Character Recognition by Digital Computer using a Special Flying-Spot Scanner

By R. L. Grimsdale and J. M. Bullingham

A method has already been described for the recognition of spatial patterns by digital computer (Grimsdale *et al.*, 1959). This paper describes how the process can be simplified and speeded up by the use of a flying-spot scanner with special coding facilities.

Introduction

It is possible to divide character-recognition systems into two broad classes, these being the parallel and serial methods. In the former, information from the whole character field is processed simultaneously, whilst in the latter some form of scanning process is used to examine points of the character sequentially. An appraisal of the two methods is similar to a comparison of parallel and serial computing circuits, where speed of operation must be measured against cost of equipment. An important difference, however, is the problem of location with the parallel system of character recognition, since in any simple device there is a two-dimensional location problem. The location problem is not so serious in the serial method since, at the worst, the character need only be correctly located in one dimension. Furthermore, if a television-type scanning raster is used it is possible to examine the information line by line, so that the precise position of part of a character relative to a scanning line is unimportant.

With the exception of the human eye, the Solartron ERA system,* the Perceptron (Rosenblatt, 1958), and similar devices, most character-recognition systems so far built use a serial processing.

The most popular devices use characters printed with magnetic ink and perform a serial scanning process by transporting the line of print under a single or multiple block of magnetic reading heads. The magnetic-ink method virtually amounts to arranging a code in a form which is easily recognizable by a human reader. Although philosophically simple, magnetic-ink systems have been used commercially with some success.

An alternative is the optical scanning method using a cathode-ray flying-spot scanner. A typical system (Greanias *et al.*, 1956) uses vertical scanning lines and notes the position and length of the intersection of these scanning lines with the figure. Thus, a capital T first gives a number of short intersections, followed by a long intersection and again a series of short intersections. The characters are then identified by a set of logical circuits.

A more sophisticated technique employs a flying-spot scanner to determine the contours of the edges of the character (see, for example, Sprick and Ganzhorn, 1959). Consider a letter C being scanned by horizontal lines with the spot progressing from left to right; if the times are t_1, t_2, t_3, t_4 at which the scanning spot meets the left-hand contour of the letter on successive scanning lines, then the quantities t_1, t_2, t_3, t_4 give the x co-ordinates of the edge for successive increments in y. It is relatively simple to generate $\frac{dx}{dy}$ and $\frac{d^2x}{dy^2}$ by electronic means, and so obtain a description of the character contours. The disadvantage of this system is that the scanner must be able to decide which of several edges is being scanned.

The present work is an extension to a method previously described (Grimsdale *et al.*, 1959), and which is outlined here. In this method, the function of the flying-spot scanner was to transfer the image of the character to the store of a computer. It then appeared as a (40 $\times$ 64) two-dimensional array of points. Points on the character were stored as ones, and blank paper as zeros. The stored pattern was then scanned internally by a computer program, horizontal scanning lines being employed for convenience. After an averaging process, the program proceeded to split the character into groups, similar to those to be described in this present paper. For example, upper case T would have been split into two groups, the horizontal line and the vertical line. Following this division,the nature of the groups was determined to establish whether they were straight or curved, etc. It was then possible to give a simple description of the figure in terms of these groups and the way in which they were connected, the description being referred to as the *statement*. The recognition process consisted of comparing the new statement with statements already in the memory of the computer.

The system had no built-in prototype patterns, it simply consisted of a mechanism for producing and comparing statements. Patterns presented to the scanner for the first time were named by the operator. On subsequent presentation of the patterns, and in spite of possible distortions, and different positioning, the computer was able to identify them.

The feature of interest in this system is the way in which the two-dimensional pattern has been reduced to a one-dimensional statement. This is done by searching for the basic features of the pattern; the process is independent of rotation, positioning, size, or even small changes in the shape of the pattern.

The work described here is an extension to this system, and describes the design of a proposed scanner to perform the scanning operations, previously performed by

* See paper by Merry and Norrie on p. 137 of this issue.

the computer. The new system is not intended to be as versatile, and must be regarded as a "character" rather than a "pattern" recognition system. It is capable of recognizing normal typewritten material and print of similar size, and is not restricted to any particular type fount. The amount of rotation permissible is small, but this is not a severe limitation because large rotations are not common.

The main part of the scanner is already constructed, but the coding circuits have so far only been simulated by a computer program. There are no special problems concerned with the construction of these, as they are only required to perform elementary logical operations.

The general arrangement of the scanner is shown in Fig. 1. The characters are scanned from top to bottom and from left to right. This arrangement is convenient when dealing with a line of print because it is easier to separate the individual characters. By causing the y-deflection timebase to pause, the vertical scanning line is divided into eight groups of five points.

The degree of resolution afforded by the 40 points in each vertical scanning line is necessary to determine the position of the edges of the character, particularly in relation to other edges. However, the total information content in each line is quite small, and it is therefore desirable to reduce the number of bits transmitted from the scanner to the computer. This can be done in two ways: by "line compression" or "line difference." With the former, a coding method is employed which condenses each of eight 5-bit "words" forming a vertical scanning line into 3-bit words. The line difference method, as its name implies, gives indications of significant changes from line to line. Although neither method gives a very great reduction in number of bits transmitted from the scanner to the computer, there is a considerable saving in the amount of subsequent processing by the computer.

Line Compression Method

The basis of this method is a simple coding of each of the 32 possible 5-bit patterns into a 3-bit representation. Although it is possible for all of the 32 combinations to occur, a successful coding process is possible because certain of the combinations have only a low probability of appearing. The exception to this is with weakly printed characters consisting of an irregular number of tiny spots or blobs. This type of defect or "noise" is partly overcome by the averaging effect of the flying-spot scanner, but neither of the two methods is particularly suitable for use with very bad printing.

The coding operation to convert each group or word of 5 bits into one of 3 bits is relatively simple. The complete coding system is shown in Fig. 2. The least-significant digit is at the top of a group. It is possible to give a descriptive meaning to each of the eight possible outputs of the codes. Individually, the presence of the digits q_0, q_1, or q_2 imply that there is a part of the character present at the top, middle, or bottom respectively of the original 5-bit word. The various combinations of the three digits q_0, q_1, q_2 can be described as

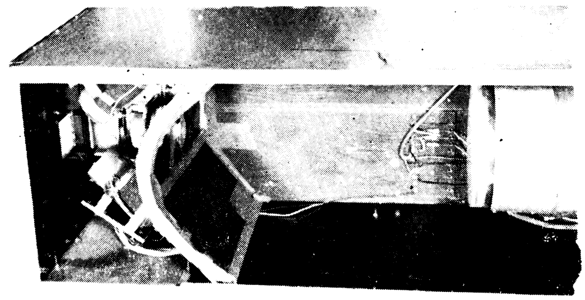

Fig. 1.—General arrangement of the scanner. Flying-spot scanner cathode ray tube on the right, and optical lens, photomultipliers and printed object on the left

```
           p0  .x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x
           p1  ..xx..xx..xx..xx..xx..xx..xx..xx
           p2  ....xxxx....xxxx....xxxx....xxxx orig-
           p3  ........xxxxxxxx........xxxxxxxx inal
           p4  ................xxxxxxxxxxxxxxxx

Top        q0  .x.x.x.x.x.x.x.x.x.x.x.x.x.x.x.x coded
Middle     q1  ..x.xxxxxxxxxxxx..x.x.x...x.x.xx out-
Bottom     q2  ................xxxxxxxxxxxxxxxx put
```

$q_0 = p_0$

$q_1 = (p_1 \text{ and } p_2 \text{ and } p_3) \text{ or } (p_1 \text{ and } \bar{p}_0) \text{ or } (p_3 \text{ and } \bar{p}_4) \text{ or } (p_2 \text{ and } \overline{(p_0 \text{ and } p_4)})$

$q_2 = p_4$

Fig. 2.—Line compression coding

Nothing, Top, Middle, Bottom, Top and Middle, Top and Bottom, Middle and Bottom, All (top, middle and bottom).

The coding system is designed to give a good representation of the input pattern in terms of the above descriptions.

The coding system can be mechanized by simple circuitry designed according to the following rules:

$$q_0 = p_0,$$
$$q_2 = p_4,$$

$$q_1 = (p_1 \text{ and } p_2 \text{ and } p_3) \text{ or } (p_1 \text{ and not } p_0) \text{ or } (p_3 \text{ and not } p_4) \text{ or } (p_2 \text{ and not } (p_0 \text{ and } p_4)).$$

It is clear that the set of coded 3-bit words representing the pattern of the symbol or character is not invariant under simple transformations of the symbol, nor is this intended.

The object of the coding system is to reduce the number of bits to be transmitted from the scanner to the computer. This reduction has the advantage of reducing the loading on the data link, and of presenting the computer with a reduced quantity of information, simplifying the subsequent operations to be performed by the computer.

Fig. 3(*a*).—Enlarged typewritten d as presented to the scanner

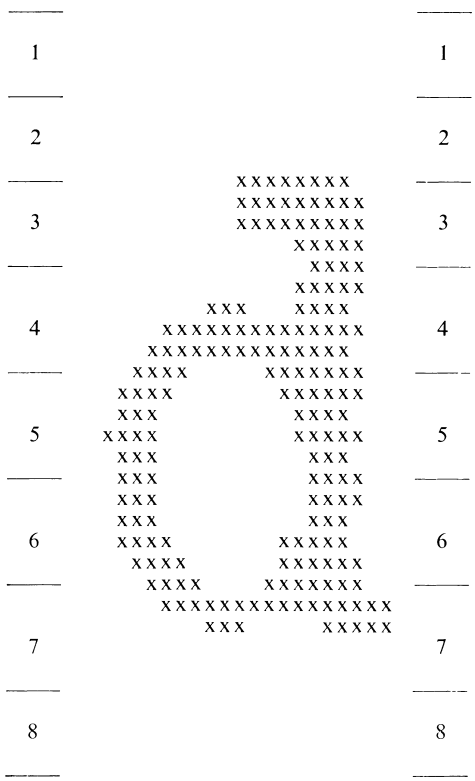

Fig. 3(*b*).—Scanner output for Fig. 3(*a*) plotted as a two-dimensional array

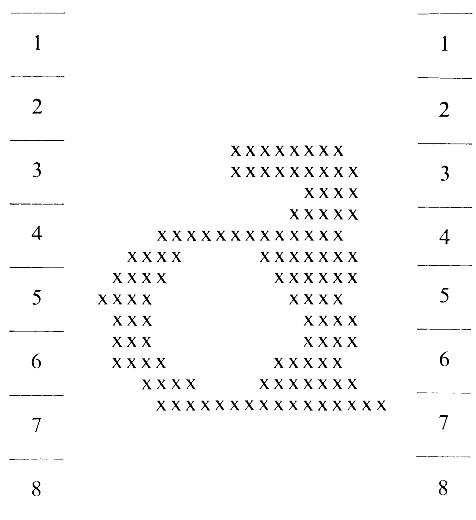

Fig. 3(*c*).—Coded representation of Fig. 3(*b*)

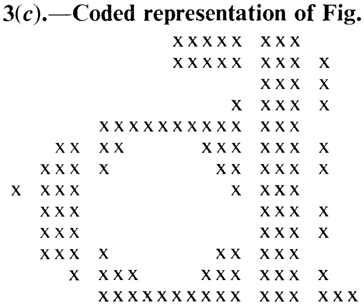

Fig. 3(*d*).—Division into groups of Fig. 3(*c*)

Computer Interpretation of the Data Compression Code

The method of interpretation of the Data Compression Code has some similarity with the Line Difference System. For clarity, however, the two systems will be described separately. Fig. 3(*a*) shows an enlarged version of a typewritten d as presented to the scanner. Fig. 3(*b*) shows the output of the flying-spot scanner plotted as a two-dimensional array, and Fig. 3(*c*) gives the coded representation.

As the code words for a particular scanning line appear in sequence, the connection between one code word and the next is investigated. This process determines the connectivity within the vertical line. Two code words are connected if the first has digit $q_2 = 1$ and the second has digit $q_0 = 1$, otherwise there is a gap and therefore no connection. The pattern is now available as sets of vertical lines as in Fig. 3(*c*). For example, a line chosen from the middle of this figure consists of the sequence

0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 .

In this line there are three distinct segments. If there are similar segments on adjacent lines they are combined to form groups. This division into groups is shown in Fig. 3(*d*). Isolated points and segments which occur on one line only can generally be ignored.

The recognition of the symbol from the groups is the same for both methods, and will be described in the description of the Line Difference method which follows.

The Line Difference Method

Although a high horizontal resolution is desirable, this results in a great deal of similarity between successive vertical scanning lines. This second method compares adjacent vertical scanning lines and reports only the differences between them. In terms of number of bits transmitted between scanner and computer this does not result in a large economy, but there is a valuable saving in the number of computer processing operations at the expense of very little extra equipment.

The Line Difference method may be described as follows. Each vertical scanning line is compared with the previous line. If there is an important difference, full details of the line are transmitted, but if they are similar only the differences are indicated.

Each vertical scanning line will have a number of segments interspersed with gaps. A line is similar to the previous one if each segment matches, or nearly matches, each segment of the previous line. If this agreement is not obtained "detailed scan" is performed, giving successively the length of each gap and each intersection.

Examples in which the line differences are reported are shown in Fig. 4. The relative positions of the tops and bottoms of the corresponding segments in the line under consideration and the previous line are compared. If the vertical positions of the points at, say, the tops of the two segments are the same, then the code word "same" is transmitted from the scanner to the computer. If, alternatively, the tops are separated by not more than two points, then the appropriate code words "rising" or "falling" are transmitted. In the example shown in Fig. 4, the difference reported between line 2 and line 1 is "falling, same," that is, the top of the intersection on line 2 is below the top of that on line 1, whereas the bottoms are at same level. Similarly the difference reported for lines 3 and 2 is "same—rising."

```
123
x
x
xxx
xxx
xxx
xxx
xxx
xxx
xxx
xxx
xxx
xxx
xxx
xxx
xxx
xx
```

Fig. 4.—"rising, falling, same" system

	A	B	C	D
Previous Line	0	1	0	1
Present Line	0	0	1	1

(*a*)

	1	2	3	4	5	6	7	8	9	10
A	S									Detail Scan
B	R	R	R						F	
C	F			F	F				R	
D									S	

(*b*)

	1	2	3	4	5	6	7	8	9	10
A	9	10	10	10	10	9	9	9	9	Detail Scan
B	6	6	6	5	10	7	10	10	4	
C	8	3	10	8	8	10	10	7	2	
D	1	1	1	1	1	10	10	10	1	

(*c*)

Fig. 5.—Coder logic

It is intended that a special coding device will be built to operate in conjunction with the scanner. This will determine the nature of the differences between adjacent vertical lines, and also indicate when a detailed scan is required. As can be seen from the following description, the logic of the unit is simple and can be implemented using standard computing circuits. To demonstrate the operation of the unit, its action has been simulated by a computer program, the results from which are given in Fig. 6 (see p. 133). The simulation program has been written with the identical logical organization of the intended coding equipment.

The coding is performed by considering pairs of points in two adjacent vertical scanning lines. If the origin (0, 0) is at the top left-hand corner of the raster, and if line 7 is being compared with line 6, then the coding operation will be performed by making the scanning spot examine the pairs of points with co-ordinates (6, 0), (7, 0) and (6, 1), (7, 1), etc., successively. There are no restrictions on the positioning of the character provided that it lies within the field of the scanner.

The coding is performed by a serial process, each step being conditional upon the results of the previous step. A detailed description of the process is given below, but briefly the operation is as follows. When the top of an intersection is encountered on one or both of the lines, in the course of scanning from top to bottom, a signal is transmitted from the scanner to the computer. If there are points on both lines the signal is "same"; if there is a point on the first and not the second the signal is "falling"; similarly, if there is a point on the second but not on the first, the signal is "rising." Further scanning establishes whether these classifications are correct. If they are incorrect, a detailed scan is given for the line.

Logical Design of Coder

The coding unit effectively consists of a ten-state device which has the option of changing state as each pair of points is scanned. The next state depends on the previous state and present configuration of the two points. Each state has a particular significance; for example, state 9 represents the starting condition, and the unit remains in this state whilst the spot is scanning white paper. In Fig. 5(*a*) A, B, C, and D represent the four conditions of the two points being scanned. The first of these two points is on the "previous" line, whilst the second is on the "present" line. A zero represents blank paper, whilst a one represents a point on the character. When the top of an intersection is detected, B indicates a falling tendency, C a rising tendency, and D indicates that the tops are on the same level. Thus when one of these conditions appears, the appropriate indication R, F, or S is transmitted to the computer, as shown in Fig. 5(*b*). This signal will subsequently be verified or, if it is countermanded, a detailed scan of the line will be performed.

The table of Fig. 5(*c*) has four rows, each one corresponding to a present condition. The new state can be found in the appropriate row by choosing the column corresponding to the present state. Thus in a typical case shown below:

```
0 0
0 0
0 1
0 1
1 1
```

the present state being initially 9, the unit will remain in this state until the pair of points on the third line have been scanned, when the state becomes 2 and R (rising) is transmitted. After the fourth line the state is 3, and after the next line it becomes state 1; the unit will remain thus if the pattern of two ones continues. It is interesting to note that with points arranged as below:

```
0 0
0 0
0 1
0 1
0 1
```

state 10 is entered at the end of the fifth line, and as a result a detailed scan is performed. All possible cases are covered by the table, and the various situations which arise are discussed below.

If the configuration of character points first encountered is D, the two tops are at the same level, and this should be followed by the D configuration until the bottoms of the intersections appear. With an initial configuration B or C, followed by D, the top description of the intersections is established. The cases of B followed by B, or C followed by C, are also permissible top configurations, provided that these do not subsequently develop into BBB or CCC configurations. In these cases the tops are too far apart for the intersections to belong to the same group. Similar considerations apply to the bottoms of intersections. A special case of note is the configuration ABBA, this representing an intersection of two points on the previous line, which has no corresponding intersection on the present line.

The results for the first few lines of the character of Fig. 3(*a*) are shown in Fig. 6. The beginning of each line is marked * and the end E. If the unit remains in state 9 for the whole scan, the output from the scanner for that line is simply * E. The first line of the character is signalled as * R 22 1 E. This can be seen to consist of a single point, which the coder first assumes is rising, but since there are no points on the previous line this assumption is incorrect and a detailed scan is given.

Both the methods described give some reduction in the amount of data transmitted from the scanner to the computer. The first method gives a reduction of 40%, the second between 10% and 50%, depending on the particular character. The principal advantage of coding

```
* E
* R 22 1 E
* R F 20 8 E
* R F E
* R F E
* R R 17 4 6 4 E
* S R F S E
* S R F S E
```

Fig. 6.—Scanner-coder output for the first seven lines of Fig. 3(*b*)

is the simplification of subsequent processing by the computer.

Interpretation of the Scanner Output by the Computer

The information presented to the computer, using the second method, is shown Fig. 6. This must be interpreted by the computer to produce a simple description of each character. The first stage of the processing is to divide the character into a number of groups. A group is defined as a two-dimensional array of points with no major change in vertical extent on successive lines. A description of these groups, in terms of length, slope, curvature, and their relative positions, forms a convenient basis for a recognition scheme.

By transmitting part of the information relating to any particular character as a continuous sequence of "rising, falling, same" descriptions, the scanner output effectively defines the groups of the character. Detail scans, however, may subdivide these groups. A detail scan may arise from one of four causes:

(i) a change in the number of intersections between adjacent scanning lines,
(ii) excessive rising or falling of group boundaries,
(iii) a splitting of groups at the beginning or ends of groups, and
(iv) the presence of imperfections.

Groups which have been subdivided by detailed scans, and which would normally be joined together by "rising, falling, same" information, must be reassembled.

The next process is to determine how the groups are connected and to find their relative positions.

Group Connections

An important feature in determining the connections between groups is an essential lack of precision. If the connections were to be specified too precisely, small changes in the original character could produce a quite different description.

The most common type of connection between groups is that of a vertical group connected to one or more horizontal groups. The position of the connection is important, and some examples are shown in Fig. 7.

It is possible to describe these positions as top, near top, top and middle, middle, etc. With this large number of divisions, however, it is difficult to design a system which will choose the appropriate description. Furthermore, a large number of classifications will make

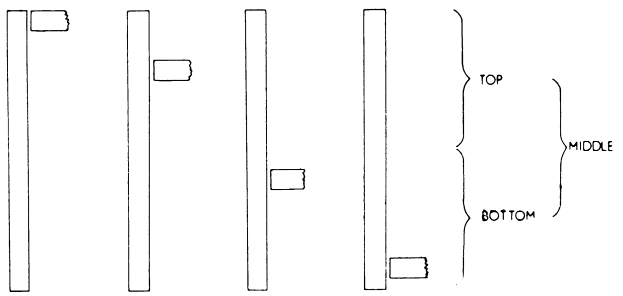

Fig. 7.—Some examples of the position of group connections

recognition difficult. To overcome these difficulties, it is proposed to have only three classifications: top, middle, and bottom, and to use a simple numerical process to specify the position. Thus the position of a connection which is near the top might be specified as Top : .64, Middle : .36, Bottom : 0 the sum of the numbers being normalized to 1. This then gives an "estimate" of the connection being at the top or at the middle. In the subsequent recognition process, some comparison has to be made with a description of a "standard character pattern," and the connections of the latter are also described in terms of these numerical values. The comparison therefore consists of comparing these estimate values and computing correlation coefficients. This and other similarly formed correlation coefficients are used to determine, in the final stage, the probability of a correct recognition decision.

Learning Mechanism

At this juncture it is interesting to note the possibility of incorporating a simple learning mechanism into the recognition process. If, in the recognition of a particular character, the unknown and standard character descriptions do not match perfectly, it is possible to make a very small change to the standard character description. If the change is legitimate then, with subsequent recognition attempts, the standard character description will gradually be adapted to match the presented pattern.

Group Descriptions

The possible forms of group description will now be considered. These fall into three broad classes: horizontal, vertical, and small groups.

The distinction between horizontal and vertical groups is determined by their *aspect ratio*, the ratio of the average width to the average height of the group. Horizontal groups will, in general, be uninterrupted sequences of "rising, falling, same" descriptions. They may, however, be divided by the occurrence of detailed scans.

The characteristics of a horizontal group are determined by the number and position of rising and falling indications. The various categories are: horizontal (flat), sloping (rising or falling), and curve (open up or open down). In cases where there is uncertainty about the category, for example where the top of a group appears to be horizontal (i.e. flat) whilst the bottom appears to be curved, estimate values are determined to indicate the degrees to which the line is to be regarded as flat or curved. These estimate values will again be used in the same way as described previously—to compute the certainty of ultimate recognition.

Vertical groups are in general the vertical lines of a character, or part of a curve. If they are part of a curve it may be possible to complete the curve by a combination of groups.

Recognition Procedure

The recognition process consists of comparing the list of features describing the unknown character with lists of features describing the standard characters. It should be noted at this point that a particular letter or numeral can be printed in more than one way (for example, the closed and open "four") and all such versions must be stored. The number of sets will be quite large, and a preselection system is necessary.

It is not intended to provide a rigid preselection system, but to make use of a "learning mechanism" to choose the best preselection features. The general scheme follows that described previously (Grimsdale *et al.*, 1959) and is outlined below.

The first step is the preparation of a "keyword" for the unknown character. This is a list of the dominant features of the pattern, the dominance of the features being determined by the learning mechanism, which will be described by taking an example. Suppose the unknown letter is d, then it is obvious that this has two dominant features: the vertical line and the almost completed circle. The reason the machine chooses these as dominant features is because they appear in several other symbols. For example, the (single) vertical line occurs in b, d, h, k, l, p, q, F, I, K, L, P, R, T, and the almost complete circle appears in c, d, e, q. Thus it can be seen that the unknown symbol is probably d or q, because these are the only two standard symbols having both the dominant features of the unknown. The next step is to perform a more detailed comparison to decide whether the descriptions of either of the two probable symbols agree sufficiently with the description of the unknown. If a high degree of agreement is obtained then the unknown symbol is identified, but if there is insufficient agreement, further comparisons must be made starting with those symbols having only one of the dominant features.

Conclusions

The present system has been designed to operate with characters produced by normal typewriters, and printed characters of similar size. There is no restriction to any particular type face or size within reason, but the characters are assumed to be nearly vertical.

A previously described system (Grimsdale *et al.*, 1959) in which all the operations were performed by a programmed computer, was very slow and required between 25,000 and 30,000 instructions to be obeyed in order to recognize a single letter. The use of a flying-spot scanner with coding facilities eliminates 75% of this processing. By limiting the present system to the recognition of printed characters it is estimated that an upper limit of 5,000 instructions would have to be obeyed to recognize one character. With an ultra-fast modern digital computer this would give a character-recognition rate corresponding to the reading rate of a photo-electric tape reader. But it must be realized that with time-sharing facilities the actual reading of information from the paper tape only occupies a fraction of the time of the central computer.

Two possible applications envisaged for such a system are the use for literary work, including such processes as translation, indexing, and the preparation of concordances, and as an input mechanism to the computer for programs and data. In considering these, the major problem appears to be that of document handling rather than recognition. Any working system must be justified on economic grounds compared with the inexpensive and versatile facilities of a human reader.

References

GREANIAS, E. C., HOPPEL, C. J., KLOOMOK, M., and OSBORNE, J. S. (1956). "The Design of the Logic for the Recognition of Printed Characters by Simulation," *Proc. I.E.E.*, Vol. 103B, p. 456.

GRIMSDALE, R. L., SUMNER, F. H., TUNIS, C. J., and KILBURN, T. (1959). "A System for the Automatic Recognition of Patterns," *Proc. I.E.E.*, Vol. 106B, p. 210.

ROSENBLATT, F. (1958). "The Perceptron," Cornell Aeronautical Laboratory, Inc., Report No. VG-1196-G-1.

SPRICK, W., and GANZHORN, K. (1959). "Recognition of Numerals by Contour Following," *Proc. I.E.E.*, Vol. 106B, p. 448.

Summary of Discussion

The Chairman, Dr. A. M. Uttley (*Head of Autonomics Division, N.P.L.*), opening the session, said that there was a linking thought between the morning and afternoon papers. If one looked at a collection of truly random dots, one would see that the kinds of patterns they were trying to recognize were quite different. Because of their non-randomness, there were strings of dots forming lines much longer than in the random dot patterns. It was the strings of dots which were considered in the paper by Dr. Clowes and which also formed the underlying thought in Dr. Grimsdale's work at Manchester University.

Mr. E. Newman (*National Physical Laboratory*), making a distinction between the machines described by Dr. Grimsdale and Dr. Clowes, said that in one sense Dr. Grimsdale's machine was the more clever of the two, in that it could distinguish between a complex figure—such as a spiral—and a straight line, whereas Dr. Clowes's could not. On the other hand, Dr. Clowes's machine did not have to be clever in this sense—since it could take no notice of a spiral anyway. But if one knew that one would never have to recognize complex figures, why risk error by using a machine which could detect them if they were there? Mr. Newman felt that a device should respond only to the class of thing it was looking for, and should ignore anything which was not relevant. He asked whether Dr. Grimsdale would comment on this point of view.

Dr. Grimsdale agreed with Mr. Newman that an important part in the design of a character recognition machine is deciding what it should and should not recognize. Both the machine described by Dr. Clowes and the system he described did in fact look for straight lines and curves as basic parts of the character. The latter system could recognize a spiral, if one were present, but it did not actually anticipate the possible presence of a spiral.

In his correlation method, Dr. Clowes had an ingenious approach to the problem of determining whether straight lines or curves are present, which was independent of the position or precise shape of the character, but it seemed difficult to determine how the various straight lines and curves join up.

Dr. Clowes said that techniques were available involving the measurement of the relative positions of different features in the character. This was essentially a cross-correlation procedure.

For a restricted alphabet, it was also possible in principle to detect the whole character in a single auto-correlation process.

The expensive part of Dr. Grimsdale's system was that one had more information than was required and it was necessary to go to considerable lengths to do away with the unwanted information.

Mr. Nadler (*Cie des Machines Bull*) said that it appeared that the tremendous amount of operations connected with character recognition cost so much on digital computers because in essence they are carried out serially. He had always hoped, when reading such work as that of Dr. Grimsdale and his associates, or the other work reported, that the intention was to determine in this way the necessary operations and then to build some special-purpose gadget which would carry out these operations simultaneously and not use a computer at all. In view of Dr. Grimsdale's remarks, this hope of his now appeared to be rather feeble; the system seemed to require the presence of the computer as an integral part of the system. Could Dr. Grimsdale clarify this point?

Dr. Grimsdale thought it was a good idea to construct a small, very fast computer, tailor-made for the pattern recognition program. Whilst this machine would be similar to a normal computer it would only require a limited number of arithmetical and logical facilities and a small working store. The program could be held in a fixed store.

He thought that parallel-recognition devices, whilst having many advantages, would be too expensive if they were to be really versatile.

The Chairman asked if he gathered from that last remark that Dr. Grimsdale did not at the moment use or propose parallel devices.

Dr. Grimsdale said that the answer was "No" because he thought it would cost too much. With, for example, 800 points from a 40 × 20 matrix, the number of connections and interconnections would be very great and it was difficult to see how a versatile parallel device could be made economically. Further, problems of registration appeared when a parallel two-dimensional array was proposed, which greatly increased the complexity.

He did not deny that a parallel device was the ultimate aim, the human eye and brain presumably working in this manner.

Mr. W. E. Norman (*IBM*) said he would like to hear some discussion on the registration of documents.

Was it not time to consider the comparable problems of uniformity of documents, the registration of documents, and type faces for the composition of a document?

Dr. Grimsdale said he thought the eye and brain should be considered as a parallel device because signals from a vast number of retinal cells were processed simultaneously.

The problem of registration was an important one. He would like to distinguish between coarse and fine registration. Coarse registration was the positioning of the character somewhere in the field of view of the recognition machine. Fine registration, which was unnecessary in the scheme he had described, was a precise positioning of the character. He considered the problem of coarse registration to be very difficult, and that no satisfactory practical solution had yet been demonstrated.

Mr. Norman asked if Dr. Grimsdale could answer his question about the material. If they tried to get accurate registration of a quarto-size sheet of paper, was it not likely to shrink and expand or be distorted by fading? I.C.I. was working on coatings. Had Dr. Grimsdale considered these dimensional difficulties?

Dr. Grimsdale did not think that the problem of paper shrinkage was very important. He did not think the coarse registration problem should be tackled by locating characters at, say, the top left-hand and right-hand corners of a page and then expecting all the other characters on the page to be correctly positioned. The method he proposed would be to search for each line of print by the flying-spot scanner.

Mr. H. McG. Ross (*Ferranti Ltd*) wished to draw attention to a point on the economics of the process. It was often suggested that one of the particular values of character recognition was that it provided a cheap means of getting data into a computer, but he wondered whether adequate consideration had been given to the costs resulting from any errors in that data. Erroneous data caused far more trouble in computer systems than in manual data-processing procedures, partly because it was very difficult to write programs to deal with incorrect data, and partly because such errors were likely to hold up the working of the whole installation.

Mention had been made of the problems of reading documents automatically and of distinguishing one character from another. But are there not much more severe difficulties (and hence greater cost) in ensuring that every character is read correctly? And is it not possible that, to attain error-free reading, the characters will have to be very well printed? Would this require much higher quality printers to be attached to cash registers, for example, with consequent increase in cost?

This matter was closely related to the point made earlier by Mr. Newman, that the document reading machine should interpret characters in the same way as did a human being. He felt strongly that this was a most important point which was directly related to the need to minimize errors; this in turn affected the complexity and cost of the document reading system and the cost of the whole character-recognition and data-gathering process.

The Chairman said that the last speaker had put forward some problems which would, he was sure, be solved before long. The Conference had heard during the day of the problem of recognizing a printed fount which the engineer could choose for himself and change to suit his purpose. A chosen fount might, however, be badly printed. He would call that stage 1. Stage 2 was to recognize a given fount which was not chosen by the engineer; this was not beyond the horizon. The further stages of recognizing many founts and of recognizing hand-printed letters and numerals were beyond the present horizon but within the next decade these problems would be solved economically.

Data Processing Society of the Netherlands

Stichting Studiecentrum voor Administratieve Automatisering

The Netherlands Automatic Data Processing Research Centre and The Netherlands Data Processing Society

The Netherlands A.D.P. Research Centre is a non-profit organization, founded 15 July 1958 on the initiative of the College of Economic Sciences of Amsterdam University in co-operation with the Netherlands Institute of Accountants. Its funds are provided by industrial and commercial companies within the Netherlands and by government and other organizations. The Centre's objects are to study the possibilities and limitations of automatic data processing by electronic and other integrating equipment, and to spread the knowledge thus acquired amongst its sponsoring bodies and among students of the co-operating universities. It seeks to co-operate with all persons and establishments at home and abroad, whose objectives concur with its aims.

The studies of this Centre do not involve technical or scientific calculations, but cover automatic data processing in the field of administration, in the fullest sense of the word.

The activities of the Centre imply courses, publications such as *Informatie* and a monthly bulletin *Review of Literature on A.D.P.*, study groups and committees, documentation and a comprehensive library. Subscriptions to the *Review* will be accepted through the British Computer Society from members of the B.C.S. at the special price of £2 per annum. A specimen copy may be obtained from The Assistant Secretary at Finsbury Court, London, E.C.2.

Apart from the centre, the Netherlands Data Processing Society was founded 18 March 1960. Members of the Society are private persons who are interested in automatic data processing. The Society has the specific aim of giving support to the activities of the Centre and to enable private persons to participate in its activities through courses, publications of library. The N.D.P. Society also organizes lectures, courses, discussions, etc., for its members. Furthermore, it aims at increasing international co-operation in the field of data processing. The ties between the Centre and the Society are very close, which appears from the fact that three members of the Board of Governors of the Centre serve also on the Council of the N.D.P. Society.