

(NASA-CR-138963) EDGE AND LINE DETECTION
IN ERTS IMAGERY: A COMPARATIVE STUDY
(Maryland Univ.) 17 p HC \$4.00 CSCL 14E

N74-29725

Unclas

G3/13 54719

22



UNIVERSITY OF MARYLAND COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

TR-312
NGR-21-002-351

June, 1974

EDGE AND LINE DETECTION
IN ERTS IMAGERY:
A COMPARATIVE STUDY

R. Eberlein
G. J. VanderBrug
A. Rosenfeld
L. S. Davis

ABSTRACT

Several local edge detection operators were applied to a set of ERTS pictures of the Monterey, Calif. area. Gradient operators performed consistently better than Laplacian operators in detecting edges. It was also found that if a grayscale normalization operation, "histogram flattening," was applied to the pictures first, the edge detector outputs were greatly enhanced. The use of interpolation for more accurate location of edges on a digital picture was also briefly investigated. Curve detection operators were applied to the edge detector outputs; this had the effect of enhancing the edges while suppressing noise.

The support of NASA's Goddard Space Flight Center, under Grant NGR-21-002-351, is gratefully acknowledged, as is the help of Andrew Pilipchuk and Shelly Rowe in preparing this report.

1. Gradients

The classical method of detecting edges -- i.e., abrupt changes in gray level -- in a picture is to apply an isotropic derivative operator such as the magnitude of the gradient

$$\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

or the Laplacian

$$\left| \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right|$$

to the given picture f [1]. For digital pictures, many different approximations to the gradient and Laplacian, based on differences rather than derivatives, have been used. One of the simplest examples is the "Roberts gradient," defined as

$$\max(|f_{ij} - f_{i+1,j+1}|, |f_{i+1,j} - f_{i,j+1}|)$$

where f_{ij} is the gray level at (i,j) . This uses first differences in the two diagonal directions, and takes their max rather than square root of sum of squares; it is an approximation to the gradient at $(i + \frac{1}{2}, j + \frac{1}{2})$.

A useful approach to the design of digital gradient operators is as follows [2]: We fit a polynomial or other standard function to the gray levels in a specified neighborhood of each point (i,j) . The gradient of the fitted function at (i,j) can be expressed in terms of the

coefficients of the function, which in turn can be expressed in terms of the gray levels at (i,j) and its neighbors. Thus, for any choice of neighborhood size and fitted function, we can define a digital gradient operator.

Suppose, for example, that we use a 3-by-3 neighborhood of (i,j) , and that we least-squares fit a second-degree polynomial in x and y to the nine gray levels

$$\begin{array}{ccc} f_{i-1,j+1} & f_{i,j+1} & f_{i+1,j+1} \\ f_{i-1,j} & f_{ij} & f_{i+1,j} \\ f_{i-1,j-1} & f_{i,j-1} & f_{i+1,j-1} . \end{array}$$

Then it turns out that the gradient of this polynomial at (i,j) , expressed in terms of the f 's, is

$$\begin{aligned} \max[& |f_{i-1,j+1} + f_{i,j+1} + f_{i+1,j+1} - f_{i-1,j-1} - f_{i,j-1} - f_{i+1,j-1}|, \\ & |f_{i-1,j+1} + f_{i-1,j} + f_{i-1,j-1} - f_{i+1,j+1} - f_{i+1,j} - f_{i+1,j-1}|]. \end{aligned}$$

We can express this more concisely by displaying the coefficients of the f 's in the two components of this gradient, namely

$$\begin{array}{ccc} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{array} \quad \text{and} \quad \begin{array}{ccc} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{array} .$$

We can similarly express the components of the gradient at $(i + \frac{1}{2}, j + \frac{1}{2})$, based on fitting a second-degree surface to a 4-by-4 neighborhood centered at $(i + \frac{1}{2}, j + \frac{1}{2})$; they are

3 3 3 3		3 1 -1 -3	
1 1 1 1		3 1 -1 -3	
	and	3 1 -1 -3	.*
-1 -1 -1 -1		3 1 -1 -3	
-3 -3 -3 -3		3 1 -1 -3	

Several of the gradient operators just described were applied to four windows (127 by 127 points each) taken from an ERTS 1 picture of the Monterey, Calif. area. Figures 1-4 show, respectively, the original pictures, their Roberts gradients, and the gradients based on second-degree least-squares fitting to 3-by-3 and 4-by-4 neighborhoods. The results, displayed on the same 0-63 gray scale as the original pictures, are very weak. They can be improved by rescaling, as in Figure 5, which shows the 4-by-4 gradient rescaled so that its highest values are now 63. However, if we re-scale each gradient output individually, we can no longer make legitimate comparisons among the different gradients. A different method of enhancing the gradient results will be described in Section 2.

* Similarly, for a 5-by-5 neighborhood centered at (i,j) we get

4 4 4 4 4		4 2 0 -2 -4	
2 2 2 2 2		4 2 0 -2 -4	
0 0 0 0 0	and	4 2 0 -2 -4	.
-2 -2 -2 -2 -2		4 2 0 -2 -4	
-4 -4 -4 -4 -4		4 2 0 -2 -4	

2. Histogram flattening

Much better edge detection results are obtained from our gradient operators if we apply a contrast enhancement operation, known as histogram flattening, to the pictures before we compute their gradients. The operation is designed to make each gray level in the enhanced picture occur equally often.

For an n -by- n picture that has m gray levels, we proceed as follows: Let S_0 be the n^2/m points of the original picture f whose gray levels are lowest; say these points have levels $0, 1, \dots, k_0$, where $k_0 \geq 0$. Then all points of f that have gray levels $0, 1, \dots, k_0-1$ get gray level 0 in the new picture \hat{f} . In addition, just enough points of f that have level k_0 are given level 0 in \hat{f} , to make up the desired total of n^2/m . These points can be chosen randomly; or we can rank the points having level k_0 according to the average gray levels of their neighbors, and choose the ones for which this average is lowest. Next, let S_1 be the n^2/m points of f having next lowest gray levels, say, k_0, k_0+1, \dots, k_1 , where $k_1 \geq k_0$. We give these points level 1 in \hat{f} , resolving ties as just described. The process continues with S_2, \dots, S_m ; at the last step, the n^2/m points of f that have the highest gray levels are given level m in \hat{f} .

The results of performing this operation on the pictures in Figure 1 are shown in Figure 6. It is seen that an increase in overall contrast has resulted. This is be-

cause most of the points in the original pictures had gray levels in the middle of the gray scale; the flattening operation had to "stretch" these in order to fill up the high and low levels on the new picture.

When the gradient operators described in Section 1 are applied to the histogram-flattened pictures, the results are much more pleasing, as shown in Figures 7-9. Note that for the upper left window, the results are not as good as those in Figure 5; this is probably because this window had heavily occupied high and low gray levels to begin with. Note also that the operators based on larger neighborhoods yield thicker edges; this is because a larger neighborhood responds to the presence of an edge over a wider range of positions.

3. Laplacians and other derivatives

The most commonly used class of digital approximations to the Laplacian are operators of the form $|f^{(s)} - f^{(r)}|$, where $f^{(k)}$ denotes the result of locally averaging f over a k -by- k neighborhood at each point. Figures 10-11 show the output of $|f^{(3)} - f^{(1)}|$ for the original Figure 1 and for its histogram-flattened version (Figure 6). As before, the latter results are much better. Figures 12-13 show analogous outputs for $|f^{(5)} - f^{(3)}|$; as we might expect, the edges are thicker for this coarser operator.

It is evident that the Laplacians are not as good edge detectors as the gradients for these pictures. This is because the edges in the pictures are step edges, to which the first-difference components of the gradient respond strongly. The Laplacians, on the other hand, being basically second-difference operators, respond better to spots than to step edges.

Although the Laplacian is not a very good edge detector, it can be used in conjunction with a coarse gradient to obtain thinned edges. This is because the Laplacians have their high values at the tops and bottoms of step (or ramp) edges, while the gradients yield peak values at the steepest points of the edges. Thus if we subtract the Laplacian from the gradient, and set negative values to zero, the gradient values should be suppressed at the tops and bottoms of edges, and preserved only right on the

edges. This effect is illustrated in Figure 14, which is the gradient Δ Laplacian of Figure 6.

The gradients defined by surface-fitting in Section 1 give less weight, at the point (i,j) , to the closer neighbors of (i,j) than to its more distant neighbors. To detect abrupt edges, one probably wants to use the opposite kind of weighting: high for near neighbors, lower for farther ones. Interestingly, this type of weighting can be observed by smoothing the picture before computing the gradient. For example, suppose that we smooth by averaging over a 3-by-3 neighborhood at each point, so that we obtain a new picture in which

$$\bar{f}_{ij} = f_{i-1,j+1} + f_{i,j+1} + f_{i+1,j+1} + f_{i-1,j} + f_{ij} + f_{i+1,j} + f_{i-1,j-1} + f_{i,j-1} + f_{i+1,j-1}.$$

If we apply to \bar{f} the gradient based on surface fitting to a 5-by-5 neighborhood, we can express the result in terms of a 7-by-7 neighborhood in f , with relative weights

2	3	3	0	-3	-3	-2
4	6	6	0	-6	-6	-4
6	9	9	0	-9	-9	-6
6	9	9	0	-9	-9	-6
6	9	9	0	-9	-9	-6
4	6	6	0	-6	-6	-4
2	3	3	0	-3	-3	-2

for the horizontal difference, and similarly for the vertical difference. The averaged version of the upper left

quadrant of Figure 1, using a 3-by-3 averaging neighborhood, is shown in Figure 15a; the gradients based on 3-by-3, 4-by-4, and 5-by-5 surface fitting, applied to the averaged picture, give the results shown in Figures 15b-d.

4. Interpolation

Our use of digital gradient operators based on interpolating continuous surfaces through the digital picture points suggests an important related question: Given a digital picture, can we locate the edges in the original scene to an accuracy greater than one digital picture point, using some type of interpolation?

To illustrate this problem, we have taken a 16-by-16 point piece of Figure 6, and blown it up to 64-by-64 in three ways, which are shown in Figure 16:

- a) Each point is simply copied 16 times, creating a 4-by-4 square of constant gray level.
- b) The points are copied into every fourth position (horizontally and vertically) in the blown-up picture, and gray levels are assigned to the intermediate points by linear interpolation, first in x , then in y .
- c) Similar to (b), except that cubic spline interpolation is used.

In the latter two cases, the blown up picture is only 61-by-61, since no extrapolation was performed.

We can now compute gradients, e.g., the 3-by-3 gradient, for blowups (b) and (c); the resulting edges, shown in Figure 17, can be regarded as interpolated between the points of the original 16-by-16 piece of picture. It appears that the spline technique gives edges that are more detailed and more sharply defined. Unfortunately, time did not permit further work on this approach.

5. Curve detection

As pointed out some years ago [3], one way to enhance the output of an edge detection operation is to apply curve detectors to it, since true region edges should be composed of curves, whereas noise output would not tend to lie on curves. It was also pointed out that still further noise suppression can be obtained by iterating the curve detection operation, since the true curve detector output should consist of curves, while the noise should not.

A simple curve detection operator can be defined as follows [3]: At each point (i,j) , we compute the average gray level in a small neighborhood, and compare it with the averages in adjacent, nonoverlapping neighborhoods on both sides, in some direction θ . Suppose that the average at (i,j) is greater than both adjacent averages in direction θ , by at least some amount t , and that the same is true for nearby points on each side of (i,j) in the directions $\theta \pm \frac{\pi}{2}$. Then a line or curve of slope $\theta + \frac{\pi}{2}$ must pass through (i,j) , since at both (i,j) and nearby points in that direction ("along" the curve), we have a "thin" locus of points whose average gray levels are at least t higher than their neighbors in the perpendicular direction ("across" the curve). Of course, this must be tested for many possible directions θ , unless we know in advance what slopes will be present. The detector output is the max, over all directions, of the average of the differences between the neighborhood averages and their adjacent

neighborhood averages.

This method of curve detection, using a 2-by-2 neighborhood at each point, was applied to the outputs of the gradient \pm Laplacian operator (Figure 14) for two of the quadrants of Figure 6. The results, for $t = 1, 2,$ and $3,$ are shown in Figures 18, 19, and 20, respectively. (These results have all been rescaled to improve their visibility.) Some noise survives when we threshold at 1, but some edges begin to break up when we threshold at 3. Much noisier results are obtained when we use single points rather than 2-by-2 neighborhoods; these results, for $t = 1, 2, 3,$ are shown in Figures 21-23.

The results of iterating the 2-by-2 curve detection scheme for one of the quadrants are shown in Figures 24 ($t = 1$ at each iteration), and 25 ($t = 2$). The second and third iterations suppress more of the noise, but some edges survive that were suppressed when we simply took $t = 3$ without iterating.

6. Conclusions

Several useful conclusions were derived, or reconfirmed, from this study:

- 1) Histogram flattening is a useful preprocessing operation for enhancing edge detector output.
- 2) Gradients perform better than Laplacians as edge detectors. However, subtracting a Laplacian from a coarse gradient can yield thinner edges.
- 3) Linear or spline interpolation can be used to locate edges to an apparent accuracy of better than one picture point.
- 4) Curve detection, possibly iterated, is a useful postprocessing operation for enhancing edge detector output.

Of course, these conclusions can only be accepted as valid for the limited class of pictures used in the study. To extend their validity to other cases, further study would be needed.

List of Figures

1. Input pictures: four 128-by-28 windows from an ERTS-1 Band 7 picture (1002-18134) of the Monterey, Calif. area.
2. Roberts gradients of the pictures in Fig. 1.
3. Gradients of the pictures in Fig. 1 based on fitting a second-degree surface to a 3-by-3 neighborhood at each point.
4. Analogous to Fig. 3, but using a 4-by-4 neighborhood.
5. Result of rescaling Fig. 4 to the range [0, 63].
6. Results of performing histogram flattening on Fig. 1.
- 7-9. Results of applying the gradient operators of Figs. 2-4 to Fig. 6 rather than Fig. 1.
10. Digital "Laplacians" $|f^{(3)} - f|$ for the pictures in Fig. 1.
11. Same for the enhanced pictures in Fig. 6.
- 12-13. Analogous to Figs. 10-11, but using $|f^{(5)} - f^{(3)}|$.
14. Result of subtracting the Laplacian $|f^{(3)} - f|$ from the 3-by-3 gradient; edges are thinner than those in Fig. 8.
15. (a) Result of 3-by-3 averaging of one of the pictures in Fig. 1. (b-d) Results of applying 3-by-3, 4-by-4,

and 5-by-5 gradients to (a).

16. Magnifications of a 16-by-16 piece of Fig. 6: (a) By copying the points (=zero-order interpolation); (b) By linear interpolation; (c) By cubic spline interpolation.
17. Results of applying the 3-by-3 gradient to Figs. 16(b-c).
- 18-20. Results of applying a curve detection operation, using 2-by-2 neighborhoods, to two quadrants of Fig. 14, with a difference threshold t equal to 1, 2, and 3, respectively.
- 21-23. Analogous to Figs. 18-20, but using single points rather than 2-by-2 neighborhoods.
24. Results of iterating the 2-by-2 curve detection operation of Fig. 18 on one quadrant: (a) twice, (b) three times.
25. Analogous to Fig. 24, for the operation of Fig. 19.

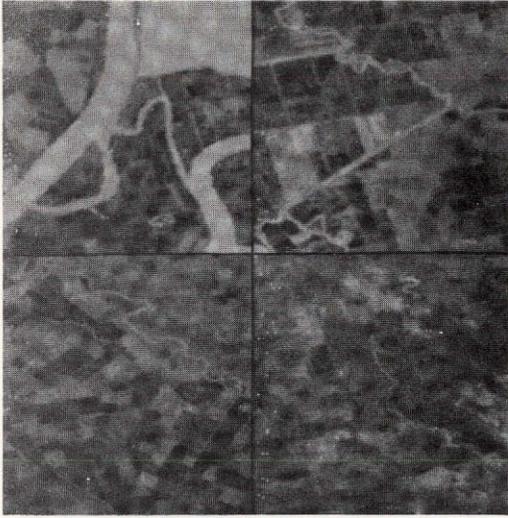


Fig. 1

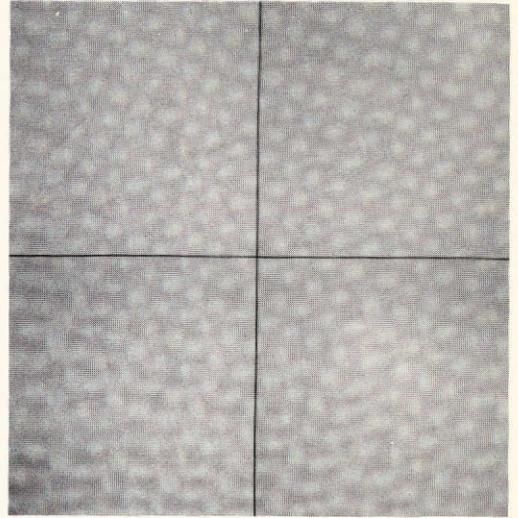


Fig. 2

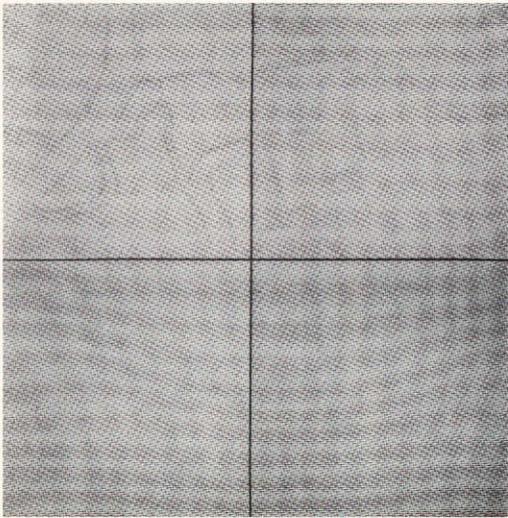


Fig. 3

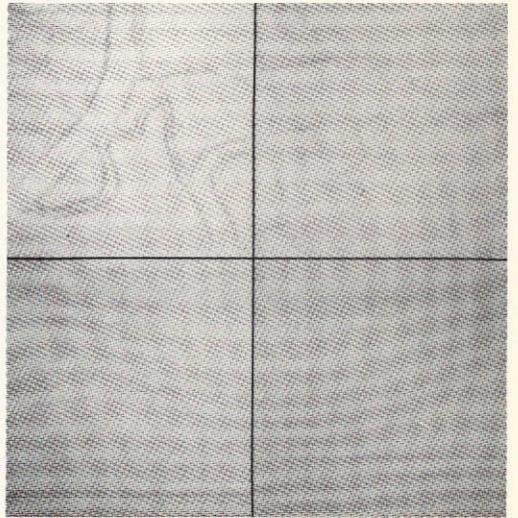


Fig. 4

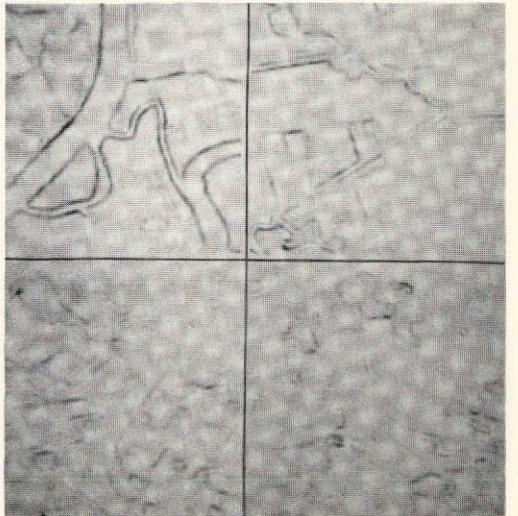


Fig. 5

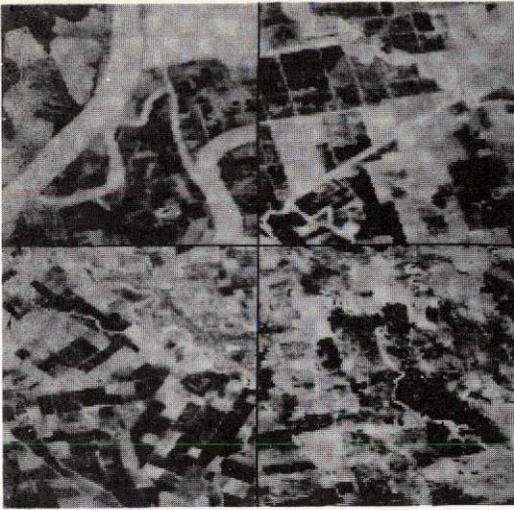


Fig. 6

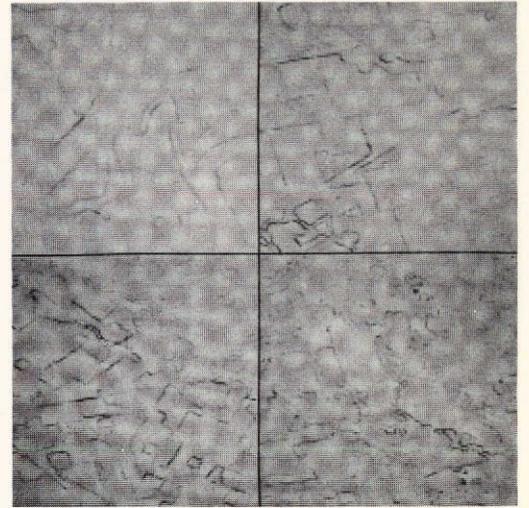


Fig. 7

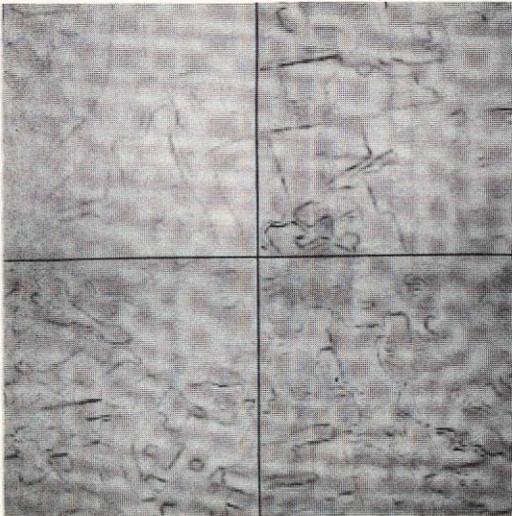


Fig. 8

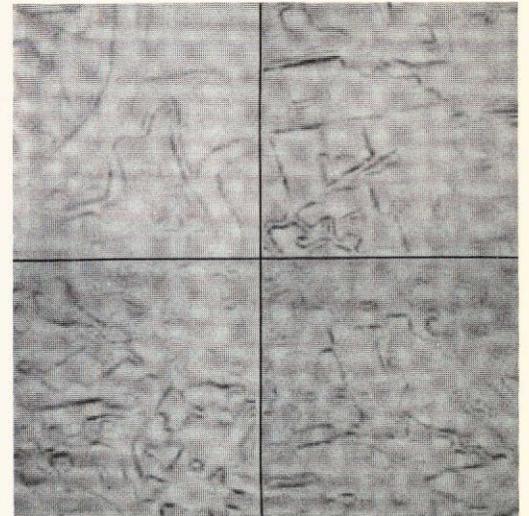


Fig. 9

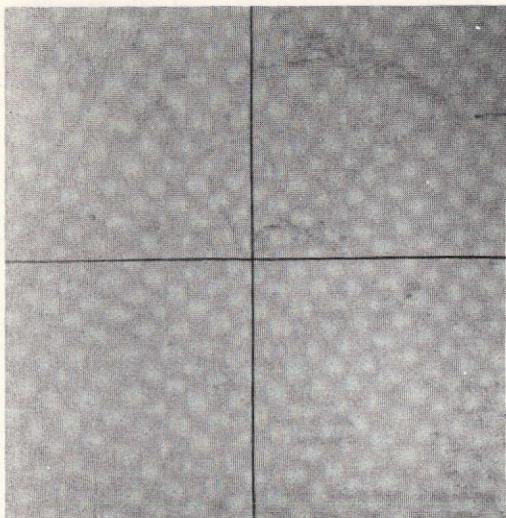


Fig. 10

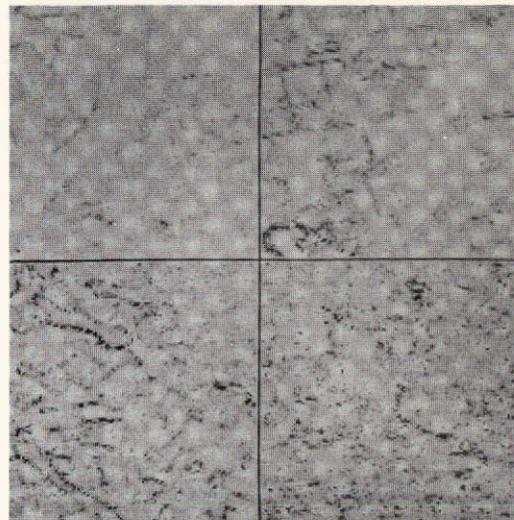


Fig. 11

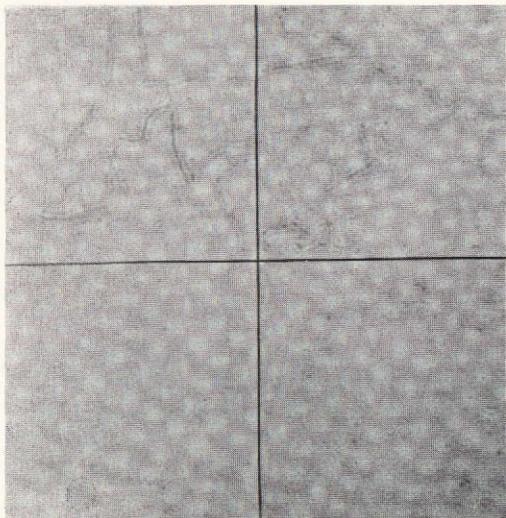


Fig. 12

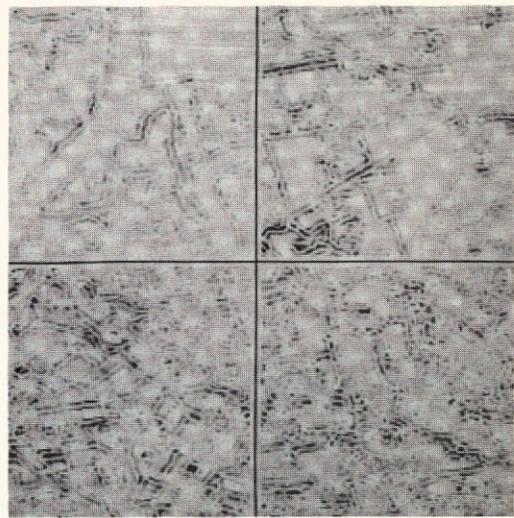


Fig. 13

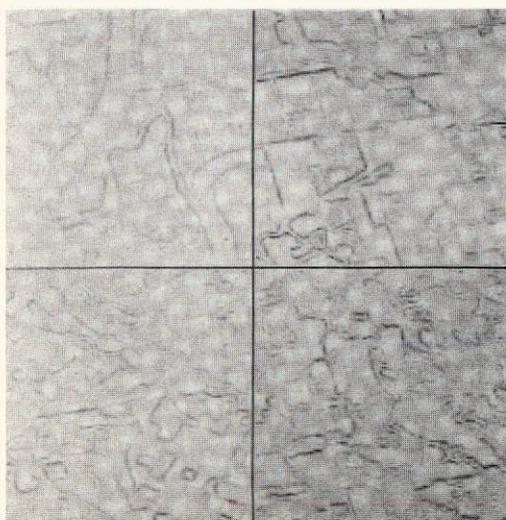


Fig. 14

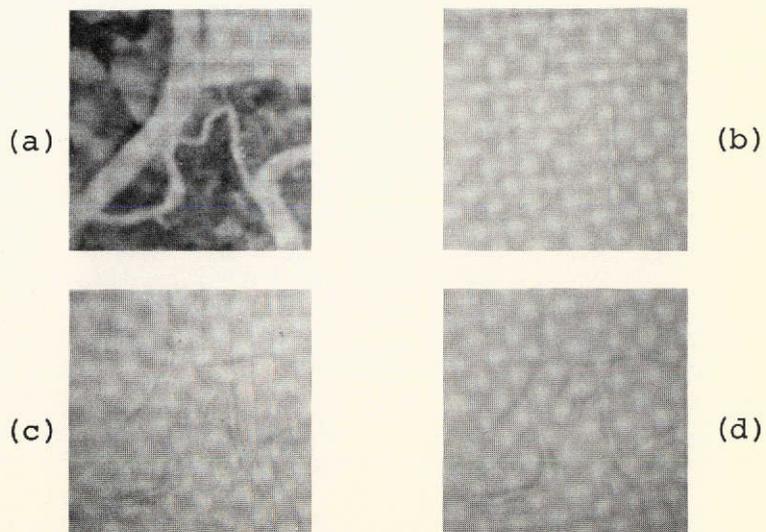


Fig. 15



Fig. 16a



Fig. 16b



Fig. 16c



Fig. 17

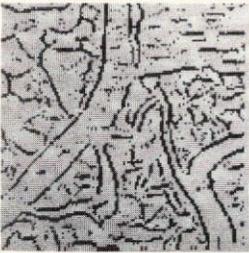


Fig. 18

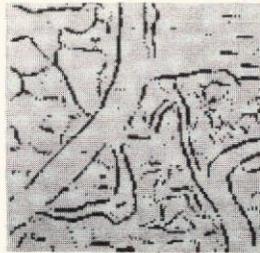


Fig. 19

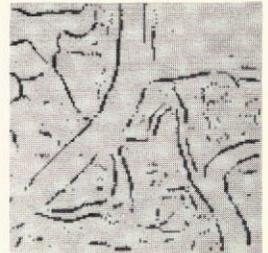
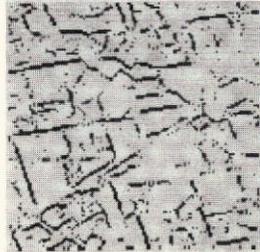


Fig. 20



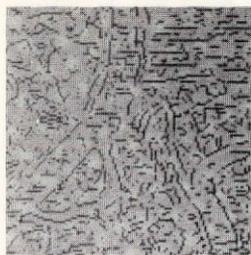


Fig. 21

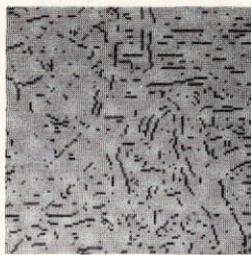


Fig. 22

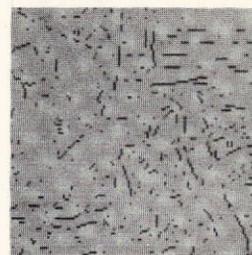


Fig. 23

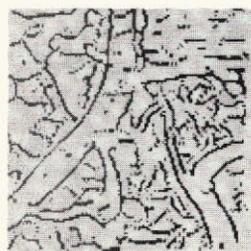
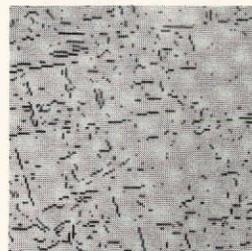
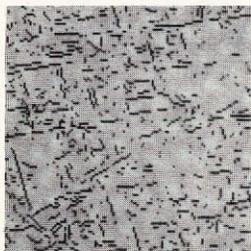
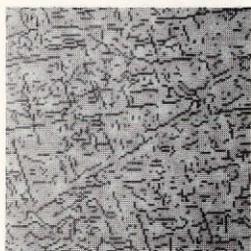


Fig. 24a



Fig. 24b

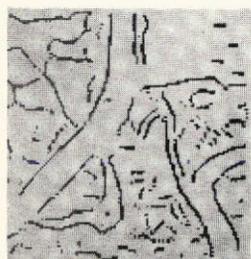


Fig. 25a

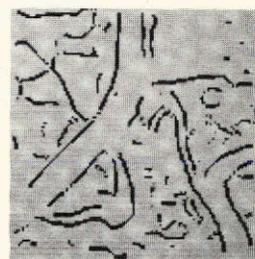


Fig. 25b

References

1. L.S.G. Kovaszny and H.M. Joseph, Image processing, Proc. IRE 43, 1955, 560-570.
2. J.M.S. Prewitt, Object enhancement and extraction, in B.S. Lipkin and A. Rosenfeld, eds., Picture Processing and Psychopictorics, Academic Press, New York, 1970, 75-149.
3. A. Rosenfeld, Y.H. Lee, and R.B. Thomas, Edge and curve detection for texture discrimination, ibid., 381-393.