D. R. Andrews
A. J. Atrubin
K. C. Hu

The IBM 1975 Optical Page Reader

Part III: Recognition Logic Development

Abstract: The design approaches which were used to specify feature measurement logic, recognition reference standards, and decision functions for a multifont character recognition system are discussed. The importance of an intuitive approach to design, as opposed to a fully automated approach, is emphasized. The nature of the problem required an intimate interaction between the designers, who investigated complex pattern recognition problems and proposed design alternatives, and the computer, which relieved the designer of routine testing and evaluation of the tentative design.

Introduction

The IBM 1975 Optical Page Reader is a multifont page reader built especially for the Social Security Administration to process employers' quarterly earning reports. These reports are prepared on a variety of print devices and contain a large number of type fonts with an uncontrolled and wide range of print quality.[1,2] This paper describes the character recognition design work done for the machine. The organization and function of the recognition system in the 1975 are described in the companion paper by Hennis.[1] Suffice it to say here that recognition is accomplished by making measurements to detect the presence or absence of character features in the shift register bit patterns, comparing these measurements against a set of stored references, and deciding on the basis of this comparison whether or not recognition has occurred.

The paper is organized more as a discussion than as a formal report on a number of experiments. Subjective decisions play a large role in the history of the development of this machine. This is necessarily so because of the nature of complex recognition problems and because of the necessity of reliance on intuitive techniques for their solutions. It was often the case that sections of the machine were judged to be working sufficiently well or not well enough when the numerical evidence from the experiments supported the opposite viewpoint. These contrary opinions were held mainly on the basis of subjective evaluations of unquantifiable factors such as the "quality" of the input data. The practical problem of acquiring enough representative input data to eliminate subjective evaluation of the quality is discussed. The intent of the paper is to give the reader an impression of the kind of problems which occur in the design of a practical recognition system. We thought that could be best accomplished in this format. The main discussion section of the paper is followed by descriptions of the measurement design procedure, the decision process in the machine, and the reference design procedure.

Nagy (Sections II, III, and IV)[3] describes methods which can be used to process binary array representations of characters in order to distinguish the classes one from another. These methods will be called "theoretical" because, in each case, the usefulness is established for certain mathematically describable inputs. The hope is that the nature of the actual input is usefully approximated by the theoretical input. This usefulness has to do with sufficient accuracy of recognition and with an economically reasonable solution. The latter is important in cases where the machine size grows to accommodate the input variability.

Nagy (Section V)[3] also discusses that aspect of pattern recognition known as feature or measurement extraction. In contrast to the techniques referred to in the previous paragraph, these methods are, almost without exception, not based on a theoretical model of the input. The process of arriving at measurements is described by Nagy as "the somewhat undignified and haphazard manipulation involved in such cases to render the problem amenable to orderly solution." The success of these intuitive methods is based on the designer's ingenuity.

The proponents of the theoretical solutions would not exclude human intervention in the design process. The role

Andrews and Atrubin are with The Systems Development Division, and Hu is with the Advanced Systems Development Division, all at the Laboratory in Rochester, Minnesota.

of the designer is quite fundamental. Sebestyen says, "While the machine operates on the measurement values and learns, within the constraints of its capabilities, how to process the measurements, it remains a human task to specify what measurements should be made on the physical world. Although this question has received some attention, pattern recognition, aside from its applications, generally does not consider this problem."[4] The choice is one of relative emphasis. Should the bulk of the machine's recognition power reside in a complex statistical decision which operates on simple measurements? Or, should it reside in complex intuitively chosen and experimentally tested measurements which are combined in a simple decision?

The theoretical methods had not been adequately tested on problems of this nature. They could prove unworkable or result in uneconomical designs. Furthermore, they required a great deal of computer time. The intuitive design methods had worked with simpler problems of this sort, but the magnitude of this problem promised much tedious labor. We chose to begin with the unknowns of the theoretical methods in order to avoid the large amount of work required with the manual methods.

This paper describes our early experiments, their shortcomings, and the midstream change of emphasis to the intuitive design methods with a successful outcome. The report also contains interpretations of the results of our various experiments. We feel strongly that a dominant human element in the design process is necessary. This conclusion will not surprise anyone who has built an economically workable character recognition machine.

The basic question answered in this paper concerns the relative importance of measurement specification and the decision procedure. We concur with Minneman who believes that the proper selection of features is of overwhelming importance relative to the precise statistical operation performed on them.[5]

Discussion

In the early stages of the recognition logic development, it was hoped that the logic could be automatically designed. The approach used was to design a large list of simple measurements, select the most useful ones, and from these, design the reference.

Many experiments were carried out in cooperation with the IBM T. J. Watson Research Center character recognition department. At this stage of the development program there were available neither real documents, nor the actual 1975 scanner. These early experiments were performed on what eventually proved to be unrealistically high quality data (video).

To facilitate the automatic generation and evaluation of measurements, the measurement complexity was very limited. Most of the work was done using as a measurement an AND of $N (N \leqslant 9)$ shift register outputs (an N-tuple). A few experiments were performed using an AND of nine 3-way OR's. An AND of nine simple threshold elements (2 out 3's) was also tried. The performance differences were insignificant, so the simple AND configuration was retained.

The first technique tried was to generate random (subject to simple geometric constraints) N-tuples and to randomly assign polarities. Another technique, similar to the first, used average character shapes as the geometric constraints. Both of these techniques allowed the generation of a large number of measurement candidates ($\sim 10^4$) in a small amount of computer time. A third technique involved point-by-point design of the N-tuple, with the object of providing a specified response for each pattern on a sample tape. This method was orders of magnitude slower than those described above and was not used extensively.

Given a list of 10 or 20 thousand measurements, the next problem was to select a subset of these (about 100) which would be useful. Again, a variety of heuristic procedures was tried, all of which operated on a design tape containing the responses of the candidate measurements to a sample of video patterns. An initial filtering of the set of measurements was usually accomplished by computing an information measure for each measurement. This information measure was related to a measurement's ability to dichotomize the set of classes constituting the alphabet. More refined selection was done by computing the information contained in pairs of measurements and by the measurement's ability to increase the distance between the closest pairs of ternary references.

Another technique tried was selection based on the measurements' ability to optimize the parameters of the distributions of difference distance in the decision. Another selection procedure was concerned with the tails of the difference distance distribution. This proceeded by attempting to recognize correctly each of the characters on the design tape. The last three selection techniques also involved simultaneous selection of a set of references.

Reference design and selection started with a simple averaging of all characters of the same class identification. This elementary prodecure did not work well, even on high quality input. The next step was to design "single font" references, where the average response for each measurement was computed over each class of each font for each of a set of line widths. This required a selection procedure and the technique used attempted to recognize correctly each character on the design tape. As the number of references required increased from 1 or 2 per class to 20 or 30, the two-stage decision process described in another section of this paper became necessary.

Just before real data became available, a configuration which worked acceptably well on some of the upper and lower case data had been achieved. The design procedures used were highly automated, with the exception of the

selection of decision parameters, which were manually adjusted. The machine had grown in the decision area from approximately 150 ternary references to a total of about 1000 references. The measurements were still automatically designed AND gates, and the references were designed and selected substantially without human intervention.

When data of greater variability was encountered, however, the performance deteriorated by an order of magnitude. Because of the automated design procedures, there was no clear idea what of each component (measurement, reference) was supposed to do. This made it difficult to decide which components were not doing their jobs. It was in this effort to modify and improve the machine that the procedures described in the body of the paper were evolved. These procedures differ from the earlier ones described in this section in that the designer plays a dominant role in the decision making function during the design process.

It was postulated that a designer, familiar with the shapes of characters, could invent a measurement that would separate a difficult-to-separate pair of characters. One persistent problem which had not yielded to any automatic technique was that of separating an O with fairly square corners from a D. The distinguishing features were subtle differences in curvature at the corners. A designer attempted to invent a logic function to separate round corners from square corners. Tentative designs were tested on a computer and video patterns of problem characters were printed. The problems were corrected and after a few iterations, the measurement worked. The most important property of this measurement was not its ability to distinguish between the few hundred difficult O's and D's on the design tape; it was, rather, the extendability to many variations of corner configurations which the designer imagined as generalizations of the few specific examples which he had on hand.

A set of logic design methods was established based on the successful operation of intuitively designed measurements. We regard the inventing of measurements by human designers as necessary to the successful construction of a recognition system of this complexity. The great variability of the input data which must be recognized was not represented on any set of design tapes which we could reasonably accumulate. (Nagy[3] says, "In practice the training set is always too small . . .") This variability was accounted for in the designers' generalizations about the problems. Furthermore, with specific tasks assigned to each measurement, the process of isolating design problems and of making corrective improvements allowed a rapid convergence to a working system.

With the major design obstacle solved, it was necessary to incorporate the measurements into a decision. Most of the decision structure of the IBM 1975 was designed prior to the selection of techniques using intuitively designed measurements. Most of the choices about decisions were governed by the already imposed hardware limitations and the pressures of the schedule. Nevertheless, two distinct points of view emerged. The essence of these viewpoints, divorced from problems arising from hardware restrictions, is given below.

The first was that, given the excellent intuitively designed measurements, the decision logic could be relatively simple. Each problem was readily isolated and solved by a measurement that was designed for a very specific purpose. References were designed in the form of simple logical combinations of the well-understood measurements. Tentative designs were easy to modify. The proponents of this method believe that the bulk of the power of the machine to separate difficult characters should reside in the measurements. Then the specific decision process used to combine these measurements is relatively less critical. Intuitive design of references is favored because of the ease of pinpointing problems and of making corrections. The decisions for the numerals and for the upper-case sans-serif alphabetics were designed this way.

The second point of view is that there are advantages to be derived from using a more complex decision procedure. The specific procedures employed make full use of the multi-level ternary reference structure as described elsewhere in this paper. The proponents of this method believe that the more complex decision allows the design to be done with fewer measurements that are not as specific as in the former case. Such less-specific measurements, then, should be more generally useful for new problems as they arise than are the more specific measurements of the other method. The decision for the upper/lower case alphabetics and for the upper-case serif alphabetics was designed this way.

The agreement, then, is that intuitively designed measurements are necessary. That is, these measurements cannot be replaced by any reasonable amount of measurement logic which can be economically designed by any existing computer program. Nor can the omission of manually-designed measurements be compensated for by an existing economically reasonable decision method. The disagreement is whether the measurements should be designed to do the bulk of the separation or just to the point where they will suffice with a complex decision procedure.

Several millions of characters were processed during the design of the machine. Since completion of the design, we have had an opportunity to observe the performance. The substitution rate is apparently consistent with the specification. However, each experimental run turns up a set of substitutions which are, for the most part, substantially different from those previously seen. This illustrates the difficulty of adequately representing the input with any reasonably obtainable set of design data. This kind of behavior leads to the question of how to model the expected input to the machine and, in fact, how to test the machine to know if it is performing properly. These are questions

which are only vaguely answered and which can provide interesting research problems.

Measurement design

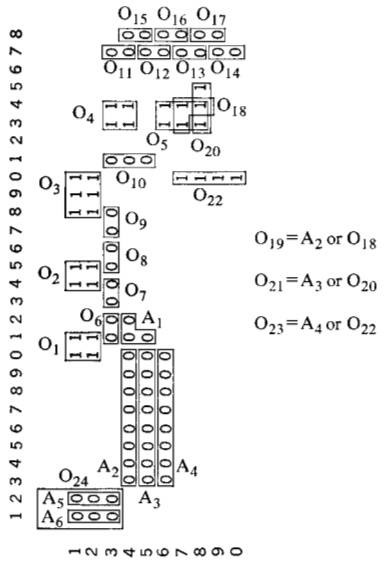
As the character pattern is shifted through the shift register, various measurement latches are set. Subsequently, the character is represented by its binary feature vector (the set of states of the measurement latches). This set of measurements must distinguish each class of character from all others. The variables which must be considered are the variety of shapes and sizes that a particular class of characters may assume, distortions and noises introduced by the printing process, and distortions introduced by the scanning process (quantizing effects, segmentation errors, etc.).

In most character recognition systems, the power to separate classes resides partly in the measurements and partly in the decision process. If the measurements are too simple, important shape information is lost and no decision process can recover the lost information. If the measurements are too complex, they attempt to do part of the separation that can be done less expensively by combining simpler measurements in the decision. A compromise was made in measurement complexity. The following kinds of features are used:

line segments (short, long, horizontal, vertical, slanted, etc.)
line endings
various corner curvatures
gaps in lines
relative positions of line segments
dimples, notches
character widths, heights
line thicknesses
blobs
overhangs.

The technique of measurement design is most conveniently described by example. It was decided that one measurement to distinguish the T from the I should be the right overhang of the upper bar of the T. The measurement should pass T's with tops of various widths, both with and without serifs, and should not pass I's. Since the top bar of the serifed I is sometimes longer than the lower bar, a decision had to be made as to what amount of difference in these lengths was sufficient to turn the measurement on. In the early iterations we decided that three bits was about right.

The solution to this problem was facilitated by the use of an IBM 1410 computer program that was written to assist in the evaluation of measurements. The proposed measurement is submitted to the computer in the form of a Boolean expression specifying the particular combination of black and white bits that should exist in the shift register if the measurement is to be satisfied. The program provides a graphic output display of the character bit pattern with



(a)

```
1 01 10 + 1 01 11 + 1 02 10 + 1 02 11 ) (1 01 14 +          5
1 01 15 + 1 02 14 + 1 02 15 ) (1 01 18 + 1 01 19 +         10
1 01 20 + 1 02 18 + 1 02 19 + 1 02 20 ) (1 03 23 +         15
1 03 24 + 1 04 23 + 1 04 24 ) (1 06 23 + 1 06 24 +         20
1 07 23 + 1 07 24 + 1 08 23 + 1 08 24 ) (  03 11 +         25
  03 12 ) (  03 13 +   03 14 ) (  03 15 +   03 16 ) (      30
  03 17 +   03 18 ) (  03 21 +   04 21 +   05 21 ) (      35
  03 27 +   04 27 ) (  05 27 +   06 27 ) (  07 27 +       40
  08 27 ) (  09 27 +   10 27 ) (  04 28 +   05 28 ) (      45
  06 28 +   07 28 ) (  08 28 +   09 28 ) (  04 11 ·       50
  04 12 ·   05 11 ) (  04 03 ·   04 04 ·   04 05 ·       55
  04 06 ·   04 07 ·   04 08 ·   04 09 ·   04 10 +       60
1 07 23 + 1 07 24 + 1 08 24 ) (  05 03 ·   05 04 ·       65
  05 05 ·   05 06 ·   05 07 ·   05 08 ·   05 09 ·       70
  05 10 + 1 08 23 + 1 08 24 + 1 08 25 ) (  06 03 ·       75
  06 04 ·   06 05 ·   06 06 ·   06 07 ·   06 08 ·       80
  06 09 ·   06 10 + 1 07 20 + 1 08 20 + 1 09 20 +       85
1 10 20 ) (  01 01 ·   02 01 ·   03 01 +   01 02 ·       90
  02 02 ·   03 02 )                                      95
```

(b)

Figure 1 Measurement logic designed to separate I from T; (a) the graphic display of the logic identifies the shift register positions which serve as inputs to the logic circuit. The boxes labeled with O's and A's represent OR and AND functions, respectively and the contents of the boxes indicate whether black (1) or white bits (0) will satisfy the functions; (b) Boolean statement of measurement logic appears in coded form as explained in the text.

the measurement pattern superimposed in a position in which a measurement match occurs if there is one. This display is useful in deciding how to modify the measurement.

Figure 1 shows the measurement for the T vs I problem at one of the late stages of its design. The logical description of the measurement is also shown. It is written as the product of a sum of products. Each variable appears as a 4-digit word in the description; the first digit of each word is either 1 or blank to indicate black or white, respectively; the second and third digits are the horizontal and vertical coordi-

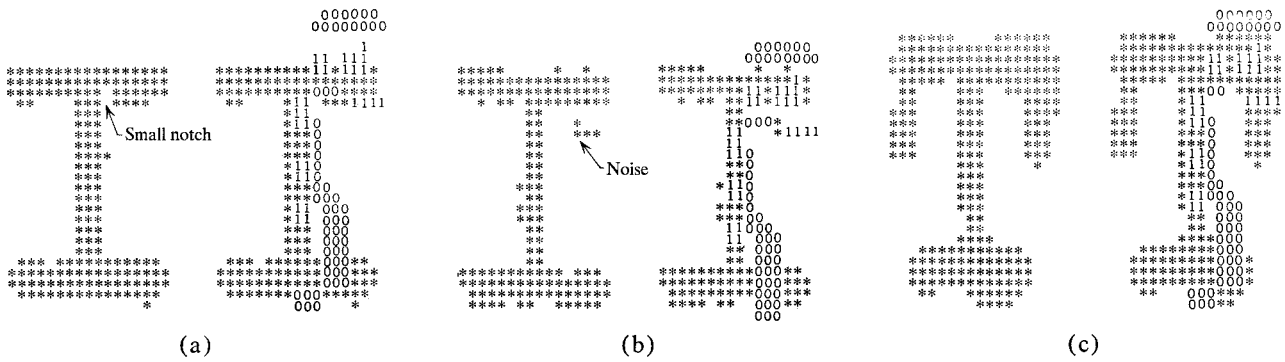
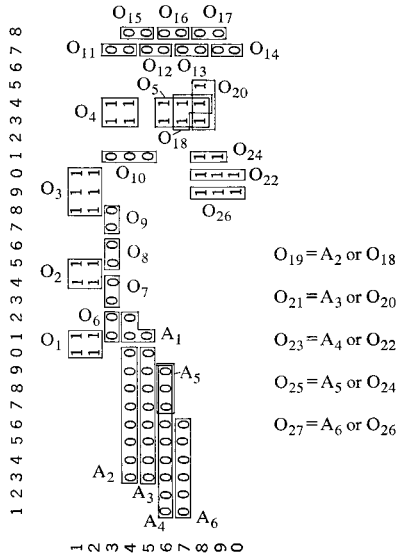


Figure 2 Character bit patterns and superimposed measurement logic defined in Fig. 1; (a) and (b) upper case I's appear as capital T to the logic because of small notch and noise, respectively, (c) pattern for T is too tall to satisfy the logic.

nates of the shift register address to be tested; and the fourth digit is a logical connective. The connectives used are • for a low level AND, a + for an OR, a)(for a high-level AND, and a) for the termination of the expression. The first 52 words in the expression loosely describe a vertical bar and a top right horizontal bar. That is, there should be at least one black bit in each of the boxes in the Figure identified by O_1, O_2, O_3, O_4, and O_5; also there should be at least one white bit in each box, O_6 to O_{17}. These bars register both T's and I's in a particular place in the array. The first part of the measurement is not used to discriminate between the T and the I. The next 11 words ensure that any character that passes the measurement and has a lower bar extending into column 4 of the array also has an upper bar extending at least to column 7. The next 11 words perform the same function shifted one column to the right. The next 12 words ensure that if the bottom bar extends into column 6, the top bar will have a vertical serif on it (that is, will dip into row 20, which is below the white registration piece in row 21). The last 6 words prevent very tall I's from falling below the white pieces in columns 4, 5, and 6.

Several problems were discovered while testing this measurement. The I of Fig. 2a passed the measurement because of the small notch on the underside of the top bar. The I of Fig. 2b passed because of the noise which gave the character the appearance of having a serif. The tall T of Fig. 2c ran into the white piece at the bottom. Some changes were made and the resultant measurement is shown in Fig. 3. The serif detector was strengthened so that the offending I's would no longer pass and the white registration piece which caused the tall T's to fail was replaced by logic that kept out tall I's in a different way. This measurement worked quite well on the design tape; however, it was simplified (at the cost of some discriminating ability) to lower its cost. The final measurement had half the words of that

Figure 3 Revised logic.




```
1 01 10 +  1 01 11 +  1 02 10 +  1 02 11 ) ( 1 01 14 +          5
1 01 15 +  1 02 14 +  1 02 15 ) ( 1 01 18 +  1 01 19 +         10
1 01 20 +  1 02 18 +  1 02 19 +  1 02 20 ) ( 1 03 23 +         15
1 03 24 +  1 04 23 +  1 04 24 ) ( 1 06 23 +  1 06 24 +         20
1 07 23 +  1 07 24 +  1 08 23 +  1 08 24 ) (  03 11 +         25
  03 12 ) (  03 13 +    03 14 ) (  03 15 +    03 16 ) (         30
  03 17 +    03 18 ) (  03 21 +    04 21 +    05 21 ) (         35
  03 27 +    04 27 ) (  05 27 +    06 27 ) (  07 27 +         40
  08 27 ) (  09 27 +    10 27 ) (  04 28 +    05 28 ) (         45
  06 28 +    07 28 ) (  08 28 +    09 28 ) (  04 11 ·         50
  04 12 ·    05 11 ) (  04 03 ·    04 04 ·    04 05 ·         55
  04 06 ·    04 07 ·    04 08 ·    04 09 ·    04 10 +         60
1 07 23 +  1 07 24 +  1 08 24 ) (  05 03 ·    05 04 ·         65
  05 05 ·    05 06 ·    05 07 ·    05 08 ·    05 09 ·         70
  05 10 +  1 08 23 +  1 08 24 +  1 08 25 ) (  06 03 ·         75
  06 04 ·    06 05 ·    06 06 ·    06 07 ·    06 08 ·         80
  06 09 ·    06 02 ·    06 01 +  1 08 20 +  1 09 20 +         85
1 10 20 ) (  06 07 ·    06 08 ·    06 09 +  1 08 21 +         90
1 09 21 ) (  07 01 ·    07 02 ·    07 03 ·    07 04 ·         95
  07 05 ·    07 06 +  1 08 19 +  1 09 19 +  1 10 19 ·        100
```

(b)

shown in Fig. 3 and failed on some I's with large amounts of extraneous black bits. Some trade-off between performance and measurement cost was made on most of the measurements selected for use in the system.

Decision process

A description of the decision process used in the 1975 follows: Denote the N-position feature vector by

$$X = (x_1, x_2, \cdots, x_N), \qquad x_i = 0, 1. \tag{1}$$

Suppose that there are P classes of characters, U_k, $k = 1, 2, \cdots, P$, to be recognized. Since there is usually more than one reference designed to represent each class of character (e.g., one reference to recognize sans-serif upper-case E's, one for pronounced-serif E's, and one for medium-serif E's), suppose that for each class U_k there are S_k references, Y_{kj}, $j = 1, 2, \cdots, S_k$. Then, denote the j^{th} reference of class U_k by

$$Y_{kj} = (y_{1kj}, y_{2kj}, \cdots, y_{Nkj}), \qquad y_{ikj} = 0, 1, d. \tag{2}$$

The distance of X from a particular reference Y_{kj} is given by

$$D\ (X,\ Y_{kj}) = A_{kj} + \sum_{i=1}^{N} (x_i,\ y_{ikj})\ , \tag{3}$$

where,

$$(x_i,\ y_{ikj}) = \begin{cases} 1, & x_i = 1 \text{ and } y_{ikj} = 0, \text{ or} \\ & x_i = 0 \text{ and } y_{ikj} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

The additive constant A_{kj} is a positive integer associated with Y_{jk}. Another constant, M_{kj}, (the multiple fixed correlation cutoff) is associated with Y_{kj}. This constant is used to limit the distance over which decisions based on Y_{kj} will be made. For each class U_k the reference Y_{kj} having the minimum distance (subject to the M_{kj} limit) from X is determined:

$$D_k = \min_j \left\{ [D(X,\ Y_{kj})]_{D(X, Y_{kj}) \le M_{kj}} \right\}\ ,$$

$$j = 1, 2, \cdots, S_k\ . \tag{4}$$

Then the class U_A having the over-all minimum distance is found:

$$D_A = \min_k \left\{ D_k \right\}\ , \qquad k = 1, 2, \cdots, P\ . \tag{5}$$

An arbitrary class is chosen from among several if these several have the same minimum. Associated with the reference yielding this minimum is a constant C_A. Excluding class U_A, the class U_B yielding the next minimum is determined:

$$D_B = \min_k \left\{ [D_k]_{k \neq A} \right\}\ . \tag{6}$$

Note that Eq. (6) does not exclude the possibility $D_A = D_B$. The decision criteria which follow are based on two more positive integers, R and T, with $R < T$:

(1) If $D_A < R$ and $D_B - D_A > C_A$, then the character is said to be recognizable and of class U_A. This may result in a correct recognition if the character is from class U_A or a substitution if the character is not from class U_A.
(2) If $D_A > T$, then no reference is close enough and the character is called unrecognizable.
(3) Otherwise, there is not confident recognition, but the character is labeled a conflict with best guess class U_A.

To conserve time the decision process is arranged in two stages. The first stage decision recognizes a large portion of all input characters. For characters not recognized at the first stage, a candidate list is obtained which directs the second stage decision to process the feature vector against a larger set of references from classes which appear in the candidate list. Sets of references and their associated parameters were designed as described in the next section.

Reference Design

Even with the best measurements, there are sufficient differences among the feature vectors of representative members of a class to necessitate using many ternary reference vectors. A variety of reference design methods was employed. Three of the methods for designing individual references were based directly on clustering techniques, that is, on ways of grouping together similar feature vectors and representing each such group by a reference. These are:

Shape classification. This is based mainly on the character shape for each type font. For instance, a particular letter of the alphabet from most of the common Pica and Elite fonts will have a (nearly) standard shape. The extent to which size and printing quality is considered varies with the subset of measurements used. This clustering is done intuitively.

Unsupervised clustering. This is based only on the feature vector representation of the characters. A large sample of feature vectors from different classes of characters is divided into a number of clusters. The variation among feature vectors of the same cluster is kept within a certain limit and the distance between clusters is kept above a certain constant. A computer program iterates by creating or deleting cluster centers and adding or deleting feature vectors from a certain cluster. References so generated, depending on the parameters used, are usually controlled to recognize

a large number of classes of characters with a relatively small number of references. This approach is mainly used to generate an initial set of references and to keep the total number of references low.

Supervised clustering. This is based on the feature vectors and their associated class identities. Clustering is done on a character class basis. The program tries to minimize the number of clusters by maximizing the number of samples to be included in each cluster while keeping the cluster tight.

The actual ternary references for these three methods are obtained by first calculating the probabilities that each element of the feature vector is in the 1-state for each cluster of characters and then quantizing the probabilities into three states as a ternary reference. If the probability is above a certain percentage, say 85%, the corresponding element in that reference vector is assigned a 1-state. If the probability is below a certain percentage, say 10%, the corresponding element in that reference vector is assigned a 0-state. Intermediate values become "don't cares," (*d*).

Another kind of reference construction depends on the designer's understanding of the measurements and how they match particular kinds of characters. The designer assigns the 1's, 0's, and don't cares to form the reference. His concept of the input extends beyond the characters on the design tape. He uses his insight into the nature of the problem (sources of noise, expected character perturbations etc.) to generalize beyond the available data. A simple decision organization is used with this technique to keep the design manageable. It is called a zero-distance decision. Only exact hits on references ($D = 0$) are used. The full decision procedure is used in the upper/lower case field and in the upper-case serif field. The zero-distance decisions are used in numeric and upper case sans-serif fields.

Individual references designed by the above procedures are not always useful for properly recognizing characters. A complete set of references must be evaluated in order to make relevant modification. The overall performance is specified by the number of failures, conflicts with best guess, and substitutions. The test set is a large sample of representative characters from real documents.

The magnitudes of the conflict rate and of the substitution rate can be changed by varying the decision parameters A, C, and M. A and M control the placement of a reference in the decision table. The distance between the feature vector and the reference is computed. To this is added A which allows partial compensation for measurements of different information value. The parameter M limits the depth in the decision table which will be used for a specific reference. This allows a reference to be used in a limited region of the decision space (that is, within a limited distance from the defined sub-cube). Finally, C is used to ensure that no reference for some other class is too close to the best match reference. The ratio of reject rate and substitution rate is set with consideration given to the costs of making whatever corrections are necessary for proper system performance.

In order to estimate the role of individual references in the context of a complete set of references the following counts were made.

Necessary for recognition. This is the number of characters which are properly recognized by the reference. Without this reference, the recognition attempt will either result in a conflict or the character will be erroneously recognized by other reference(s).

Necessary for conflict (or no error). This is the number of characters which will be in conflict because of the reference. Without this reference, the characters will be recognized incorrectly by other references.

Substitutions introduced by the reference. This is the number of characters incorrectly recognized by this reference.

Conflicts introduced by the reference. This is the number of characters which result in conflict due to the reference. Without this reference, the characters will be recognized correctly.

Necessary for recognition in the absence of another reference of the same class identity. This is the number of characters that will be recognized by this reference if another reference designed to recognize the same class of characters has been removed from the set of references. This indicates the potential of a reference.

To study the recognition results after single modifications would take too much time. The five classifications above tend to describe the role of each reference in the total decision scheme. Many simultaneous changes can be made fairly safely employing this information.

Conclusion

The major question which was answered during the development of the recognition system for the IBM 1975 concerned the role of the human designer. Could his intuition and extensive participation in the design procedure be replaced to any large degree by elaborate computer processing of the design data? We were unable to accomplish this. Furthermore, we are convinced that the designer's role is the dominant factor in the design procedure. An automatic algorithm to replace the designer would have to be qualitatively different from the kinds that are presently available.

Acknowledgments

The authors would like to acknowledge the contribution of Messrs. R. G. Casey, C. N. Liu, and G. Shelton of the T. J. Watson Research Center, and Messrs. D. Bachman, G. Benson, M. Bond, W. Naylor, and B. Steele of The IBM

development laboratory at Rochester. These people were involved at various times during the development of both the design techniques and particular components which make up the 1975 recognition system.

References

1. R. B. Hennis, "The IBM 1975 Optical Page Reader, Part I: System Design," *IBM J. Res. Develop.* **12,** No. 5, 346–353 (1968), this issue.
2. M. R. Bartz, "The IBM 1975 Optical Page Reader, Part II: Video Thresholder," *IBM J. Res. Develop.* **12,** No. 5, 354–363 (1968), this issue.
3. G. Nagy, "State of the Art in Pattern Recognition," *Proc. IEEE* **56,** No. 5, 836–862 (1968).
4. G. S. Sebestyen, "Decision-Making Processes in Pattern Recognition," Macmillan Co., New York, N. Y., 1962, p. 3.
5. M. J. Minneman, "Handwritten Character Recognition Employing Topology, Cross Correlation, and Decision Theory," *IEEE Trans. Systems Science and Cybernetics,* SSC-2, No. 2, 95 (1966).

Received October 24, 1967.