

Linear Skeletons from Square Cupboards

C. Judith Hilditch

Medical Research Council,
Clinical and Population Cytogenetics
Research Unit

INTRODUCTION

The problem of reducing the line-like elements of a digitized picture to idealized thin lines is of general interest in pattern recognition. As early as 1957 the idea of obtaining a thin-line representation of certain patterns was suggested (Kirsch *et al.* 1957); recently McCormick (1963) and Narasimhan (1964) have described computer programs for doing this (for use in particular on bubble chamber photographs), and similar work has been done in character recognition, for example by Deutsch (1967). Blum (1964) has put forward an approach for dealing with more general shapes. In this the boundary of a shape is considered as being the source of a wavefront. The points at which wavefronts originating at different parts of the boundary first meet form a 'skeleton' which, with a function giving the time taken for the wavefront to reach each point of the skeleton, completely defines the original shape. Programs for generating this skeleton for digitized pictures have been described by Rosenfeld and Pfaltz (1966), and also by Philbrick (1966). A technique which we at the MRC have implemented for reducing line-like shapes to idealized thin lines is similar to the last of these, in that it involves working inwards from the boundary of the shape under consideration, removing all points except those which are considered part of the skeleton. It is also similar to a stripping routine described by Izzo and Coles (1962) and Preston (1961) in a rather different context.

AIMS

The problem with which we have been primarily concerned is the automatic analysis of chromosome spreads, a typical example of which is shown in figure 1.

The aim of the algorithm described in this paper is to reduce such a picture to a 'skeleton' of idealized thin lines which satisfy not only the obvious

requirement that they should lie approximately along the centre of each line-like part of the picture but also satisfy, as nearly as is possible in a discrete space, the definition of a line as 'that which has length without breadth' while still retaining the connectivity of the original.

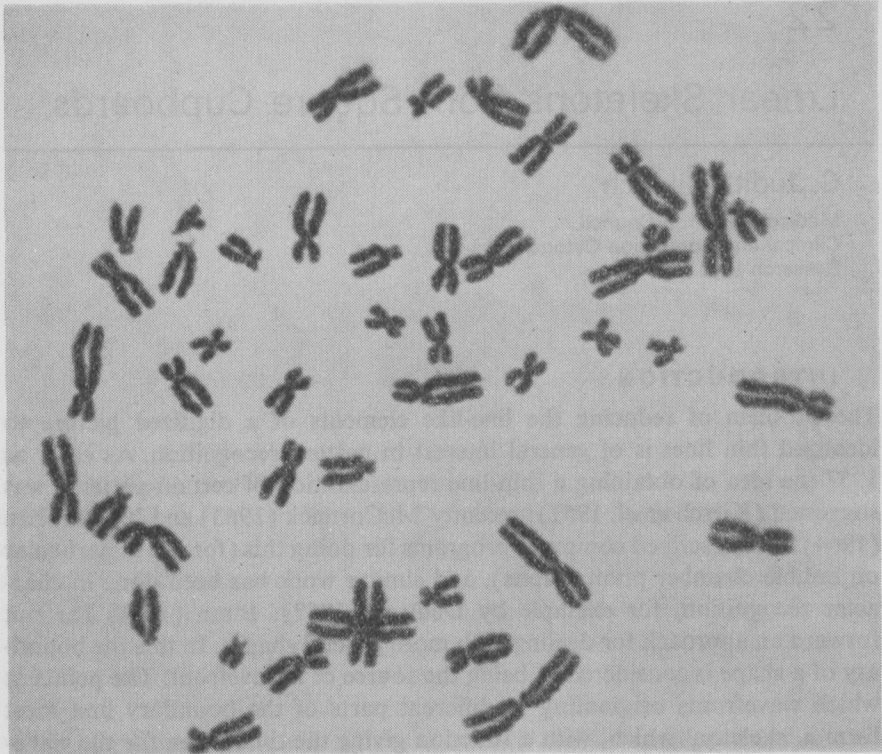


Figure 1. A typical human chromosome spread

THE PICTURE

Before describing how this has been done it is necessary to describe more fully the form which the picture takes. Let J be the set of all pairs of integers. Then a picture is defined as a function f on a subset P of J . In particular P is usually rectangular, i.e., $P = \{(i, j) \mid 1 \leq i < m, 1 \leq j < n\}$, and the elements of P can be regarded as the elements of a matrix, or as points with integer coordinates in the Euclidean plane. Usually the function f will initially take values such that each element or point represents in some way the darkness of that part of the picture. Subsequent transformations of the picture will, however, alter the values of the points and hence of course their significance.

The work described here is concerned not with whole pictures, which in our case consist of about a quarter of a million points, but with sub-pictures, that is restrictions of f to a subset Q of P ; formally we denote this by $f|_Q$. Q may for example be the set of all points of P with value greater than some threshold

or a connected component of such points. The problem which we are considering here is that of producing a skeleton for such a sub-picture which satisfies all the requirements for a skeleton listed above.

CONNECTIVITY

At this stage it is necessary to define what we mean by connectivity in the case of a digitized picture. Each point of a picture is considered to have eight neighbours, these being the eight points which differ from it by one in either or both co-ordinates. For convenience these neighbours are numbered n_1, n_2, \dots, n_8 as shown in figure 2. The numbers are taken modulo eight so that, for example, neighbour nine is the same point as neighbour one.

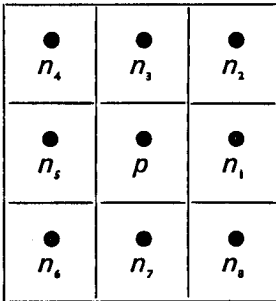


Figure 2. The eight neighbours n_1 to n_8 of a point p

Following Rosenfeld and Pfaltz (1966) we say that a subset of a digitized picture is *connected* if for any two points p and q of the subset there exists a sequence of points

$$p = p_0, p_1, p_2, p_3, \dots, p_{n-1}, p_n = q,$$

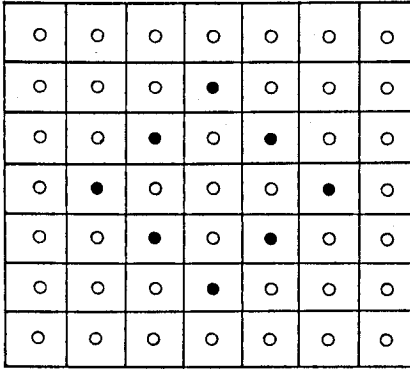
such that p_i is a neighbour of p_{i-1} , $1 \leq i \leq n$. This corresponds with the usual concept of connectivity in the Euclidean plane, if one considers the area defined by taking each point of the subset as the centre of a closed unit square.

However, as pointed out by Rosenfeld and Pfaltz (1966), if one then considers whether the complementary subset of the picture is connected, paradoxical situations can arise. For example, both the set of black points and the set of white points in figure 3a are connected, as is illustrated in figure 3b. This paradox is not resolved by considering a point as having only four neighbours, namely n_1, n_3, n_5 and n_7 . In this case we simply have the reverse situation that neither the set of black points nor the set of white points is connected.

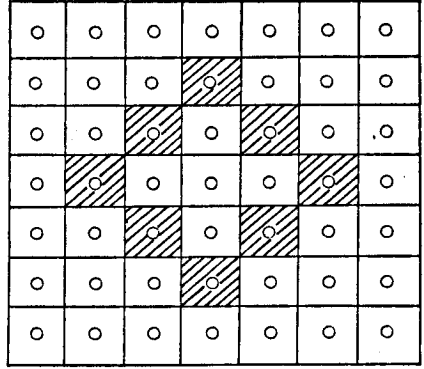
The paradox arises because the connectivity of a subset of the picture and the connectivity of its complementary subset are both being considered at the same time. In practice one is usually interested only in the connectivity of a

PATTERN RECOGNITION

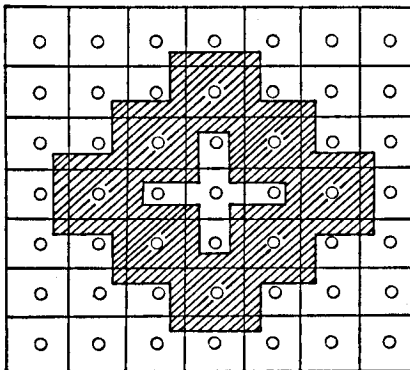
given subset, say the black points in figure 3a. In this case confusion can be avoided by considering the connectivity of the area defined by this subset in the Euclidean plane and the connectivity of the complementary area, rather than the area defined by the complementary subset.



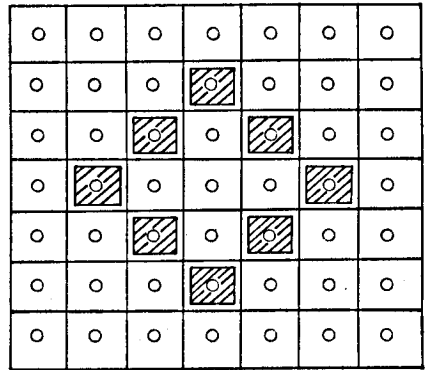
3a



3b



3c



3d

Figure 3. A paradoxical situation in which both the set of black points and the set of white points are connected. For a full description see text

This is made clearer by taking each point of the subset as the centre of a square with side slightly greater than the distance between adjacent points; the area defined by the set of black points is then given by the shaded part of figure 3c and clearly is connected, whereas its complement clearly is not. On the other hand the area defined by the set of white points (given by the unshaded part of figure 3d) is also connected and its complement is not. We choose to

consider each point of a given subset of the points of a picture as representing a square of slightly greater than unit area, in this way, and define the subset as connected if the area which it represents is connected in the usual sense in the Euclidean plane.

This is equivalent to defining connectivity on the neighbourhood basis with the points of the given subset each having eight neighbours n_1 to n_8 , but allowing the points of the complementary subset to have only four neighbours n_1, n_3, n_5 and n_7 .

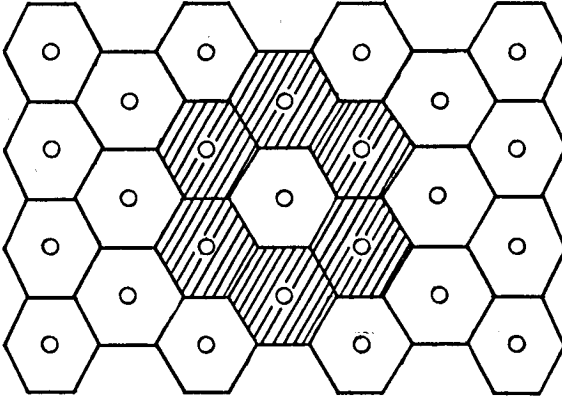


Figure 4. In the hexagonal case each point has six neighbours, and the paradoxical situations cannot arise

It may be interesting to note an alternative approach which avoids this difficulty altogether. This requires alternate lines of the picture to be displaced by half a unit, and each point taken to have six neighbours, as shown in figure 4. This is equivalent to considering each point as the centre of a hexagon, and in this case it is clear that the paradoxical situations do not arise.

REQUIREMENTS TO BE MET BY THE SKELETON

We are now in a position to consider in greater detail the requirements to be met by the skeleton and how these are satisfied. The requirements are:

1. Thinness

First, the skeleton is required to consist of thin lines. This is achieved simply by eroding away the subset by successively removing points which lie on its edge until all that remains is lines which are one point wide. The algorithm is intended for use on a general purpose digital computer in which the points must be treated sequentially and not in parallel. However, if the points are removed sequentially – a point being removed from the subset as soon as it has been found to lie on the edge of the subset – then subsequent nearby points will appear to lie on the edge and will be deleted in their turn. The

PATTERN RECOGNITION

result is that the skeleton will tend to be biased towards one side or the other of the original subset, depending on the order in which the points are taken.

2. Position

The second condition is that the idealized thin lines should lie along the centres of the line-like parts of the subset. To achieve this the process of removing points on the edge of the subset is made essentially parallel. A point is removed only if it lies on the edge of the initial subset, regardless of which other points have been removed. This means that several passes through the points of the subset are required. At each pass the outer layer of points is removed to give a 'thinner' subset for the next pass.

However, once this thinning process has reached a stage where some or all of the subset has been reduced to thin lines of points, these lines must not be thinned away to nothing. This is ensured by the next two conditions.

3. Connectivity

The third requirement is that the process should not alter the connectivity of the subset. This is achieved by testing each point that is to be removed to find whether its removal will alter connectivity. If it does, then the point is retained even though it lies on the edge.

However, if this is performed in parallel, a difficulty arises in the case where the subset has been reduced to a line that is two points wide, such as is shown in figure 5.

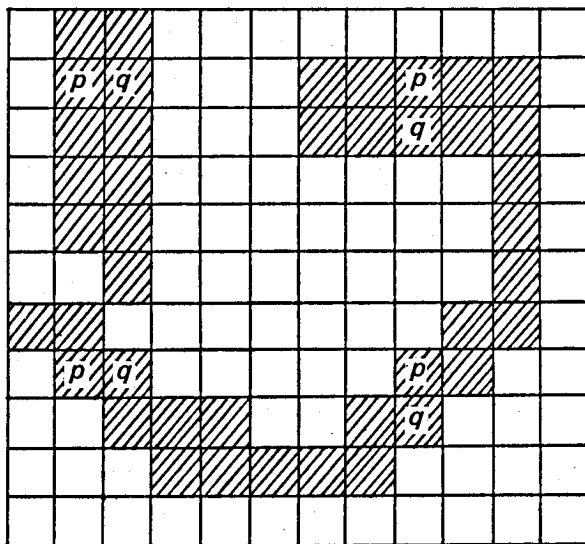


Figure 5. In each of the cases shown above neither the deletion of point p nor the deletion of point q will disconnect the area of shaded points; however, in each case the deletion of both p and q will,

In this case the removal of point p would not alter the connectivity of the subset, neither would the removal of point q . Unfortunately, the removal of both together most certainly will. This difficulty is overcome by taking advantage of the fact that the points are really being dealt with sequentially. Suppose, for example, that point p in figure 5 is tested before point q . Since the removal of p does not alter the connectivity of the subset, this point is removed. Subsequently, when point q is tested, the fact that p has been removed can be taken into consideration and the point q retained. This means that for a point to be removed, not only must its removal alone not alter connectivity, but if any one of its neighbours has been removed, then the removal of the two points together must not alter connectivity either.

4. Stability

Finally, it is necessary that as soon as a satisfactory skeleton, or part of a skeleton, is obtained, this should be stable and not be eroded away by subsequent passes. For most points of the skeleton this is ensured by the connectivity condition. However, a point which lies at the tip of a thin line can be removed without altering connectivity. An additional condition is therefore required to the effect that such a point may not be removed, otherwise the lines of the skeleton will gradually become shorter and shorter. Also, in order that an approximately circular subset should not disappear entirely, we need a condition which ensures that the last remaining point of such a subset is not removed.

THE ALGORITHMS

To meet these requirements the various algorithms that we have implemented each require several passes. At the start of a pass a subset Q of some sub-picture $f_0 | P$ is given¹; at the end of the pass a subset Q' of Q , which is one layer 'thinner', has been defined. Initially, Q is the subset to be reduced to idealized thin lines; the resulting Q' then becomes Q for the second pass, and so on; eventually a Q' is obtained which is a skeleton of the required form. Thereafter, no further points can be removed, and this fact is used to recognize when no more passes are required.

We have found it convenient to assume that the subset Q is defined as those points of P for which f_0 takes one of a given set of values, I , say, and that all other points of P take one of a set of values N . In particular, the picture usually takes the form of either a characteristic function with points in Q having value one and all other points zero, or f_0 takes values on the non-negative integers, points in Q having values greater than some threshold and other points less. However, this assumption causes no loss in generality since it can be extended to a subset Q consisting of points having *any* given property simply by making f_0 the characteristic function for these points.

¹ Since the algorithm requires the values of the neighbours of all points of Q to be defined, P should contain at least all neighbours of points of Q .

Throughout a pass through the points of the picture a record of the initial subset Q must be maintained. However, it has been found convenient, rather than retaining the original picture $f_0 | P$ as well as constructing point by point a new picture $f_1 | P$ which will define Q' , to reserve a third set of values R for points which have been removed ($R \cap I = \phi$ and $R \cap N = \phi$). Only one picture is then required, and as each point $q \in P$ is tested the picture is changed by setting that point to a new value $f_1(q)$, where $f_1(q) \in R$ if the point is removed, or, if it is not, $f_1(q) \in I$ or $f_1(q) \in N$, depending whether the point was in Q in the first place or not.

At the stage when point p of the picture has just been set the value of any other point q is denoted by $f^{(p)}(q)$ (or $f(q)$ for short), where $f^{(p)}(q) = f_0(q)$, if q follows p , and $f^{(p)}(q) = f_1(q)$, if q precedes or is p . At this stage we have a *partially thinned picture* $f^{(p)} | P$; the initial subset Q is given by

$$Q = \{q \in P \mid f^{(p)}(q) \in I \cup R\}$$

and the *partially thinned subset* $Q^{(p)}$ is defined by

$$Q^{(p)} = \{q \in P \mid f^{(p)}(q) \in I\}$$

When all points of the picture have been tested

$$f^{(p)} = f_1 \text{ and } Q^{(p)} = Q' = \{q \in P \mid f_1(q) \in I\}$$

The picture is then ready for the next pass to begin, except that all points with values in R must first be reset to have values in N , or R and N must be redefined to the same effect.

It is sometimes useful, also, to reserve a subset U of I for the values of points which for some reason may not be removed at the current pass.

The crucial part of the algorithm is the determination of whether or not a point should be removed. This is independent of the choice of sets I, N, R and U and is described in full below.

CONDITIONS FOR THE REMOVAL OF A POINT

A point p will be removed, that is, it will be set to a new value $f_1(p)$ in R , if and only if it satisfies all the following conditions:

1. It belongs to Q and its removal is allowed, i.e.,

$$f(p) \in I - U;$$

2. It lies on the edge of Q , that is, at least one of its axially adjacent neighbours n_1, n_3, n_5 and n_7 does not belong to Q ; i.e., if

$$\mu(p) = a_1 + a_3 + a_5 + a_7 \text{ (where } a_i = 1 \text{ if } f(n_i) \in N, a_i = 0 \text{ otherwise)}$$

then we have the edge condition

$$\mu(p) \geq 1;$$

3. It is not the tip of a thin line, that is, it has more than one neighbour which belongs to Q , i.e., if

$$v(p) = \sum_{i=1}^8 (1 - a_i) \quad (a_i \text{ defined as above})$$

then we have the 'not tip' condition

$$v(p) \geq 2$$

4. It is not the last remaining point of small 'circular' subset, that is, it has at least one neighbour in Q which has not been removed; i.e., if

$$w(p) = \sum_{i=1}^8 c_i \quad (\text{where } c_i = 1 \text{ if } f(n_i) \in I, c_i = 0 \text{ otherwise})$$

then we have the condition

$$w(p) \geq 1$$

5. Its removal does not alter connectivity, that is, the area of the Euclidean plane defined by the subset Q can be continuously deformed to give the area defined by the subset $Q - \{p\}$. This will be so if and only if the set comprising those neighbours of p which are in Q consists of one connected component. (The case where the removal of p would produce a 'hole' is prevented by condition 2 above).

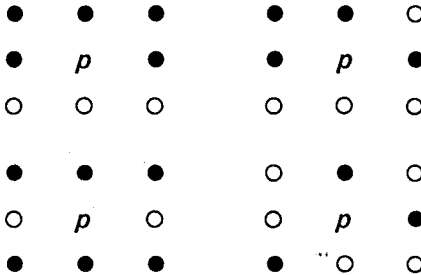


Figure 6. Crossing number. The crossing number at point p with respect to the set of black points is one in the two upper cases and two in the two lower cases. Note in the two right-hand cases that neighbour one is connected to neighbour three, so the fact that neighbour two is white does not cause an increase in crossing number.

The number of components is calculated by considering the number of times a 'bug' taking a walk around p by way of its neighbours would have to cross from outside to inside the subset. This number is called the *crossing number* of p with respect to the subset, and is generally equal to the number of components, but would be zero in the case where the 'bug' did not need to leave the subset at all. Note that to conform with our definition of connectivity the 'bug' must go from one axially adjacent neighbour of p to the next axially adjacent one, by-passing the diagonal if by so doing it avoids leaving the subset. Some examples are given in figure 6.

PATTERN RECOGNITION

This crossing number $X(p)$ is calculated as follows:

$$X(p) = \sum_{i=1}^4 b_i \text{ (where } b_i = 1 \text{ if } f(n_{2i-1}) \in N \text{ and either } f(n_{2i}) \in I \cup R \text{ or } f(n_{2i+1}) \in I \cup R \text{ and } b_i = 0 \text{ otherwise).}$$

The connectivity condition is then given by

$$X(p) = 1$$

6. Its removal in conjunction with any one of its neighbours that has been removed does not alter the connectivity of Q ; that is, if neighbour n_i of p has been removed, i.e., $f(n_i) \in R$, then if the value of this neighbour is temporarily altered so that $f(n_i) \in N$ and it thus appears that n_i is not in Q , then the new crossing number at p , $X_i(p)$, say, is still one. This gives us the additional 'two thick line' condition

$$f(n_i) \notin R \text{ or } X_i(p) = 1 \text{ (} i = 1, \dots, 8 \text{)}$$

In general, the sequence in which the points are being tested is known, and it is therefore only necessary to test this last condition for those neighbours which precede p , since f can only take a value in R for a point *after* it has been tested. For example, if we start at the top of the picture and work from left to right along each line, it is only necessary to test the condition for $i=2,3,4$ and 5 . In addition, if i is even and p satisfies all the preceding conditions, then the condition that either $f(n_i) \notin R$ or $X_i(p) = 1$ is automatically satisfied and need not be tested. For, if $f(n_i) \in R$, then, since p has crossing number equal to one and has more than one neighbour in Q , and n_i belongs to Q , it follows that either n_{i-1} belongs to Q , or n_{i+1} belongs to Q , or both. If both belong to Q then deleting neighbour i does not alter the crossing number at p since the 'bug' can in this case 'cut the corner'. If on the other hand only one of these is in Q then deleting neighbour i cannot possibly alter the crossing number, so in each case we have $X_i(p) = 1$. It is thus only necessary to test this last condition for $i=3$ and $i=5$, i.e.,

$$f(n_i) \notin R \text{ or } X_i(p) = 1, \text{ for } i=3 \text{ and } i=5$$

APPLICATION TO BINARY PICTURES

The algorithm was initially implemented for use on pictures taking the form of a characteristic function. In this case points belonging to the subset Q have value one, all other points have value zero, and when a point is removed its value is set to minus one, i.e., the algorithm is applied with

$$\begin{aligned} I &= \{1\} \\ N &= \{0\} \\ \text{and } R &= \{-1\} \end{aligned}$$

Between each pass it is necessary to set those points which have been removed and so have value -1 to zero ready for the next pass.

This can be made more efficient in the following way. Consider a point p of the picture and suppose that at one pass neither this point nor any of its neighbours is removed. Then this point cannot possibly be removed at the next pass since its situation with regard to the conditions for removal has in no way changed. If all such points are set to two, therefore, and we define

$$U = \{2\}$$

this will avoid the unnecessary testing of the neighbours, crossing number, etc., of these points at the next pass. This is achieved by setting the value of *all* points of Q which are not removed to two; then at the end of the pass, when all points which have been removed are reset from -1 to zero, their neighbours with value two are reset to one.

Figure 7 shows the results obtained by applying this algorithm to several pictures.

An alternative method of improving the efficiency of the algorithm is by avoiding the need to reset the values of all points which have been removed, between passes, by re-defining the sets N and R instead. This is done by setting the value of those points which are removed at the m th pass to $-m$ instead of -1 and defining I , N and R as follows:

$$\begin{aligned} I &= \{1\} \\ N &= \{f \mid 0 \geq f > -m\} \\ R &= \{-m\} \end{aligned}$$

Thus, after each pass, N and R are re-defined by incrementing m , and the next pass can take place immediately.

SHORTCOMINGS OF THESE IMPLEMENTATIONS

With chromosome images we found that the type of approach described above had two disadvantages. First, the outlines of chromosomes tend to be rather noisy, with the adjacent arms of a chromosome not completely separate but touching in several places, and this results in a skeleton with many spurious branches, as illustrated in figure 8b. This can usually be overcome without much difficulty by a simple smoothing of the boundary and by expanding any 'holes' produced by touching arms until they are connected with the exterior of the object. Application of the algorithm then usually produces a skeleton with few or no spurious branches, as in figure 8c.

A second more serious shortcoming is that the resulting skeleton is completely defined by the original outline of the object. If the picture is essentially black and white and its outline well defined – as it frequently is in character recognition – this is exactly what is required. However, with chromosomes the density drops off gradually from the central ridge of each arm outwards, and the choice of boundary is somewhat arbitrary. In this case what is

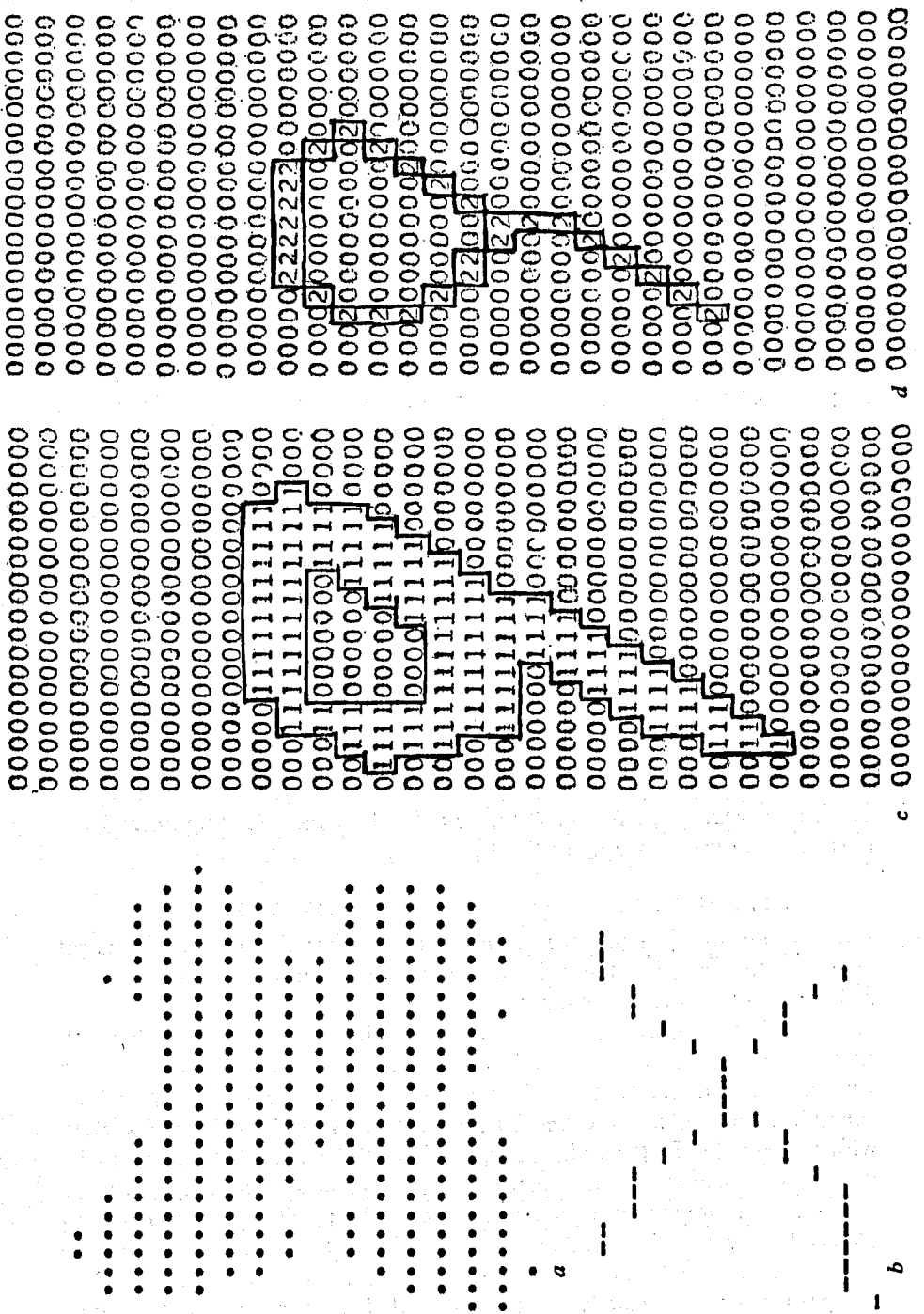


Figure 7. Binary pictures of a chromosome and of a hand-written digit, and the skeletons obtained from them.

required is that the skeleton should lie along the higher density ridges of the picture rather than in a central position determined by outline alone. An example illustrating this is shown in figure 9.

The way in which the algorithm can be varied in order to achieve this, for pictures in which f takes values that are positive integers representing the densities of the points, is given below.

IMPLEMENTATION FOR PICTURES WITH A RANGE OF DENSITY VALUES

It is possible to obtain a skeleton which tends to lie along ridges of higher density in a picture by removing only points at low density in the first few passes and allowing points with higher density to be removed later. This is done by restricting removal in any one pass to points with a given value. Thus, if d is the value of points which may be removed in the current pass, the algorithm is applied with I, N, R and U defined as follows:

$$\begin{aligned} I &= \{ \text{positive integers} \} \\ N &= \{ 0 \} \\ U &= \{ \text{all positive integers except } d \} \\ R &= \{ -1 \} \end{aligned}$$

Initially d is set to one. After each pass those points which have been removed are set to zero and the algorithm is re-applied until a pass is made in which no points are removed. At this stage any troughs of points with value one, which are not completely surrounded by points with a higher value, will have been deleted. d is now increased to two and the process repeated. In this way points with a low value lying between ridges of higher value points are deleted first, thus, for example, separating touching arms. When no further points with value two can be removed the program starts on value three, and so on. However, whenever the deletion of a point with value d uncovers a point with a lower value (i.e., a neighbour of a deleted point has value $g, 0 < g < d$) d is reset to g at the next pass. Nevertheless, it may be desirable to consider that whenever there is an area of points with value d_1 completely surrounded by points with a higher value, d_2 say, then this represents a closed curve surrounding the d_1 points. To achieve a skeleton which reflects this, all that is necessary is that when all points with value less than or equal to d_2 which can be removed, have been, then any remaining points at level d_1 are deleted thus producing 'holes'. Thinning then continues at level $d_1 + 1$.

Unfortunately, application of the algorithm is very time-expensive since it requires a complete pass to be made through *all* the points of the picture even though there may be only very few points remaining with the current value d (and in general the greater the number of values the less the efficiency). This has been overcome by initially making a list of all points of the picture with each value. Therefore, when removing points with a given value, it is necessary to consider only the points given in the list for that value.

	292	302	312	322	332	342	352	362
	I	I	I	I	I	I	I	I

```

96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136

```

a

	291	301	311	321	331	341	351
--	-----	-----	-----	-----	-----	-----	-----

```

95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137

```

b

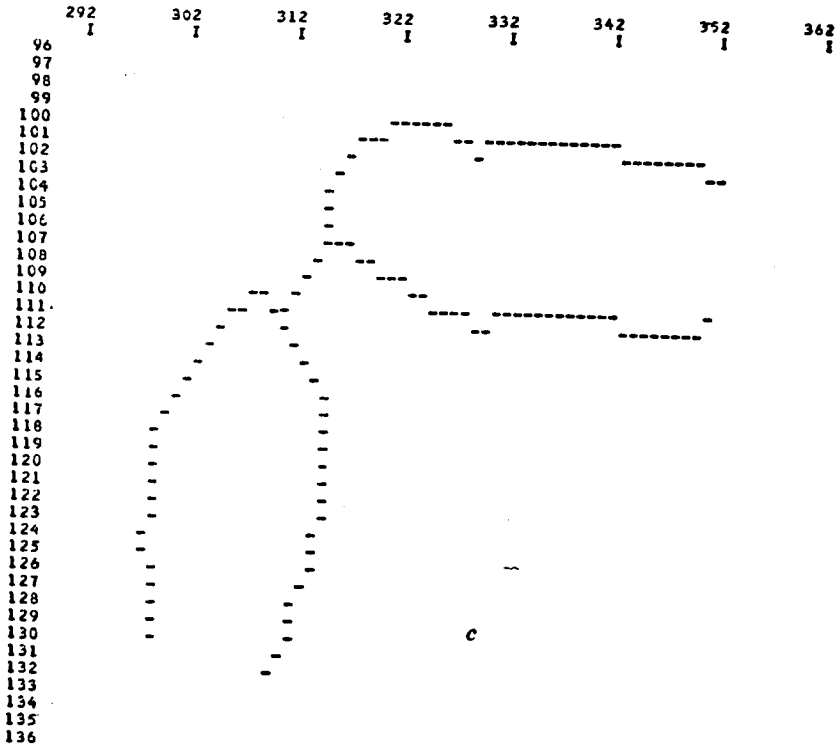


Figure 8. (a) A digitization of a chromosome. The chromosome is digitized on a seven-level density scale, and in the line printer representation shown here the characters are chosen so that the density at each point corresponds approximately to the overall darkness of the printed character.
 (b) The skeleton obtained from this chromosome by using the binary application of the algorithm on its characteristic function.
 (c) The skeleton obtained by pre-processing the picture before taking the characteristic function.

PATTERN RECOGNITION

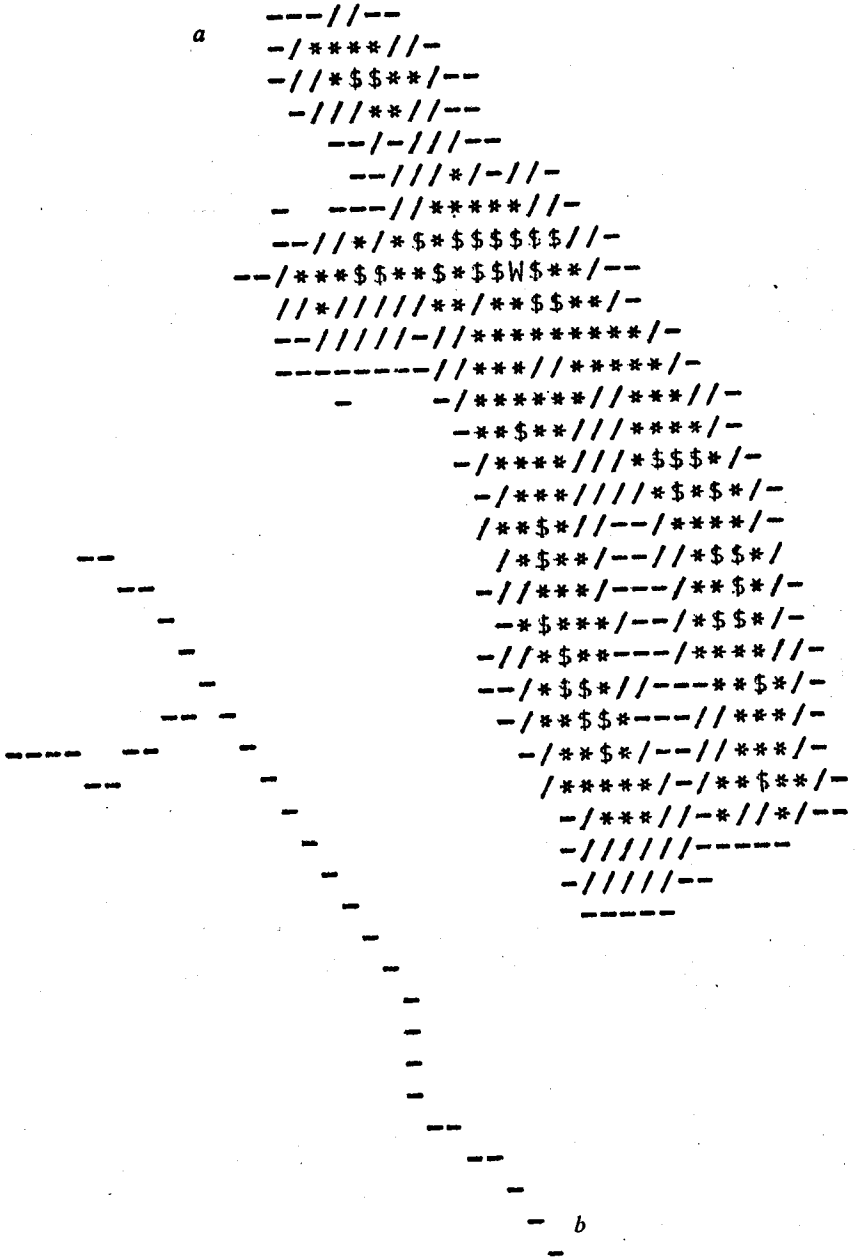


Figure 9. An example of a chromosome and its skeleton obtained by using the binary application of the algorithm. In this case it would be better if the skeleton were to lie along the high density ridges of the picture, rather than in a central position with respect to area alone.

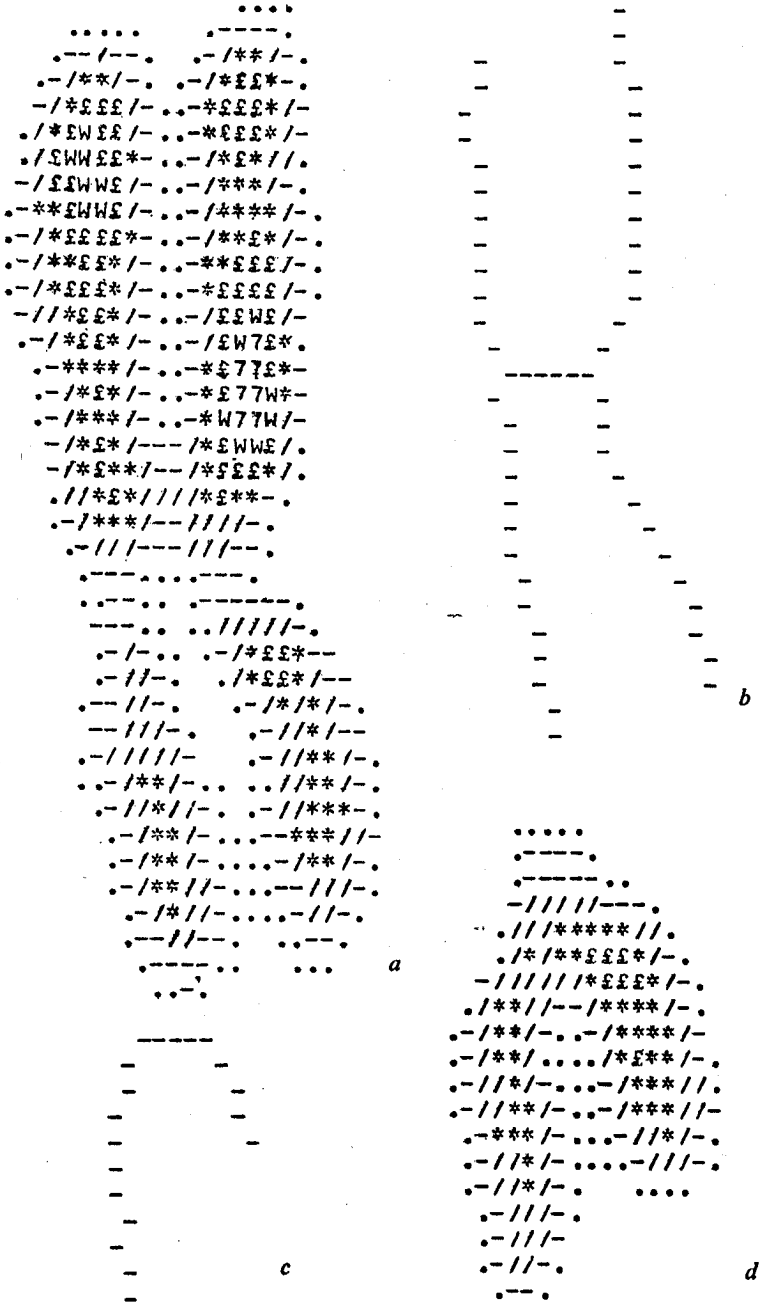


Figure 10. Some chromosome pictures and skeletons obtained using the version of the algorithm intended for pictures covering a range of density values.

PATTERN RECOGNITION

The success of this algorithm is illustrated in figure 10. As well as being less sensitive to the choice of boundary for the picture it is also found to be much less sensitive to noise – the only pre-processing required to obtain satisfactory skeletons being a simple smoothing, by setting each point to a weighted average of itself and its neighbours.

Acknowledgements

Thanks are due to all members of the Clinical and Population Cytogenetics Research Unit, Pattern Recognition and Automation Section under the direction of Dr Denis Rutovitz.

The computing was done using the IBM 7090 computer at Imperial College in conjunction with a Fidac Scanner.

REFERENCES

- Blum, H. (1964) *A Transformation for Extracting New Descriptors of Shape*. Boston: Air Force Cambridge Research Laboratories.
- Deutsch, E.S. (1967) Computer simulation of a character recognition machine. *The Post Office Electrical Engineering Journal*, 60, 39–44 and 104–9.
- Izzo, N. & Coles, W. (1962) Blood-cell scanner identifies rare cells. *Electronics* 52.
- Kirsch, R.A., Cahn, L., Ray, C. & Urban, G.J. (1957) Experiments in processing pictorial information with a digital computer. *Proc. Eastern Computer Conference*, 221–9.
- McCormick, B.H. (1963) The Illinois pattern recognition computer—Illiac III. *IRE Trans. Electronic Comp.* EC-12, 5.
- Narasimhan, R. (1964) Labelling schemata and syntactic descriptions of pictures. *Information and Control*, 7, 151.
- Philbrick, O. (1966) *A Study of Shape Recognition Using the Medial Axis Transform*. Boston: Air Force Cambridge Research Laboratories.
- Preston, K. (1961) The Cell Scan system—a leukocyte pattern analyser. *Proc. Western Computer Conference*, 173.
- Rosenfeld, A. & Pfaltz, J.L. (1966) Sequential operations in digital picture processing. *Journal of the ACM*, 13, 471–94.