	HHH--HHH	HHH--HHH	--H-HH--
HHHHHHH-	HHHHHHH-	-HHHHHHH	HHHHHHHH	H------H

REFERENCE TEMPLATES

--HHHHHH	HHHHHHHH	---HHHHH	-HHHHHH-	HHH----H
HHHHHHHH	HHHH-HHH	HHHHHHH-	HHHH--HH	----HH-H
HHH-----	HHH---HH	HHHHH---	HH----HH	--HHH-HH
HHH-----	HHH----H	HHH-----	H-----HH	-HH---HH
H-------	HH----HH	HH---HHH	HHHHHHHH	-HHHHHHH
HHH-----	HH----HH	HH--HHHH	-HHHHHH-	H-HHHHHH
HHH-----	HH-HHHH-	HHH-HHHH	--HHH---	HHHHHHHH
HHHHHHHH	HHHHH---	HHHHHHHH	--HHHH--	HH---HHH

TEST TEMPLATES

Figure 26. CDGO (16 x 12) templates and difference templates for sample set 1 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

REFERENCE TEMPLATES

```
---HHHHHHH---   HHHHHHHHHH---   ---HHHHHHHH--   ----HHHHHHHH-   HHHH-----HH-
-HHHHHHHH---    HHHHHHHHHHH--   --HHHH-HHHH-    --HHHHHHHHHH-   HH----H--HH-
-HHH---HHH--    HH----HHHHH-    --HH-----HH-    -HHHHH---HH-    HHHHHHHHH-H-
HHHH---HHH--    HH------HHH-    -HHH------HH    HHHH-----HH-    H-HH---HHHHH
HH----------    HH--------HH    -H--------HH    HHHH-----HHH    H-HH-----HHH
HH----------    HH--------HH    HH--------HH    HH-------HHH    ---------HHH
HH----------    HH--------HH    HH--------HH    H--------HHH    -H-------HHH
HH----------    HH--------HH    HH----------    HH--------HH    ----------HH
HH----------    HH--------HH    HH----------    HH-------HHH    ---------HHH
HH--------HH    HH--------H-    HH----HHH---    HH-------HHH    ------HHHHHH
HH--------HH    HH-------HH-    HH---HHHHHH-    HH-------HHH    -----HHHHH-H
HH--------HH    HH------HHH-    HH------HHH-    HHH----HHHH-    --H----HHH-H
HHHH----HHH-    HH----HHHHH-    HH-------HH-    HHHH--HHHHH-    --HH--HHH---
-HHHH--HHHH-    HH--HHHHHH--    -HHH-----HH-    HHHHHHHHHH--    H-HHHHHHH-H-
-HHHHHHHH---    HHHHHHH-----    -HHH---HHHH-    --HHHHH-----    HH--HHHHHHH-
--HHHHH-----    HHHH--------    -HHHHHHHH---    ----HH------    HHHHHHHHH---
```

DTEMP

TEST TEMPLATES

```
-HHHHHHHHH--    HHHHHHH-----    --HHHHHHHH--    ---HHHHHH---    HHH----HHH--
-HHHHHH-HHH-    HH--HHHHHHH-    --HHHH-HHHH-    -HHHHHHHHH--    HHHH--HH--H-
-HHH------HH    H------HHHH-    -HHH-----HHH    -HHHH--HHHH-    HHHHH--HHH-H
-HH-------HH    HH-------HHH    -HHH------HH    -HHH-----HH-    H-HH-----H-H
HHH---------    H---------HH    HH--------H-    HHH-------HH    -HH-------HH
HH----------    H---------HH    HH----------    HHH-------HH    -HH-------HH
HH----------    HH--------HH    HH----------    HH--------HH    ----------HH
HH----------    HH--------HH    HH-----HH---    HH--------HH    -------HH-HH
HH----------    HH---------H    HH--HHHHHHHH    HH--------HH    ----HHHHHHHH
HH----------    HH--------HH    HH--HHHHHHHH    HH--------H-    ----HHHHHHHH
HH----------    HH--------HH    HH--------HH    HH--------H-    ----------HH
HH----------    HH-------HHH    HHH-------HH    HH-------HH-    --H------HHH
HHHH------HH    HH-------HHH    -HHH-----HHH    HH------HHH-    H-HH----HH-H
-HHH----HHHH    HHH----HHHH-    -HHHHH--HHHH    HHH----HHHH-    H--HHH-H---H
-HHHHHHHHHHH    HHHHHHHHHH--    -HHHHHHHHHHH-    HHHHHHHHHHH-    H---------HH
-HHHHHHHHHH-    HHHHHHHH----    ---HHHHHH---    HHHHHHHHH---    HHH-----HHH-
```

DTEMP

Figure 27. CDGO (16 x 12) templates and difference templates for sample set 2 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

REFERENCE TEMPLATES

```
----HHHHHH--   -HHHHHHHHH---   ----HHHHHHHH   ---HHHHHHHH-   -HHH-----HHH
----HHHHHHH-   HHHHHHHHHHH-   --HHHHHHHHHH   --HHHH-HHHHH   HHHH--H----H
--HHHH---HHH   -HHH-----HHH   --HHHH----HH   --HHHH---HHH   -H--HH---H--
-HHHH----HHH   HHH-------HH   -HHHHH----HH   -HHH------HH   H--HHH---H--
-HH------HHH   HHH-------HH   -HHH-------H   -HH-------HH   H--H-----HH-
HHH------HHH   HHH-------HH   HHH---------   -HH-------HH   H--------HHH
HHH---------   HHH-------HH   HH----------   -H--------HH   H-H-------HH
HH----------   HHH-------HH   HH--HHH----H   HH--------HH   --H-HHH---HH
HH----------   HH--------HH   HH--HHHHHHHH   H--------HH-   -H--HHHHHHHH
HH----------   HH-------HHH   HH--HHHHHHHH   HH-------HH-   ----HHHHHHHH
HH----------   HH-------HH-   HHH----HHHHH   H--------HH-   -HH----HHHHH
HHH---------   HH-------HH-   HHH----HHHHH   H--------HH-   -HH----HHHHH
HHH--------H   HH-----HHHH-   -HHHHHHHHHH-   HH-----HHHH-   H-HHHHHHHHHH
-HHH-----HHH   HH----HHHHH-   -HHHHHHHHHH-   HHH---HHHH--   H-HHHHHHH-HH
--HHHHHHHHH-   HHHHHHH-----   --HHHHHHHHHH   -HHHHHH-----   HH-----HHHHH
--HHHHH-----   HH----------   ---------HHH   -HHHHH------   HHHHHHH--HHH
```

TEST TEMPLATES

```
------HHHHH-   -HHHHHHHHHH-   -------HHHHH   -----HHHHH--   -HHHHHH---HH
--HHHHHHH-HH   -HHHHHHHHHH-   -----HHHHHHH   ----HHHHHH--   -HHHH----HHH
--HHHHH---HH   -HHH-----HHH   ---HHHHHH-HH   ----HHH-HH--   -HHHHHHHHHHH
--HHH-----HH   -HHH------HH   -HHHHHH----H   ---HHH--HHH-   -HH-HHH-HHHH
--HH------HH   -HHH-------H   -HHHHH------   --HHH----HH-   -H--HH---HHH
-HH-------H-   -HHH-------H   -HHHH-------   --HHH-----HH   -H-HH-----HH
HH----------   HHH--------H   -HH---------   -HHH------HH   H-HH------HH
HH----------   HHH--------H   HHH---HHHHHH   -HH-------HH   H-H---HHHHHH
HH----------   HHH--------H   HH-----HHHHH   HH--------HH   --H----HHHHH
HH----------   HH--------HH   HHH----HHHHH   HH--------H-   --H----HHHHH
HH----------   HH--------HH   HHH----HHHHH   HH-------HH-   --H----HHHHH
HH----------   H---------HH   HHH----HHHHH   HH------HHH-   -HH----HHHHH
HH----------   H--------HH-   HHHHHHHHH-HH   HH-----HHH--   -HHHHHHHHHHH
HH----------   HH-----HHHH-   -HHHHHH---HH   HH----HHH---   H-HHHHHHHHHH
HHH--HHHH---   HHHH---HHH--   ----------HH   -HHHHHHH----   HHHHHHHHHHHH
-HHHHHH-----   HHHHH-------   ----------HH   -HHHHH------   HHHHHHH---HH
```

DTEMP

DTEMP

Figure 28. CDGO (16 x 12) templates and difference templates for sample set 3 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

REFERENCE TEMPLATES

```
-----HHHHHH-   HHHHHHHHHHHH-   ----HHHHHHHH-   ----HHH-----   HHHHH--HHHH-
----HHHHHHHH-   ----HH-HHHH-   --HHHHHHHHHH   --HHHHHHHHHH-   --HH--H----H
----HHHHH-H-   ----HH---HH-   --HHHHHHH-HH   --HHHHHHHHH-   --HH--HHHH-H
--HHHHH---HH   ----HH---HHH   -HHHHH----HH   -HHHHHHHHHH-   -HHH--HHHH-H
--HHHH----HH   ----HH----HH   -HHHH-------   -HHHH---HHH-   -HHH-H--HHHH
-HHH------HH   ----HH----HH   HHHH--------   -HHHH---HHH-   HHHHHH--HHHH
-HHH------H-   ----HH-----H   HHH---------   HHH-------HH   HHHHHH----HH
-HHH--------   ----HH----HH   HH---HHHHHH-   HHH-------HH   HHHHHHHHHHHH
HH----------   ----H-----HH   HH----HHHHHH   HH--------HH   HH--H-HHHHHH
HH----------   --HHH-----HH   HH----HHHHHH   HH--------HH   HHHHH-HHHHHH
HH----------   --HHH----HH-   HH---HHHHHHH   HH--------HH   HHHHHHHHHHHH
HH----------   --HHH----HH-   HH-----HHHHH   HH--------HH   HHHHH--HHHHH
HHHH--------   --HHH--HHHH-   HHHHHHHHHHH-   HHH------HHH   HH-HHHHHHHHH
-HHHHH------   --HHH--HHHH-   HHHHHHHHHHH-   HHH----HHHH-   HH-HHHHHHHH-
--HHHHHHHHH-   --HHHHHHHHH-   HHHHHHHHHHH--   HHHHHHHHHHH-   HH--------H-
-----HHHHHH-   --HHHHHH----   -HHHHHH-----   -HHHHHHHH---   -HHHH--HHHH-
```

TEST TEMPLATES

```
----HHHHHHH-   HHH---------   ----HHHHHHH-   ----HHHHH---   HHH-HHHHHHH-
----HHHHHHHH   HHHHHHHH-HH-   --HHHHHHHHH-   -HHHHHHHHH--   HHHH----H-HH
--HHHHHHH-HH   HHHHHHHHHHH-   --HHHHHHHHH-   -HHHHHHHHH--   HH-------HHH
-HHHHH-----H   HH-----HHHH-   HHHHHH------   HHHHHHHHHHH-   H-HHHHHHHHHH
-HHHH------H   HH------HHH-   HHHHHH------   HHHHHHH--HH-   H-HHHHH-HHHH
HHHH-------H   HH-------HHH   HHHH--------   HHHHHHH---HH   --HHHHH--HHH
HHH---------   HH--------HH   HH----------   HHH-------HH   --H-------HH
HHH---------   HHH------HHH   HHHHHHHHHHH-   HHH------HHH   ---HHHHHHHHH
HHH---------   HHH-----HHHH   HHHHHHHH----   HH------HHH-   --HHHHHHHHHH
H----------   HHH----HHHHH   HHHHHHHHHHH-   HH------HHH-   -HHHHHHHHHHH
H----------   HH---HHHHHH-   HH--HHHHHHHH   HH-----HHHH-   -H--HHHHHHHH
HHH---------   HH---HHHHHH-   HH--HHHHHHHH   HH---HHHHHH-   --H-HHHHHHHH
HHHH--------   HHHHHHHHHH--   HHH----HHHHH   HHH-HHHHHH--   ---HHHHHHHHH
HHHHHH------   HHHHHHHH----   HHHH-HHHHHHH   HHHHHHHHHH--   ----H-HHHHHH
HHHHHHHHHHHH   HHHH--------   HHHHHHHHHHH-   HHHHHHHHH---   ----HHHHHHHH
----HHHHH---   HH----------   -HHHHHHH----   --HHHH------   HHHHHHHHH---
```

DTEMP

```
HHHHH--HHHH-
--HH--H----H
--HH--HHHH-H
-HHH--HHHH-H
-HHH-H--HHHH
HHHHHH--HHHH
HHHHHH----HH
HHHHHHHHHHHH
HH--H-HHHHHH
HHHHH-HHHHHH
HHHHHHHHHHHH
HHHHH--HHHHH
HH-HHHHHHHHH
HH-HHHHHHHH-
HH--------H-
-HHHH--HHHH-
```

DTEMP

```
HHH-HHHHHHH-
HHHH----H-HH
HH-------HHH
H-HHHHHHHHHH
H-HHHHH-HHHH
--HHHHH--HHH
--H-------HH
---HHHHHHHHH
--HHHHHHHHHH
-HHHHHHHHHHH
-H--HHHHHHHH
--H-HHHHHHHH
---HHHHHHHHH
----H-HHHHHH
----HHHHHHHH
HHHHHHHHH---
```

Figure 29. CDGO (16 x 12) templates and difference templates for sample set 4 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

REFERENCE TEMPLATES

```
--HHHHHHHH---   HHHHHHH------   ----HHHHHHHH-   ---HHHHHHHHH-   HHHH--HHHHH-
---HHHHHHHHH-   HHHHHHHHHHH--   -HHHHHHHHHHH-   --HHHHHHHHHHH   HH--------HH
--HHH--HHHH-    HH----HHHH--    -HHHH-------    --HHHH--HHHH    HHHHHHHHHHHH
-HH------HH-    HH-----HHHH-    -HHHH-------    --HHH-----HH    HHHHH--HHHHH
-HH-------H-    HH-----HHHH-    HHHH--HH----    --HHH-----HH    HHHHH-HHHHHH
HH----------    HH-------HHH    HHHH-HHH----    -HHH------HH    H-HH-HHH-HHH
HH----------    HH--------HH    HH---HHHHH--    HHH-------HH    --H--HHHHHHH
HH----------    HH--------HH    HH---HHHHH--    HHH-------HH    --H--HHHHHHH
H----------     HH--------HH    HH--HHHHHHHH    HH--------HH    -H--HHHHHHHH
HH----------    HH--------HH    HH--HHHHHHHH    HH--------HH    ----HHHHHHHH
HH----------    HH-------HHH    HHHH-HH-HHHH    HH--------H-    --HH-HH-HHHH
HH----------    HH-------HH-    HHHH------HH    HH-------HH-    --HH-----HHH
HHH---------    HH-----HHHH-    HHHHH-----HH    HHH-----HHH-    --HHH--HHHHH
HHHHH-----HH    HH----HHHH--    HHHHH-----HH    HHHH---HHHH-    --HHH-HHHHHH
-HHHHHHHHHHH    HHHHHHHHH---    -HHHHHHHHHHH-   -HHHHHHHH---    H--------HHH
----HHHHHHH-    HHHHHHH-----    ---HHHHHHHH-   -HHHHHHHH---    HHHH---HHHH-
```

TEST TEMPLATES

```
--HHHHHHHHHH   HHHHHHH-----   ----HHHH----   ---HHHH-----   HHHH---HHHHH
-HHHHHHHHHHH   HHHHHHHHH---   -HHHHHHH----   -HHHHHHHHH--   H-------HHHH
HHHHHHHH---H   H---HHHHH---   HHHHHH------   HHHHHH-HHHH-   -HHH--HHHHHH
HHHHH-------   HH-----HHHH-   HHHHH-------   HHHHHH-HHHH-   --HHHH-HHHH-
HHHH--------   HH-----HHHH-   HHHH--------   HHHH-----HH-   --HH---HHHH-
HHH---------   HH-------HH-   HHH-HHH-----   HHHH-----HHH   --HHHHH--HHH
H----------    HH--------H-   H---HHHHH---   HH--------HH   -H---HHHHH-HH
H----------    HH--------HH   H---HHHHHHH-   HH--------HH   -H--HHHHHHHH
H----------    HH--------HH   H---HHHHHHHH   H---------HH   -H--HHHHHHHH
H----------    HH-------HHH   H---HHHHHHHH   H---------HH   -H--HHHHHHHH
HHH---------   HH-----HHHH-   HHH-HHH--HHH   HH-------HHH   --H-HHHHHHHH
HHH---------   H------HHHH-   HHH--HH---HH   HH-------HH-   -HH--HHHHHHH
-HHHHH------   HH--HHHHHHH-   HHHHHHHHHHHH   HHHH---HHHHH   H-HHHHHHHHHH
-HHHHHHH----   HH--HHHHHHH-   -HHHHHHHHHHH   HHHH---HHHH-   H-HHHHH-HHHH
--HHHHHHHHHH   HHHHHHHHH---   -HHHHHHHHHHH-   HHHHHHHHHH--   HH-------HHH
-------HHHHH   HHHH--------   ----HHHHH---   HHHHHHHHHH---   HHHHHHHHHHHH
```

DTEMP

Figure 30. CDGO (16 x 12) templates and difference templates for sample set 5 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

REFERENCE TEMPLATES

```
----HHHHHHHH  HHHHHHHHHH--  --HHHHHHH---  ----HHHH----  HHHH----HHHH
-HHHHHHHHHHH  HHHHHHHHHHH-  --HHHHHHH---  -HHHHHHHHHH-  HH-------HHH
HHHHHHHHHHHH  HHHHHHHHHHHH  --HHHH--H---  -HHHHHHHHHH-  HH----HH-HHH
HHHHH-------  HH------HHHH  -HHHH-------  HHHHHHHHHHHH  H-HHHHHHHHHH
HHHHH-------  HH------HHHH  -HHH--------  HHHH-----HHH  H-HHH---HHHH
HHHHH-------  HH--------HH  HHHH--------  HHHH-----HHH  --HHH----HHH
HHHH--------  HH--------HH  HHH---------  HHH-------HH  --HH------HH
HHHH--------  HH--------HH  HHH---------  HHH-------HH  --HH------HH
H----------  HH--------HH  HH----------  H---------HH  -H--------HH
H----------  HH--------HH  HH-----HHHHH  H---------HH  -H-----HHHHH
H----------  HH--------HH  HH-----HHHHH  HHH-------HH  -HH----HHHHH
H----------  HH--------HH  HH-----HHHHH  HHH-------HH  -HH----HHHHH
HHHH-----HH-  HH----HHHHH-  HHHH---HHHHH  HHH------HHH  --HH--HHH--H
HHHH-----HH-  HH------HHH-  HHHH------HH  HHHHH--HHHH-  --HHH--HHH-H
HHHHHHHHHHH-  HHHHHHHHHHH-  -HHHHHH--HHH  HHHHHHHHHHH-  H------HH--H
HHHHHHHHHH--  HHHHHHHHHH--  ----HHHHHHH-  -HHHHHHHHHH-  HHHH------H-
```

TEST TEMPLATES

```
----HHHHHHHH  --HHHHHHHHH-  ----HHHHHHH-  ---HHHHHHHH-  --HH-------H
----HHHHHHHH  HHHHHHHHHHHH  ----HHHHHHH-  --HHHHHHHHH-  HHHH-------H
----HHHHHHHH  HHHHHH-HHHHH  --HHHHHHHHH-  --HHHH---HH-  HHHH--HHH--H
HHHHH----HH-  HHHH------HH  -HHHHHHH----  -HHH-----HHH  H---HHHH-HHH
HHHHH-------  HHHH------HH  -HHHHHHH----  -H--------HH  H-HHHHHH--HH
HHHHH-------  HHHH------HH  HHHHHH------  HH--------HH  --HHHH----HH
HHHH--------  HHHH------HH  HHHHH-------  H---------HH  -HHHH-----HH
HHHH--------  HHHH------HH  HHHHH-------  H--------HH-  -HHHH----HHH
H----------  HH--------HH  HHH----HHHHH  HHHH------HH  -HHH---HHHHH
H----------  HH--------H-  HHH----HHHHH  -HHHHHHHHHH-  HHHHHHHHHHHH
HHHH--------  HH--------H-  HHH---HHHHHH  -HHHHHHHHHH-  H-HHHHHHHHHH
HHHH--------  HH--------H-  HHH---HHHHHH  ---HHHHH----  HHHHHHHHHHHH
HHHHH-------  HH---HHHHHH-  HHHH--HHHHHH  ----HHHH----  HHHHHHHHHHHH
HHHHH-------  HH---HHHHH--  HHHH-----HHH  ----HHH-----  HHHHHHHHHHHH
HHHHHHHH---H  HHHHHHHH----  HHHHHHHHHHHH  ----HHHH----  HHHH----HHHH
HHHHHHHHHHHH  HH----------  HHHHHHHHHHHH  -----HHHH---  HHHHHHHHHHHH
```

DTEMP

DTEMP

Figure 22a compares the two array sizes using one DTEMP and shows the 8 x 8 template to be the better of the two for this data.

Figure 22b shows the results of multi-step testing as per the functional diagram previously discussed as Figure 20. Note that the number of errors has remained the same for the 8 x 8 arrays while there is one less error for the 16 x 12 arrays. Multi-level tests generally produced fewer errors than single level tests.

The persistent failure of the D from set 3 to be recognized can be explained by viewing Figure 24. The difference between the reference and test sample is so great that the D-to-D comparison produced very low CS values. As a consequence the test D always compared more favorably with some other character.

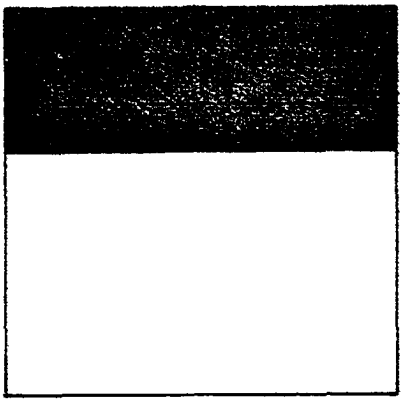
The D and 0 templates shown in Figures 23-30 exhibit the small differences between these two characters. Note that even with these very small differences, it is possible to separate these two characters on some sample sets.

The SAC templates used for the multi-level tests are shown in Figure 31. Those shown in Figure 31a were used in obtaining the results shown in Figure 22b. Additional tests were performed on the 8 x 8 templates using the SAC templates of Figure 31b. SAC templates were used at both levels of testing. These tests produced minor improvements in separation distances, but the improvement was not sufficient to reduce the number of errors.

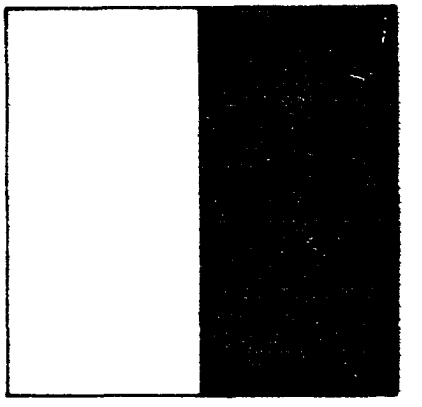
Figure 32 illustrates the improvement possible with the addition of a SAC template to the comparison process. The templates used in these

Figure 31a. First design set of SAC templates for CDGO subgroup. White areas are comparison areas

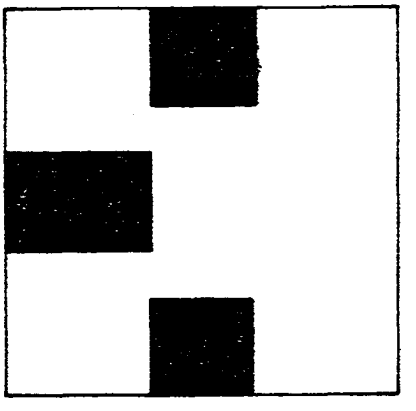
Figure 31b. Second design set of SAC templates for CDGO subgroup



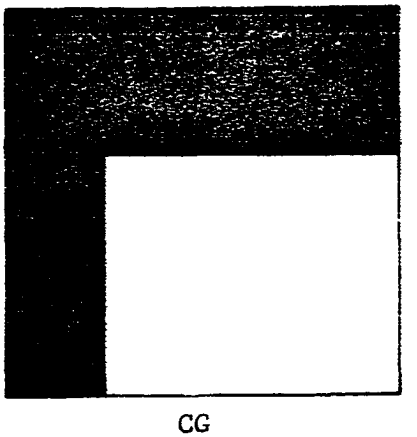
CG



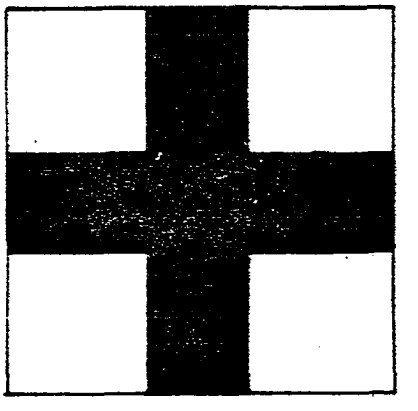
DO



CDGO

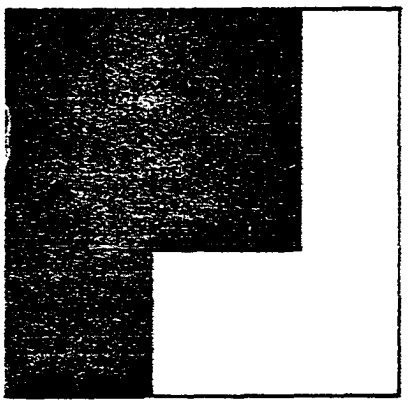


CG



DO

		8 x 8 ARRAY					
	SET NO.	1 DTEMP UNKNOWN			1 DTEMP, SAC UNKNOWN		
		F	P	R	F	P	R
DECISION	1	F	P	R	F	P	R
	2	<u>P</u>	P	R	F	P	R
	3	<u>F</u>	P	R	F	P	R
	4	F	F	R	F	F	R
	5	<u>P</u>	P	R	F	P	R



SAC_{FPR}

Figure 32. Comparison of FPR subgroup tests using DTEMP and DTEMP with SAC template. Underlined characters are recognition errors. Comparison takes place on white area of SAC

tests are those shown previously in Figures 16-18. Comparison was checked on the white area of the SAC template shown in Figure 32.

The recognition of a P as an F in set 4 is not an error because the P has an open loop and thus looks like an F. Figure 17 shows the open loop P of sample set 4.

Figure 33 lists the results of tests performed on the subgroup FNPRW. The templates used for these tests are shown in Figures 34-41. The use of 8 x 8 templates produces one less error than the 16 x 12 templates. Note also that the use of multiple difference templates as previously discussed offers no reduction in the number of errors.

Observe from Figures 35 and 39 that the differences between the reference and test samples for W in sample set 3 are so large as to preclude recognition.

Due to false differences between reference and test templates for R, this character was recognized as a P when testing was performed with 16 x 12 arrays. P and R were then tested again with DTEMP = $D_R^P = D_P^R$. There was still no improvement. A SAC template was added to the tests and the errors were reduced as shown in Figure 33b.

<u>Processing times</u> The algorithm was written in PL/1 and tested on a 360/65. The processing times using 8 x 8 templates at all stages are as shown below:

Preprocessing one character	1.7 seconds
Stage 1-Matching unknown against 52 reference templates	4.3 seconds
Stage 2 testing	1.5 seconds
	7.5 seconds

Figure 33a. Comparison of FNPRW subgroup tests. Underlined characters are recognition errors

Figure 33b. Improvement in PR subdivision tests using SAC template. Underlined characters are recognition errors

		8 x 8 ARRAY									
	SET NO.	1 DTEMP					10 DTEMP				
		UNKNOWN					UNKNOWN				
		F	N	P	R	W	F	N	P	R	W
DECISION	1	F	N	P	R	W	F	N	P	R	W
	2	P	N	P	R	W	P	N	P	R	W
	3	F	N	P	R	N	F	N	PR	R	N
	4	F	N	F	RW	W	F	N	F	RW	W
	5	P	N	P	R	N	P	N	P	R	N

		16 x 12 ARRAY									
	SET NO.	1 DTEMP					10 DTEMP				
		UNKNOWN					UNKNOWN				
		F	N	P	R	W	F	N	P	R	W
DECISION	1	F	N	P	P	W	F	N	P	P	W
	2	F	N	P	P	W	F	N	P	P	W
	3	F	N	P	P	N	F	N	P	P	N
	4	F	N	F	P	W	F	N	F	P	W
	5	P	N	P	R	W	P	N	P	R	W

		16 x 12 ARRAY			
	SET NO.	DTEMP		DTEMP, SAC	
		UNKNOWN		UNKNOWN	
		P	R	P	R
DECISION	1	P	P	P	P
	2	P	P	P	PR
	3	P	P	P	R
	4	P	P	P	R
	5	P	R	P	R

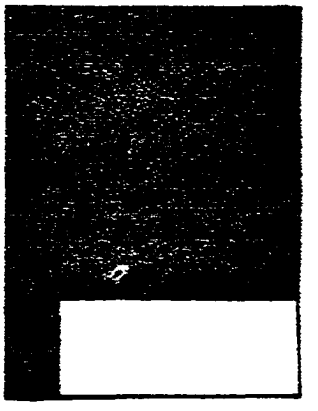


Figure 34. FNPRW (8 x 8) templates and difference templates for sample sets 1 and 2 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SET 1

F	N	P	R	W	DTEMP
HHHHHHHH	------HH	HHHHHHH-	HHHHHHH-	H------H	HHHHHHHH
HH------	HH----HH	HH---HHH	H----HHH	H--H---H	-H-H-HHH
HH------	HHH---HH	HH-----H	H-----HH	HH-HH-HH	-HHHH-HH
HHHHHHH-	HHHH---H	HHHHHHHH	HH--HHHH	HH-HHHHH	--HHHHHH
HHHHHHHH	HH-HHH-H	HHHHHHHH	HHHHHHH-	-H-HHHH-	H-H---HH
HH------	HH--HHHH	HHHHHH--	HH-HHHH-	-HHH--H-	H-HHHHHH
HH------	HH---HHH	HH------	H---HHHH	-HHH--H-	HHHHHHHH
HH------	HH----HH	HH------	H-----HH	-HH---H-	HHH---HH

REFERENCE TEMPLATES

HHHHHHHH	H-----HH	HHHHHHH-	HHHHHHH-	-------H	HHHHHHHH
HH------	HHH---HH	H----HHH	H-----HH	H------H	-HH--HHH
HH------	HHHH--HH	H------H	H-----HH	H--HH--H	-HHHH-HH
HHHH----	HHHHH-HH	HH----HH	HH----HH	HH-HH-HH	--HHH-HH
HHHHHHH-	HH-HHHHH	HHHHHHH-	HHHHHHH-	-HHHHHHH	H-H----H
HH------	HH---HHH	HH------	HHHHHH--	-HHHHHH-	H-HHHHHH
HH------	HH---HHH	HHH-----	HH--HHH-	-HHH-HH-	H-HHHHHH
HH------	HH----HH	HHH-----	HH---HH-	--H---H-	HHH--HHH

TEST TEMPLATES

SAMPLE SET 2

F	N	P	R	W	DTEMP
HHHHHHHH	-HH----H	-HHHHHHH	-HHHHHHH	H------H	HHHHHHH-
HH------	-HHH---H	-HH---HH	HHHH---H	H------H	HHHH--HH
H-------	-HHHH--H	-HH--HHH	HH----HH	H-----HH	HHHHHHHH
HH--HH--	-HHHH-HH	HHHHHHHH	HHHHHHHH	H-HH--HH	HHHHHHHH
HHHHHH--	HH-HH-HH	HHHHH---	HHHHH---	HHHH-HH-	--H-HHHH
H-------	HH--HHH-	HH------	H--HH---	HHHHHH--	-HHHHHH-
HH------	H---HHH-	H-------	H--HHH--	HH-HHH--	-H-HHHH-
HH------	H----HH-	H-------	H---HH--	H--HH---	-H-HHHH-

REFERENCE TEMPLATES

HHHHHHHH	--HH--HH	HHHHHHHH	HHHHHHHH	H------H	HHHHHHH-
HHHHH---	-HHH--HH	HHH---HH	HH-----H	H------H	HHHHH-HH
HH----H-	-HHH--HH	HHHH-HHH	HH--HHHH	HH-H--HH	H-HHHH-H
HHHHHHH-	-HHHH-H-	HHHHHH--	-HHHHH--	HH-H--HH	H-H-HHHH
HHHHH---	-H-HHHH-	HH------	HHH-----	HHHH-HH-	H-HHHHH-
HH------	HH--HHH-	H-------	HHHH----	HHHHHH--	-HHHHHH-
H-------	H---HH--	H-------	H-HHH---	HH--HH--	-HHHHH--
H-------	H----H--	H-------	H--HH---	----H---	H--HHH--

TEST TEMPLATES

Figure 35. FNPRW (8 x 8) templates and difference templates for sample sets 3 and 4 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SET 3

F	N	P	R	W	DTEMP
-HHHHHHH	--HH--HH	-HHHHHHH	HHHHHHHH	H--HH-HH	HHH-HH--
-HHHHHH-	--HH--HH	-HHHHHHH	HH---HHH	H--HHHHH	HHHHHH-H
-HHH----	--HH--HH	HHH---HH	HHHHHHHH	HHHHHHH-	HH-HHHHH
-HH--HH-	-HHHH-H-	HHH----H	HHHHHHHH	HHHHHH--	H--HHHHH
-HHH--H-	HHHHHHH-	-HHHHHHH	HHHHHHH-	HHHHH---	H---HHHH
HHHHHHH-	HH-HHH--	HHHHHHHH	HH--HHH-	HHHHH---	--HH-HHH
HHHHHHH-	H---HH--	HH------	HH---HHH	HHHHH---	-HHHHHHH
H-------	----HH--	HH------	HH-----H	HHHHH---	HHHHHH-H

REFERENCE TEMPLATES

--HHHHHH	--HH----	HHHHHHH-	HHHHHHH-	H--H--HH	HHH-HHHH
--HHH---	--HH--HH	HHHHHHHH	HHHHHHHH	H-HH--HH	HH--HHHH
--HH----	-HHH--HH	-HHHHHHH	HHH-HHHH	HHHH--HH	HH-HHHHH
-HH-HH--	-HHHH-HH	-HHHHHHH	HHH-HHHH	HHHHH-HH	H--H-HHH
-HHHHH--	-HHHHHH-	-HHHHHHH	HHHHHHHH	-HHHH-H-	H----HHH
HHH-HH--	HHHHHHH-	-HHH----	HHHHHHH-	-HH-HHH-	H--HHHH-
HH------	HHH-HH--	-HH-----	HHHHHHH-	-HH-HH--	H-HHHHH-
H-------	H---HH--	HHH-----	HH--HHHH	-HH-HH--	HHH-HHHH

TEST TEMPLATES

SAMPLE SET 4

F	N	P	R	W	DTEMP
-HHHHHHH	HHH----H	HHHHHHH-	HHHHHHHH	H--HH-HH	HHHHHHHH
HHHHH---	HHH---HH	HHH--HHH	HHH---HH	H--HH-HH	-HHHHHHH
HHH-----	-HHH--HH	HH-----H	HHH-HHHH	HH-HH-HH	H-HHHHHH
HHHHHHH-	HHHHH-HH	HH---HHH	HHHHHH--	HHHHHHHH	--HHHHHH
HHHHHHH-	HHHHHHHH	H---HHHH	HHHHHH--	HHH-HHH-	-HHH--HH
HH------	HH-HHHH-	HHHHHHH-	H---HHH-	HH--HH--	-HHHHHH-
HH------	HH--HHH-	HHHH----	HH---HH-	-H--HH--	H-HHHHH-
HH------	------H-	HH------	H-----H-	-H------	HH----H-

REFERENCE TEMPLATES

HHHHHHH	HHH---HH	--HHHHHH	HHHHHHH-	H------H	HHHHHHHH
HHHHHH--	HHH---HH	-HHH---H	HH--HHHH	H--HH--H	HHHHHHHH
-HH------	HHH---HH	-HHH----	HH----HH	H--HH--H	HHHHH-HH
HHHHHHH-	HHHH--HH	-HHH--HH	HHH-HHHH	HHHHHHHH	H--HHH-H
HHHHHHHH	HHHHH-HH	-HHHHHHH	HHHHHHH-	HHHHHHHH	H----H-H
HH-HHHH-	HH-HHHH-	-HH-----	HH-HHH--	-HH--HH-	H-HHHHH-
HH------	HH--HHH-	-HH-----	HH--HHH-	-HH--HH-	H-H-HHH-
HH------	HH--HHH-	-HH-----	HH---HH-	-HH---H-	H-H-HHH-

TEST TEMPLATES

Figure 36. FNPRW (8 x 8) templates and difference templates for sample set 5 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

SAMPLE SET 5

F	N	P	R	W	DTEMP
HHHHHHHH	-HH---HH	-HHHHHHH	HHHHHHH-	HH----HH	H-HHHH-H
HHHHHH--	HHH---HH	HHHHH-HH	HHHHH-HH	HH----HH	--HHHHHH
HHHHHHH-	HHH---HH	HHH---HH	HHH--HHH	HHHH--HH	---HHH-H
HHHHHHHH	HHHHH-HH	HHH-HHHH	HHH-HHHH	HHHHHHHH	---H-H--
HHH-----	HHHHHHHH	HHHHHHHH	HHHHHHH-	HHHHHHH-	---HHHHH
HHH-----	HHHHHHHH	HH------	HHHHHHH-	HHH-HHH-	--HHHHHH
HH------	HH--HHH-	HH------	HHHHHHH-	HHH--HH-	--HHHHH-
H-------	HH---HH-	HH------	HH-HHHH-	-H---HH-	HH-HHHH-

REFERENCE TEMPLATES

HHHHHHHH	------HH	HHHHHHHH	HHHHHHHH	HH-----H	HHHHHHH-
HHHH----	HHHHH--H	HHHH--HH	HHHH-HHH	HH-----H	--HHHHHH
HHH-----	HHHHH-HH	HH----HH	HHH---HH	HH----HH	--HHH-HH
HHHHHHHH	HHHHH-HH	HHHHHHHH	HH---HHH	HHHHH-HH	--HHHH--
HHHHHHHH	HHHHHHHH	HHHHHHH-	HHHHHHH-	HHHHHHHH	-------H
HH------	HHHHHHH-	HHHHH---	HHHHHHH-	HHHHHHHH	--HHHHHH
HH------	HHH-HHH-	HH------	HHHHHHH-	-HHHHHHH	H-HHHHHH
HH-----	HH---HH-	HH------	HH---HH-	-HH---HH	H-H--HHH

TEST TEMPLATES

Figure 37. FNPRW (16 x 12) templates and difference templates for sample set 1 are shown ordered left to right. "H" areas of DTEMP are comparison areas

```
HHHHHH HHHHHH   ---------HHH   HHHHHHHHHHH--   HHHHHHHHHHH--   HH--------HH   HHHHHHHHHHHH
HHH-HHHHHH-H    H--------HHH   HHHHHHHHHHHH-   HHHHHHHHHHHH-   HH--------HH   -HHHHHHHHHHH
HH----------    H--------HHH   HH-----HHHHH-   H------HHHHH-   HH--------HH   -H-----HHHHH
HH----------    HH-------HHH   HH-------HHH    H--------HHH    HH---H----HH   -H---H---HHH
HH----------    HH------HHH    HH--------HH    H---------HH    HH---H----HH   -HH--H---HHH
HH----------    HHHH-----HHH   HH--------HH    H---------HH    HH---HH---HH   -HHH-HH--HHH
HHHHHHHHHHH--   HHHHH-----HH   HH--------HH    H-------HHH-    HH---HH--HHH   -HHHHHHHHHHH
HHHHHHHHHHH-    HHHHHH----HH   HHHHHHHHHHHH    HH----HHHHH-    -HH-HHHHHHHH-   H-HHHHHHHH-H
HHHHHHHHHHH-    HHH-HHHH--HH   HHHHHHHHHHHH-   HHHHHHHHHH--    -HH-HHHHHHHH-   H--H----HHHH
HH----------    HH---HHHH-HH   -HHHHHHHHHHH-   HHHHHHHHH---    -HH-HH-HHHH-   H-HHHHHHHHHH
HH----------    HH----HHHHHH   -HHHHHHHH---    HH--HHHHH---    -HHHHH---H--   H-HHHHHHHHHH
HH----------    HH-----HHHHH   -HH---------    HH---HHHHH--    --HHHH---H--   HHHHHHHHHHHH
HH----------    HH-----HHHHH   -HH---------    H-----HHHHH-    --HHH----H--   HHHHH-HHHHHH
HH----------    HH-------HHH   -HH---------    H-------HHHH    --HHH----H--   HHHHH---HHHH
HH----------    HH-------HH-   -HH---------    H--------HHH    --HHH----HH-   HHHHH----HHH
HH----------    HH--------H-   -HH---------    H----------    ---------H--   HHH------HH-
```

REFERENCE TEMPLATES

DTEMP

```
HHHHHHHHHHHH    HH-------HH-   HHHHHHHHHHHH-   HHHHHHHHH---    ----------H   HHHHHHHHHHHH
HHHHHHHHHHHH    HH-------HH-   HHHHHHHHHHHH-   HHHHHHHHHHH-    ----------HH   HHHHHHHHHH-H
HH----------    HH-------HHH   HH-----HHHHH    H-------HHH-    HH---------H   -H-----HHHHH
HH----------    HHHH-----HHH   H---------HH    HH-------HH-    HH---H----HH   -HHH-H---HHH
HH----------    HHHH-----HHH   H---------HH    HH--------H-    HH---HH---HH   -HHHHHH--HHH
HH----------    HH-HHH---HHH   H---------HH    HH--------HH    HH---HH---HH   -H-HHHH--HHH
HH----------    HH--HHH--HHH   HH-------HHH    HH--------HH    HH---HH---HH   ----HHH--HHH
HHHHHH------    HH--HHHH--HH   HH-------HHH    HHH------HH-    -HH-HHHH--HH   H-HHHHHH-HHH
HHHHHHHHHH--    -H---HHHH-HH   HHHHHHHHHHHH-   HHH----HHHH-    -HHHHHHHH-HH   H-HHHHH--HHH
HHHHHHHHHH--    -H-----HHHHH    HHHHHHHHHHHH-   HHHHHHHHHH--    -HHHHHHHHHH-   H-HHHHH---HH
HH----------    -H-----HHHHH    HHHH--------    HHHHHHHHH---    --HHH--HHHH-   HHHHHHHHHHHH
HH----------    -H-----HHHHH    HH----------    -HH----HH---    --HHH--HHHH-   HHHHH--HHHHH
HH----------    -H------HHHH    HH----------    -HH----HH---    --HHH---HHH-   HHHHH--HHHHH
HH----------    -H-------HHH    HHHH--------    -HH----HHH--    --HHH----HH-   HHHHH--HHHHH
HH----------    -H-------HHH    HHHH--------    HHH-----HHH-    ---HH----HH-   HHHHH---HHHH
HH----------    -H----------    HHHH--------    HHH-----HHH-    ---------H--   HHHH----HHH-
```

TEST TEMPLATES

DTEMP

Figure 38. FNPRW (16 x 12) templates and difference templates for sample set 2 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

```
H-HHHHFFHHHH  --HH------HH  -HHHHHHHHHHH-  ----HHHHHHHH  ----------HH  HHHHHHHHHH-H
HHHHHHHHHHHH  --HHF------HH  -HHHHH---HH-  -HHHHHHHHHHH  HH--------HH  HHHHHHHHHH-H
HHH---------  --HHH-----HH  -HHHH-----HH  -HHHHH----HH  HH--------HH  HHHHHH----HH
HH----------  --HHFH----HH  -HHH------HH  -HH-------HH  HH--------HH  HHHHHH----HH
H----------  --FFHH----HH  -HHH------HH  -HH-------HH  HH--------HH  HHHHHH----HH
H----------  --HHHHH---HH  -HHH---HHHHH  HHH-----HHHH  HH-------HHHH  HHHHHHHHHHHH
HH----------  -HHHHHH---HH  -HHH----HHH-  HHH----HHHH-  HH--H----HH-  H-HHHHHHHHHH
HH----HHHH--  -HHHHHH--HH-  -HHHHHHHH---  HHHHHHHHH---  HH-HHH--HHH-  H-HHHHHHHHH-
HHHHHHHHH---  -HH--HHH-HH-  -HHHHHH-----  HHHHHHH-----  HH-HHH--HH--  H-HHH-HHHHH-
HHHHHHHH----  HH---HFH-HH-  HHHH--------  HHH-HHH-----  HHHHHH-HHH--  --HHHHHHHHH-
H----------  HH----HH-HH-  HHH---------  HH--HHH-----  HHHHHH-HH---  -HHHHHHHHHH-
HH----------  HH----HHHH--  HH----------  H---HHHH----  HHHHHHHHH---  -HHHHHHHHH--
HH----------  HH----HFHH--  HH----------  H----HHH----  HH---HHHH---  -H---HHHHH--
HH----------  HH-----HHH--  HH----------  H-----HHH---  HH---HHH----  -H---HHHHH--
HH----------  HH-----HHH--  HH----------  H-----HHH---  HH---HH-----  -H---HHHHH--
HHH---------  --------HH--  HH----------  -------HH---  HH----H-----  HHH---HHHH--
```

REFERENCE TEMPLATES DTEMP

```
-HFHHHFHFHHH  ----------HH  -HHHHHHHHHH-  -HHHHHHHHHH-  H----------H  HHHHHHHHHHHH
HHHHHHHHHHH-  ---HH-----HH  -HHHH---HHH-  -HHHH-----HH  HH----------H  HHHHHHHHHHHH
HHHHHHHH----  --HHHH----HH  -HHH------HH  -HHH------HH  HH----------H  HHHHHHHH--HH
HH----------  --HHFH----HH  -HHH------HH  -HH-------HH  HH--------HH  HHHHHH----HH
-H----------  --HHHH----H-  -HHH-----HHH  -HH------HHH  HH---H----HH  HHHHHH---HHH
-HH------H--  --HHHHH--HH-  -HHHHH-HHHH-  -HH---HHHHHH  HH--HH----HH  HHHHHHHHHHHH
-H-------H--  --FH-HF--HF-  -HHHHHHHH---  -HHHHHHHH---  HH--HH---HHH  HHHHHHHHHHHH
-HHHHHHHHH--  --HH-HH-HHH-  HHHHHHHHHH---  -HHHHHHH----  HH--HH---HH-  HHHHH-HHHHH-
-HHHH------  -HH--HH-HH--  HHH---------  -HHH--------  HH-HHH---HH-  H-HHHHH-HHH-
-HHHHHHH---  -HH---HHHH--  HHH---------  HHHHH-------  HHHHHHH-HHH-  H--HHHHHHHH-
HH----------  -HH---HHHH--  HH----------  HHHHH-------  -HHHHHHHHH--  H-HHHHHHHH--
HH----------  HH----HHHH--  HH----------  HH-HHH------  -HHHHHHHH---  H-HHHHHHHH--
HH----------  HH-----HH---  H----------  HH--HH------  -HH--HHHH---  HHH-HHHHH---
HH----------  HH-----HH---  H----------  HH--HHHH----  -HH---HH----  HHH-HHHHH---
HH----------  HH-----HHH--  H----------  H----HHH----  ------HH----  HH---HHHHH--
HH----------  HH-----HH---  H----------  H----HHH----  ------HH----  HH---HHHH---
```

TEST TEMPLATES DTEMP

Figure 39. FNPRW (16 x 12) templates and difference templates for sample set 3 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

```
-HHHHHHHHHHH  ----HH----HH  --HHHH------  HHHHHHHHHHHH-  HH--------HH  HHHHHHHHHHHH
-HHHHHHHHHH-  ----HH----HH  -HHHHHHHHHHH  HHH----HHHHH  HH--HHH--HH-  HHHHHHHHHH-H
-HHHHHHHHHH-  ----HH----HH  -HHHHHHHHHHH  HHH-----HHHH  HH--HHH--HH-  HHHHHHHHHH-H
-HHHH-------  ----HH----H-  -HHHHHHHHHHH  HHH-------HH  HH--HHH-HHH-  HHHHHHHHHHHH
-HHHH-------  ----HH----H-  -HHH-----HHH  HHH-----HHHH  HH--HHH-HHH-  HHHHHHH-HHHH
-HHH--------  ---HHH---HH-  -HHH------HH  -HHHHHHHHHHH  HHHHHHHHHHH-  HHH-HHHHHHHH
-HHH--------  -HHHHH---HH-  -HHH------HH  -HHHHH--HHHH  HHHHHHHHH---  H---HHHHHHHH
-HHHH--HHHH-  -HHHHHH--HH-  -HHH------HH  HHHHHHHHHHH-  HHHHHHHHH---  H---HHHHHHHH
-HHHH-----H-  -HHHHHH--HH-  -HHHH---HHHH  -HHHHHHHHHH-  HHHHHHHH----  H----HHHHHHH
-HHHH-----H-  HHHH-HHHHH--  -HHHHHHHHHHH  HHHHHHHHHH--  HHHHHHHH----  H---HHHHHHHH
HHHHHHHHHHH-  HHHH-HHHH---  -HHHHHHHHHH-  HHH---HHHHH-  HHHHHHH-----  H--HHH-HHHH-
HHHHHHHHHHH-  HHH--HHHH---  -HHHHHHHHHH-  HHH---HHHHH-  HHHHHHH-----  H--HHH-HHHH-
HHHHHHHHHHH-  HH----HHH---  HHH---------  HH-----HHHHH  HHH-HHH-----  --HHHHHHHHHH
HH----------  H-----HH----  HHH---------  HH------HHHH  HHH-HHH-----  --H-HHHHHHHH
HH----------  H-----HHH---  HHH---------  HH--------HH  HHH-HHH-----  -HH-HHHHH-HH
H-----------  ------HHH---  HH----------  HH--------HH  HH--HHH-----  HH--HHHHH-HH
```

REFERENCE TEMPLATES DTEMP

```
----HHHHHHHH  ----HH------  HHHH--------  HHHHH-------  ----------HH  HHHHHHHHHHHH
---HHHHHHHHH  ----HH----H-  HHHHHHHHHHHH-  HHHHHHHHHH--  H---HH----HH  HHHH--HHHHHH
----HHHH----  ---HHH----HH  HHHHHHHHHHHH-  HHHHHHHHHHH-  HH--HH----HH  HHHH--HHHHHH
---HHH------  ---HHH----HH  --HHHHHHHHHH  -HHH-HHHHHHH  HH--HH----HH  HHHHH-HHHHHH
---HHH------  ---HHH----H-  --HHH---HHHH  -HH----HHHHH  HH--HH----H-  HHHHHH-HHHHH
---HHH------  --HHHH---HHH  --HHHHHHHHHH  -HHH-----HHH  HH-HHH---HH-  HHH-HHHHHHHH
--HHH--HH---  --HHHHH--HH-  --HHHHHHHHH-  -HH----HHHH-  -HHHHH---HH-  -H-HHHHHHHH-
--HHH--HH---  -HHHHHH-HHH-  --HHHHHHHHH-  -HHH---HHHHH  -HHHHHH---H-  -H--HHHHHHHH
--HHHHHHH---  -HHHHHHHHHH-  --HHHHHHH-H-  -HHHHHHHHHH-  --HHHHHH-HH-  -H------HHH-
-HHHHHHHH---  -HHHHHHHHHH-  --HHHHHHH-H-  -HHHHHHHHHH-  --HHHHHH-HH-  -H------HHH-
HHHHHHHHH---  HHHHHHHHHH--  --HHHH------  -HHHHHHHHHH-  --HHH-HHHHH-  HH---HHHHHH-
HHH---------  HHHH-HHHH---  --HHH-------  -HHHHHHHH---  --HHH-HHHH--  HH-HHHHHHH--
HHH---------  HHHH-HHHH---  --HHH-------  HHHHHHHHHH--  --HHH-HHHH--  HH-HHHHHHH--
HHH---------  HH----HHH---  --HH--------  HHH---HHHHH-  --HH---HH---  HHHH--HHHHH-
HH----------  HH----HHH---  -HHH--------  HHH---HHHHHH  --HH--HHH---  HHHH--HHHHHH
H-----------  HH----HH----  -HHH--------  HH-------HHH  --HH--------  HHHH--HH-HHH
```

TEST TEMPLATES DTEMP

Figure 40. FNPRW (16 x 12) templates and difference templates for sample set 4 are shown ordered from left to right. "H" areas of DTEMP are comparison areas

```
------HHHHH  --H-------HH  ----HHHH----  --HHHHHHHHHH-  -----HH---HH  --HHHHHHHHHH
-HHHHHHHHHH-  -HHH------HH  HHHHHHHHHHHH-  -HHHHHHHHHHH  H----HH---HH  HHHHHHHHHH-H
-HHHHHH-----  -HHH------HH  HHHH---HHHHH  -HHHH-----HH  HH---HH---HH  H-HHHHHHHHHH
-HHH--------  -HHHH-----HH  HHH-----HHHH  -HHH-----HHH  HH---HH---HH  H-HHHHH-HHHH
-HHH--------  -HHHH-----HH  HHH--------HH  -HH-----HHHH  HH---HH--HH-  H-HHHHH-HHHH
-HHH--------  --HHHHH---HH  HHH-------HH  -HHH-HHHHHH-  HH--HHHH-HH-  HHHHHHHHHHHH
HHHHHHHHHHH-  --HHHHH---HH  HHH-------HH  HHHHHHHHHH---  -HH-HHHH-HH-  HH-HHHHHHHHH
HHHHHHHHHHH-  HHHHHHH---HH  HHH-----HHHH  HHHHHHHHH---  HH-HHHH-HH--  --HHHHHHHHHH
HHHHHHHHHH--  HHHHHHH---H-  HH----HHHHHH  HHHHHHH-----  HHHH--HHHH--  --HHHH-HHHHH
HHHHHH------  HHH-HHHHHHH-  H-----HHHHH-  HHHHHHHHH---  HHHH--HHHH--  -HHHHHHHHHH-
HH----------  HH--HHHHHHH--  HHHHHHHHHH--  H----HHHH---  -HH---HHH---  HHHHHHHHHH--
HH----------  HH--HHHHHHH-  HHHHHH------  H-----HHHH--  -HH---HHH---  HHHHHHHHHHH-
HH----------  HH----HHHHH-  HHHHHH------  HH-----HHHH-  -HH----HH---  H-HHHHHHHHH-
HH----------  HH-----HHHH-  HH----------  HH------HHH-  -HH----HH---  H-H----HHHH-
HH----------  --------HHH-  HH----------  HH------HHH-  -HH----H----  HHH----HHHH-
HH----------  ---------HH-  HH----------  ---------HH-  -HH---------  HHH------HH-
```

REFERENCE TEMPLATES

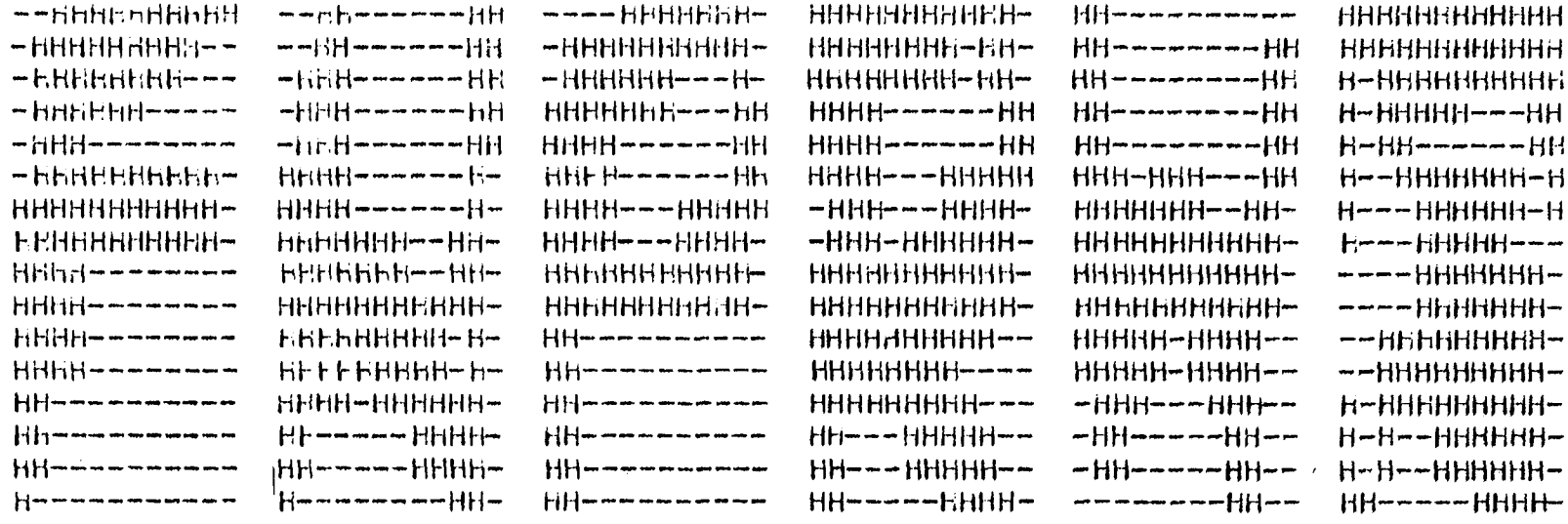
DTEMP

```
H------HHHHH  -HH-------H-  --HHHHHHHHHH  -HHHHHHHH---  ----------HH  HHHHHHHHHHHH
HHHHHHHHHHHH  -HHH------HH  ---HHHHHHHHH  HHHHHHHHHHH-  HH--------HH  HHHHHHHHHH-H
HHHHHHHHH---  -HHH------HH  -HHH-------H  HHH----HHHH-  HH---HH---HH  H-HHHHHHHHHH
HH----------  -HHH------HH  --HHHH------  HHH-----HHHH  HH--HHH---HH  HHHHHHH-HHHH
HH----------  -HHH------HH  --HHHH------  HHH-------HH  HH--HHH---HH  HHHHHHH---HH
HH----------  --HHH-----HH  --HHHH------  HHH-------HH  HH--HHH---HH  HHHHHHH---HH
HH----------  -HHHH-----H-  --HH-----HHH  -HH----HHHHH  HH--HHHH--HH  HHHHHHHHHHHH
HHHHHHHHHHH-  -HHHHH----H-  --HHHH---HHH  -HHH--HHHHHH  HH--HHHH--HH  HHHHHHHHHH-H
HHHHHHHHHHH-  HHH-HHH--HH-  --HHHHHHHHHH  -HHHHHHHHH--  HH-HHH-H--H-  HHHH--HHHHHH
HHHHHHHHHHH-  HHH-HHH--HH-  --HHHHHHHHHH  HHHHHHHHH---  -HHHH--HHHH-  HH-H-HHHHHHH
HHH-HHHHHH--  HHH-HHHHHHH-  --HH--------  HHH-HHHHH---  -HHH---HHHH-  HH-HHHHHHHH-
HH----------  HHH--HHHHHH-  --HH--------  HHH---HHH---  -HHH---HHHH-  HHHH-HHHHHH-
HH----------  HH----HHHHH-  --HH--------  HHH---HHHH--  -HHH----HH--  HHHH--HHHHH-
HH----------  HH-----HHH--  -HHH--------  HH-----HHH--  --HH-----H--  HHHH---HHH--
HH----------  HH-----HHH--  -HHH--------  HH------HHH-  --HH-----HH-  HHHH---HHHH-
HH----------  HHH----HHH--  -HHH--------  HH-------HH-  --HH-----H--  HHHH---HHHH-
```

TEST TEMPLATES

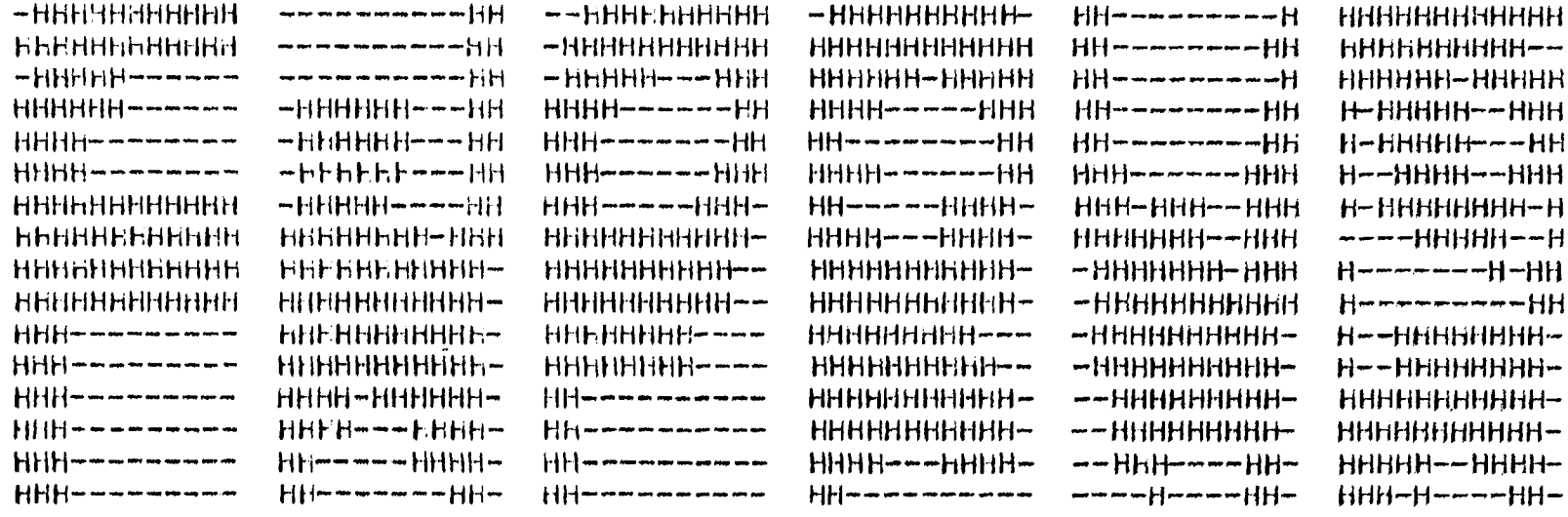
DTEMP

Figure 41. FNPRW (16 x 12) templates and difference templates for sample set 5 are shown ordered from left to right. "H" areas of DTEMP are comparison areas



REFERENCE TEMPLATES

DTEMP



TEST TEMPLATES

DTEMP

The set of 52 templates includes the alphameric set plus a symbol set. The figure for stage 2 testing assumes a six-member subgroup with multi-level testing. Processing time for 16 x 12 arrays can be approximated as 3 times the above total on the basis of the 3 to 1 ratio in array size.

No attempt has been made to determine the exact time needed to preprocess or perform second stage testing with the special recognition processor. It is possible, however, to compare the stage 1 testing operations. The special processor can compare two 64 bit arrays in 10 microseconds or 52 arrays in 0.52 milliseconds. Two 16 x 12 arrays can be compared in 30 microseconds or 52 arrays in 1.56 milliseconds. The operating times show the special processor reducing the recognition time by a factor of 10^3 to 10^4 when compared with the PL/1 - 360/65 system. Since the time spent in stage 1 represents over one-half the recognition time, there will be no difficulty in meeting a recognition rate of 100 characters per second.

<u>The threshold problem</u> One additional topic should be discussed; viz., the decision threshold. There is a need within the system for both an absolute and a relative decision threshold. An absolute threshold is needed for protection against very noisy characters that are steered to the wrong subgroup. To implement this protection a minimum value would first have to be selected for CS_{max}. A character completing the comparison tests at stage 2.1 with a comparison score less than this CS_{min} would be treated as an invalid subgroup member and rejected. The possibility exists also that the unknown is an allowed member of the subgroup but that it has such a high noise content as to preclude a

reasonable chance of being recognized.

A value of CS_{min} is needed for single member subgroups A, M, and ϕ. It is dangerous to identify each unknown entering these subgroups as A, M, or ϕ. Instead the subgroup testing should be multi-level and a function of the CS value. If the comparison of the unknown and the A reference template does not produce a CS value greater than CS_{min}, the system would request additional testing. These tests would use the reference templates for A and its nearest neighbors in character identification space even though these characters had a separation distance greater than d. The tests performed on the five data sets indicated that a value of CS_{min} could be determined for A, M, and ϕ when using 16 x 12 templates. For 8 x 8 templates a value of CS_{min} could be determined for A and M.

The possibility of determining a value of CS_{min} to guard against the entrance of invalid characters into a subgroup is not so clear. A few subgroups were tested with some invalid characters. These characters were close to the subgroup in character space but would not normally be steered to the subgroup. The results of the testing were inconclusive. The CS values for the invalid characters were approximately those of the allowed members. The issue is clouded however by the large differences between reference and test samples. These differences cause low comparison values for allowed members of the subgroup. A better sample, in terms of style consistency, should provide reference templates offering a higher comparison score for allowed members and a lower score for invalid members.

A relative threshold is needed for the difference between the CS values representing the first and second choices of the recognition system. This decision threshold would be used at stages 2.1 and 2.2. For example, in stage 2.1 the threshold would prevent a C from being directed to the DO subdivision of the CDGO subgroups. At level 2.2 it would prevent a D-O confusion if the separation distance between D and O is too small.

An examination of the test results indicates that a threshold could be established for this data. The separation distances are small but should be improved with better reference templates.

DISCUSSION OF RESULTS

After partitioning five sample character sets and conducting a series of subgroup tests, a close examination of the results indicated that further testing of the sets would be of little value. In examining the characters that failed, it was found that while some were due to failures in the scanning operation, many more were caused by extreme differences between the first and second sample sets produced by the same person. These differences were not the minor translational or rotational variations that might be expected. The changes were much larger; large enough to be characterized as style changes. Thus to continue to test the sample sets in their present format would be of limited value. The value was further limited due to the fact that the testing would have to continue to be done on a large computer system with high operating costs.

The results of this investigation indicate that a larger sample will have to be used to construct a users reference templates. The larger sample will produce a reference template that is more tolerant of the second order variations which occur in characters produced by the human system. Of even more importance is the fact that the results point to the need for a partially tutored population. This tutoring should be directed toward style consistency rather than to a single style format. It is this lack of style consistency which has been the largest source of recognition failures.

To what degree the user must be style consistent will have to be determined during the next phase of the recognition system project of

which this thesis is a part. The literature that is available on training a population for character recognition is usually directed towards training in simple style restrictions. No emphasis is made on style consistency throughout the character set beyond that consistency represented by the style restrictions.

A set of style restrictions will have to be determined to train a population in style consistency. However, instead of a rigid standard applicable to all, each person will have to analyze his own printing and arrive at a set of style restrictions. The human system does not require style consistency to recognize characters because it is a sophisticated recognition system which recognizes not by viewing isolated characters but by viewing characters in their contextual relationships.

Crook and Kellogg (12) suggest that some human systems will not be able to produce input documents of sufficient quality to be read by a recognition system. Whether this is due to a lack of motivation or due to some other aspect of the human system is unexplained. The recognition scores shown in Figure 14 offer some indication of the variability with which human systems produce machine readable characters.

For those unable to produce handprinted characters suitable for recognition it may be necessary to obtain style consistency through the use of a typewriter. The page reader built for this investigation could be modified to accept a typewritten input document. The optical system would have to be modified to accept the smaller sized character and the mechanical system would have to be altered to accept a different line spacing. A new input document would also have to be designed. These

modifications could be accomplished quite easily within the present framework of the system.

Even though it is not reasonable to continue testing with the data in its present form, the results do not cast doubt on the validity of the algorithm. The present data format does however preclude exhaustive testing of the subgroup tests and a determination of system performance that could be expressed, as is commonly done, in terms of % correct, % in error, and % of rejects.

The subgroups designed for this algorithm on the basis of testing five data sets show a general validity, i.e., the subgroup members have similar shape characteristics. These similarities are apparent when viewing the characters both as raw and transformed data and when considering their commonly degraded forms.

At the beginning of this investigation a decision was made to represent a character as a 32 x 24 bit array. The ability to recognize on the 8 x 8 templates indicates that there is sufficient granularity in the 32 x 24 bit representation.

The investigation shows that the 16 x 12 templates are more appropriate than the 8 x 8 templates for partitioning the character sets. The larger template produces tighter grouping as evidenced by reduced subgroup populations.

At the second stage of testing the 8 x 8 template offers fewer errors than the 16 x 12 template. The 8 x 8 representation of a character provides a template which is less sensitive to size, position, and density noise.

The question of the appropriate array size would appear to be answered here. However, it is possible that the 8 x 8 array may not be appropriate for all subgroup tests. After a character has passed stage 2.1 it may be necessary to change from an 8 x 8 representation back to a 16 x 12 representation. This could be used, for example, in separating characters that are naturally very close in character space. D and 0 or B and 8 are good examples of this situation. These characters normally differ only in the left side corners and are very difficult to separate. The fine detail of the 16 x 12 template may be required here.

Various forms of second stage tests offer a wide range of testing capability which can be designed to fit the subgroup population. The tests have shown the value of the selected area comparison template in reducing the noise in the difference templates. These tests offer added recognition power by allowing subgroups to request additional testing if the comparison scores do not fit a decision threshold.

It may appear that the amount of testing done in the second stage of the system is unnecessarily redundant. Experience with handprinted characters from untutored subjects indicates that the lack of style consistency requires significant redundancy.

The discussion of processing time shows that this algorithm can be used on the special recognition processor and meet the design recognition rate of 100 characters per second.

Even though testing has been confined to the alphameric character set, the algorithm does not restrict the expansion of the character set. The system would be able to accept the additional symbols shown

previously in Figure 2 with the exception of the common parentheses. These degenerate too easily into a 1 or a C. Instead the rectangular set of parentheses shown in Figure 2 should be used. These are from the ASA standard recognition character set.

The smaller symbols would be diverted at stage 1 on the basis of their size as is now done for the 1. They would then be directed to subgroup tests designed for these small symbols.

The system can in principle accept any symbol provided that it was produced with an appropriate degree of style consistency and provided that it was not a minor variation of a character already in the set. As a preliminary step to recognizing a special symbol, a partitioning check would have to be performed on the users sample set including the special symbol. This would determine into which subgroups the special symbol would fall. These subgroups would have to have their testing routines altered to accommodate the special symbol. This goes beyond the usual definition of an individualized system in that it requires the existence of an individualized subgroup testing procedure. Still this does not go beyond the capability of the recognition system. These tasks can be accomplished with no difficulty. If the system is to ask that a set of reference templates be activated in order to recognize an individuals handprinting, it is also possible to request that special subgroup tests be read in to accommodate the individuals special symbols.

While it is not possible to attach a precise performance score to this recognition system, the current literature provides a guide as to what the recognition performance must be. Commercial systems for reading

machine printed material have error rates of 0.1% to .01% or better depending on the amount of control exercised in preparing the input documents. A system which reads handprinted numbers offers a reject rate of a fraction of a percent (43), but again this will vary with the degree of control exercised in input document preparation.

The human as a character recognizer has error rates that vary from 3 to 5% for isolated characters (36). Human error rates for very noisy characters have been reported as high as 11% (37). A keypunch operator generally is considered to have an error rate of approximately 1%. The keypunch operator attains this performance through the use of contextual information, training and motivation.

The above error rates then show that if a handprinted character recognition system is to be of significant value it must attain the 99% recognition rate of the keypunch operators for which the system is commonly offered as a replacement. Also the system would be of even greater utility if it could approach the performance of systems designed for recognizing machine printed characters.

Current efforts in the field of recognizing handprinted characters have turned to contextual information to improve recognition rate. Duda and Hart (17) show a 3 to 1 reduction in error rate for one system using FORTRAN contextual information to read coding sheets. Duda and Hart suggest further improvements can be made in their programs. Nagy (38) reports that a recognition system will have to have a recognition rate of 95% before contextual analysis will increase the rate to 99% and become competitive with keypunching.

Two reports indicate that the performance possibilities for individualized systems should be about 97% (37, 38). The use of context should increase this recognition rate to over 99%.

Finally then it has been shown that it is possible to recognize handprinted characters on a small special purpose processor, but that this system will require a population that has been tutored with respect to style consistency. The current literature shows that the individualized system used here offers a better chance of success (with a full alphameric character set) than the non-individualized system. The literature also indicates the value of context as an aid to recognition.

If the proposed system can reach a recognition rate of 95 to 97% using templates from a tutored population, a human can provide the contextual analysis for those characters that the system would classify as rejects. This could increase the system performance to the necessary 99%.

BIBLIOGRAPHY

1. Alt, F. L. Digital pattern recognition by moments. Association for Computing Machinery Journal 9: 240-258. 1962.

2. Anderson, William H. The IBM 1288 optical page reader. IEEE Computer Group News 2, No. 6: 20-23. Nov. 1968.

3. Andrews, D. R., Atrubin, A. J., and Hu, K. C. The IBM 1975 optical page reader. Part III: Recognition logic development. IBM Journal of Research and Development 12: 364-371. 1968.

4. Bledsoe, W. W. and Browning, I. Pattern recognition and reading by machines. AFIPS Eastern Joint Computer Conference Proceedings 16: 225-232. 1959.

5. Bomba, J. S. Alpha-numeric character recognition using local operations. AFIPS Eastern Joint Computer Conference Proceedings 16: 218-224. 1959.

6. Bonner, R. E. A "logical pattern" recognition program. IBM Journal of Research and Development 6: 353-359. 1962.

7. Bonner, R. E. Pattern recognition with three added requirements. IEEE Trans. on Electronic Computers 15: 770-781. 1966.

8. British Computer Society Document Handling and Character Recognition Committee, 1966. Character recognition 1967. London, England, author. 1967.

9. Brown, Richard M. On-line computer recognition of handprinted characters. IEEE Trans. on Electronic Computers 13: 750-752. 1964.

10. Casey, R. and Nagy, G. Recognition of printed Chinese characters. IEEE Trans. on Electronic Computers 15: 91-101. 1966.

11. Chow, C. K. An optimum character recognition system using decision functions. IRE Trans. on Electronic Computers 6: 247-254. 1957.

12. Crook, M. N. and Kellogg, D. S. Experimental study of human factors for a handwritten numerical reader. IBM Journal of Research and Development 7: 76-78. 1963.

13. David, E. E., Jr. and Selfridge, O. G. Eyes and ears for computers. IRE Proceedings 50: 1093-1101. 1962.

14. Dimond, T. L. Devices for reading handwritten characters. AFIPS Eastern Joint Computer Conference Proceedings 12: 232-237. 1957.

15. Doyle, Worthie. Recognition of sloppy, hand-printed characters. AFIPS Western Joint Computer Conference Proceedings 17: 1737-1752. 1959.

16. Duda, R. O. and Fossum, H. Pattern classification by iteratively determined linear and piecewise linear discriminant functions. IEEE Trans. on Electronic Computers 15: 220-232. 1966.

17. Duda, Richard O. and Hart, Peter E. Experiments in the recognition of hand-printed text: Part II - Context analysis. AFIPS Fall Joint Computer Conference Proceedings 33, Part 2: 1139-1149. 1968.

18. Fischer, George L., Jr., Pollock, Donald K., Raddack, Bernard, and Stevens, Mary Elizabeth, eds. Optical character recognition. Washington D.C., Spartan Books. 1962.

19. Fu, King-Sun, Chien, Y. T., and Cardillo, Gerald P. A dynamic programming approach to sequential pattern recognition. IEEE Trans. on Electronic Computers 16: 790-803. 1967.

20. Geanias, E. C., Meagher, P. F., Norman, R. J., and Essinger, P. The recognition of handwritten numerals by contour analysis. IBM Journal of Research and Development 7: 14-21. 1963.

21. Grimsdale, R. L., Sumner, F. H., Tunis, C. J., and Kilburn, T. A system for the automatic recognition of patterns. IEE Proceedings 106B: 210-221. 1959.

22. Groner, Gabriel F. Real-time recognition of handprinted text. AFIPS Fall Joint Computer Conference Proceedings 29: 591-601. 1966.

23. Hennis, R. B. The IBM 1975 optical page reader. Part I: System design. IBM Journal of Research and Development 12: 346-353. 1968.

24. Highleyman, W. H. An analog method for character recognition. IRE Trans. on Electronic Computers 10: 502-512. 1961.

25. Highleyman, W. H. Linear decision functions with application to pattern recognition. IRE Proceedings 50: 1501-1514. 1962.

26. Holt, A. W. Character recognition using curve tracing. U.S. Patent 3,142,818. July 28, 1964.

27. Holt, A. W. Comparative religion in character recognition machines. IEEE Computer Group News 2, No. 6: 3-11. Nov. 1968.

28. Holt, A. W. What was promised - what we have - and what is being promised in character recognition. AFIPS Fall Joint Computer Conference Proceedings 33, Part 2: 1451-1459. 1968.

29. Kamentsky, L. A. and Liu, C. N. Computer-automated design of multifont print recognition logic. IBM Journal of Research and Development 7: 2-13. 1963.

30. Kanal, Laveen N., ed. Pattern recognition. Washington, D.C., Thompson Book Company. 1968.

31. Kuhl, F. Classification and recognition of hand-printed characters. IEEE International Convention Record, Part 4: 75-93. 1963.

32. Leimer, John J. Design factors in the development of an optical character recognition machine. IRE Trans. on Information Theory 8: 167-171. 1962.

33. Lewis, P. M., II. The characteristic selection problem in recognition systems. IRE Trans. on Information Theory 8: 171-178. 1962.

34. Liu, C. N. and Shelton, G. L., Jr. An experimental investigation of a mixed-font print recognition system. IEEE Trans. on Electronic Computers 15: 916-925. 1966.

35. Minneman, Milton J. Handwritten character recognition employing topology, cross correlation, and decision theory. IEEE Trans. on Systems Science and Cybernetics 2: 86-96. 1966.

36. Munson, John H. Experiments in the recognition of hand-printed text: Part I - Character recognition. AFIPS Fall Joint Computer Conference Proceedings 33, Part 2: 1125-1138. 1968.

37. Munson, John H., Duda, Richard O., and Hart, Peter E. Experiments with Highleyman's data. IEEE Trans. on Computers 17: 399-401. 1968.

38. Nagy, George. State of the art in pattern recognition. IEEE Proceedings 56: 836-862. 1968.

39. Nagy, G. and Shelton, G. L., Jr. Self-corrective character recognition system. IEEE Trans. on Information Theory 12: 215-222. 1966.

40. Rabinow, J. Development in character recognition machines at Rabinow Engineering Company. In Fischer, George L., Jr., Pollock, Donald K., Raddack, Bernard, and Stevens, Mary Elizabeth, eds. Optical Character Recognition. Pp. 27-50. Washington, D.C., Spartan Books. 1962.

41. Rabinow, J. The present state of the art of reading machines. In Kanal, Laveen N., ed. Pattern Recognition. Pp. 3-29. Washington, D.C., Thompson Book Company. 1968.

42. Roberts, L. G. Pattern recognition with an adpative network. IRE International Convention Record 1960: 66-70. 1960.

43. Rohland, Williams, Traglia, Patrick J., and Hurley, Patrick J. The design of an OCR system for reading handwritten numerals. AFIPS Fall Joint Computer Conference Proceedings 33, Part 2: 1151-1162. 1968.

44. Sheinberg, Israel. Optical character recognition for information management. In Kanal, Laveen N., ed. Pattern Recognition. Pp. 31-40. Washington, D.C., Thompson Book Company. 1968.

45. Spinrad, R. J. Machine recognition of hand printed. Information and Control 8: 124-142. 1965.

46. Stevens, M. E. Automatic character recognition - a state of the art report. National Bureau of Standard Tech. Note 112. 1961.

47. Teitelman, W. Real time recognition of hand-drawn characters. AFIPS Fall Joint Computer Conference Proceedings 26: 559-575. 1964.

48. Uhr, Leonard. Pattern recognition. In Uhr, Leonard, ed. Pattern Recognition. Pp. 365-381. New York, N.Y., John Wiley and Sons, Inc. 1966.

49. Uhr, L. and Vossler, C. A pattern recognition program that generates, evaluates, and adjusts its own operators. AFIPS Western Joint Computer Conference Proceedings 19: 555-569. 1961.

50. Ungar, S. H. Pattern detection and recognition. IRE Proceedings 46: 1737-1750. 1959.

51. Uyehara, G. U. A stream-following technique for use in character recognition. IEEE International Convention Record, Part 4: 64-74. 1963.

52. Wilson, Robert A. Optical page reading devices. New York, N.Y., Reinhold Publishing Corporation. 1966.

53. Zingg, Roy James. Structure and organization of a pattern processor for handprinted character recognition. Unpublished Ph.D. thesis. Ames, Iowa, Library, Iowa State University of Science and Technology. 1968.

ACKNOWLEDGMENTS

The author wishes to thank Dr. R. M. Stewart for his encouragement and his valuable critical commentary. The author thanks Dr. A. V. Pohm, who as project leader was a constant inspiration and source of guidance. The author wishes to express his thanks to Professor R. A. Sharpe for his suggestions in the area of programming techniques. The author thanks Dr. W. B. Boast and Dr. C. S. Maple for their generous support in terms of computer time.

The author also wishes to thank Terry McConnell and his group for their assistance in constructing the reader. Thanks also to Rennie, Dick, and Brent for their assistance in constructing the electronics for the system. Finally, thanks to Sheila for typing the rough draft for this thesis.

This work was partially supported by the Iowa State Affiliates Program in Solid State Electronics.