

CIRCULAR SCAN

VISION FLASH 23

by

Patrick H. Winston and Jerome B. Lerman

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

Robotics Section

March 1972

ABSTRACT

Previous feature point detectors have been local in their support and have been universally designed for objects without appreciable texture. We have invented (or perhaps reinvented) a scheme using correlation between concentric or osculating circles which shows some promise of being a first step into the texture domain.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

Vision flashes are informal papers intended only for internal use.

Origin

One of the three or four efforts receiving greatest effort within the Vision group is what we call the Hour-Glass Problem. Abstractly speaking, the Vision system has taken on the proportions of a classical hour glass. We think of the highly heterarchical PLANNER code as occupant of the upper bulb of the hour glass and the mildly heuristic, very complicated, machine language code, Horn-Binford line finder as occupant of the lower. The sole communication between the two has been through the line drawing offered up through the neck of the hour glass. This style of organization has been an annoyance preventing smooth acceptance of suggestions from our nicely progressing higher-level algorithms by the lower level line finding functions.

WIZARD is a new program that circumvents the Horn-Binford line finder and, being in PLANNER, offers great hope for conveying higher level conjecture down to the lower level work horses. As described in detail in Vision Flash #24, in preparation, WIZARD appeals to a few very basic primitive modules:

1. a line verifier
2. a vertex finder
3. a suspects finder that identifies the directions that lines are likely to enter a given vertex

Our first vertex finder operates much like a complete line finder in miniature. Its first activity is to scan five or so concentric circles around the point given as the center of a neighborhood believed to contain a vertex. Feature points are established in this data by something we call the Maryland function. This is the product of first differences defined as follows:

If $I(\theta)$ is the intensity around a scan as a function of θ , the first difference function is $D(\theta, \alpha) = I(\theta + \frac{1}{2}\alpha) - I(\theta - \frac{1}{2}\alpha)$ and the product of first differences is simply

$$D(\theta, \alpha, \beta, \dots) = D(\theta, \alpha) \times D(\theta, \beta) \times \dots$$

where the α, β, \dots are parameters. The feature points are identified as those points where the absolute value of this function is a local maximum.

Once the feature points are identified, a mildly boring program, with a large number of obscure parameters, attempts to fit lines and identify potential vertexes.

This little pass-oriented micro system works, but it is inadequate for a number of reasons. For one thing, lines entering the system of circles obliquely are difficult to pull out of the feature point array since the feature points do not simply cluster around some particular θ . Still less appealing is the fact that lines created from weak scratches count equally with genuine edges as far as the poor line grouping and vertex conjecturing program is concerned. Finally, the whole program collapses, as all other line finders do, when faced by an object with a textured surface.

These inadequacies motivated the conjectures described here in this tentative and preliminary working paper.

Our first idea is illustrated in Figure 1. Three circles are involved, with centers at $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3) . The circles have intensity functions of the form $I(\theta)$. Now if a vertex lies within the smaller circle on the left as shown, we claim that $I_2(\theta)$ is more like $I_1(\theta)$ than is $I_3(\theta)$, given some measure of the notion more like. The reason is simply that the angles at which lines exit are about the same for the left small circle and the big circle, but are quite different for the right small circle.

Given this, one adds upper and lower small circles, stirs a little, and out comes a vertex-finding hill climber with some peculiar features outlined in later sections.

Now the measure we use for similarity is based on the notion of the cross correlation of two functions defined as

$$R(f,g) \equiv \int f(\theta)g(\theta)d\theta$$

$R(f,g)$ tends to have larger values when f and g are similar and smaller values when they are not similar. (To see this intuitively, consider two functions f and g which take on values of either +1 or -1 and which have zero means. If f and g are dissimilar, they will have opposite signs about half the time and the integral of their products will be small.)

For our purposes here we substitute for f and g the intensity functions modified by subtracting out the mean intensities:

$$I(\theta) - \text{avg}[I(\theta)]$$

Some Theory

Readers familiar with elementary linear systems will recognize that our similarity measure is strongly related to the distribution of power in the Fourier transforms of the intensity functions. To review briefly, let a double arrow indicate that the θ function on the left transforms to the frequency function on the right. Then by definition

$$f(\theta) \leftrightarrow \int f(\theta)e^{-jw\theta}d\theta \equiv F(w)$$

$f(\theta)$ can then be obtained from $F(w)$ by taking the inverse transform,

$$f(\theta) = \int F(w)e^{jw\theta}dw$$

We can then define the cross correlation of f and g as

$$R_{fg}(\alpha) = \int f(\theta)g(\theta-\alpha)d\theta$$

which again by appeal to linear system theory, we note to be given by

$$R_{fg}(\alpha) \leftrightarrow F(\omega)G^*(\omega)$$

or

$$R_{fg}(\alpha) = \int F(\omega)G^*(\omega)e^{+j\omega\alpha}d\omega$$

Now our similarity measure is

$$\begin{aligned} \int f(\theta)g(\theta)d\theta \\ = R_{fg}(0) \\ = \int F(\omega)G^*(\omega)d\omega \end{aligned}$$

or in other words, we are just computing the total energy in the so-called cross power spectrum.

In subtracting out the means before we cross correlated, we are doing what amounts to elimination of the d.c. component in the forier transforms. We must do this one way or another or else we will have a luminosity detector, not a vertex finder. We are interested in good match in the variation of the signals around the circles, not in their average strength.

Curiously, we can do the same thing another way by differentiating the signals. Recall that

$$\begin{aligned} f(\theta) &\leftrightarrow F(\omega) \\ \Rightarrow \\ f'(\theta) &\leftrightarrow j\omega F(\omega) \end{aligned}$$

then clearly

$$\int f'(\theta)g'(\theta) = -\int \omega^2 F(\omega)G^*(\omega)d\omega$$

So the net result is that the average intensities are rejected but the noise which tends to be in the higher frequencies gets accentuated.

Noise Suppression

If we knew in advance that the spatial noise in the scene occupies a particular frequency band, we could suppress whatever correlated noise squeaks through by going to the trouble of actually computing the transforms and attenuating the noise frequencies in the integration through the use of some particular weighting function.

Curiously, some noise suppression will take place by virtue of the fact that the circles are of different diameter. Figure 2 suggests why this is true intuitively. Since the larger circles cover more arc length with θ than do smaller ones, the larger circles will encounter more of the granular noise blobs and the spectrum of the noise will cluster around higher frequencies than it will for the smaller circles. Thus the integral of the product of the two spectra is less influenced by the noise than it would be if the circles were of the same size. The effect is similar in the case of elongated textural noise such as is found in wood.

Alternate Configurations

Rather than the system of circles shown in Figure 3, Shirai has suggested the system of Figure 4, which is perhaps easier to understand intuitively and may be better technically. It is easy to hill climb either way.

For a large capture diameter and sharp peak on vertexes or lines, one could use a system of more than two circles as shown in Figure 5 by taking the product of all the intensities,

$$\int f(\theta)g(\theta)h(\theta) \dots d\theta$$

$\frac{C}{(\Delta I)^2}$
↑

$$\frac{C}{(\Delta I)^2} = 2 \left[\cos^{-1} \frac{V}{R} - \frac{1}{\pi} \left(\cos^{-1} \frac{V}{R} \right)^2 \right]$$

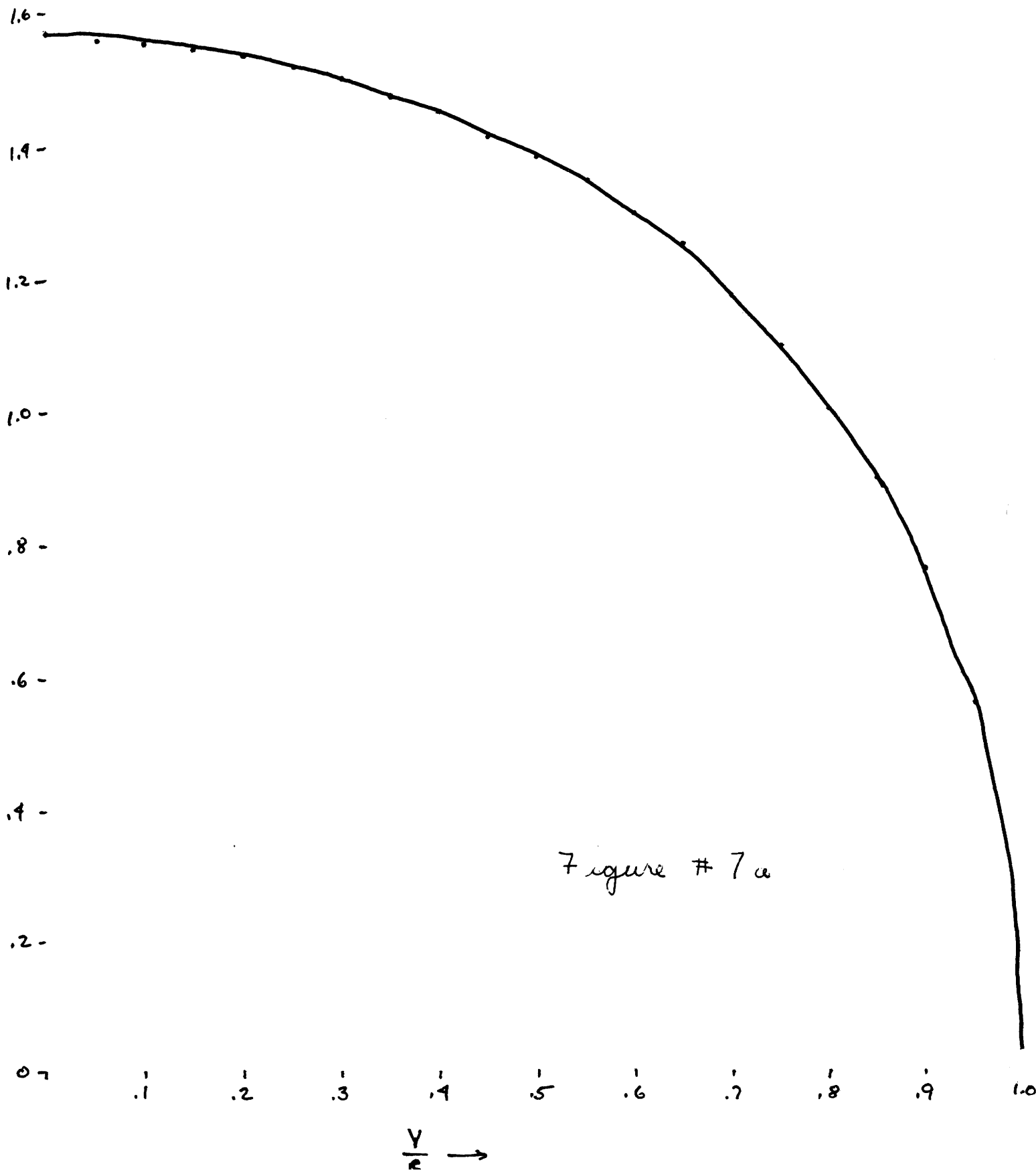


Figure # 7a

$\frac{C}{(\Delta I)^2}$
↑

$$\frac{C}{(\Delta I)^2} = \frac{2}{\pi} \cos^{-1}\left(\frac{\frac{1}{2}d+x}{R}\right) \cos^{-1}\left(\frac{\frac{1}{2}d-x}{R}\right)$$

1.6 -

1.4 -

1.2 -

1.0 -

.8 -

.6 -

.4 -

.2 -

7

5

10

15

20

25

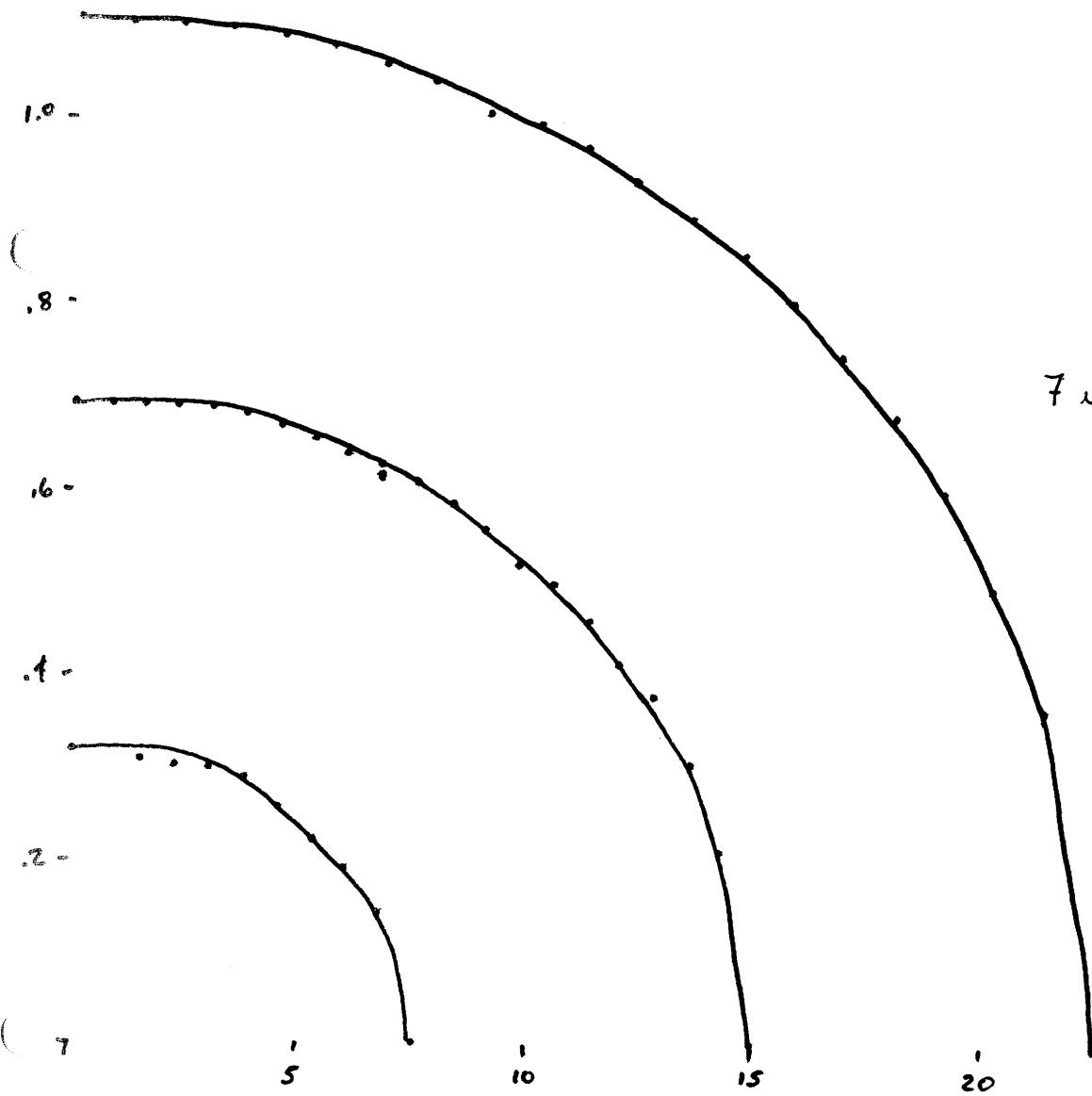
30

~ $d = \frac{R}{2}$ $R = 30$.

~ $d = R$ $R = 30$.

~ $d = 1.5 R$ $R = 30$

Figure #7b



Another variation on the theme would be the use of an array of overlapping circles of the same size with cross correlations taken between adjacent pairs. See Figure 6.

If the radii of the circles is r and the distance between their centers is $d < r$, and the line in question is perpendicular to the line between the centers and is located at x , where $x=0$ is the point midway between the centers, then it can be shown that

$$\int fgd\theta = \frac{2}{\pi}(\Delta I)^2 \cos^{-1}\left(\frac{x+\frac{1}{2}d}{r}\right) \cos^{-1}\left(\frac{\frac{1}{2}d-x}{r}\right)$$

Of course for lines parallel to the line between the centers, the separation does not matter and if y is the distance of the line from the centers, then

$$\int fgd\theta = 2(\Delta I)^2 \left[\cos^{-1}\frac{y}{r} - \frac{1}{\pi} \left(\cos^{-1}\frac{y}{r} \right)^2 \right]$$

Obviously, then, by increasing the distance d between centers one can make the pair more and more unidirectional in sensitivity.

Figure 7 shows what these functions look like for various d .

Note that the response is proportional to the square of the intensity of the step. The obvious mechanism for implementing such correlations is by eye jiggle.

Textured Surface

Two circles will have general reaction to something like wood texture. But the amplitude is increased as before by edges that are signaled by general changes in luminosity across them. Thus there is some hope that such a picture function might be useful in finding edges on textured objects which clobber all known line finders, other than the new one by Shirai.

First Results

Figure 8 shows the results of a concentric circle scan on the inside fork vertex of a wooden block. Figure 9 shows what the Horn-Binford line finder does.

Figure 10 shows the results of the same type of scan on the fork of a white block, artificially textured with the pattern shown in figure 11. Figure 12 shows the Horn-Binford result.

We include figure 13 to hint at the sort of intensity profiles involved. Those plotted are found on the indicated circle, with 0° corresponding to the x axis.

Shockingly, Shirai's new line finder finds all 9 lines of the object. It will be described in a forthcoming vision flash.

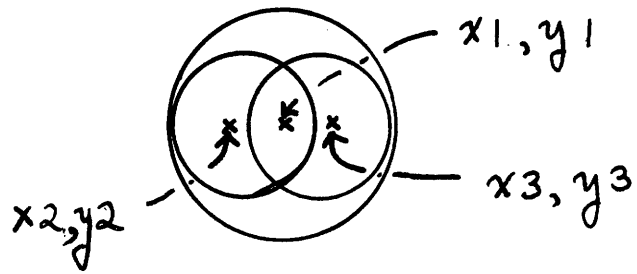


Figure #1

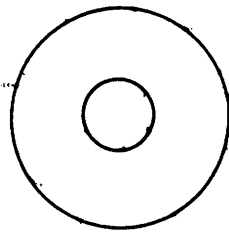


Figure #2

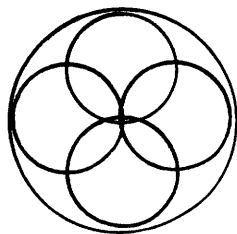


Figure #3

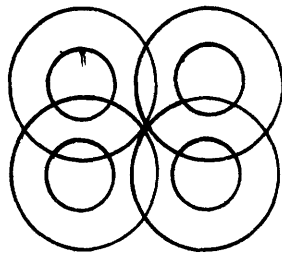


Figure #4

1

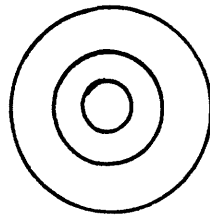


Figure #5

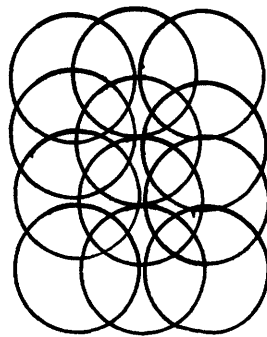


Figure #6

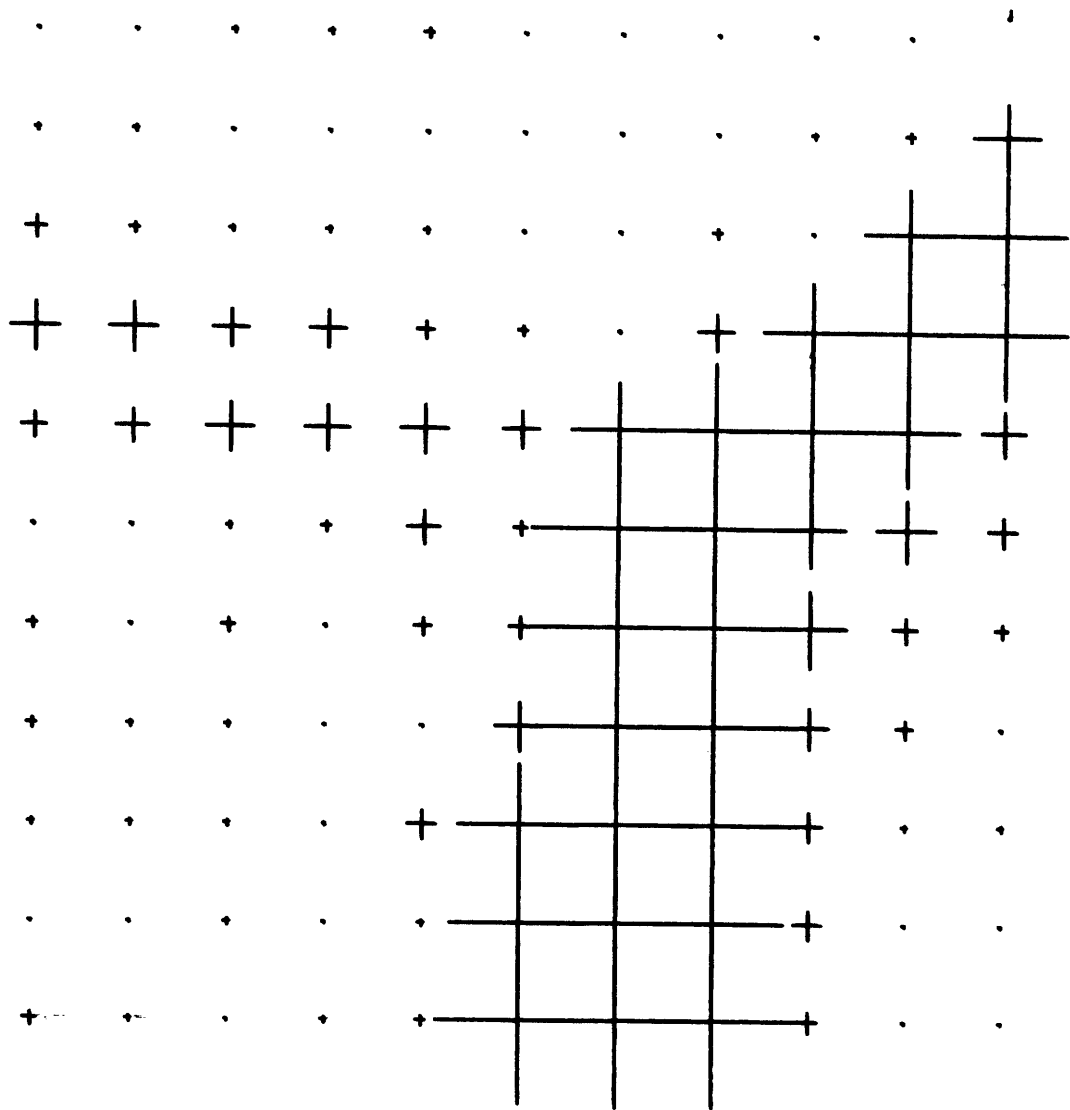


Figure #8

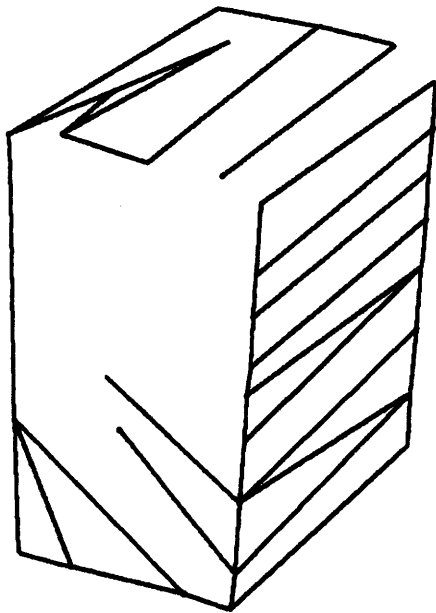


Figure #9

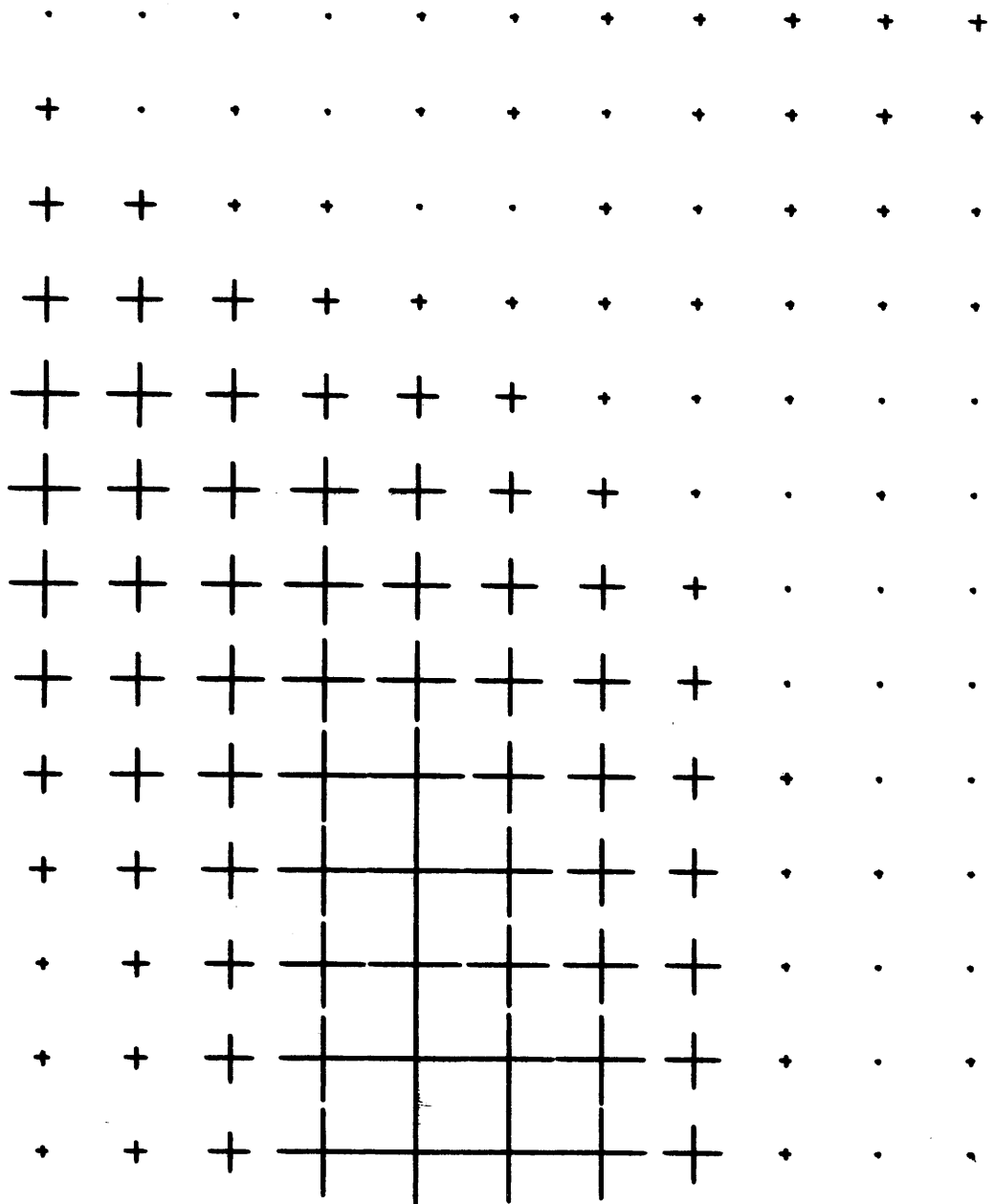


Figure #10

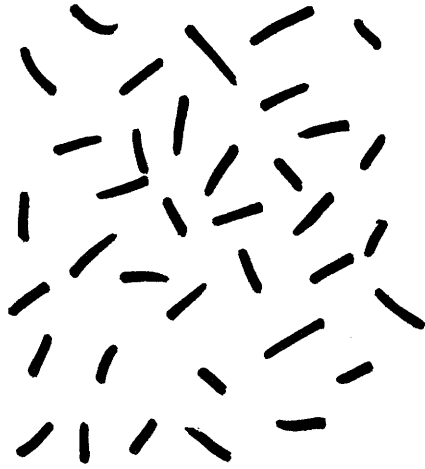


Figure #11

feature points



final result

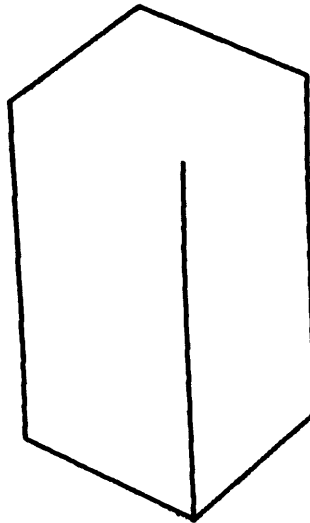


Figure #12

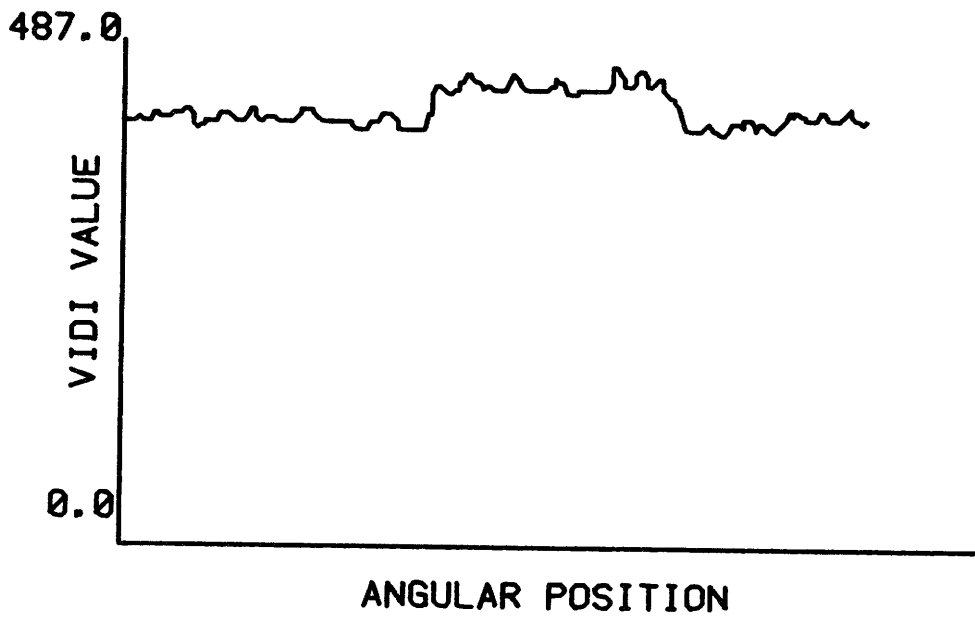
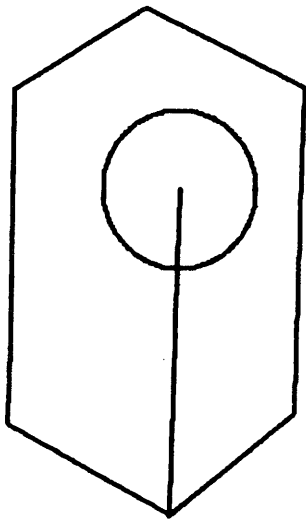


Figure #13