EXTRACTION OF THE LINE DRAWINGS OF 3-DIMENSIONAL OBJECTS BY SEQUENTIAL ILLUMINATION FROM SEVERAL DIRECTIONS

Yoshiaki Shirai
Saburo Tsuji

Electroteehnical Laboratory
Tokyo, Japan

Osaka University, Osaka, Japan

Abstract

The extraction of the line drawing of 3-dimensional objects presented here is developed for the eye of the intelligent robot studied in the Electrotechnical Laboratory. This procedure consists mainly of two processes. First, four line drawings of the same objects illuminated from four different directions are sequentially obtained. Second, by applying 2-dimensional logical operations to these line drawings, a complete one is extracted. This method is applicable to more then two polyhedrons in a scene which is difficult by usual methods heretofore presented because of too small difference of light intensities between two different planes or the effects of shadows.

1. Introduction

The recognition of 3-dimonsional objects is necessary for intelligent robots that manipulate objects in relation to visual information. Up to now the recognition of a simple polyhedron based on its line drawing has been studied by lighting it so that the effect of shadow is eliminated.(1), (2) The extraction of a complete line drawing is, however, very difficult because of too small differences of light intensities between two different planes or the effects of shadows. The improved method presented here for extracting the line drawing consists of two steps. First, by lighting the objects from several directions sequentially, the line drawing for each illumination is extracted. Next, only useful information is extracted from the obtained line drawings. This method has been developed for the eye of the intelligent robot studied in the Electrotechnical Laboratory.[(3)]

The outline of the whole procedure is shown in Fig.l. Each block in Fig.l is described in the following sections.

2. Data input

A vidicon TV camera is used as the visual input device. A preprocessor (Fig.2) is employed to transfer the video signal to a computer. This preprocessor divides a picture frame into 256 x 256 picture elements and digitizes the video signal of 64 x 64 elements to 6 bits (64 levels of light intensity) in one scanning period. The region of a picture frame to be sampled is controlled by a computer.

3. Extraction of a line drawing illuminated from one direction

The procedure to obtain a line drawing, which is represented by a set of equations of lines, consists of (1) smoothing, (2) spatial differentiation, (3) filtering 1, *(A)* thinning, (5) filtering 2, (6) tracking and (7) line fitting. Each stage is described below.

(1) Smoothing

This stage is to eliminate the noise of the vitlicon camera and the preprocessor, by applying a 3 x 3 nonlinear filter to the picture elements. Let the element of the 3x3 matrix be denoted as in Fig.3 and the value of the picture element corresponding to the matrix element i be denoted by C(i). The filter is to compare C(0) with the other picture elements C(i), i=l,...,8 , and if necessary correct C(o). The criterion of the correction is based on the fact that the object is a polyhedron. Therefore, the fine structure of brightness as shown in Fig.4 may not exist. If the condition shown in Fig.4 holds, C(0) is corrected as follows.

If $C(i)>C(0)$ for $i=i_1,\ldots,i_n$,

$$C(0)=\min\{C(i_1)\ldots,C(i_n)\}$$

If $C(i)<C(0)$ for $i=i_1,\ldots,i_n$,

$$C(0)=\max\{C(i_1)\ldots,C(i_n)\} .$$

The filter is applied to every picture element repeatedly until the correction is no longer required. This smoothing operation eliminates the change of brightness caused by noise, but preserves the resolution of the picture which is lost by the usual smoothing methods.

(2) Spatial Differentiation

The 2-dimensional difference of light intensity at each picture element D and the direction of the gradient K are calculated by the 3 x 3 array as follows.

$$D_x=(C(2)+C(3)+C(4))-(C(6)+C(7)+C(8))$$
$$D_y=(C(4)+C(5)+C(6))-(C(8)+C(1)+C(2))$$
$$D'=(D_x+D_y)/3+k_b\ (k_b=\text{constant})$$
$$\alpha= \arctan\ (D_y/D_x)$$
$$D=\begin{cases} D' & \text{if}\quad D'>0 \\ 0 & \text{if}\quad D'\leq 0 \end{cases}$$

The direction of the gradient is digitized as

$$K = (20/\pi)\alpha \qquad (\text{mod. } 20)$$

Thus the value of D is large at the picture element corresponding to the edge of the object, and the direction of the edge is nearly perpendicular to the direction of the gradient. The result of spatial differentiation is illustrated in Fig.5.

(3) Filtering 1

If a picture includes an edge, the picture elements with large 1) constitute a line whose width is larger than 2 elements. Based on this fact, the noise is eliminated by the 3x3 filter. Let D(i) denote the value D of the C(i), filter 1 corrects D(o) to 0 if among its adjacent I)(i)(i=1,...,8), more than five D(I) take value 0.

(A) Thinning

The edges of the objects are detected by filtering 1 such that the D of each element distributes like a ridge whose center corresponds to an edge. The thinning is to detect the peak of the D and correct the other D to 0. The algorithm is to detect the segment of the]ine formed by the picture elements and to examine sequentially the adjacent element toward its perpendicular direction to determine whether or not the D of the element is to be corrected. For a detailed description, the definitions are given in the following:

a) The direction perpendicular to K is dentoed by L.

b) Two elements which are adjacent to an element in the direction L are denoted by ℓ and ℓ' respectively, and those which are adjacent in the direction K are denoted by k and k'.

c) Let K(i) denote the value K of the element i; the element 0 is the line segment with error d if and only if the following two conditions are satisfied.

1. $|K(\ell)-K(0)| \leq d$, 2. $|K(\ell')-K(0)|\leq d$.

The correction of D(k) and D(k') is based on the 3 x 3 arrays of elements where k and k' are defined in (b). The correction algorithm of D(k) is as follows.

[1] If the element is the line segment with error 3, then [2]; otherwise no correction.

(2) If $|K(k)-K(0)|\le 3$, then (3); otherwise no correction.

3 If $D(k)\le Da$ (Da is constant), then (4); otherwise no correction.

4 If $D(k)<Da$, then (5); otherwise (6).

5 If $|K(k)-K(0)|\le 2$, then $D(k)=0$; otherwise (6).

6 If the element k is the line segment with error 3, then D(k)=0; otherwise no correction.

The correction of D(k') is just the same except k is replaced by k' an the above algorithm. This rather complicated algorithm is required in order to preserve the true]ine segments near the crossing point of more than two lines.

(5) Filtering 2

For the elimination of the noise from the obtained set of D and K, the 3 x 3 filter is applied to the each element. The operation of the filter is to correct the 1) of the element to 0, (1) if more than 7 adjacent elements' 1) are 0, or (2) if both $D(\ell)$ and $D(\ell')$ are 0. In case (l), the isolated points are eliminated, while in case (2), those points which do not constitute a line are eliminated. This obtained result is illustrated in Fig.6.

(6) Tracking

A set of elements which constitutes a line is picked up by tracking the element sequentially based on the 1) and K of the element. The starting point of the tracking must be on the line. Scanning each element by the 3 x 3 array, the starting point is decided as the center element of such an array to satisfy the following two conditions.

1. $|K(\ell)-K(0)|\le 1$ or $|K(\ell')-K(0)|\le 1$
2. $|K(i)-K(0)|\le 1$ for i such that $i\in\{\ell,\ell-1,\ell+1\}$ and $i\in\{\ell',\ell'-1,\ell'+1\}$

From the starting point the element is tracked toward the direction L(0), and then toward the opposite direction until the terminal point is found. The tracking algorithm is shown in Fig.7, where initial value α_0=L and n_0=constant.

(7) Line Fitting

From the sets of elements each of which constitutes a line, the equation of the line is obtained by the least squares method. In order to simplify the calculation without a great deal of error, the equation is represented as y=ax+b or x=ay+b such that $a\le 1$. The obtained line drawing is illustrated in Fig.8.

4. Synthesis of the Desired Line Drawing

The desired line drawing is defined such that it includes all the edges of the polyhedron and no others. In order to get the desired line drawing, the authors proposes the following 2-dimentional operations. They are the logical 0R(+) and logical AND(.) of two line drawings. Let A and B denote the two line drawings; the function of each operation is as follows.

1. OR: This operation synthesizes the line drawing that includes all the edges included in both A and B. Although the lines in A and B which correspond to the same edge do not necessarily occupy the same place in the drawing, they must be regarded as the same edge. For example, the result of OR operation of the line drawings A (Fig.8) and B shown in Fig.9 (this is denoted by A+B) is shown in Fig.10.
2. AND: This operation synthesizes the line drawing that includes only those edges that are included in both A and B. The result of the operations of A and B (denoted by A.B) is illustrated in Fig.11.

The philosophy of synthesis of the desired line drawing is to extract the useful information from 4 line drawings which are obtained by lighting the objects from different directions. Let A, B and C denote the line drawings obtained by lighting from different directions, if the position of the shadow in A is different from that in B, the result of A.B includes no lines caused by the effect of the shadow. Although each A, B or C alone may not include edges whose adjacent planes have the same light intensities, every edge is included in either A, B or C. Therefore, the result of (A+B+C) includes every edge. On the other hand, if the objects are lighted from the frontal direction, the obtained line drawing 1) is free from the effect of the shadow. Thus, the operation sequence to synthesize the desired line drawing E from I) and the other 3 line drawings obtained by lighting from different direction A, B and C consists of the next 6 steps.

1 $F=A\cdot B$, 2 $E=F+D$,
3 $F=A\cdot C$, 4 $E=F+E$,
5 $F=B\cdot C$, 6 $E=F+E$

The result of this sequence using A(Fig.8) , B(Fig.9), C(Fig.I2) and i)(Fig.I3) is illustrated in Fig.14.

The block "Set the lighting" in Fig.1 is, therefore, to get the line drawings A, B, C and D, and the block "Indicate the logical operation" is to perform the above described sequence.

5. The procedure of the 2-dimensional logical operations

There are mainly two methods for the previously defined logical operations such as the one at the line drawing level and at the intensity array level. For simplicity of the procedure, the latter method is adopted here.

The procedure of the logical operations previously defined is as follows. First, the line drawing is mapped to the array of the differentiation. The element of the array that corresponds to a line segment of the line drawing is given value D] and the direction of the line L, and the adjacent element is given D_2 ($D_2<D_1$) and L. The result of this operation is just the same as obtained in spatial differentiation described in 3 - (2), except that it includes no noise. The operation OR of A and B utilizes the two arrays thus mapped from A and B, respectively, to make a new array by adding the two arrays. Then, by applying the previously described thinning, tracking and line fitting to the new array, the result of OR is obtained. The operation AND is just the same as OR except that the new array is made such that if 1)=0 at one of the corresponding elements of two arrays, then the value of the corresponding new element is 0. Thus these operations regard the two lines, in A and B, as the same edge even if their positions are slightly different from each other. This method is very simple because only new algorithm to be developed is to map the line drawings to the array of differentiation. The other algorithm such as thinning, tracking and line fitting are just the same as described in section 3.

6. Cone1usions

The extraction of the line drawing by sequential lighting from several directions is proved to be applicable to a scene where objects are placed behind other objects. The desired line drawing of a few types of prisms and pyramids are obtained by lighting from the left, right, up and front sequentially. The calculation time is about 10 times as long as that of obtaining a lino drawing illuminated from one direction. Though it takes at present about 2 minutes by NEAC-3100 (cycle time is 2μ sec, 32K core memory), a large part of the calculation can be processed by special purpose hardware. Application to more complicated objects may be possible if the present resolution of a picture (64 x 64) is improved.

Acknowledgements

The authors would like to express appreciation to Mr. M. Suva and Y. Suzuki who provided them with an excellent hardware system.

Referenc cs

1. J. McCarthy et al. "A computer with hands eyes and ears", AFIPS Conf. Proc, Vol.33 (1968).
2. N.J. Nilsson "A Mobile Automaton", International Joint Conf. on Artificial Intelligence (1969).
3. S. Tsuji et al. "Industrial Eye" Conf. of Inst, of Electric Eng. (1970).
4. Y. Suzuki et al. "Industrial Eye 1", Joint Conf. of four Electrical Institutes (1968).

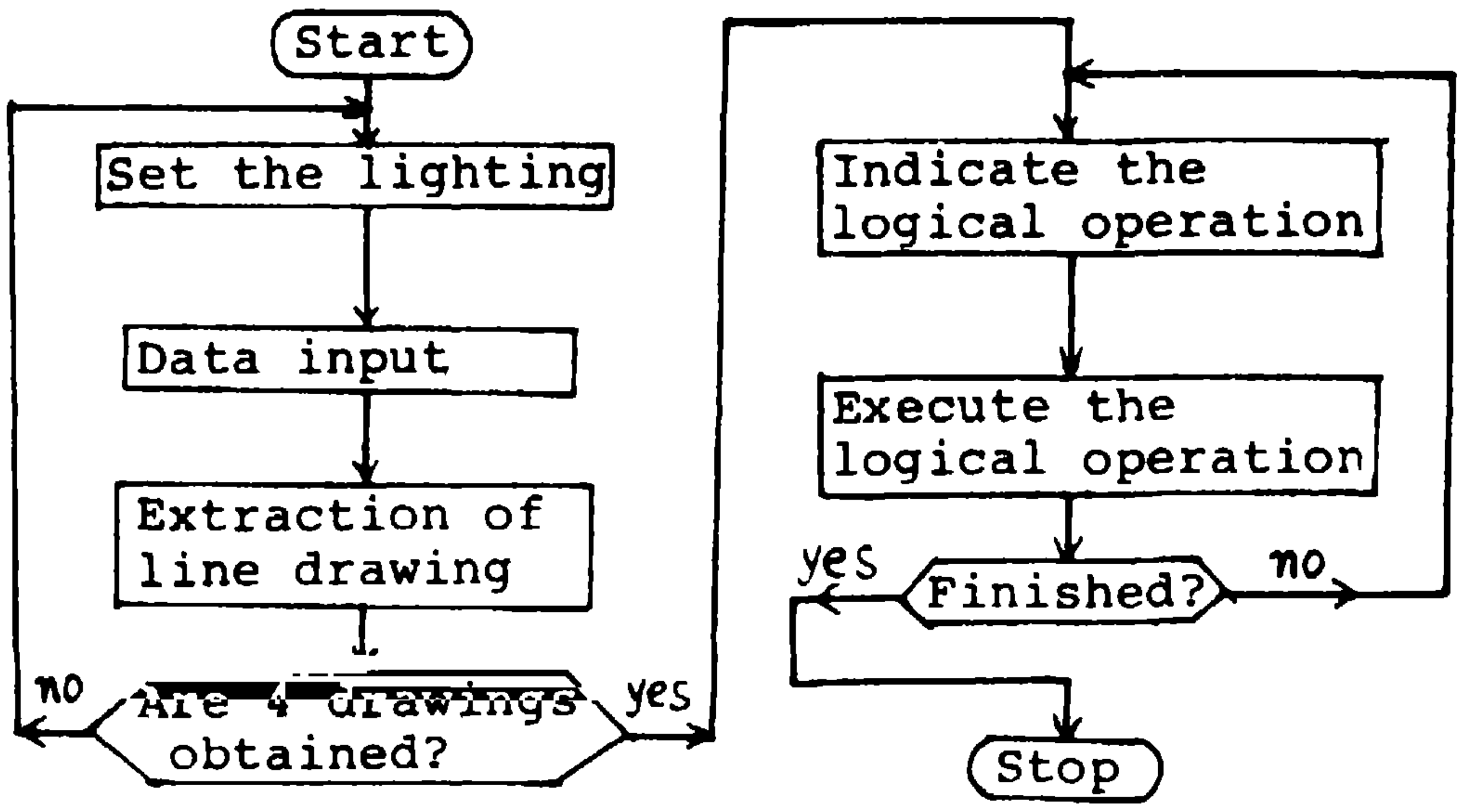

Start
Set the lighting
Data input
Extraction of line drawing
no
Are 4 drawings obtained?
yes
Indicate the logical operation
Execute the logical operation
yes
Finished?
no
Stop

Fig.1. Steps of the whole procedure

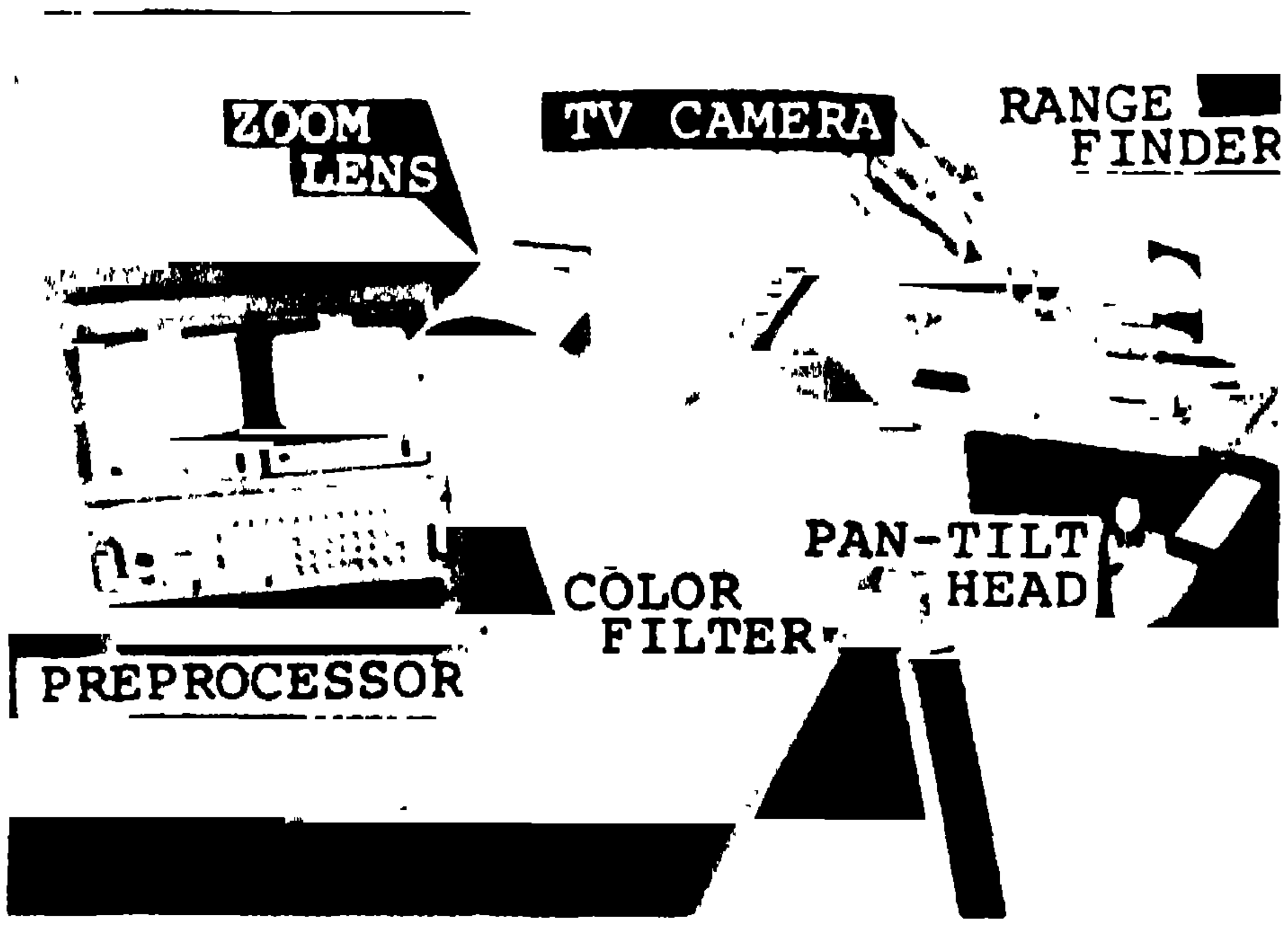
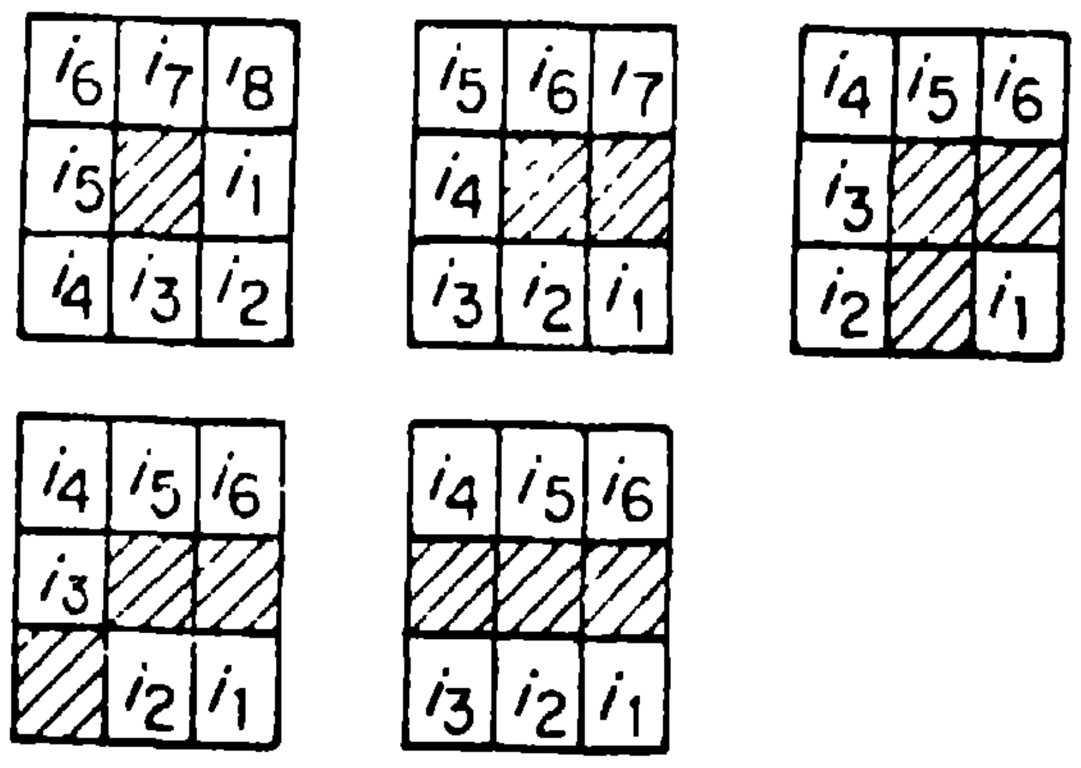


Fig.2. Visual input devices

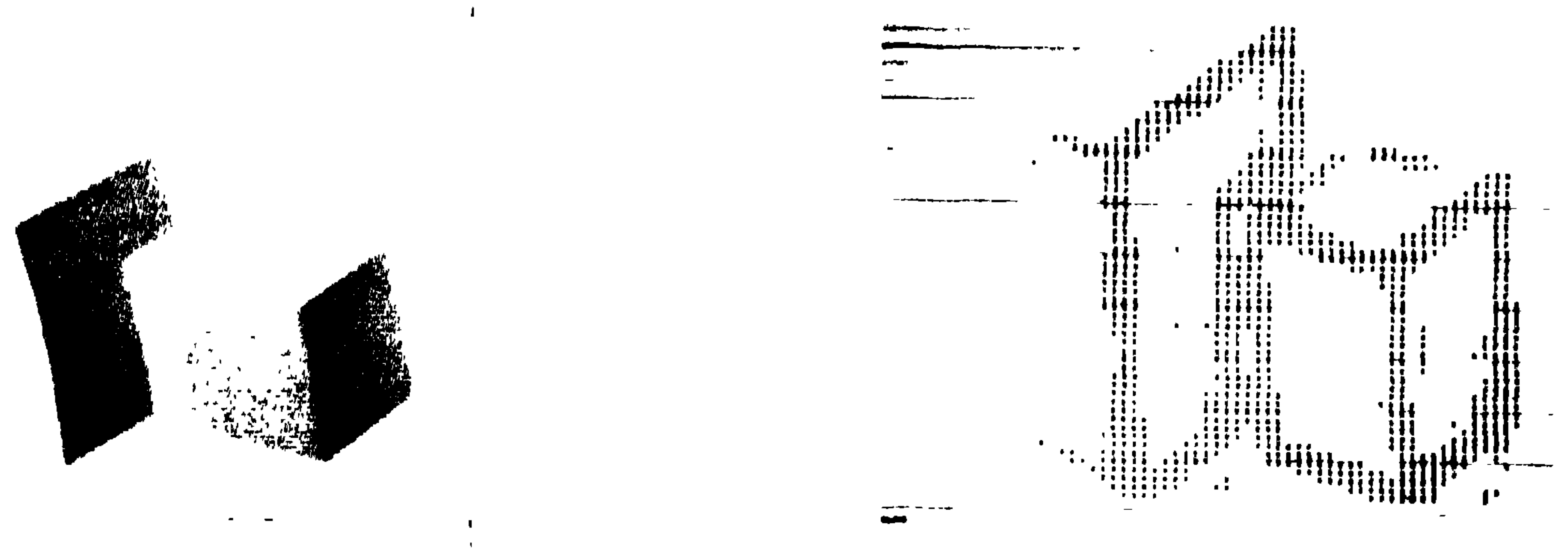
6	7	8
5	0	1
4	3	2

Fig.3. Number of array element



$C(i_n) > C(0)$ or $C(i_n) < C(0)$

Fig.4. The pattern of the picture elements to be corrected



(A) Picture data

x represents the element with D>0
(B) Result of operation

Fig.5. Spatial differentiation

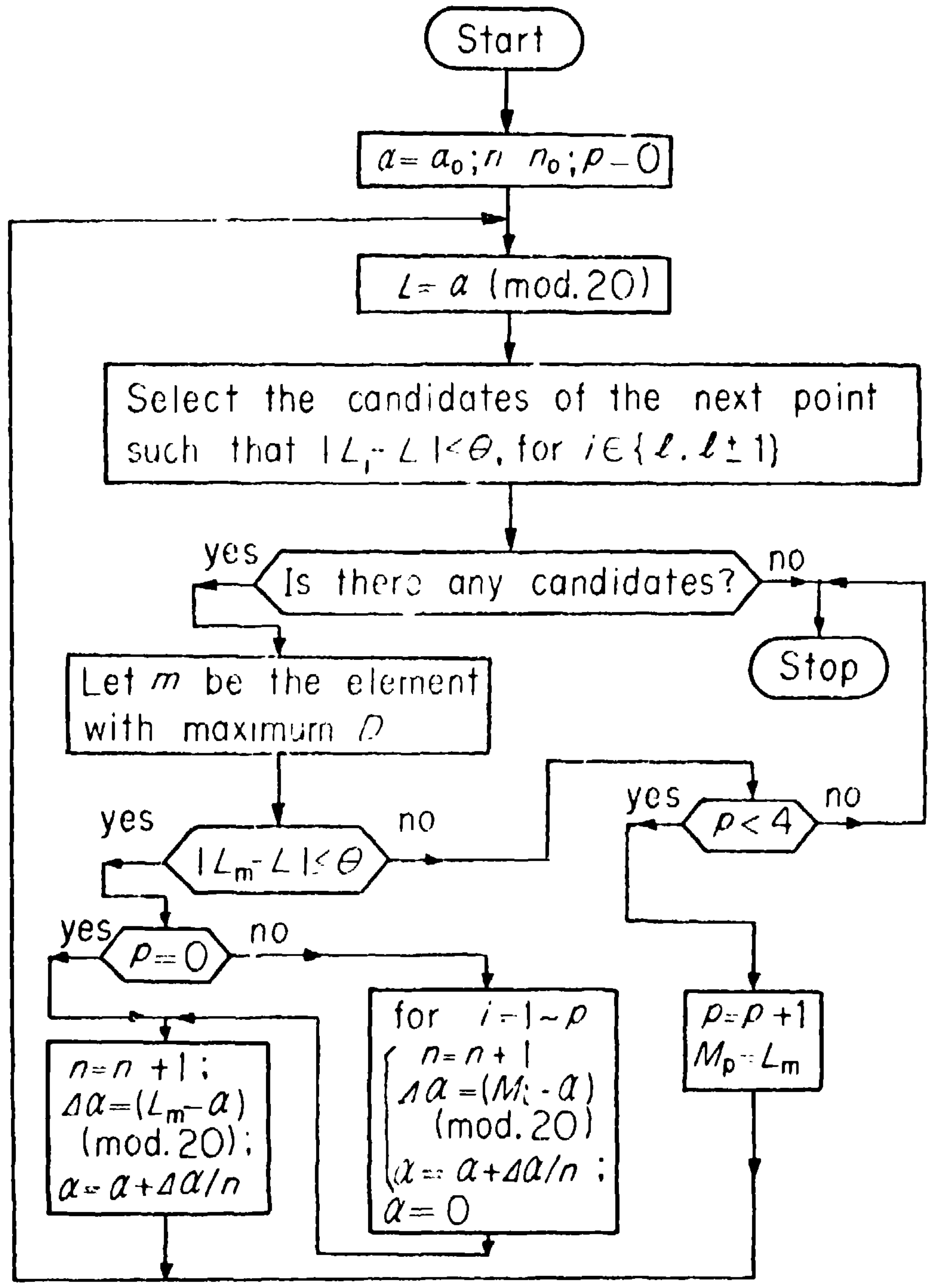
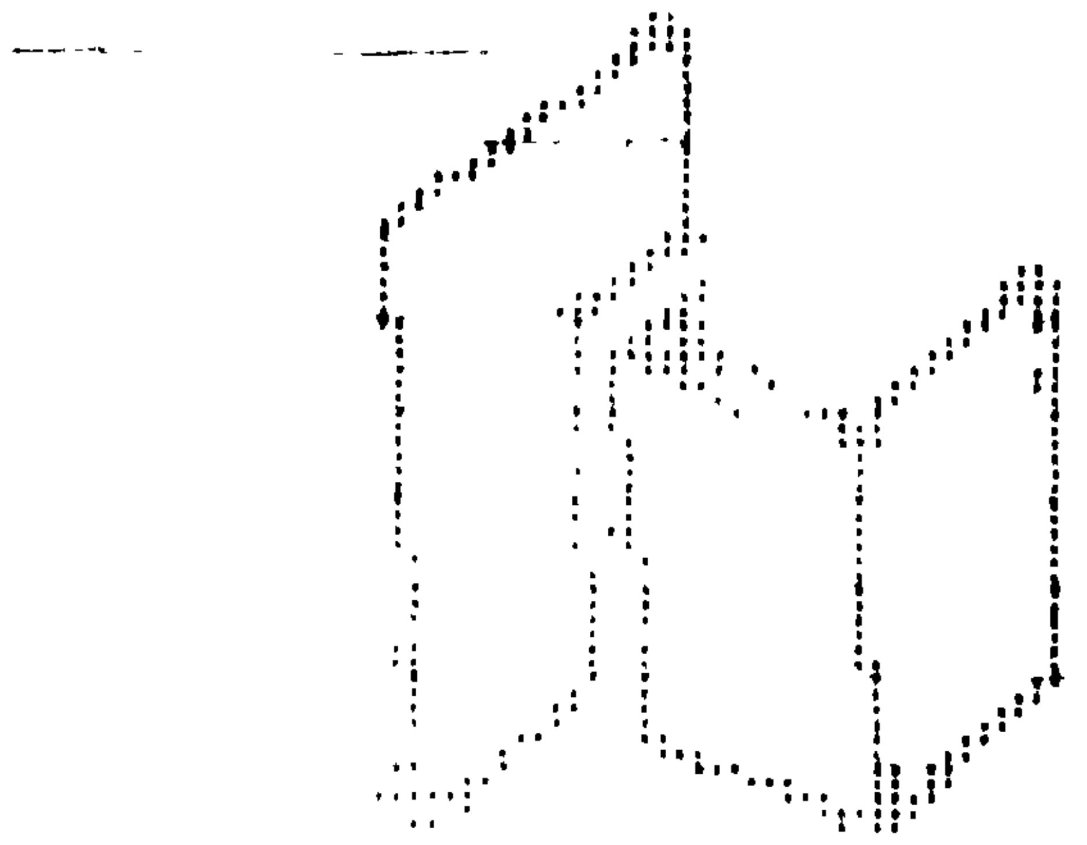
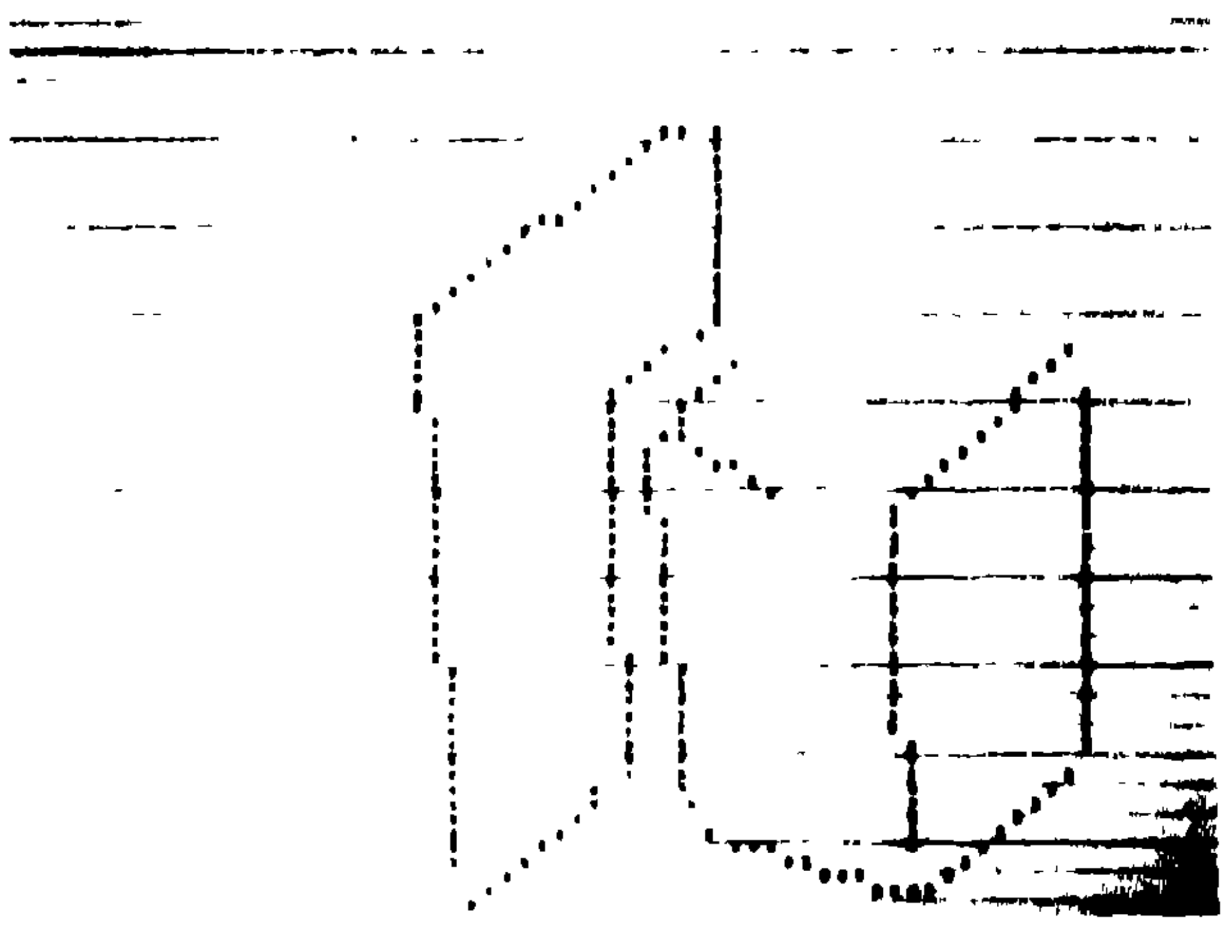

Start
$a = a_0; n = n_0; p = 0$
$L = a$ (mod.20)
Select the candidates of the next point such that $|L_i - L| < \theta$, for $i \in \{l, l \pm 1\}$
Is there any candidates?
yes
no
Stop
Let m be the element with maximum D
$|L_m - L| < \theta$
yes
no
$p < 4$
yes
no
$p = 0$
yes
no
$n = n + 1$; $\Delta a = (L_m - a)$ (mod.20); $a = a + \Delta a / n$
for $i = 1 \sim p$ $n = n + 1$ $\Delta a = (M_i - a)$ (mod.20) $a = a + \Delta a / n$; $a = 0$
$p = p + 1$ $M_p = L_m$

Fig.7. Flow chart of the tracking



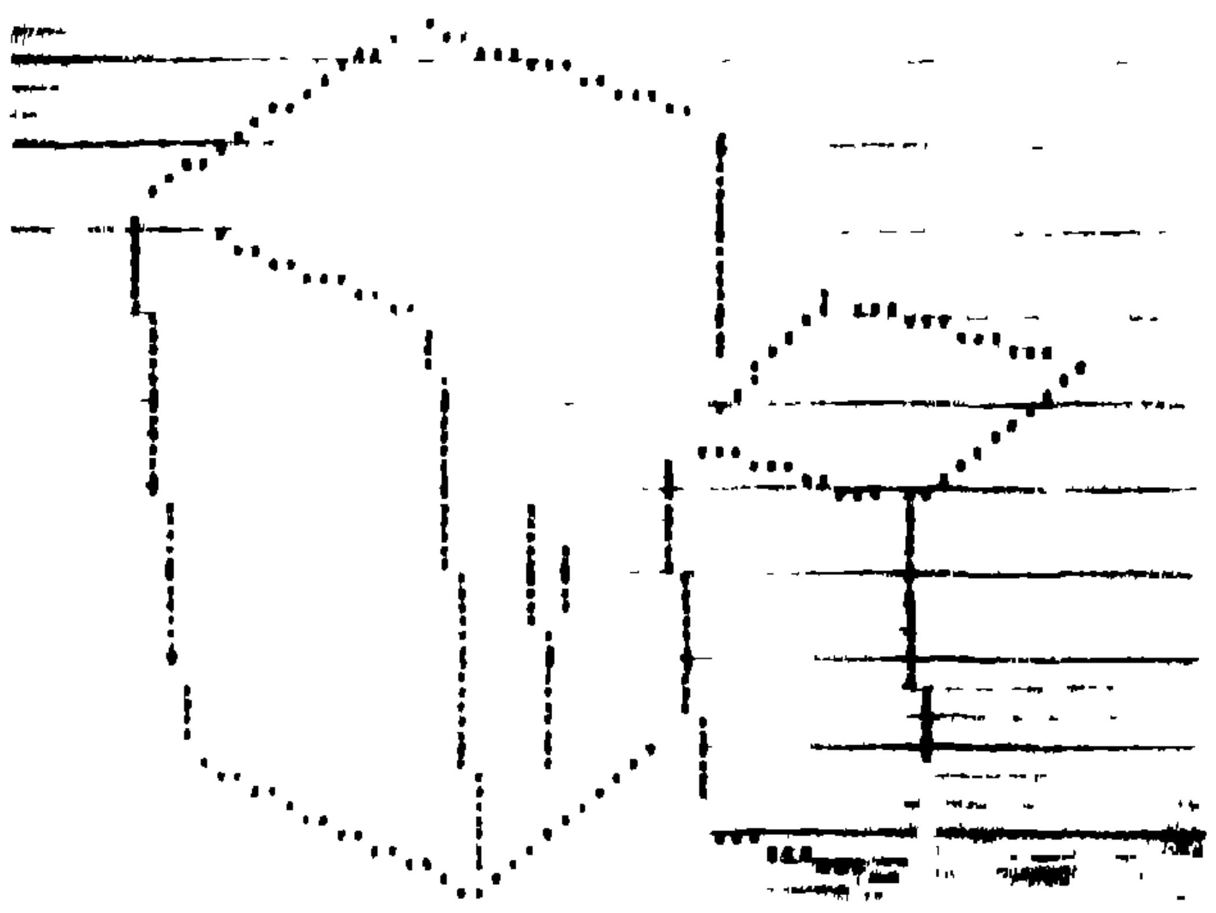
x represents the element with D > 0

Fig.6.Output of filtering 2



right lighting

Fig.8. Line drawing A



left lighting

Fig.9. Line drawing B

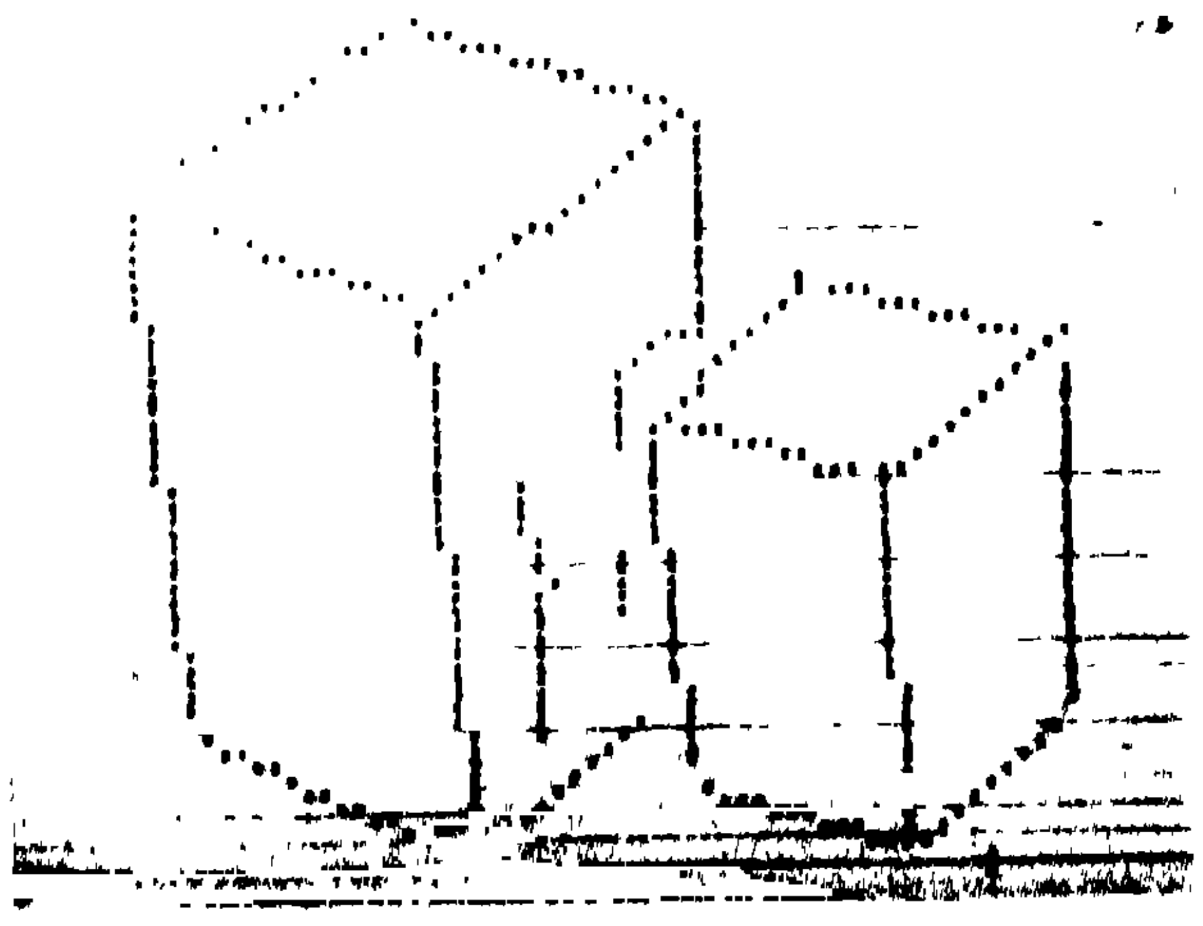


Fig. 10. Line drawing A+B

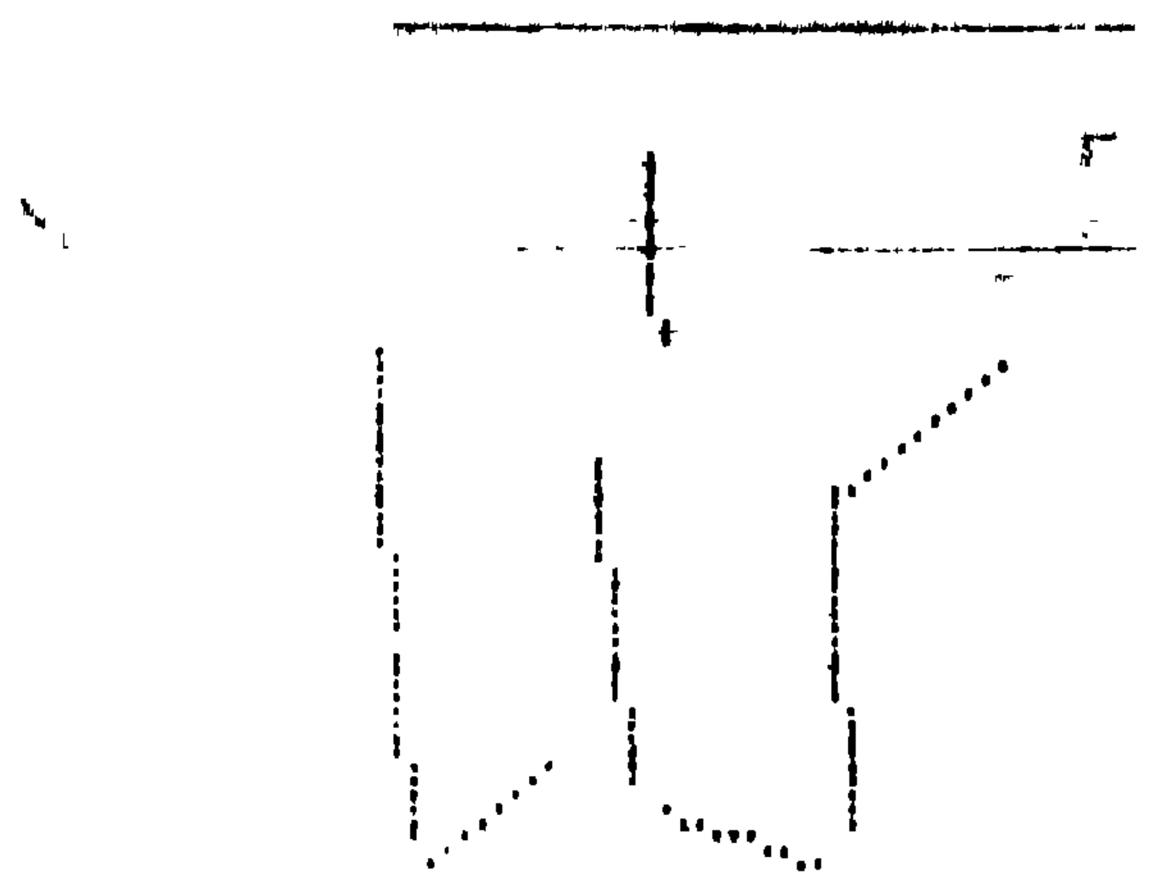
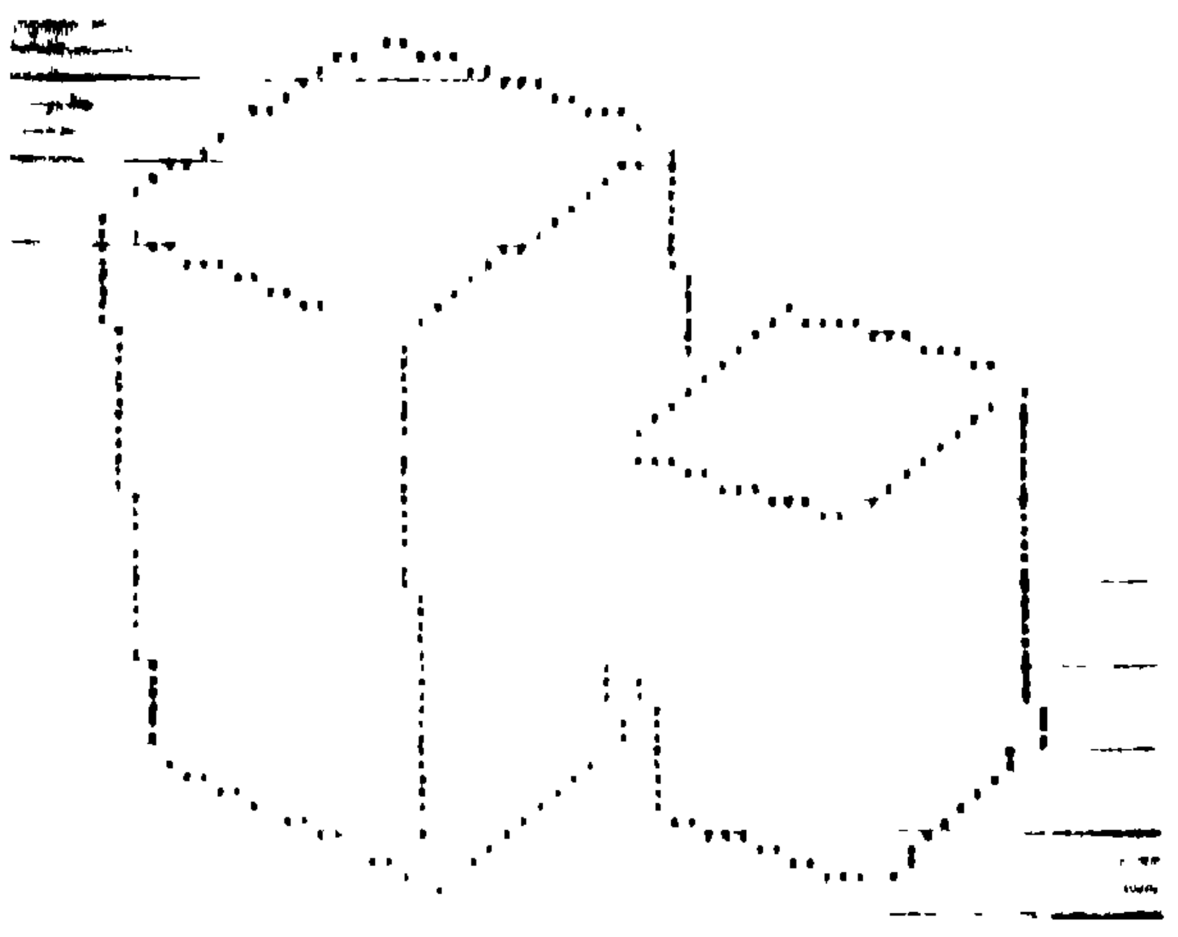
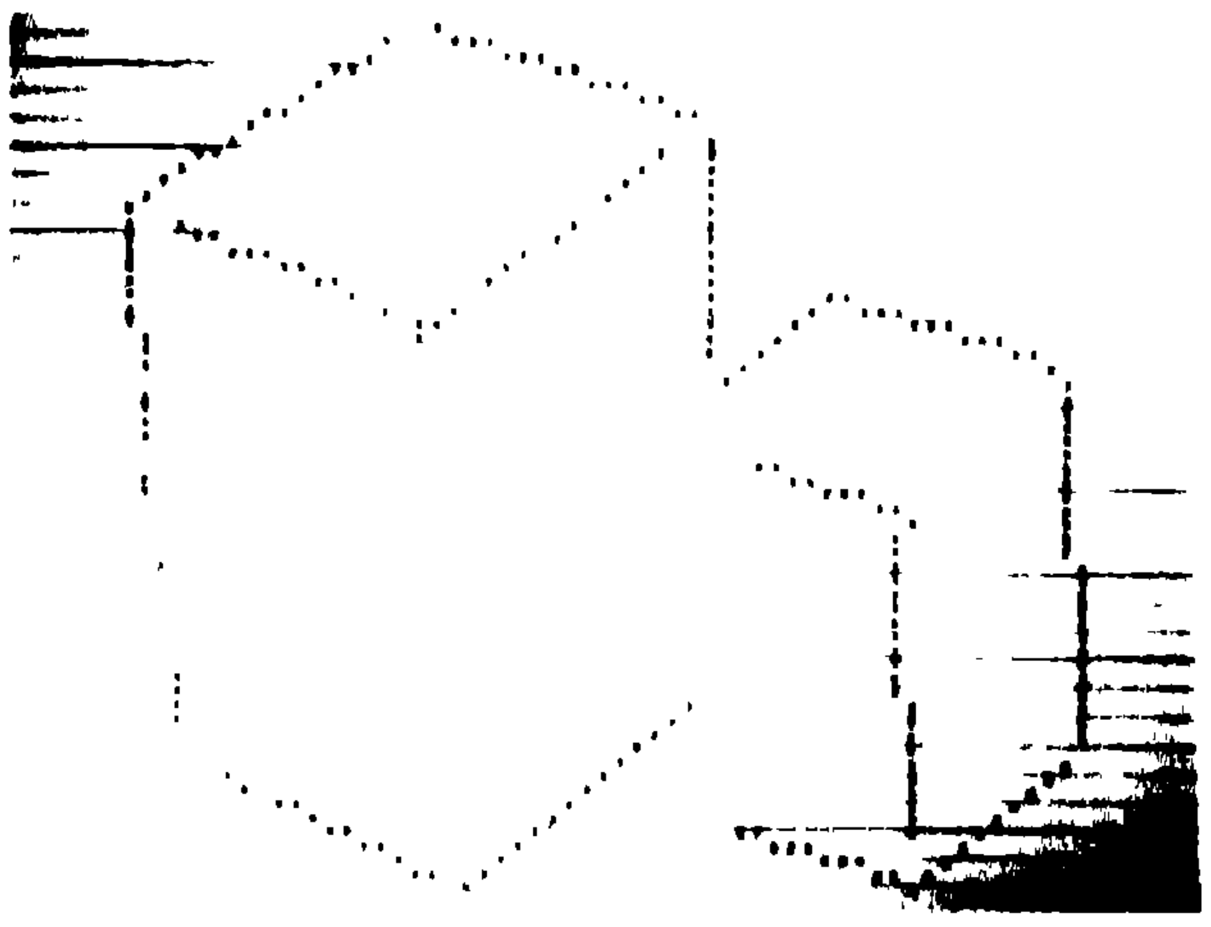


Fig.11. Line drawing A•B



up lighting
Fig.12. Line drawing C



front lighting
Fig.13. Line drawing D

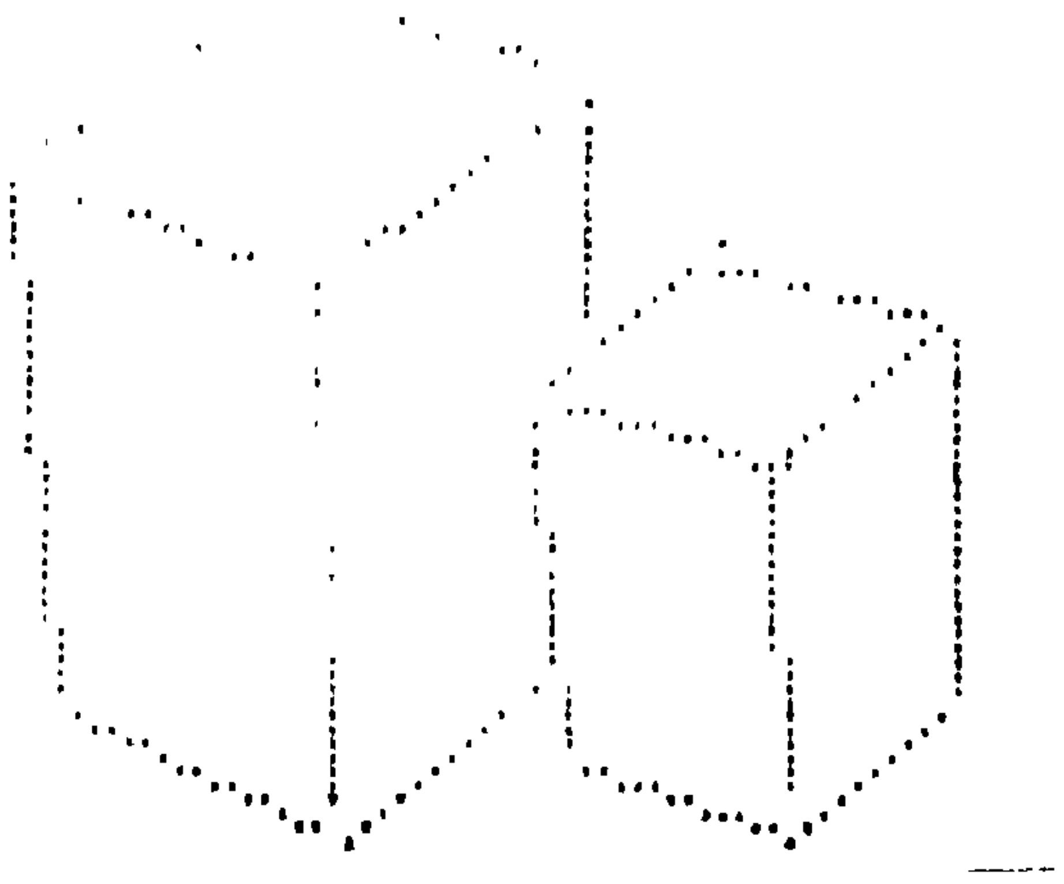


Fig.14. Obtained line drawing E