Parallel Recognition of Idealised Line Characters

By J. R. Ullmann

Autonomics Division, National Physical Laboratory, Teddington, Middlesex, England

With 10 Figures in the Text

(Received September 1, 1964)

Summary. The aim is to design a machine which is able to learn a number of idealised characters and to recognise them, irrespective of their size, position and context on an infinite retina. If the number of characters which such a machine can possibly learn to recognise is astronomical, it is not practical to use separate templates for every possible character. It is more economical to use, instead, templates for various parts, called features, of characters. In recognising a number of characters simultaneously, without scanning, the question arises of how to tell which feature belongs to which character of figure on the retina. In particular, if a given character is not present but all its features are included in nonsense figures simultaneously present on the retina then the machine must not indicate the presence of the given character. The technique which overcomes this difficulty employs overlapping features which must be mutually consistent for recognition. This consistency technique is assessed by comparison with a more conventional technique, and the work is restricted to closed line characters which are not subject to deformations or mutilations.
1. Introduction

The designer of a practical character recognition machine does not generally know in advance exactly what deformations and mutilations the characters will be subject to. He may therefore attempt to design a machine which, having made a record of a number of examples of each character, correctly recognises new examples. The present paper deals with an idealisation of this problem in so far as a machine is required, after learning, to give correct output signals for retinal patterns which have never occurred before; and this idealisation does not reduce the problem to triviality. The purpose of this paper is to put forward a logical idea rather than to propose a complete scheme for practical character recognition.

We describe two logical schemes, scheme A and scheme B, which learn to recognise idealised line characters on an infinite retina. The principal difference is that scheme B can recognise learnt characters when they are present simultaneously on the retina in a new combination which has not itself been learnt, while scheme A cannot. Scheme A employs a conventional "property list" (Minsky and Selfridge, 1961) recognition technique, related to that of Bomba (Bomba, 1959). Scheme A is intended to serve as a yardstick in terms of which scheme B may be assessed.

Scheme B employs a technique which we call a "consistency" technique because it is based on the use of overlapping features which must be mutually consistent for a character to be counted as recognised. The iterative array of logical circuits which this technique requires is not without precedent in the character recognition literature, (e. f. the feature-detecting iterative networks of Unger (Unger, 1958) and Kamentsky (Kamentsky, 1959).

2. The Problem

It is convenient to regard schemes A and B as alternative designs for a machine $M$. Machine $M$ has an infinite binary retina, $S$, and an output array which consists of a row, $T$, of binary logical units. At any time, $T_p$ is the set of units in $T$ which are in state ‘1’. For all $i$, if the $j$th unit in $T$ is in state ‘1’ we say that $t_i \in T_p$, i.e. that $t_i$ belongs to $T_p$.

A binary pattern is an array of 0’s and 1’s and changing any digit makes a different pattern. Patterns are written on $S$ by a generator machine $G$, and these patterns always consist of closed line figures, the lines being rows of 1’s on a background of 0’s. For our purposes, a figure is a collection of straight lines in which every line is met at each end by at least one perpendicular line. Fig. 2, (a), (b), (c) and (d) show some possible figures; but $G$ is designed not to generate examples such as those in Fig. 2 (e), (f), which have loose ends.

The restriction to closed line figures (i.e. no loose ends) is imposed because scheme B depends upon it. The restriction to straight lines and right-angles is not necessary and is imposed only for the sake of simplicity — removing this restriction would not help to put forward the basic idea of this paper.

$G$ is connected (see Fig. 1) to an array $T''$ of binary units, which correspond 1:1 to those in $T$. When the $j$th unit in $T''$ is in state ‘1’ we say that $t_j$ belongs to $T_p$. The comparator $C$ indicates whether or not $T_p = T_p$.

Figures generated by $G$ belong to one of two classes, $P$ and $Q$, defined as follows. Whenever $G$ writes a figure belonging to $P$ on $S$, $G$ also fires a unit in $T''$, and we call such a figure a variant of a character. Figures belonging to $P$ which by translation across the retina, or alteration of the lengths of the lines in a figure without alteration of their proportions, can be made identical, are defined as variants of the same character.

For example if any of the figures in Fig. 3 is a variant of a character, then all the others are variants of the same character. The lengths of the lines are indicated by the numbers written beside them; these are their lengths on $S$ but are not drawn to scale in this and the following diagrams.

For each character there is one corresponding $T''$ unit which is fired by $G$ whenever a variant of this character is written on $S$: this is true whatever other figures are written at the same time on $S$. For example, the $j$th $T''$ unit fires whenever $G$ writes a variant of the $j$th character on $S$. For a figure belonging to $Q$ there is no corresponding $T''$ unit — one may say that figures belonging to $Q$ are nonsense figures.

From a source of noise machine $G$ derives random choices of an arbitrary number, $k$, of characters, and an unlimited number of nonsense figures. For each character $G$ arbitrarily allocates a corresponding $T''$ unit.
After being connected up as in Fig. 1, machine M is subject first to a learning phase and then to an operating phase. During the learning phase, G writes in turn one variant of each of the k randomly generated characters on S and at the same time fires the corresponding T' unit, and also sends a "learn" signal to M. The "learn" signal causes M to store a record of which T' unit is firing and also certain abstractions from the retinal pattern. When this information has been stored for a variant of a character it is convenient to say that the character has been learnt. Note that during the learning phase G never writes more than one figure at once on the retina. The learning phase terminates when M has learnt each of the k characters randomly generated by G.

During the operating phase G writes figures belonging to P and Q, one or more at a time, on S, and sends no "learn" signal to M. Suppose for example that k = 6, so that the states of the T' units can be represented by a six bit pattern. Then corresponding patterns written by G on S and T' might be as in the successive rows of Fig. 4.

For example, in the second row of Fig. 4 the pattern 001000 on T' signifies that, counting from the left, the third T' unit is firing, that T' = {t_3}, and that a variant of the third character is written on S. The pattern 001010 on T' in the fourth row of Fig. 4 signifies that corresponding to the pattern on S in the fourth row, T' = {t_3, t_5}. This paper is concerned with the problem of designing M so that during the operating phase, for a given pattern on S, T' = T_P. Thus after learning, M is required to predict what T' units are fired by G when G writes a pattern on S which has never been written on S before. We stipulate that the design of M is to be as economical in hardware and fast operating as possible.

Since, in this theoretical work, the retina is infinite, a scanning technique which looked for variants of characters in successive small regions of the retina would take an infinitely long time and thus not be acceptable.

In formulating the design problem for M it is not essential to describe a machine G; but in terms of G it is very easy to see how figures belonging to P differ from figures belonging to Q, and this is why we have introduced G.

3. Scheme A

In a retinal pattern, a line-pair is a pair of perpendicular lines which meet at a point. Thus a line-pair comprises two rows of 1's and we only say that the line-pair is present on the retina if all of these 1's are present. The ratio of the lengths of the two lines in a line-pair is the proportion of the line-pair. In each of the sets L_1, L_2, ..., L_i, ..., of line-pairs, all the line-pairs are of the same proportion and orientation though of different retinal position and absolute size. Thus for every line-pair proportion and orientation we define a corresponding line-pair set L_i, where i lies between unity and infinity.

At a given time, V is the set of line-pairs in the retinal pattern which are not completely overlapped by any other line-pair. Suppose for example that the only figure on the retina is that shown in Fig. 5.

For this retinal pattern, V is the set of line-pairs (AB, BC), (BC, CD), (CD, DA), (DA, AB), (AE, EG), (EG, GF), (GF, FA), (FD, FG), (GE, EB). But for example the line-pairs (AE, AF), (AE, AD), (AF, AB) do not belong to V because they are completely overlapped by (AB, AD).

When scheme A learns a variant of a character it simply stores a record of the value of j for which t_j ∈ T_P, and for what values of i L_i ∩ V = A, where A is the empty set. For all j from 1 to k for the j-th character we define J_j as the set of line-pair sets L_i such that L_i ∩ V = A.

Fig. 4. Examples of Generator outputs during operating phase

During the operating phase the condition for t_j ∈ T_P in scheme A is that L_i ∩ V = A for all L_i such that L_i ∩ J_j and L_i ∩ V = A for all L_i such that L_i ∩ J_j. In the following pages we write for example "for all L_i ∩ J_j" to mean "for all L_i such that L_i ∩ J_j" because this occurs frequently.

It is important that in scheme A, when a variant of the j-th character is present on the retina and at the same time any line-pair which belongs to V belongs also to L_i such that L_i ∩ J_j, then t_j ∈ T_P. Scheme B is of interest mainly because it does not suffer from this defect. It sometimes happens in scheme A that t_j ∈ T_P when no variant of the j-th character but only a nonsense figure is present on the retina, and we compare below in a later section the performance of schemes A and B in this respect.

4. Scheme B

We say that a line l_2 is common to two adjacent line pairs (l_1, l_2), (l_1, l_3) if one end of l_2 is met by one end of the perpendicular line l_4 and the other end of l_2 is met by one end of the perpendicular line l_5, e.g. in Fig. 5, the line AB is common to the adjacent line-pairs (DA, AB), (AB, BC). In a closed line figure, every line which is not completely overlapped by any other line is common to at least two adjacent line-pairs.
We define a set $W_i$ by saying that a line-pair belongs to $W_i$ when and only when $\lambda$ is present on the retina and both of the lines in $\lambda$ are common to at least two adjacent line-pairs which are present on the retina.

When scheme B learns a variant of a character it stores a record of the value of $j$ for which $t_j \in T_F$, and for what values of $i$ $L_i \subseteq W_i \neq \emptyset$. For all $j$, from 1 to $k$, for the $j$th character we define $H_j$ as the set of line-pair sets $L_i$ such that $L_i \subseteq W_j \neq \emptyset$.

During the operating phase scheme B computes which line-pairs belong to successive sets in the series of sets $W_0, W_1, W_2, \ldots$ which are defined as follows. For a given retinal pattern we define $K_\alpha$ as the union of the sets $H_j$ for all $j$ such that in every $L_i \subseteq H_j$ at least one line-pair is present in the retinal pattern. A line-pair $\lambda$ belongs to $W_0$ when and only when $\lambda \subseteq W_0$ and also $\lambda \subseteq L_i$ such that $L_i \subseteq K_0$. $K_1$ is the union of the sets $H_j$ for all $j$ such that $W_0 \cap L_i \neq \emptyset$ for all $L_i \subseteq H_j$. A line-pair $\lambda$ belongs to $W_1$ when and only when $\lambda \subseteq W_0$ and both of the lines in $\lambda$ are common to at least two line-pairs which belong to $W_0$, and also $\lambda \subseteq L_i$ such that $L_i \subseteq K_1$. $K_2$ is the union of the sets $H_j$ for all $j$ such that $W_0 \cap L_i \neq \emptyset$ for all $L_i \subseteq H_j$. A line-pair $\lambda$ belongs to $W_2$ when and only when $\lambda \subseteq W_0$ and both of the lines in $\lambda$ are common to at least two line-pairs which belong to $W_1$, and also $\lambda \subseteq L_i$ such that $L_i \subseteq K_2$.

We define $W_\infty$ as the first member of the series $W_0, W_1, W_2, W_3, \ldots$ such that $W_\infty = W_{(w+1)}$. In scheme B the condition for $t_j \in T_F$ is $L_i \subseteq W_\infty \neq \emptyset$ for all $L_i \subseteq H_j$.

In the following section we compare the performance of schemes A and B, and in a later section we give a logical design for scheme B which does not employ very many more logical units than that for scheme A. (These designs have not been built.)

5. Comparison of Performance for Schemes A and B

(a) Simple Line Figures. We say that a simple line figure is a figure in which no line is common to more than two adjacent line-pairs. Thus Fig. 2 (a), (c) are examples of simple line figures, and Fig. 2 (b), (d) are not. If $G$ is restricted to generate only simple line figures, it is then much easier to compare the performance of schemes A and B and the loss of generality does not matter for the purpose of the present paper. So in the present section (i.e. section 5) we consider only simple line figures.

In a simple line figure the two lines in any line-pair belonging to $V$ are common to two adjacent line-pairs which belong to $V$. Therefore any line-pair belonging to $V$ belongs to $W_i$. And no line-pair belonging to $W_i$ is completely overlapped by any other line-pair. Thus any line-pair belonging to $W_i$ belongs to $V$, and so we see that for simple line figures $W_i \subseteq V$, and in this case $H_i = J_i$ for all $j$ from 1 to $k$.

(b) Only one Learnt Character. It is convenient first to consider the case where during the learning phase altogether only one character is learnt. When a variant of this character is present alone on the retina, $H_i$ is the set of all line-pair sets $L_i$ such that $L_i \subseteq V \neq \emptyset$.

A necessary but insufficient condition for $t_i \in T_F$ in schemes A and B is $L_i \subseteq V \neq \emptyset$ for all $L_i \subseteq H_1$. Scheme A requires also that $L_i \subseteq V \neq \emptyset$ for no $L_i \subseteq H_1$. And in scheme B $t_i \in T_F$ only if a figure on the retina contains no line-pairs belonging to $L_i$ such that $L_i \subseteq H_1$. When only one figure is present on the retina and only one character has been learnt, we see that the performance of scheme B is identical to that of scheme A; i.e. for a given retinal pattern $T_F$ is the same for both schemes.

Suppose for example that the one character learnt by schemes A and B is that shown in Fig. 6 (a). Then if any of the figures shown in Fig. 6 (b), (c), (d) are written as the only figures on the retina, $t_i \in T_F$ in both schemes: this is because the set of line-pair sets for each of the four figures is the same. These are examples of erroneous recognition, but it is easy to construct an unlimited number of figures which would not be erroneously recognized.

(c) Many Learnt Characters. When schemes A and B have learnt many characters, scheme B may make erroneous recognitions which would not be made by scheme A. For example, suppose the $i$th and $j$th characters have been learnt and that the retinal pattern includes a figure which contains a line-pair in every $L_i$ such that $L_i \subseteq (H_i \cup H_j)$, and no line-pair in any $L_i$ such that $L_i \subseteq (H_i \cup H_j)$. In this case in scheme A, $t_i \in T_F$ and $t_j \in T_F$, while in scheme B (erroneously), $t_i \in T_F$ and $t_j \in T_F$. On the other hand when variants of the $i$th and $j$th characters are present simultaneously on the retina, in scheme B (correctly) $t_i \in T_F$ and $t_j \in T_F$, while in scheme A, $t_i \in T_F$ and $t_j \in T_F$ (unless $H_i \subseteq H_j$ or $H_j \subseteq H_i$). In scheme B if a variant of the $i$th character is present on the retina then $t_j \in T_F$ whatever nonsense figures and variants of characters are present on the retina at the same time, whereas this is not true in scheme A. If no variant of the $i$th character is present as a whole on the retina but all its features are included in figures which are present simultaneously on the retina, in scheme B $t_j \in T_F$ only if all of these figures are also recognized.

We now give a less trivial example of the operation of scheme B. Suppose the learnt character variants are the figures shown in Fig. 7(a), and that the
retinal pattern is shown in Fig. 7(b). From the definition of \( W_0 \) we see that \( W_0 \) is the set of line-pairs not completely overlapped by any other line-pair in Fig. 7(c). There are two line-pairs which are present in Fig. 7(b) but not in Fig. 7(c) because they are absent from Fig. 7(a). Similarly, Fig. 7(d), (e), (f) correspond to \( W_1, W_2, W_3 \), respectively. Thus \( W_n = W_0 \) and the nonsense figures in the retinal pattern are not recognized.

<table>
<thead>
<tr>
<th>Variants of Learnt Characters</th>
<th>Variants of Learnt Characters</th>
<th>Variants of Learnt Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_0 )</td>
<td>( W_1 )</td>
<td>( W_2 )</td>
</tr>
<tr>
<td>( W_3 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7. \( W \) sets at successive stages of iteration of scheme B

- Threshold x unit
- 'And' unit
- 'inhibit' gate - fires if \( a \& \sim b \)
- Time delay
- Inverter
- Threshold x trigger unit: once fired, continues to fire, irrespective of input.

Fig. 8. Key to logical notation

6. Logical Design

(a) Notation. It is not practical to show complete diagrams of the highly repetitive schemes, and instead incomplete diagrams are accompanied by written specifications. Figs. 9 and 10 show only very few logical units and connections, and a key to the notation used is given in Fig. 8.

Let the line-pairs in \( L_{ij} \) be \( \lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{ij}, \ldots \)

For every \( \lambda_{ij} \) there is a corresponding logical unit, and a few of these are shown in Figs. 9 and 10 and marked e.g. \( \lambda_{32}, \lambda_{68} \). A \( \lambda_{ij} \) unit only fires if the corresponding line-pair is present on the retina — otherwise it does not receive the signal marked "from retina". For simplicity the retina is not shown.

Fig. 9. Part of logical design of scheme A

For each \( \lambda_{ij} \) unit there are several auxiliary units in Fig. 10 and all of these with a similar function are marked similarly, e.g. \( a_1, a_2, a_3 \), etc., and this applied for other frequently repeated auxiliary logical units.
When the threshold of a threshold unit is equal to the total number of inputs the unit is of course effectively an "and" unit. We have only introduced units marked "k" because of the embarrassing problem of saying how many inputs there are to the units $T_1, T_2, ...$ in the array $T$.

(b) Scheme A. Corresponding to each $\lambda_{ij}$ unit is an "or" unit called an $a_i$ unit. When a $\lambda_{ij}$ unit fires it sends signals to the $a_i$ units for all line-pairs completely overlapped by $\lambda_{ij}$. Thus a $\lambda_{ij}$ unit only persists in firing if $\lambda_{ij} \in V$. The units $L_1, L_2, ...$ correspond to the line-pair sets $L_1, L_2, ...$, and the unit $L_i$ is an "or" unit connected to the $\lambda_{ij}$ units for all $j$. Thus the $L_i$ unit persists in firing if $L_i \cap V = \Lambda$.

The units connected between $T_j, T_j'$ and $L_1, L_2, ...$ units ensure that when the $T_j$ unit fires, and the "learn" signal is received, a short time later $a_i$ trigger units corresponding to $L_i$ such that $L_i \cap V = \Lambda$ fire. Subsequently the $T_j$ unit fires only if $L_j \cap V = \Lambda$ for all $L_i \in J_j$ and $L_j \cap V = \Lambda$ for all $L_i \in J_j$. The time delay in the "learn" signal is to prevent trigger units being fired until the inhibition of line-pair units by line-pair units for completely overlapping line-pairs is completed.

(c) Scheme B. In scheme B, corresponding to each $\lambda_{ij}$ unit there is an $a_4$, an $a_5$, and two $a_6$ units. Whenever the retinal pattern is changed a "set up" signal is received by all the $a_i$ units, so that $\lambda_{ij}$ units can fire for all line-pairs in the retinal pattern. Both during the learning and operating phase the "set up" signal is then switched off, so for an $\lambda_{ij}$ unit to persist in firing, the corresponding $a_6$ unit must be firing.

During the learning phase the "learn" signal fires all the $a_i$ units, and so for an $a_6$ unit to fire it is only necessary that the two $a_6$ units connected to it fire. One $a_6$ unit corresponds to each line in the line-pair and is connected to all line-pair units for adjacent line-pairs to which the corresponding line is common. Thus only line-pair units such that $\lambda_{ij} \in W_e$ persist in firing.

The units connected between $T_j, T_j'$ and $L_1, L_2, ...$ units ensure that when the $T_j'$ unit fires and the "learn" signal is received, a short time later $a_i$ trigger units corresponding to $L_i$ such that $L_i \cap W_e = \Lambda$ fire and $a_5$ trigger units corresponding to $L_i$ such that $L_i \cap W_e = \Lambda$ fire. The time delay in the "learn" signal is to ensure that only line-pair units such that $\lambda_{ij} \in W_e$ are firing when the trigger units are fired.

During the operating phase an $a_7$ unit only fires if the corresponding line-pair set unit $L_i$ belongs to $H_j$ such that the units $L_i$ are firing for all $i$ such that $L_i \in H_j$. After the "set-up" signal is switched off and after iteration terminates, the remaining firing line-pair units correspond to line-pairs belonging to $W_e$. The unit $T_j$ only persists in firing if $L_i \cap W_e = \Lambda$ for all $L_i \in H_j$. To see clearly the computation of membership of $W_e, W_1, W_2, ...$ etc., we could by trivial modification make the design purely synchronous in operation. But the asynchronous design shown appears to reach the same $W_e$, although we have not found how to prove this formally.

7. Comments

(a) Restriction to right-angles. To deal with non-rectangular line figures, one may introduce line-pair sets for all pairs of meeting lines — one such set for every angle between lines, proportion and orientation. To deal with curved figures one may introduce curve-pairs, in which each of the two curves is an arc of a circle, and curve-pair sets: in any curve-pair set the ratio of radii of curvature of the two arcs and the ratio of the arc lengths and the angle between the tangents at the point of intersection of the curves are constant. The curve-pair sets can be treated logically like the line-pair sets in schemes A and B.

(b) Short Lines. Lines on the retina, in the foregoing discussion, are rows of 1's. Consider a square in which each side comprises only three 1's: it is not possible to represent on the retina a square with sides only half this length. We have stipulated that the retina is infinite mainly to make this difficulty unimportant. (Another reason for using an infinite retina is to eliminate the plausibility of scanning.)

(c) Learning one Character at once. If, during the learning phase, many variants of characters were usually present on the retina, it would be possible to find by Bayesian inference which $T'$ unit was associated with each. But this would not help to put across the basic idea of this paper.

(d) Loose ends. If the retina were large but finite, the number of line-pair sets in schemes A and B would be very great. Apart from this economy difficulty, scheme B suffers from the restriction that variants of characters must include no loose ends. It is hoped to overcome this difficulty in future work and in fact a series of more general problems have been formulated elsewhere (ULLMANN, 1964).

Acknowledgements. The work described above has been carried out as part of the research programme of the National Physical Laboratory and this paper is published by permission of the Director of the Laboratory.


Anschrift: J. R. Ullmann
Autonomies Division
National Physical Laboratory
Teddington, Middlesex
England

Druck der Universitätsdruckerei F. Stärzler AG., Würzburg