An Application of Relaxation Labeling to Line and Curve Enhancement

STEVEN W. ZUCKER, MEMBER, IEEE, ROBERT A. HUMMEL, AND AZRIEL ROSENFELD, FELLOW, IEEE

Abstract—A relaxation process is described and is applied to the detection of smooth lines and curves in noisy, real world images. There are nine labels associated with each image point, eight labels indicating line segments at various orientations and one indicating the no-line case. Attached to each label is a probability. In the relaxation process, interaction takes place among the probabilities at neighboring points. This permits line segments in compatible orientations to strengthen one another, and incompatible segments to weaken one another. Similarly, no-line labels are reinforced by neighboring no-line labels and weakened by appropriately oriented line labels. This process converges, in only a few iterations, to a condition in which points lying on long curves have achieved high line probabilities, while other points have high no-line probabilities. There is some tendency, under this process, for curves to thicken; however, a thinning procedure can be incorporated to counteract this. The process is effective even for curves of low contrast, and even when many curves lie close to one another.

Index Terms—Curve detection, line detection, picture processing, relaxation, scene analysis.

I. Introduction

MOST approaches to locating lines or curves in pictures begin by applying local line detectors [10]. These detectors are operators which perform measurements on a small neighborhood of the original picture. A strong response from a detector is usually interpreted as an indication of the presence of a line segment in that neighborhood. Once detected, these line segments are joined into more global lines or curves.

When the curves in the picture are relatively perfect, i.e., when they are without gaps and when they differ everywhere from a background which is noise-free, the single interpretation of the detector's response is reliable. Curves in real world imagery, however, are not that clean. The local detectors often return responses which are due primarily to noise effects: a strong response may occur when no line segment is present, or, on the other hand, a weak response may occur when a segment is present. There is no one-to-one relationship between a detector response and the existence of a line segment. Thus, in order to interpret the detector responses correctly, additional information must be used.

Manuscript received January 13, 1976; revised May 10, 1976. This research was supported by the National Science Foundation under Grant DCR-72-03610-A03.

S. W. Zucker was with the Computer Science Center, University of Maryland, College Park, MD 20742. He is now with the Department of Electrical Engineering, McGill University, Montreal, P.Q., Canada.

R. A. Hummel and A. Rosenfeld are with the Computer Science Center, University of Maryland, College Park, MD 20742.

In limited domains, such as the blocks world, semantic knowledge about the physical structure of blocks can be used as an aid in the line detection (Shirai [13]; see also Zucker *et al.* [18]). More generally, the pictorial context can be used to disambiguate a detector's response. One way to do this is to evaluate detectors of many different sizes and suppress nonmaximal responses (Rosenfeld [12]). Other methods use the orientation information to bridge gaps between segments (e.g., O'Gorman and Clowes [8]). This enables tracking algorithms to travel from one strong detector response to another and thus extract the legitimate lines or curves (e.g., Horn [4]). However, since the raw detector responses are often ambiguous (or incorrect), and since tracking algorithms are highly sequential, many wrong turns are taken. To recover, significant backtracking is necessary. This lack of global perspective can be overcome by the application of optimization techniques, but only at greatly increased computation cost (e.g., Martelli [6] and Montanari [7]). Such techniques require, in principle, knowledge of the space of all possible curves in the picture before the optimum can be selected.

One way to improve the various methods for finding curves would be to improve the output of the local line detectors. One of the purposes of this paper is to develop a parallel relaxation technique which enhances these detector outputs. At the same time it aids in changing the detectors' responses into a symbolic form (see Marr [5] for a discussion of the use of symbols in low-level vision). The parallelism is appropriate because the context around every point is useful in interpreting the response of the detector centered at that point. The symbols correspond to assertions about the existence of line segments with given orientations. The relaxation process then attempts to accumulate evidence useful for deciding which label assertions are most appropriate. Placing the detector responses in this symbolic form makes the subsequent task of following curves much more tractable.

II. Relaxation Labeling

The present approach to curve enhancement derives from a previous theoretical study of labeling by relaxation processes [9]. In that study relaxation was shown to be an effective technique for resolving labeling ambiguities by using contextual information. For curve enhancement, the ambiguities in detector responses are resolved by using

information from the region surrounding each detector. Before the new approach to line enhancement is detailed, however, a general review of relaxation labeling will be given.

Relaxation operates on objects with labels attached to them. Then, through the use of appropriately defined compatibility relationships between labels, some labels are strengthened and some are weakened or eliminated. More specifically, let $a = \{a_1, a_2, \cdots, a_n\}$ be a set of objects, perhaps derived from a picture. Let $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_m\}$ be the set of labels which indicate possible interpretations for these objects. For example, in the curve enhancement application, the objects are the individual picture points and the labels correspond to interpretations of these picture points either as part of a curve with some orientation, or as not part of a curve. For another example see Waltz [17].

Attached to each label λ is a quantity $p_i(\lambda)$ which denotes the probability (or, more precisely, an estimate of the probability) that λ is the correct label for the object a_i. These probabilities must always satisfy the condition:

$$\sum_{\lambda \in \Lambda_i} p_i(\lambda) = 1, \qquad i = 1,2,\cdots,n \tag{1}$$

where Λ_i is the label set for object a_i.

In addition to the foregoing, a neighborhood relation is specified on the set of objects. The relaxation process updates the estimated probabilities by iterating an operator which we may call F. At each iteration, the new probabilities $p_i^{(k+1)}(\lambda)$ are functions of both the old probabilities $p_i^{(k)}(\lambda)$ on the given object, and the old probabilities $p_j^{(k)}(\lambda')$ on the neighboring objects.

There are three basic kinds of contributions that one label λ' can make to another label λ on a neighboring object:

1) if λ' and λ are semantically compatible, i.e., if they co-occur frequently with one another on neighboring objects, then $p(\lambda')$ should contribute positively to the computation for $p(\lambda)$;

2) if λ' and λ are semantically incompatible, i.e., if the occurrence of one implies that the other cannot also occur, then $p(\lambda')$ should contribute negatively to $p(\lambda)$;

3) if there is no relation between the two labels, then one should not influence the other.

In general there are many intermediate situations, and compatibility functions must be used to weight the interactions between pairs of labels. Compatibility functions behave much like correlation functions between events of the form "a_i has label λ" and "a_j has label λ'." A compatibility function can be defined as a mapping

$$r_{ij}: \Lambda_i \times \Lambda_j \rightarrow [-1,1]$$

such that for condition 1) above, r_{ij} approaches 1; for condition 2), r_{ij} approaches -1; and, for condition 3), r_{ij} is near 0. Thus, both positive and negative influences can propagate during relaxation. (See Arbib [1] for a discussion of a possible role such cooperating and competing processes may play in brain theory.)

The updating process can now be expressed in terms of these compatibility functions. Let $q_i^{(k)}(\lambda)$ represent the correction applied to $p_i^{(k)}(\lambda)$ in the $(k+1)$st iteration to obtain $p_i^{(k+1)}(\lambda)$:

$$p_i^{(k+1)}(\lambda) = \frac{p_i^{(k)}(\lambda)[1 + q_i^{(k)}]}{\sum_{\lambda} [p_i^{(k)}(\lambda)[1 + q_i^{(k)}]]} \tag{2}$$

The denominator in (2) is a normalization factor necessary to guarantee that the sum of probabilities (1) remains equal to 1. (See [9] for a more detailed discussion of this equation.) This correction can be defined as a weighted sum of the probabilities on the labels attached to neighboring objects acting through the compatibility functions:

$$q_i^{(k)}(\lambda) = \sum_j d_{ij} \left[\sum_{\lambda'} r_{ij}(\lambda,\lambda') p_j^{(k)}(\lambda') \right]. \tag{3}$$

The coefficients (d_{ij}) weight the total influence that object a_j can have on a_i, subject to $\Sigma_j d_{ij} = 1$. Thus, in general, the final distribution of probabilities on the label sets is dependent both upon the initial assignment of the probabilities and upon the compatibility functions between labels.

The relaxation process operates in this way until limiting values are obtained for the label probabilities. A limiting probability value of 1 denotes an unambiguous label and a value of 0 denotes an impossible label. In some cases, the limiting values may be strictly between 0 and 1, but we have not been able to obtain a mathematical characterization of the cases in which this is true; for further discussion of this, and some examples (involving contradictory initial evidence), see [9].

III. Line and Curve Enhancement

In the following approach to line enhancement, a picture is considered to be an array of intensity values. This array specifies a set of objects, the picture points, with the standard 8-neighbor relations defined between them [10]. Associated with each picture point is a set of $(K+1)$ labels. The first K labels $\lambda_1, \lambda_2, \cdots, \lambda_K$ correspond to unit line segments at orientations $\theta_1, \theta_2, \cdots, \theta_K$, respectively. The final label λ_{K+1} corresponds to the case in which no line is present. For the current implementation, $K = 8$ distinct orientations were selected, with $-\pi/2 < \theta_k < \pi/2$. Since line segments are undirected, orientations are symmetric modulo π.

The initial probability for each label is obtained by evaluating the nonlinear detector (Rosenfeld [11]) at every picture point in the eight orientations (Fig. 1). If the detector's response is strong for only one orientation θ_k, then the initial probability $p_i^{(0)}(\lambda_k)$ is set to be high and the

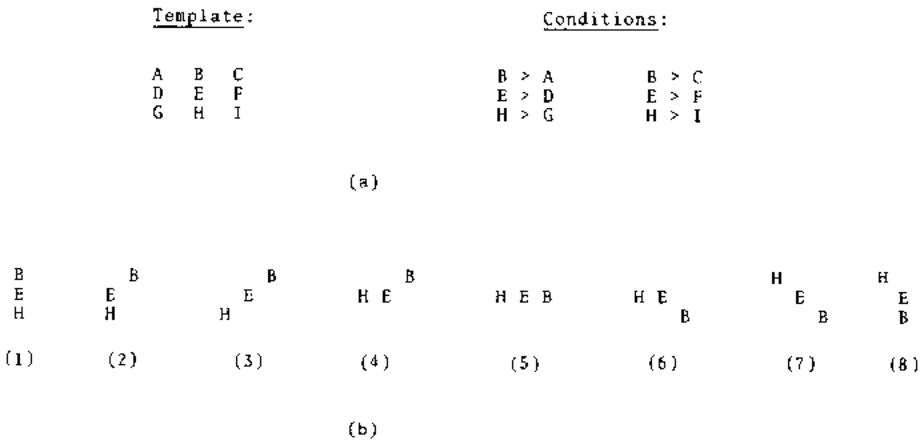


Fig. 1. (a) Nonlinear line detector for the vertical orientation. (Each letter denotes the average of a two-by-two pixel block.) When the conditions do not hold, the response is zero; when they do hold, the response is $(B + E + H) - \frac{1}{2}(A + D + G + C + F + I)$. (b) Eight orientations of the detector.

probabilities on the other labels $\{p_i^{(0)}(\lambda_q), q \neq k)\}$ at that position are set to be low. These low probabilities are not set to 0, however, because the strong response could have been caused by a noise point. If there is no strong response from any of the detectors at some position, then the no-line probability is set high. It is never set to 1, because the picture point may be a gap in a curve. For the many intermediate situations, the probabilities should be distributed over all the labels.

To define this process more precisely, let $\mu_k(x,y)$ denote the output of the detector with orientation θ_k at position (x,y). Let $p_{(x,y)}(\lambda_k)$ denote the probability of label λ_k at (x,y). Then the initial probabilities can be obtained by scaling the detector responses at each position (x_0,y_0). These scaled responses must then be normalized so that (1) is always satisfied:

$$p^{(0)}_{(x_0,y_0)}(\lambda_k) = \frac{\max_{n=1}^{8} \mu_n(x_0,y_0)}{\max_{m=1}^{8} \max_{(x,y)} \mu_m(x,y)} \cdot \frac{\mu_k(x_0,y_0)}{\sum_{q=1}^{8} \mu_q(x_0,y_0)},$$

$$k = 1,2,\cdots,8$$

$$p^{(0)}_{(x_0,y_0)}(\lambda_9) = 1 - \frac{\max_{n=1}^{8} \mu_n(x_0,y_0)}{\max_{m=1}^{8} \max_{(x,y)} \mu_m(x,y)}.$$

An additional small constant correction must also be applied to the above expressions to insure that no probability is initially set to zero. Scaling the detector responses by the maximum over the entire picture guarantees that the initial probabilities are assigned conservatively. This assumes, in effect, that the strongest detector response is the most reliable.

To update the probabilities, the compatibility relations $r_{ij}(\lambda,\lambda')$ between labels must be specified [see (3)]. Since there are nine labels attached to each point, every pair of points could require an associated 9×9 matrix of compatibilities. But, because of the symmetry inherent in lines

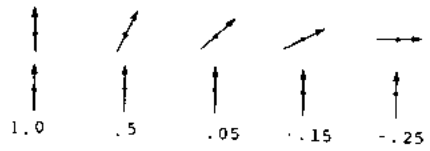


Fig. 2. Compatibility weights between line labels.

and curves, the compatibilities depend only on the relative orientations of the neighboring segments. These compatibilities can be specified in the following way. If two neighboring line segments are oriented in the same direction or close to the same direction, they add support to one another [condition 1) in Section II]. If, on the other hand, two neighboring segments are oriented perpendicularly to one another, they subtract support [condition 3)]. All other pairs of line segments are distributed between these two extremes. Fig. 2 shows one such smooth distribution of compatibility weights between pairs of line labels.

In the current implementation, the updating of the line label probabilities can be expedited by examining only a portion of the neighborhood around each point. For example, suppose that the label probability for a line segment in direction θ_k is being updated. Then, in the 5×5 neighborhood surrounding the given point, the positions which enter the computation are those which lie approximately in the directions θ_k and $-\theta_k$ (Fig. 3). The total contribution from each point within this core of influence can be given even more weight, if necessary, by changing the values of the coefficients (d_{ij}) in (3). For instance, to bias the process in favor of straight lines, the contribution from points lying directly on the orientation axis could be emphasized more heavily than those just off axis.

To update the probability of the no-line label at a point, all eight of its closest neighbors are examined. The no-line label is supported positively by adjacent probabilities for no-line labels and negatively by adjacent probabilities for line labels oriented toward the point. This latter condition tends to suppress the no-line label at points which are actually small gaps in lines, and, in so doing, it indirectly enhances the appropriate line label.

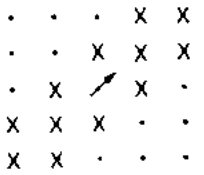


Fig. 3. The neighboring points X used for updating the probability on the label →. (For this orientation, twelve points are necessary.)

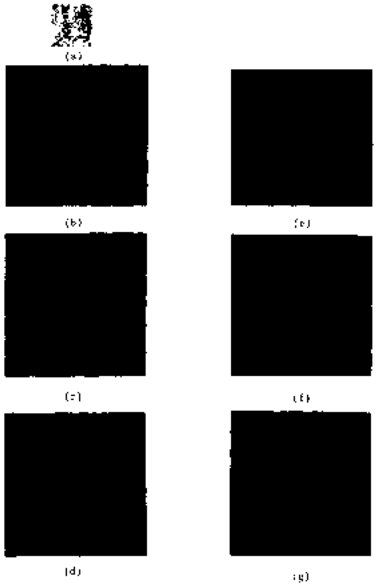


Fig. 4. (a) Computer-generated line in noise. (b) Initial probability assignments for (a), obtained from nonlinear line detector responses. (c)–(g) Iterations 1–4, and 8 of the relaxation process applied to (b).

All of the compatibility functions discussed in this section were defined between pairs of labels attached to adjacent pictorial positions. In Section V a new compatibility function will be defined which relates a label probability not to another single label probability, but rather to the difference in probability between two labels attached to positions once removed from adjacency. This more complex compatibility function will be used to prevent thickening of the lines.

IV. Experiments with Satellite Imagery

In order to evaluate the line enhancement process on a meaningful sample set, satellite terrain images were selected. Satellite imagery is rich in what geologists refer to as linear features, e.g., rivers, roads, vegetation alignments, and geological faults [14]. These characteristics provide a rich source of applications, as well as a common ground for integrating this present study into earlier work (e.g., VanderBrug [15], Bajcsy and Tavakoli [2], [3]).

However, before the more complex terrain results are presented, a simpler example is given to illustrate the basic operation of the process. In Fig. 4(a) a computer generated line is embedded in random noise. Fig. 4(b) displays the initial probability assignments obtained from the nonlinear detector responses.[1] In this larger output format, each position in the original image [Fig. 4(a)] corresponds to a small square region. The line label (at each position) that has the highest probability is displayed inside the corresponding square with an intensity proportional to the probability distributed on it.

For this picture the local detectors return significant responses not only from positions on the visually distinct line, but also from within the noise pattern (see VanderBrug [15] for a detailed discussion of the operation of various line detectors on similar pictures). In addition, gaps and other orientation anomalies exist in the maximal responses along the line. It is clear from this display that any sequential tracking algorithm would have substantial difficulty operating on these raw detector responses.

The rest of Fig. 4 shows eight iterations in the relaxation process. Only one long sequence of aligned segments is enhanced. After enhancement, this sequence can be tracked easily by standard algorithms [10]. Furthermore, the probabilities attached to the line segments (i.e., labels) in this sequence are converging to 1. This indicates a high

[1] The double response along the line is the result of local smoothing (over 2 × 2 pixel neighborhoods) which precede the line detection operation (see Fig. 1).

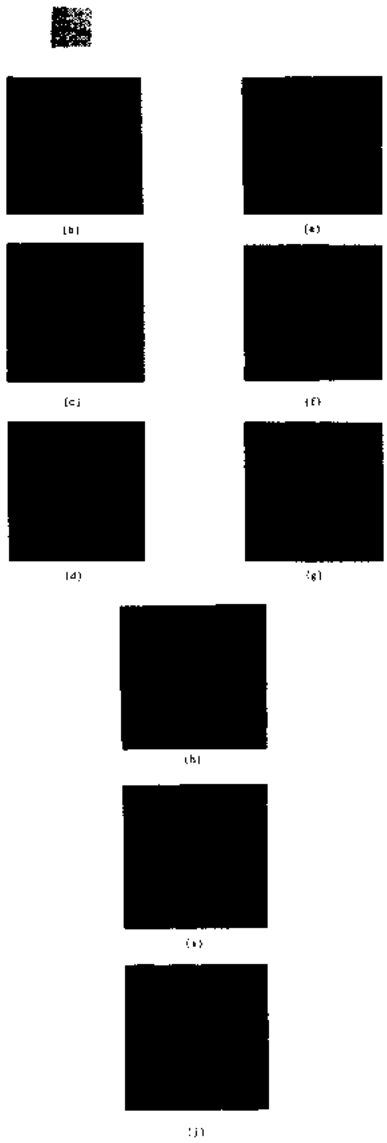


Fig. 5. (a) LANDSAT subimage containing river. (b) Initial probability assignments for (a). (c)–(g) Iterations 1–5 of the relaxation process applied to (a). (h)–(j) Iterations 6–8 of the relaxation process applied to (a).

level of confidence in them. The other labels which were initially possible at these positions have become either extremely unlikely or impossible.

In addition to the single long sequence in the final iteration, there are several isolated clusters of line segments. These clusters are a result of noise-based detector responses which occasionally align by chance. To eliminate them, it becomes necessary to use a process with a more global perspective than relaxation, such as a tracking algorithm.

The compatibility coefficients of Fig. 2, which were used for the straight line example, also work well for the curved lines which occur in terrain pictures. Fig. 5(a) shows a winding river in a LANDSAT image of Kentucky. The rest of Fig. 5 displays eight iterations of the relaxation process. Note especially how the orientations become aligned around the curves.

The robustness of the compatibility coefficients for both the straight line example and the curved line example also extends over more complex classes of pictures. This will be shown using additional examples. This robustness can be accounted for by considering the relaxation process not as a single process over the nine labels, but rather as two subprocesses in competition with one another. One subprocess consists of all the line labels attempting to gather support from their neighbors, and the other subprocess consists of the no-line labels attempting to gather support. When the former subprocess dominates at a point, a line label is placed there; when the latter subprocess dominates, a no-line label is established. All of the mixed coefficients (i.e., those between line pairs with different orientations, or those between line/no-line pairs) enter into the computations for both of these subprocesses. Thus, it is not their specific values which are important but rather their proportionality relationships.

In some specific applications, robustness may not be

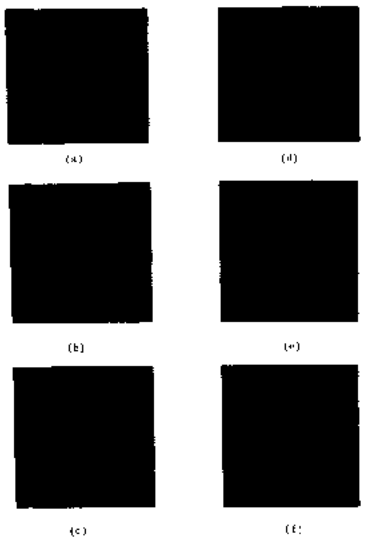


Fig. 6. (a)–(f) Iterations 1–5 and 8 of a modified relaxation process, biased in favor of straight lines, applied to Fig. 4(a).

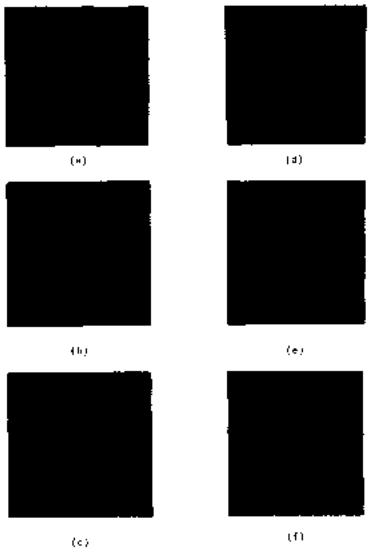


Fig. 7. (a)–(f) Iterations 1–5 and 8 of a modified relaxation process, without noise cleaning capability, applied to Fig. 4(a).

desirable. In this event the proportionality relationships between the compatibilities must be changed dramatically to favor the special circumstances. For example, to favor only very straight lines, the coefficients of Fig. 2 might be changed to the proportions (1.0, 0, −0.1, −0.17, −0.22). This removes all of the intermediate positive support and replaces it with zero or negative support. The result of applying these new coefficients to the line in noise example of Fig. 4(a) is shown in Fig. 6. Almost all of the smaller clusters have been eliminated. For the remaining examples in this paper, however, the standard proportions (Fig. 2) are used.

The no-line/no-line compatibility plays a more crucial role in line enhancement than the mixed compatibilities. It regulates the no-line subprocess with respect to the line subprocess. By reducing the no-line/no-line compatibility to an ineffective level, all competition from the line subprocess is removed. This enables every cluster of no-line labels to be reinforced (Fig. 7). Increasing the compatibility back to a functional level leads to more victories for the no-line label and performs, in a practical sense, "noise cleaning." Appropriate values for this compatibility can be selected from a rough estimate of pictorial noise;[2] it is at about the same level of effectiveness for most of the examples in this paper.

Another notable feature of the results in Fig. 7 is that the lines have thickened extensively. Thickening occurs when an incorrect line label, with a nonzero probability on it, is supported by several strong labels in its immediate neighborhood. Points bordering thick lines are especially susceptible.

In Fig. 7 thickening occurred because every line label

[2] To see the negligible effect of small variations (approximately 15 percent) in this compatibility, compare Fig. 10(g) and (h).

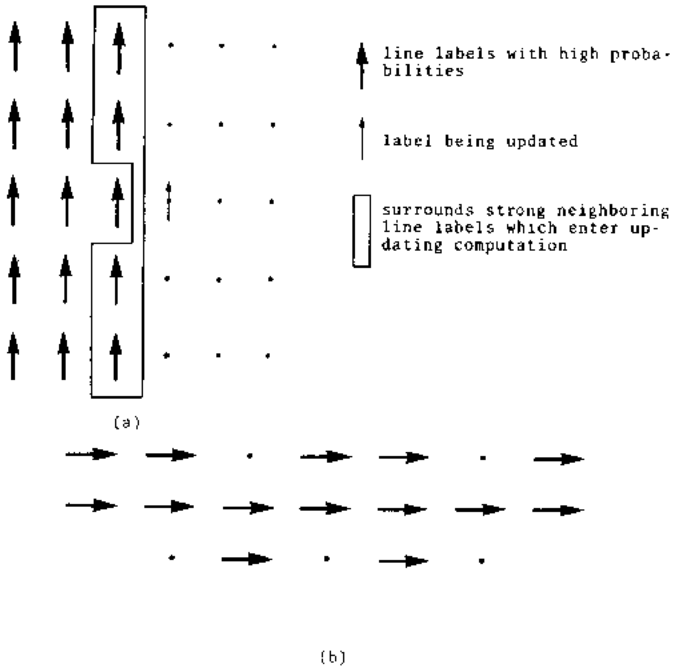


Fig. 8. Conditions for line thickening: (a) Points near thick line in noisy background. (b) Points in concavities of line of varying thickness.

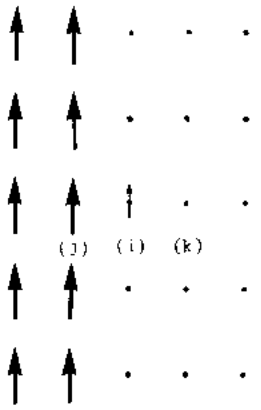


Fig. 9. Neighbors (j) and (k) of a point (i) in the direction perpendicular to a thick line used to detect thickening.

received only positive support—there was no competition from the no-line subprocess. However, with the no-line/no-line compatibility at a functioning level, thickening may occur if the total positive support that a line label receives from its line neighbors exceeds the total negative support it receives from it no-line neighbors. Such circumstances often arise for points bordering thick lines in noisy background [Fig. 8(a)] and for points lying in the concavities formed by lines which vary in thickness [Fig. 8(b)]. In order to counter this tendency, a thinning algorithm, developed in the next section, has been incorporated into the relaxation process.

V. Curves Irregular in Thickness and in Contrast

Thickening occurs when a strong line or curve incorrectly dominates the points which border it. Suppose, for example, that a point (i) borders a thick line in direction θ. One way to detect when thickening may occur at (i) is to examine its neighbors (j) and (k) which lie on an axis perpendicular to the thick line (Fig. 9). The point (j) lying just inside the line should have a high probability on its line label λ_θ in the direction θ, and the point (k) lying outside should have a low probability on its λ_θ. In terms of the actual label probabilities, this translates into the condition that the absolute difference

$$D_{j,k}(\lambda_\theta) = |p_j(\lambda_\theta) - p_k(\lambda_\theta)|$$

is large, where (j) and (k) are the points neighboring (i) in the directions $\theta - \pi/2$ and $\theta + \pi/2$, respectively.

This observation on the distribution of label probabilities can be incorporated into the updating computation by defining a compatibility coefficient $r_{i,D_{j,k}}(\lambda_\theta,\lambda'_\theta)$ between $p_i(\lambda_\theta)$ and $D_{j,k}(\lambda'_\theta)$. This compatibility should be such that

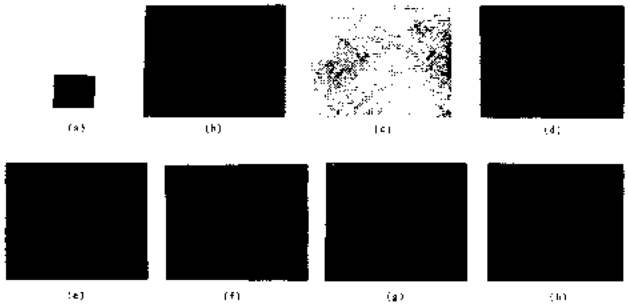


Fig. 10. Enhancement of faint curves. (a) SKYLAB subimage containing concrete and asphalt roads. (b) Initial probability assignments for (a). (c) "Pseudo-complement" of (b) to emphasize the low probabilities (dark lines). (d)–(g) Iterations 0, 2, 4, 6, and 7 of a modified relaxation process, incorporating thinning capability, applied to (a). (h) Iteration number 7 for a relaxation process, like that in Fig. 10(d)–(g), but with the no-line/no-line compatibility increased 15 percent.

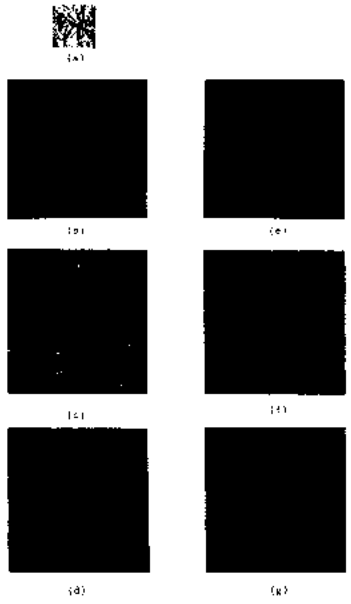


Fig. 11. (a) LANDSAT image containing many lines (geological linear features). (b)–(g) Iterations 0, 1, 2, 3, 5, and 7 of the same process used in Fig. 10, applied to (a).

$D_{j,k}(\lambda'_\theta)$ contributes negatively to the computation of $p_i(\lambda_\theta)$. When the difference $D_{j,k}(\lambda'_\theta)$ is large, the label λ_θ at (i) is weakened, and the tendency for lines to thicken is countered.

For the last two examples (Figs. 10 and 11) in this paper, thinning was in operation during the relaxation process. These figures contain many places where thickening could be a problem, because the lines and curves vary a great deal in width. Note, however, that the effect of thickening is minimal (also compare Fig. 12 with Fig. 11(g) to see the result of eliminating the thinning step).

Fig. 10 illustrates another property of lines in real world imagery—variability in intensity. Fig. 10(a) is a Skylab photograph of a suburban area in Maryland. It contains a road (traveling from the lower left to the upper right) whose composition changes from concrete to asphalt. The accompanying change in reflectivity for this road makes the upper portion almost indistinguishable from the background. Also there is a dim section in the lower, left-hand portion. Both of these sections, especially the upper continuation, are very difficult to find using traditional techniques such as thresholding [10]. Relaxation, however, performs very noticeable enhancement: it eliminates the intensity changes, and it makes the continuation through the upper right-hand corner much more prominent.

VI. Interactions Between Curves

All the examples that have been discussed so far have focused on pictures which contain only a few long lines and curves. If a picture contains many lines and curves, there is a possibility that these may lie close enough together to influence each other unduly. Since the size of the effective context around each picture point grows with the number of iterations, it would seem increasingly likely, during relaxation, that this interaction would occur. As it happens, however, the lines and curves do maintain their autonomy.

Fig. 12. The seventh iteration of the same process as that in Fig. 11, except without thinning.

For example, one type of competitive interation between two lines occurs at a junction where they cross (see Fig. 10). Both lines exert enough influence over the junction so that segments oriented in both directions survive. The problem of determining which line should dominate at the intersection is resolved by preserving both. This result is consistent with the ambiguity inherent in such situations.

A more extreme example is shown in Fig. 11. The abundance of lines remain clearly distinct, interacting only at the junction and crossover points. The correct resolution of the distinct lines can be attributed to the strong no-line labels separating them. These no-line intermediates, strengthened further by thinning, effectively stop the cross influences from propagating.

VII. Conclusions

For relaxation labeling processes, the final distribution of probabilities on the label sets is dependent both upon the initial probabilities and upon the compatibility functions (see Section II). Such processes are appropriate for line enhancement because the knowledge they require can be represented in terms of compatibilities (or constraints) between the possible line segments. The examples in this paper demonstrate that, for line enhancement, a reasonable set of compatibilities is both readily obtainable and applicable to many different types of pictures. Moreover, since lines are often strongly determined by their pictorial context, it appears that the initial probability distributions need to be only rough estimates. As the initial estimates become more accurate, the convergence of probabilities seems to require fewer iterations. However, the minimum number of iterations is determined largely by "worst case" detector responses, i.e., by the largest contexts required to interpret the most ambiguous responses properly.

For most of the examples studied, eight iterations were sufficient for overall enhancement. This implies that the distance over which influences must propagate in order to enhance a label at a given position is rather small, although it is larger than just the immediate neighborhood of each point. Allowing the process to continue for large numbers of iterations often produced the undesirable result of thickening existing lines near junctions.

One previous study did suggest iterating a local operator to enhance the output of line detectors (Rosenfeld *et al.* [11]). However, that suggestion was based on the observation that the output of line detectors applied over a picture should itself be line-like. Thus, finding the lines in this new feature picture (using local line detectors) would have the effect of enhancing the lines in the original picture. Unlike relaxation, this approach did not fully realize the utility of local context for line enhancement.[3]

Relaxation labeling is controlled by the algorithm for updating the label probabilities. For the line enhancement application, a simple algebraic summation over all neighbors was used at first. However, in order to provide thinning, it was necessary to extend this to include (difference) relationships over some of the neighboring label sets. Thus, in an elementary way, some of the structural information contained in the distribution of the labels over the neighborhood was used.

By incorporating conditional expressions into the updating algorithm, more of this information could be used. This would allow the process to become increasingly sensitive to the relations between labeled objects in their contextual neighborhoods. Biases for various structural combinations, e.g., Y-branches, could be introduced in this way, and simple dependence on numerical values reduced even further.

Acknowledgment

The authors wish to thank Sunny Banvard for her help in preparing this paper.

References

[1] M. A. Arbib, "Segmentation, slides, and cooperative computation," Center for Systems Neuroscience, University of Massachusetts, Amherst, MA, unpublished, 1975.
[2] R. Bajcsy and M. Tavakoli, "Computer recognition of bridges, islands, rivers, and lakes from satellite pictures," in *Proc. Machine Processing of Remotely Sensed Data*, Oct. 1973, pp. 2A-54–2A-68.
[3] ——, "Computer recognition of roads from satellite pictures," in *Proc. 2nd Int. Joint Conf. on Pattern Recognition*, Aug. 1974, pp. 190–194.
[4] B. K. P. Horn, "The Binford-Horn line finder," Artificial Intelligence Lab., Massachusetts Inst. Tech., Cambridge, MA, AI Memo 285, Dec. 1973.
[5] D. Marr, "On the purpose of low-level vision," Artificial Intelligence Laboratory, M.I.T., December, 1974.
[6] A. Martelli, "Contour detection in noisy pictures using heuristic search methods," in *Proc. 1st Int. Joint Conf. on Pattern Recognition*, 1973, pp. 375–388.
[7] U. Montaari, "On the optimal detection of curves in noisy pictures," *Commun. Assoc. Comput. Mach.*, vol. 14, pp. 335–345, May 1971.
[8] F. O'Gorman and M. B. Clowes, "Finding picture edges through collinearity of feature points," in *Proc. 3rd Int. Joint Conf. Artificial Intelligence*, Aug. 1973, pp. 543–555.
[9] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 420–433, June 1976.
[10] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976.
[11] A. Rosenfeld, R. B. Thomas, and Y. H. Lee, "Edge and curve detection for texture discrimination," in *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York, 1970.

[3] Some experimental work on a simplified relaxation approach to line enhancement has been done by G. J. VanderBrug [16].

[12] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, pp. 562–569, May 1971.
[13] Y. Shirai, "Analyzing intensity arrays using knowledge about scenes," in *The Psychology of Computer Vision*, P. H. Winston, Ed. New York: McGraw-Hill, 1975.
[14] N. W. Short and P. D. Lowman, Jr., "Mineral resources, geological structure and landform surveys," in *Proc. 3rd Earth Resources Symp.*, vol. II, pp. 147–169, Dec. 1973.
[15] G. J. VanderBrug, "Line detection in satellite imagry," *IEEE Trans. Geosci. Electron.*, vol. GE-14, pp. 37–43, Jan. 1976.
[16] ——, "Experiments in iterative enhancement of linear features," *Computer Graphics and Image Processing*, 1976, to be published.
[17] D. Waltz, "Understanding line drawings of scenes with shadows," in *The Psychology of Computer Vision*, P. H. Winston, Ed. McGraw-Hill, New York, 1975.
[18] S. W. Zucker, A. Rosenfeld, and L. S. Davis, "General purpose models: Expectations about the unexpected," in *Proc. 4th Int. Joint Conf. Artificial Intelligence*, Sept. 1975, pp. 716–721.

Steven W. Zucker (S'71–M'75) received the B.S. degree in electrical engineering from Carnegie-Mellon University, Pittsburgh, PA, in 1969, and the M.S. and Ph.D. degrees in biomedical engineering from Drexel University, Philadelphia, PA, in 1972 and 1975, respectively.

From 1974 to 1976 he was a Research Associate at the Picture Processing Laboratory, Computer Science Center, University of Maryland, College Park. He is currently an Assistant Professor in the Department of Electrical Engineering, McGill University, Montreal, P.Q., Canada. His research interests include image processing, pattern recognition, and artificial intelligence.

Dr. Zucker is a member of Sigma Xi and the Association of Computing Machinery.

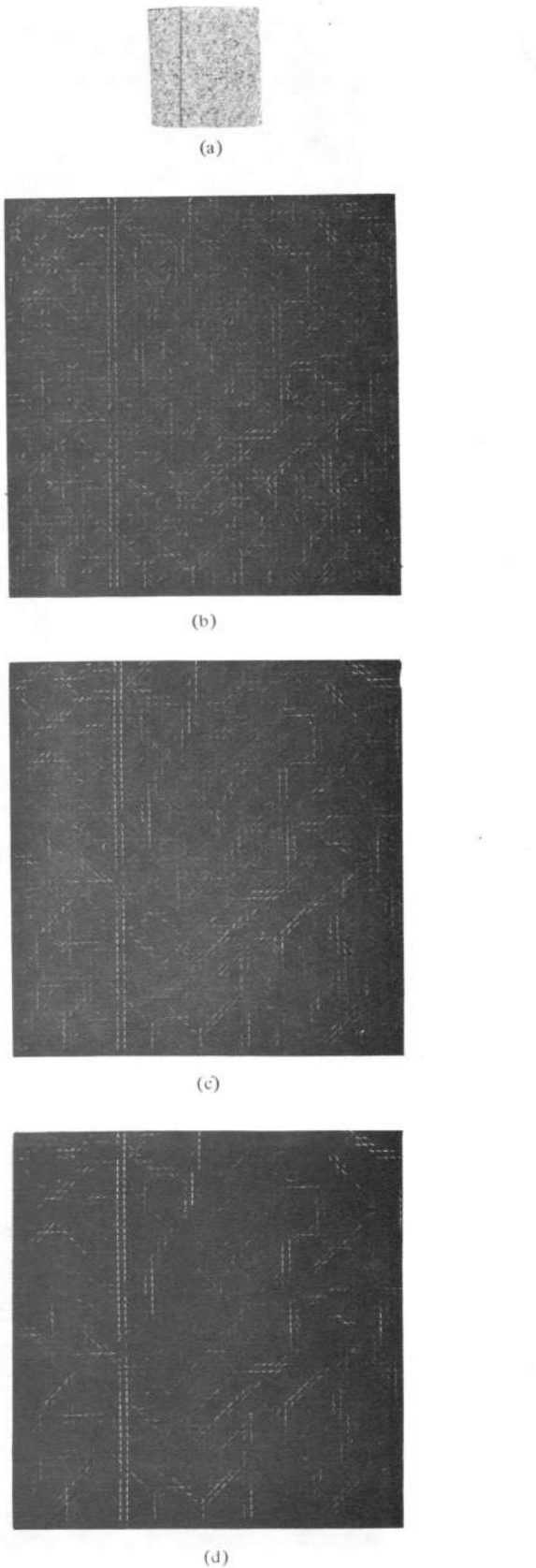
Robert A. Hummel was born in Houston, TX, on May 6, 1953. He received the B.A. degree in mathematics from the University of Chicago, Chicago, IL, in June 1975.

Periodically from 1972 to 1975, he was employed at the University of Maryland Picture Processing Laboratory. His research included topics in digital picture enhancement, sampling, and applications of relaxation techniques for scene labeling. He was also a Bush Foundation graduate fellow in mathematics at the University of Minnesota. He is presently at the Computer Science Center, University of Maryland, College Park.

Mr. Hummel is a member of Phi Beta Kappa and an associate member of Sigma Xi.

Azriel Rosenfeld (M'60–F'72), for a photograph and biography please see this issue, p. 393.

versus 5646 for the optimal algorithm. Hence, for the larger variable problems, the suboptimal lookahead scheme with smaller values of l is more efficient than the optimal algorithm and yields more optimal subsets than backward sequential selection.



(a) (b) (c) (d)

Fig. 4. (a) Computer-generated line in noise. (b) Initial probability assignments for (a), obtained from nonlinear line detector responses. (c)–(g) Iterations 1–4, and 8 of the relaxation process applied to (b).

Correction to "An Application of Relaxation Labeling to Line and Curve Enhancement"

STEVEN W. ZUCKER, ROBERT A. HUMMEL, AND AZRIEL ROSENFELD

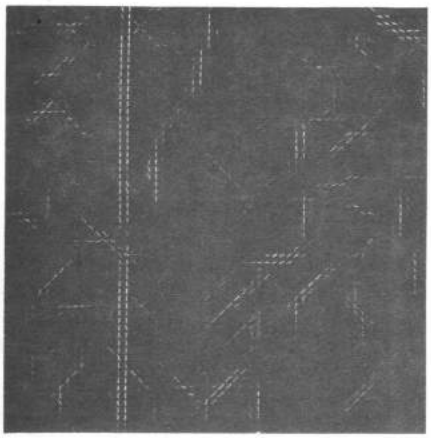
In the above paper[1] Figs. 4–7, and 10–12 were inadvertently misrepresented. They are correctly displayed here.

Manuscript received June 8, 1977.

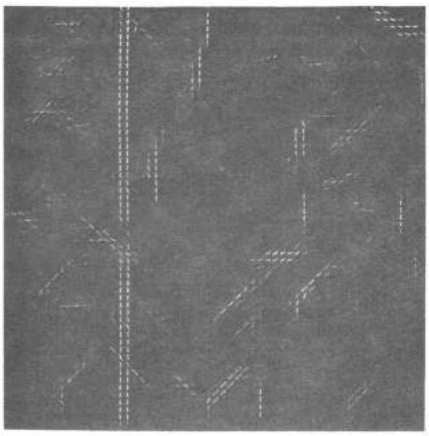
S. W. Zucker was with the Computer Science Center, University of Maryland, College Park, MD 20742. He is now with the Department of Electrical Engineering, McGill University, Montreal, P.Q., Canada.

R. A. Hummel and A. Rosenfeld are with the Computer Science Center, University of Maryland, College Park, MD 20742.

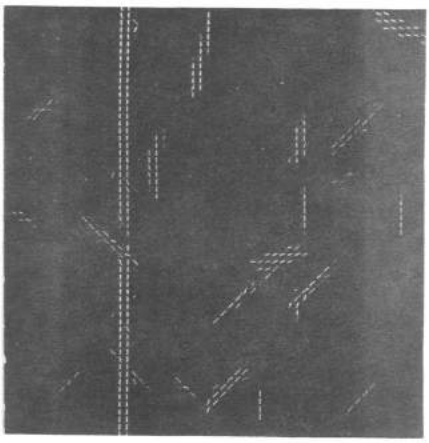
[1] S. W. Zucker, R. A. Hummel, and A. Rosenfeld, *IEEE Trans. Comput.*, vol. C-26, pp. 394–403, Apr. 1977.



(e)



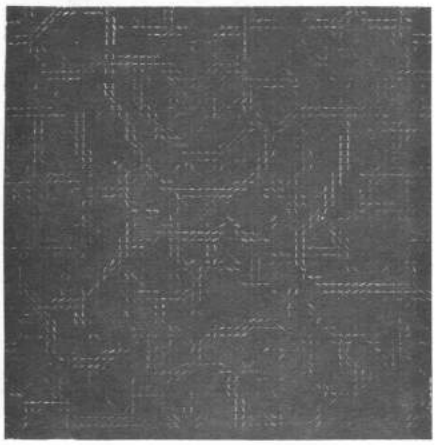
(f)



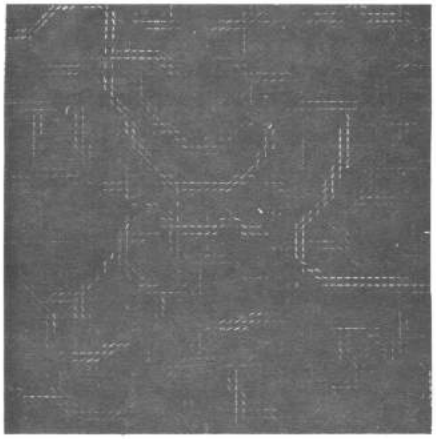
(g)

Fig. 4. (Continued.)

(a)

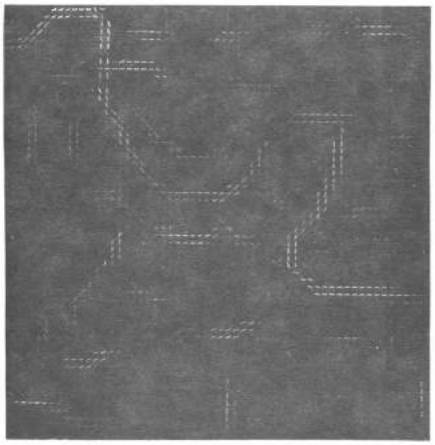


(b)

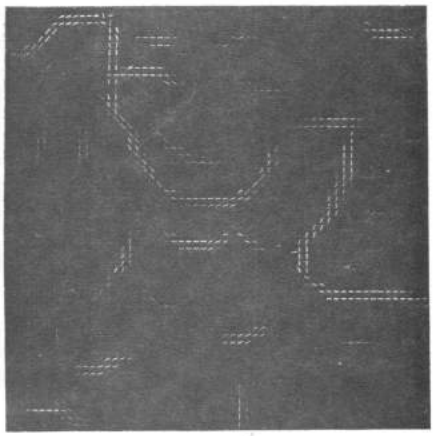


(c)

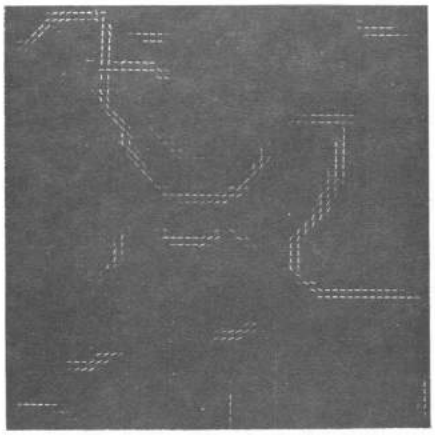
Fig. 5. (a) LANDSAT subimage containing river. (b) Initial probability assignments for (a). (c)–(g) Iterations 1–5 of the relaxation process applied to (a). (h)–(j) Iterations 6–8 of the relaxation process applied to (a).



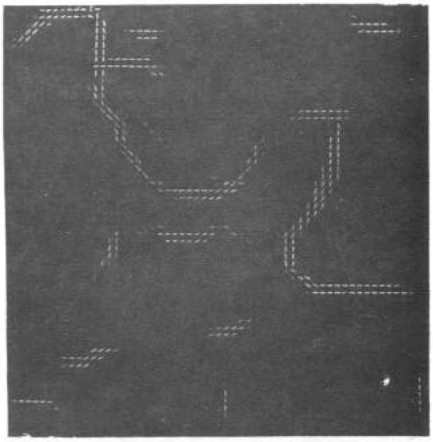
(d)



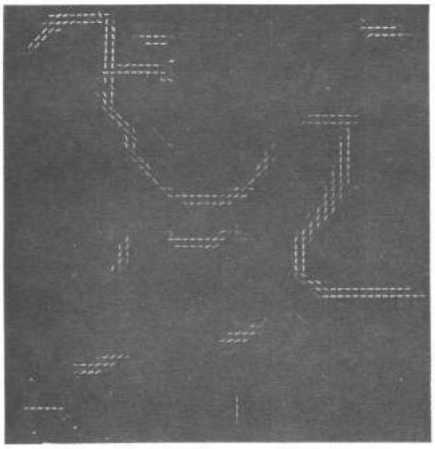
(e)



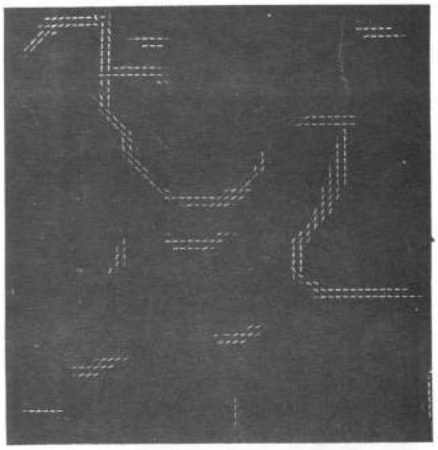
(f)



(g)

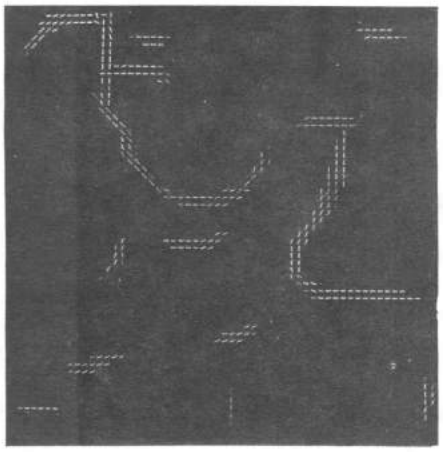


(h)



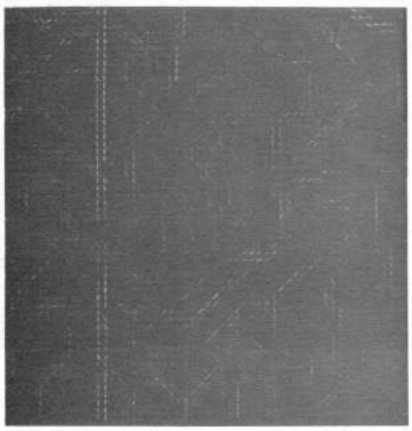
(i)

Fig. 5. (Continued.)

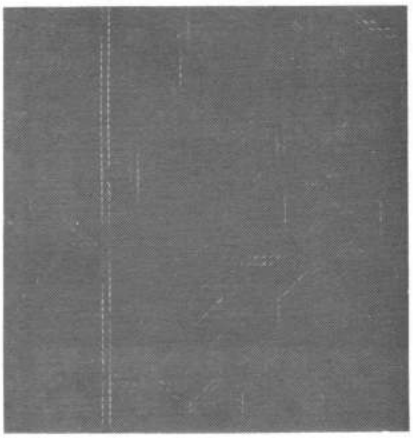


(j)

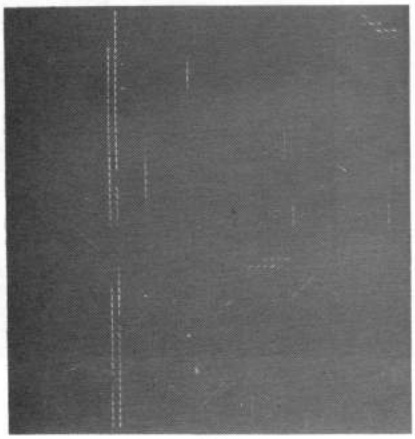
Fig. 5. (Continued.)



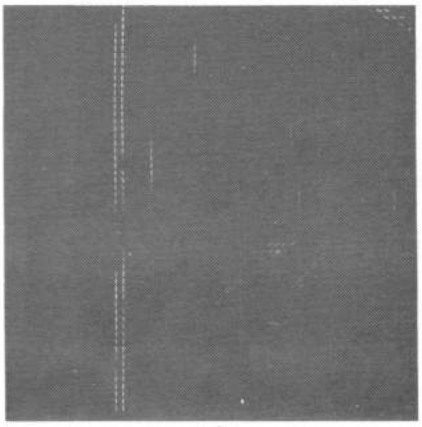
(a)



(b)

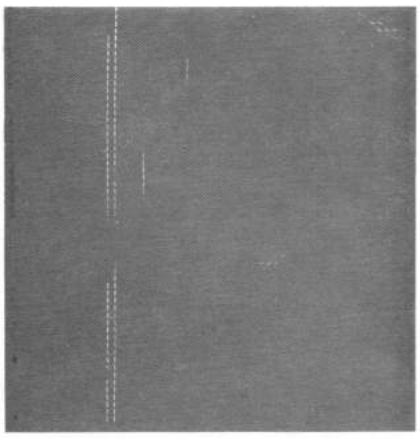


(c)

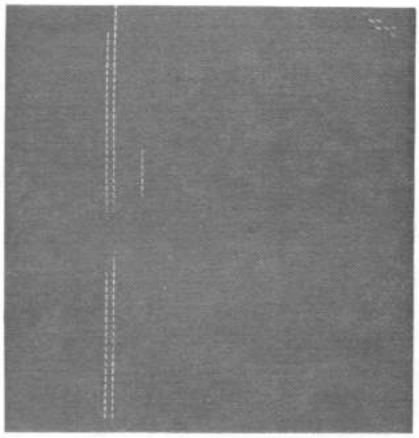


(d)

Fig. 6. (a)–(f) Iterations 1–5 and 8 of a modified relaxation process, biased in favor of straight lines, applied to Fig. 4(a).

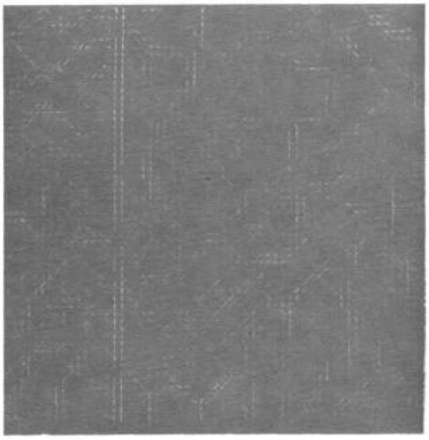


(e)



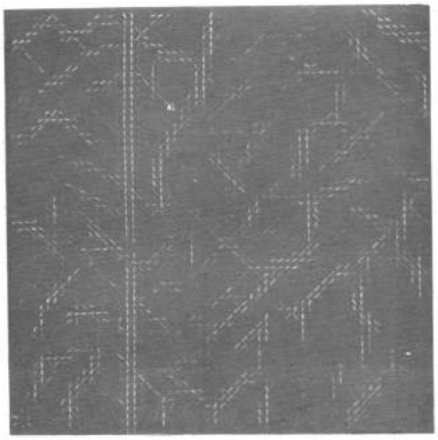
(f)

Fig. 6. (Continued.)



(a)

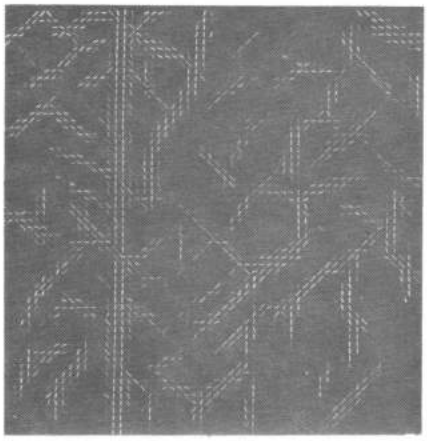
(b)



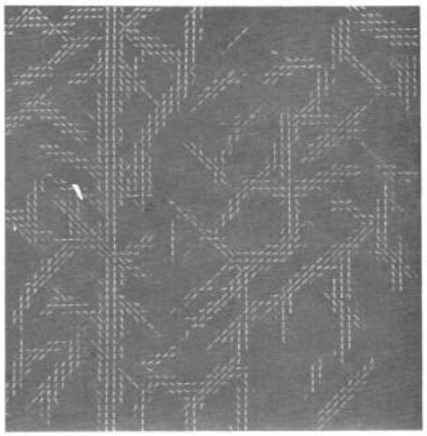
(c)

(d)

Fig. 7. (a)–(f) Iterations 1–5 and 8 of a modified relaxation process, without noise cleaning capability, applied to Fig. 4(a).



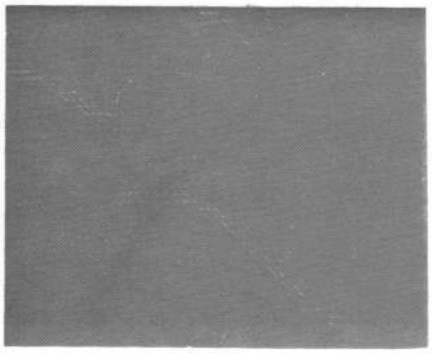
(e)



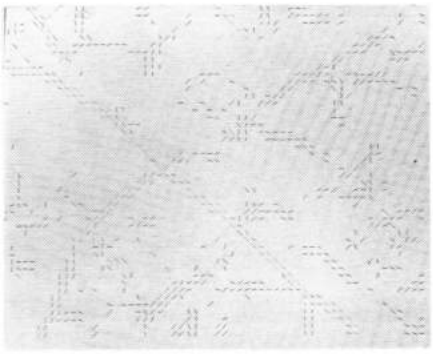
(f)

Fig. 7. (Continued.)

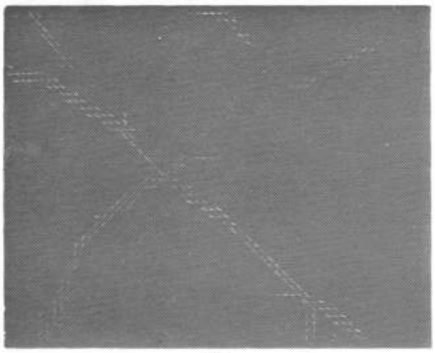
(a)



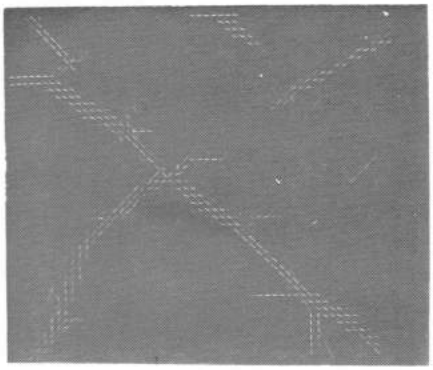
(b)



(c)

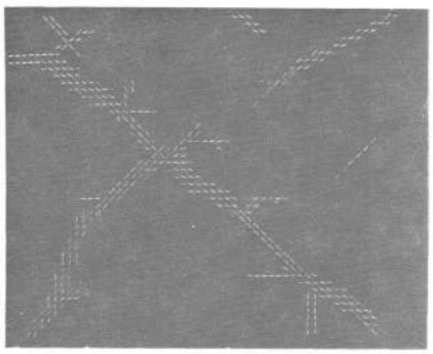


(d)

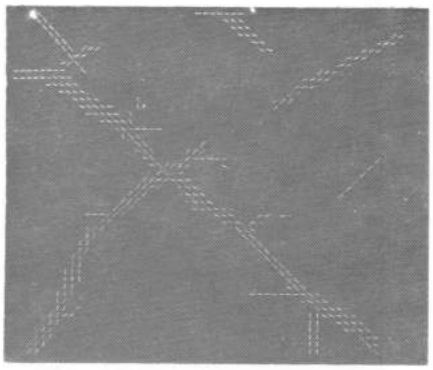


(e)

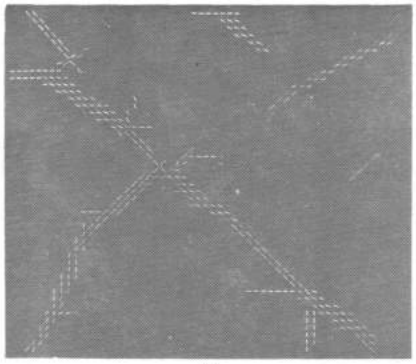
Fig. 10. Enhancement of faint curves. (a) SKYLAB subimage containing concrete and asphalt roads. (b) Initial probability assignments for (a). (c) "Pseudo-complement" of (b) to emphasize the low probabilities (dark lines). (d)–(g) Iterations 0, 2, 4, 6, and 7 of a modified relaxation process, incorporating thinning capability, applied to (a). (h) Iteration number 7 for a relaxation process, like that in Fig. 10(d)–(g), but with the no-line/no-line compatibility increased 15 percent.



(f)

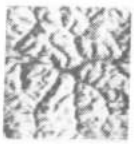


(g)

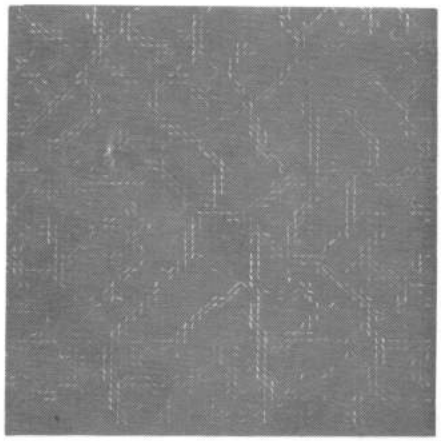


(h)

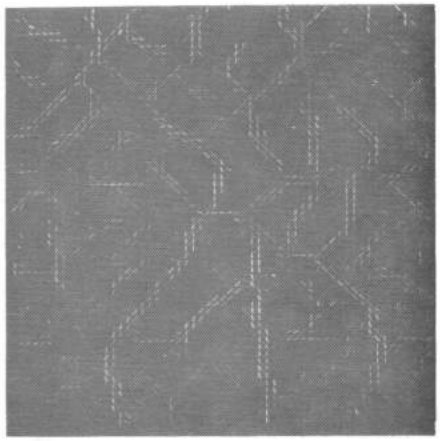
Fig. 10. (Continued.)



(a)

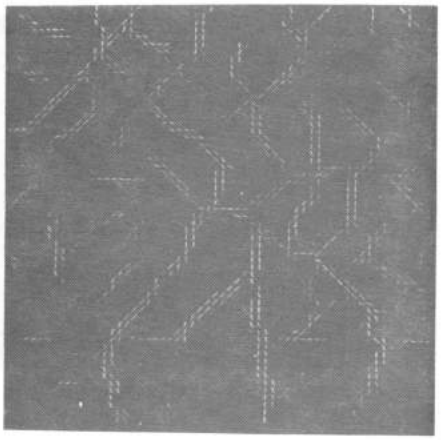


(b)

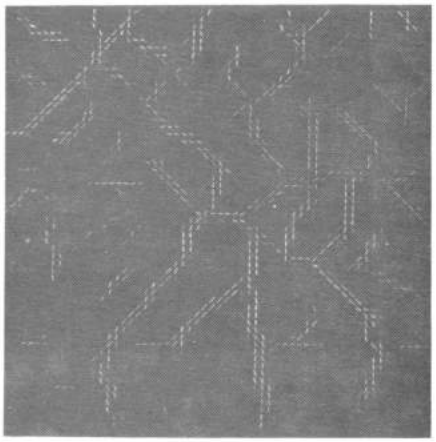


(c)

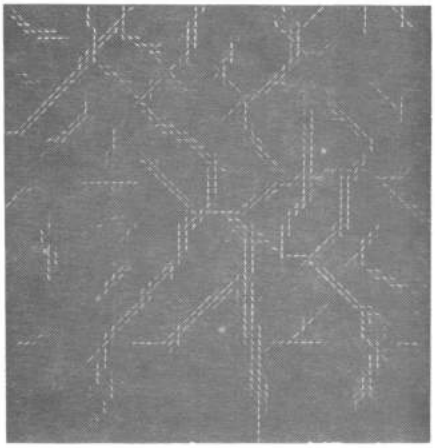
Fig. 11. (a) LANDSAT image containing many lines (geological linear features). (b)–(g) Iterations 0, 1, 2, 3, 5, and 7 of the same process used in Fig. 10, applied to (a).



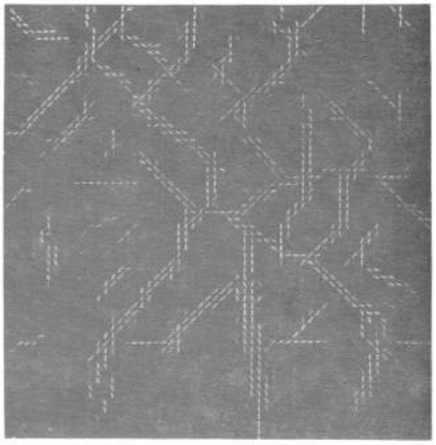
(d)



(e)



(f)



(g)

Fig. 11. (Continued.)

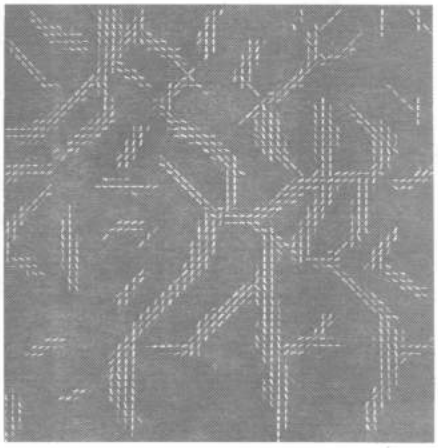


Fig. 12. The seventh iteration of the same process as that in Fig. 11, except without thinning.