NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

RELAXATION METHOD APPLIED TO
LOFARGRAM

by

Y.H.Yang

June 1990

Thesis Advisor   C.H. Lee

91-03064

91 6 21 016

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified		1b Restrictive Markings
2a Security Classification Authority		3 Distribution/Availability of Report
2b Declassification/Downgrading Schedule		Approved for public release; distribution is unlimited.
4 Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)

6a Name of Performing Organization Naval Postgraduate School	6b Office Symbol (if applicable) 33	7a Name of Monitoring Organization Naval Postgraduate School
6c Address (city, state, and ZIP code) Monterey, CA 93943-5000		7b Address (city, state, and ZIP code) Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization	8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number

8c Address (city, state, and ZIP code)	10 Source of Funding Numbers

	Program Element No	Project No	Task No	Work Unit Accession No

11 Title (include security classification) RELAXATION METHOD APPLIED TO LOFARGRAM

12 Personal Author(s) Y. H. Yang

13a Type of Report Master's Thesis	13b Time Covered From To	14 Date of Report (year, month, day) June 1990	15 Page Count 74

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)
Field	Group	Subgroup	Relaxation method applied to lofargram

19 Abstract (continue on reverse if necessary and identify by block number) When the signal to noise ratio is low, the detection of a track in a lofargram by computer is difficult. Actually, operators often face this kind of problem when detecting an acoustic signature in the noisy sea environment. Furthermore, operators must keep on tracking lofargram to identify a given target. This problem could be handled by the automation of lofargram processing using filtering or image processing techniques. This technique will suppress the background and emphasize the spectral lines in the lofargram. Targets can be tracked by an automatic lofargram processing system up to a certain point at which the system should alarm the operator. The enhancement processing of lofargrams using a relaxation method could be one part of the automatic system of lofargram processing to provide available target information for good decisions. The objective of this thesis is to enhance spectral lines of the lofargram by using the relaxation method which is an iterative approach to line detection. This technique makes probabilistic decisions at every point of the lofargram for each iteration. Decisions are adjusted at successive iterations based on the results of the previous iterations. This thesis tested algorithms using relaxation method for lofargrams. Some experimental results of the lofargram processing are presented. Experimental results showed that an edge relaxation method can yield better results than the line relaxation method. However, a double line detected from an edge is still undesirable and requires future work.

20 Distribution/Availability of Abstract ☒ unclassified/unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual C. H. Lee	22b Telephone (Include Area code) 408-646-2907	22c Office Symbol 62Le

DD FORM 1473,84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Relaxation Method Applied to Lofargram

by

Y. H. Yang
Lieutenant, Korean Navy
B.S., Naval Academy of Korea, 1983

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

from the

NAVAL POSTGRADUATE SCHOOL
June 1990
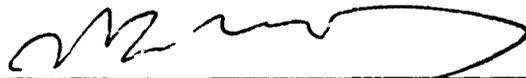
Author: _____
Y. H. Yang
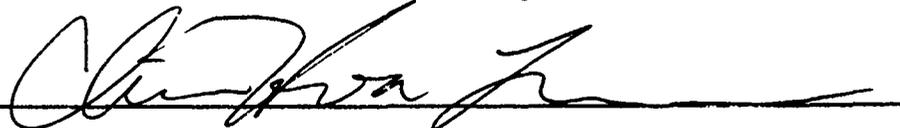
Approved by: _____
C. H. Lee, Thesis Advisor
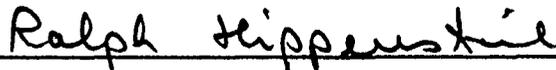
_____
R. Hippenstiel, Second Reader
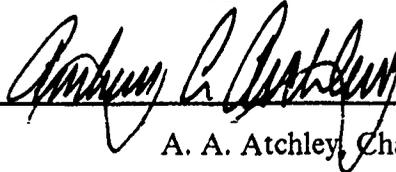
_____
A. A. Atchley, Chairman,
Engineering Acoustics Academic Committee

ABSTRACT

When the signal to noise ratio is low, the detection of a track in a lofargram by computer is difficult. Actually, operators often face this kind of problem when detecting an acoustic signature in the noisy sea environment. Furthermore, operators must keep on tracking lofargram to identify a given target. This problem could be handled by the automation of lofargram processing using filtering or image processing techniques. This technique will suppress the background and emphasize the spectral lines in the lofargram. Targets can be tracked by an automatic lofargram processing system up to a certain point at which the system should alarm the operator. The enhancement processing of lofargrams using a relaxation method could be one part of the automatic system of lofargram processing to provide available target information for good decisions. The objective of this thesis is to enhance spectral lines of the lofargram by using the relaxation method which is an iterative approach to line detection. This technique makes probabilistic decisions at every point of the lofargram for each iteration. Decisions are adjusted at successive iterations based on the results of the previous iterations. This thesis tested algorithms using relaxation method for lofargrams. Some experimental results of the lofargram processing are presented. Experimental results showed that an edge relaxation method can yield better results than the line relaxation method. However, a double line detected from an edge is still undesirable and requires future work.

iii

TABLE OF CONTENTS

v

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. WHAT IS A LOFAR

The term "LOFAR" is defined as "search technique using omnidirectional sonobuoys". It is an acronymn for "LOw Frequency Analysis and Recording." Every ship including submarine create peculiar noises in the sea. The noise spectrum can be used to distinguish different types of ships. If we know the noise frequency spectrum pattern radiated from a certain ship, it is possible to identify this ship by examining the lofargram. This is the basic motive of using lofar.

For determining the characteristic of noise spectra, a frequency-time analyzer can be used, which is similar to those used for speech analysis. It yields a plot of frequency against time and shows the intensity of the sound in the bandwidth of the spectrum by darkening the record paper. This paper plot is called a lofargram.

Before the appearance of the digital lofargram, thermal burning of record paper was used. But, the conventional paper gram was inconvenient to handle compared to digital lofargram. Furthermore, an undesirable smell was created during operation because of the burning on the long strip of paper using the old method. Therefore, the old method is almost replaced by the digital lofargram. The digital lofargram is obtained by digitizing the analog signal and displaying a gray scale on the computer screen or on a paper.

Figure 1 adopted from [Ref. 1] is an example of a lofargram of the noise of a large surface ship obtained from a hydrophone. As the ship passed by, tracks are left in the lofargram. The frequency scale along the horizontal axis extends from 0 to 150 Hz and the recording duration of the vertical axis was approximately 1/2 hour. The constantly spaced vertical line components marked by arrows are blade-rate lines. The lines marked X are of unknown origin. Figure 2 adopted from [Ref. 2] indicates the block diagram of a passive sonar to get the lofargram.

Generally, the lofargram has characteristics as follows

1. Signal to noise ratio is generally low.

2. A constant frequency tonal produces a darkening over time periods, which would appear as a vertical line on the lofargram display.

3. Multi-lines could be displayed on the lofargram depending on the spectrum band-width.

4. Slant lines or slightly curved lines are due to the Doppler shift could be shown on the lofargram.

1
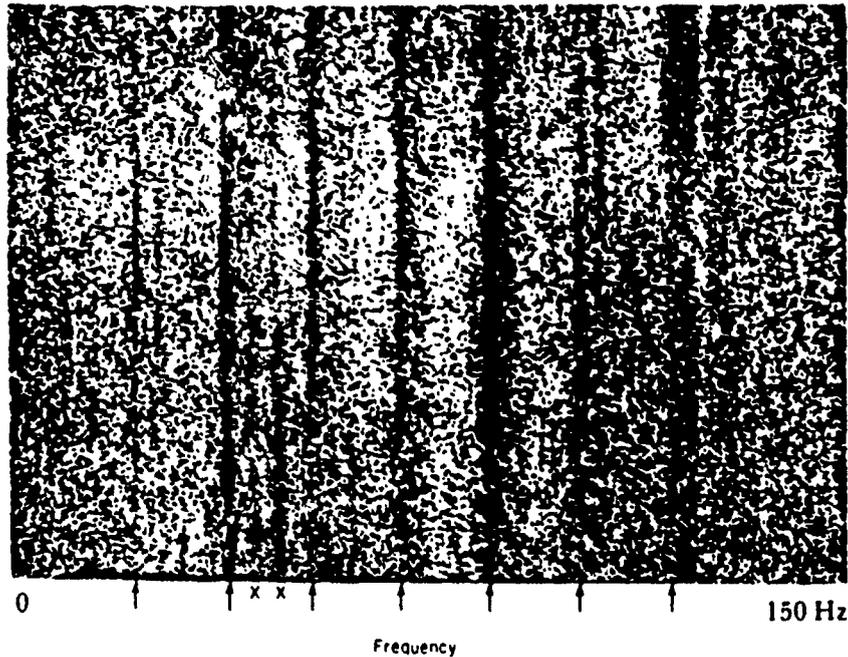
Figure 1.   Lofargram of a surface ship at a speed·of 11 knots
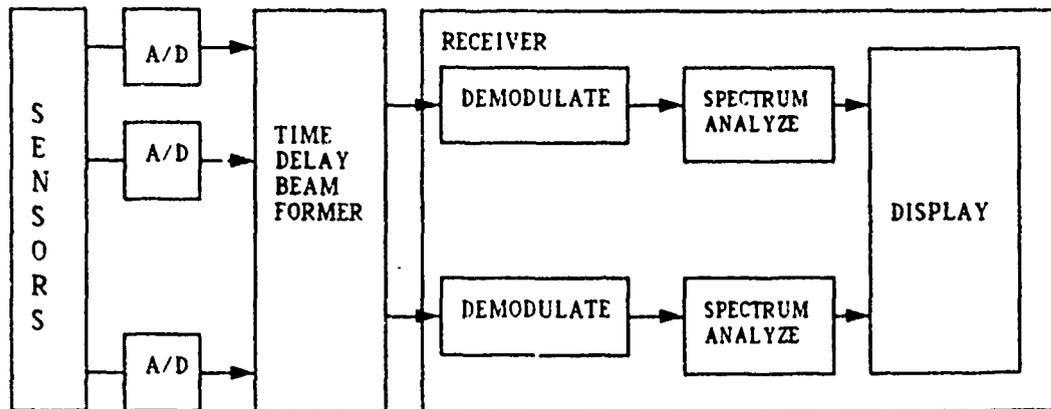
Figure 2.   Passive sonar system block diagram

B.   OBJECTIVE OF LOFARGRAM PROCESSING

When the SNR is lower than 3db, the interpretation of lofargram is difficult. Actually, operators often face this kind of situation in the noisy sea environment. Furthermore, operators must keep on tracking lofargram to identify target. This problem could be handled by the automation of lofargram processing using filtering or image processing techniques which can suppress the background and emphasize the spectral lines in the lofargram. Target can be tracked by an automatic lofargram processing system up

2

to a certain point then the system alarms the operator. The enhancement processing of a lofargram using the relaxation method could be one part of the automatic system of lofargram processing to provide target related information for good decisions.

The objective of this thesis is to enhance the spectral lines of the lofargram by using the relaxation method which is an iterative approach to line detection. This technique makes probabilistic decisions at every point of the lofargram for each iteration. Decisions are adjusted at successive iterations based on results of the previous iterations.

This thesis will present some experimental results of the line enhanced lofargrams using the relaxation method.

C. WHAT IS EDGE AND LINE DETECTION

Edges are defined as the discontinuity of intensity in an image, and are basic parts of the image information in general. Both the edge and line detection are fundamental techniques of image processing which are used to detect boundary of an object. But, there is an important difference between edge and line detection. Edge detection aims at finding the local discontinuities in an image. These discontinuities are of interests because they are likely to occur at the boundaries of objects. The output of an edge detection process will be some local line shown as boundaries of an object. Figure 3 shows the step edge and output profile of an edge detection process.

(a)   (b)
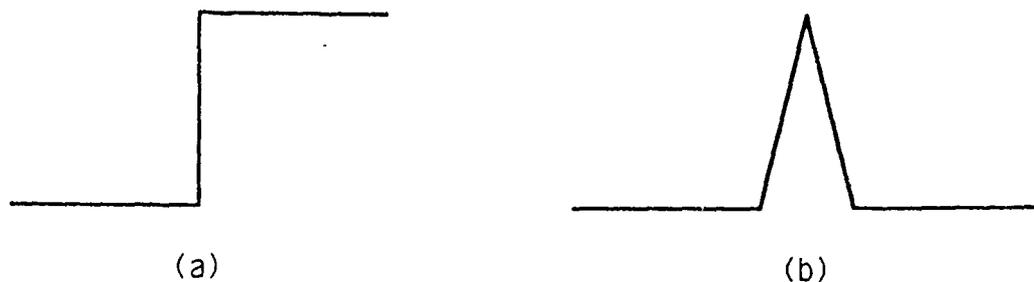
Figure 3.   Gray level profile of a) edge and b) line

In real images, with noise and surface imperfections, gaps between local lines are expected. Therefore, a process to organize the local edges into aggregates is needed in a line detection technique. Line detection is a process where edges in an image are aggregated to form object boundaries. These boundaries may not be known a priori. But, in many cases, they can be approximated well by piecewise linear segments. However, it is not feasible to simply fit linear segments to all the edges in an image and discard the poor fits. It is first necessary to aggregate the edges lying along a single line or along

3

another known curve. Proximity and the directions of an edge or more detailed descriptions of the edges can be used for such aggregation. Therefore, edge detection and line detection can be put in sequence. Basic ideas of edge and line detection are explained in [Ref. 3].

Usually edge and line detection algorithms can be divided into the following categories. [Ref. 4]

1. Detection of element.

   Differential-type operator.

   Model-fitting method.

2. Enhancement of lines.

   Enhancement in image space.

   Enhancement in parametric space (Hough transform).

3. Connection of elements.

   Connections using graph search techniques.

Category 1 refers to the process to detect line or edge elements by applying local operators such as Roberts [Ref. 5] , Prewitt [Ref. 6] and Sobel [Ref. 7]. An approximation of the gradient for a digital picture from Roberts's operator is given by

$$R(i,j) = \nabla g(i,j)$$

$$= \sqrt{\{g(i+1,j+1)-g(i,j)\}^2+\{g(i,j+1)-g(i+1,j)\}^2} \qquad (1.1)$$

In Figure 4 $g(i,j)$ denotee the intensity of the image iof the pixel (i,j), which the direction of the gradient is given by

$$\theta = -\frac{\pi}{4} + \tan^{-1}\left[\frac{g(i,j+1)-g(i+1,j)}{g(i+1,j+1)-g(i,j)}\right] \qquad (1.2)$$

The Prewitt operator defines the magnitude of a gradient from Figure 5 by

$$S = \sqrt{S_x^2+S_y^2} \qquad (1.3)$$

$$S_x = (a_2+a_3+a_4)-(a_0+a_7+a_6) \qquad (1.4)$$

$$S_y = (a_6+a_5+a_4)-(a_0+a_1+a_7) \qquad (1.5)$$

and the direction of the edge is given by

4

$$\theta = \tan^{-1}(\frac{S_x}{S_y})\tag{1.6}$$
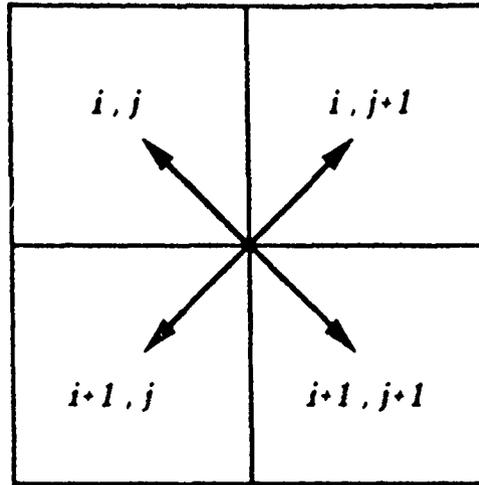
Figure 4.   Robert's gradient operator

The Sobel operator defines a gradient from figure 5 by

$$S = |\ (a_0 + 2a_1 + a_2) - (a_6 + 2a_5 + a_4)\ |$$

$$+\ |\ (a_0 + 2a_7 + a_6) - (a_2 + 2a_3 \cdot a_4)\ |\tag{1.7}$$

Generally, a differential type operator doesn't perform well compared to a model fitting method. We are going to discuss this problem in the experiment part of this thesis.

Category 2 is a smoothing processes which eliminates noise components and emphasize edges. In this category, the iterative method performs edge enhancement by applying iterative processing based on relationships of its neighbour elements in an image space. Category 3 is a process which connects small line segments into longer lines. Knowledge or constraints related to the sequence of elements to be detected are used in an evaluation function. Then, the connection of elements is regarded as a graph search problem. In this case, two techniques are used: one directly proceeds with the line extraction while searching the edge elements for brightness data, and the other applies a local edge operator to the entire image before the connection of elements is done. [Ref.3]

a_0	a_1	a_2
a_-	$i\,.\,j$	a_3
a_6	a_5	a_4
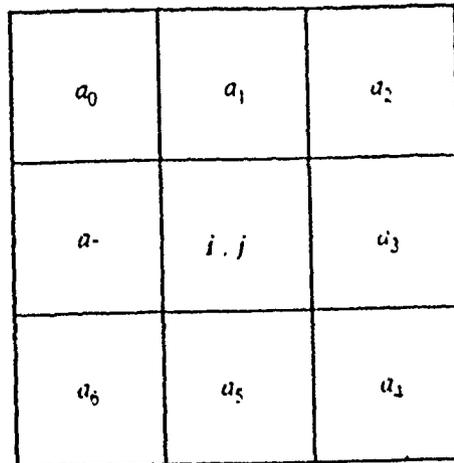
Figure 5. Apixel and its eight neighbors

In this chapter, the general ideas of line and edge detection as used in lofargram processing are explained. Chapter II introduces the relaxation technique, while a modified algorithms using relaxation technique is introduced in Chapter III. Experimental results of lofargram processing are given in Chapter IV. Chapter V is the conclusion of this thesis.

II. RELAXATION METHOD

A. GENERAL BACKGROUND

Relaxation techniques were first used by Waltz [Ref. 8] for the description of solids and were expanded by Zucker [Ref. 9] for a variety of applications. Relaxation is an iterative approach to assign each pixel to categories by assigning the neighboring pixels in a "compatible" way. This means that each pixel has interaction with its neighboring pixels, and the degree of interaction is considered in terms of the compatibility of each pixel to its neighboring pixels. This method takes a set of probabilities for each pixel to belong to a possible class, and uses an iterative technique to update the probabilities.

An important element of the method is a set of compatibility measures $c(k,s;l,t)$ that gives the compatibility of assigning pixel k to class s and assigning the neighboring pixel l to class t. It is assumed that $c(k,s;l,t)$ is in the range $(-1,+1)$ where -1 implies a strong incompatibility, +1 implies a strong compatibility, and 0 implies neutrality. $p^0(k,s)$ is an initial estimate of the probability that pixel k belongs to class s. Let $q^r(k,s)$ is the increment of the probability and N be the number of neighbors for each pixel; a neighborhood of 3x3 or 5x5 is usually employed. Then, a sequence of estimates $p^r(k,s)$ is computed iteratively as follows

1. For each pixel k, compute the "neighbor compatibility" between pixel k and its neighbor pixel l in class s, which results in the probability of pixel k in class s.

$$q^r(k,s) = \frac{1}{N} \sum_l \sum_t c(k,s;l,t) p^r(l,t) \qquad (2.1)$$

where the summation is over all possible neighbor pixels l and all classes t. If only two classes(for example, edge and non-edge) are considered, then there are only four cases of neighbor compatibility $c(k,e;l,e)$ means the compatibility of an edge pixel k being a neighbor of an edge pixel l. $c(k,e;l,n)$ means the compatibility of an edge pixel k being a neighbor of a non-edge pixel l . $c(k,n;l,e)$ means the compatibility of a non-edge pixel k being a neighbor of an edge pixel l . $c(k,n;l,n)$ means the compatibility of a non-edge pixel k being a neighbor of a non-edge pixel l . Therfore, equation (2.1) can be expanded as follows

7

$$q^r(k,e) = \frac{1}{N} \sum_l c(k,e;l,e)p^r(l,e) + c(k,e;l,n)p^r(l,n) \tag{2.2}$$

$$q^r(k,n) = \frac{1}{N} \sum_l c(k,n;l,e)p^r(l,e) + c(k,n;l,n)p^r(l,n) \tag{2.3}$$

2. Update the probabilities $p^{r+1}(k,s)$

$$p^{r+1}(k,s) = \frac{p^r(k,s)(1+q^r(k,s))}{\sum_s p^r(k,s)(1+q^r(k,s))} \tag{2.4}$$

For two classes;

$$p^{r+1}(k,e) = \frac{p^r(k,e)[1+q^r(k,e)]}{p^r(k,e)[1+q^r(k,e)]+p^r(k,n)[1+q^r(k,n)]} \tag{2.5}$$

Equation (2.5) gives the probability that pixel k is an edge. Then the probability that pixel k is a non-edge is defined by

$$p^{r+1}(k,n) = 1 - p^{r+1}(k,e) \tag{2.6}$$

As the process is iterated, these probabilities must stay in the range from 0 to 1. The formulas given above are the basic expression of the relaxation technique. This concept can be expanded and modified for other applications. This technique has the following characteristics:

1. High compatibility pixels tend to reinforce each other and the low compatibility pixels tend to discourage each other.

2. The degree of reinforcing or discouraging done by a neighbor is proportional to its own probability of assignment of that class.

3. The probabilities $p(k,s)$ remain in the range from o to 1 and sum to 1 over all of class s.

B. APPLICATIONS OF RELAXATION METHOD

1. Line enhancement

One of the applications in which relaxation techniques have been proven useful is the detection of long smooth curves in an image. Initially, line detection operators are applied to the image, and their outputs are used to determine an initial probability

of the point lying on a curve with a given orientation. These probabilities are then iteratively reinforced: the probability of a point lying on a curve is reinforced by the probabilities of the other points lying on curves that smoothly continue it. After iterations of the procedure, points that lie on smooth curves tend to have high probabilities of being curve points, while the other points do not.

An application of the relaxation technique to the detection of major edges in an image will be presented in the next chapter in detail. Initially, a gradient edge operator is applied to the image. This provides information about edge strength and orientation at each point. These probabilities are then iteratively reinforced.

· The general idea of the edge reinforcement process is as follows.

a. Magnitude and direction

First, the magnitude and direction of the image gradient are computed for each pixel k. The magnitude of pixel k, divided by the maximum of the magnitudes over the entire image, defines the initial probability $p^o(k,s)$ of a pixel k being an edge.

b. Reinforcement of edges

The reinforcement process defines a new edge probability of a pixel k in terms of the old probabilities at the pixel k and its neighbor pixels. The computation of the new edge probability can be broken into steps as follows.

(1) Interactions between edge and non-edge. The interaction between edge and edge, edge and non-edge, non-edge and edge, non-edge and non-edge depend on the edge direction at the two positions, on the orientations of the line joining the points, and on the distance between them.

(2) Computation of the new probability. To compute the new edge probability obtained from the iterations, first, compute weighted sums of the edge and non-edge probability increments, $q'(k,e)$ and $q'(k,n)$. These sums are then normalized to lie in the range$(-1,+1)$ according to (2.4). They are used to update the edge and non-edge probabilities, while the updated values are normalized so that they can sum to 1.

The real issue in this thesis is to search for good line detection techniques applicable to lofargrams, and not to edge detection. Therefore, we are going to concentrate on the reinforcement of the relaxation method as applied to lofargram.

2. Labeling

There is another application using the relaxation technique, which is called "labeling." The objective is to assign labels to objects in an image. [Ref. 10]

9

a. Components of labeling

In a finite set of relations between objects, the objects usually correspond to entities to be labeled. The objects are often geometrically or topologically related to each other. An input scene is thus a relational structure of all objects. In the simplest case, each object is to be assigned with a single label. Labels may be weighted with "probabilities" indicating the "probability of an object having that label. "Constraints determine what labels may be assigned to an object. A basic labeling problem is then: Given a finite input scene (relational structure of objects), a set of labels, and a set of constraints, find a consistent labeling.

There are several types of labeling using relaxation.

b. Discrete labeling

It is a parallel iterative algorithm adjusting all object labels. All possible labels to each object are assigned in accordance with constraints. Iterations are performed until a globally consistent labeling is found. In parallel, from each object's label set all the labels that are inconsistent with the current labels of the rest of the relational structure are eliminated.

c. Linear labeling

The labeling process starts with an initial assignment of weights to all objects. Weight are reminiscent of probabilities, reflecting the "probability that a label is correct." Here p refers to probability-like weight rather than to the value of a probability density function. Let a relational structure with n objects be given by a_k, $k = 1,..n$, each with m discrete labels $\lambda_{1,...,m}$ and let $p_k(\lambda)$ denote the weight, or the "probability" that the label λ is correct for the object a_k and restricted as follows

$$0 \leq p_k(\lambda) \leq 1 \qquad (2.10)$$

$$\sum_\lambda p_k(\lambda) = 1 \qquad (2.11)$$

The linear labeling operator is based on the compatibilities of the labels, which serve as the constraints. A compatibility p_{kl} looks like a conditional probability.

$$\sum_\lambda p_{kl}(\lambda \mid \lambda') = 1 \qquad (2.12)$$

10

The $p_k(\lambda \mid \lambda')$ may be interpreted as the conditional probability that object a_k has label λ given that another object a_l has label λ'. The operator iteratively adjusts label weights in accordance with other weights and the compatibilities. A new weight $p_k(\lambda)$ is computed from old weights and compatibilities as follows.

$$p_k(\lambda) = \sum_l c_{kl}[p_k(\lambda \mid \lambda')p_l(\lambda')] \qquad (2.13)$$

The c_{kl} are coefficient such that

$$\sum_l c_{kl} = 1 \qquad (2.14)$$

d. Nonlinear labeling

If the compatibilities are allowed to take on both positive and negative values, we can express strong incompatibility and strong compatibility. Denote the compatibility of the event "label λ on a_k" with event "label λ' on a_l" by $r_{kl}(\lambda, \lambda')$. If the two events occur together often, r_{kl} should be positive. If they occur together rarely, r_{kl} should be negative. If they are independent, r_{kl} should be 0. The compatibilities are based on correlations

$$Cov(X,Y) = p(X,Y) - p(X)p(Y) \qquad (2.15)$$

$$\sigma^2(X) = p(X,X) - (p(X))^2 \qquad (2.16)$$

$$cor(X,Y) = \frac{Cov(X,Y)}{\sigma(X)\sigma(Y)} \qquad (2.17)$$

The r_{kl} is used to obtain the positive or negative change in weight.

$$q_k^r(\lambda) = \sum_l c_{kl} \sum_{\lambda'} r_{kl}(\lambda, \lambda')p_k^r(\lambda') \qquad (2.18)$$

The $q_k^r(\lambda)$ is the increment of the weight. The probability change is as follows

11

$$p_k^{r+1}(\lambda) = \frac{p_k^r(\lambda)[1+q_k^r(\lambda)]}{\displaystyle\sum_\lambda p_k^r(\lambda)[1+q_k^r(\lambda)]} \qquad (2.19)$$

Basically, labeling is different with the detection problem. Labeling is a process to assign the correct label to the correct position while detection is a process to take the desired portion of a scene having a mixed feature. Therefore, labeling is a different technique for line detection.

The general concept of the relaxation technique and its applications were discussed in this chapter. In the next chapter, line and edge detection algorithms using relaxation technique will be discussed in details.

III. RELAXATION ALGORITHMS APPLIED TO LINE DETECTION

There are many algorithms already developed for line detection. In this thesis, the objective is to select and test some of them for the lofargram processing.

A. EDGE ENHANCEMENT(RXEG)

This algorithm is derived from the original idea developed by Schachter, et al to reinforce the continuous edges detected by a gradient operation. RXEG is an acronymn for "RelaXation method for EdGe detection".Edges reinforce other edges that interact with nearby non-edge points in specified ways. The gradient of the edges are also iteratively adjusted. [Ref. 11]

1. Initial edge probability

A digital gradient operation is applied to the given image f. If we denote the x and y components of the gradient by $\Delta_x f$ and $\Delta_y f$, then the magnitude and direction of the gradient are given by

$$mag = \sqrt{(\Delta_x f)^2 + (\Delta_y f)^2} \qquad (3.1)$$

$$\theta = \tan^{-1}\left(\frac{\Delta_y f}{\Delta_x f}\right) \qquad (3.2)$$

The mag of the gradient indicates the strength of the edge. The angle θ of the gradient is perpendicular to the edge direction. Edge direction is obtained by adding 90° to the θ angle of the gradient.

We defined the "probability" of an edge at a given point k by

$$p(k,e) = \frac{mag(k)}{\max(mag(l))} \qquad (3.3)$$

where the max is taken over the entire image. The probability of a nonedge at a pixel k is defined as $p(k,n) = 1 - p(k,e)$.

2. Pixel and neighbor interaction

The centered pixel k could have four kinds of interactions with its neighbor pixel l such as edge/edge, edge/non-edge, non-edge/edge and non-edge/non-edge.

13

a. Edge/edge interaction

k is the centered pixel position, and l is one of the neighbor pixel position. Let α be the edge direction at k, β the edge direction at l, γ be the direction of the line joining k to l, D the chessboard distance from k to l, i.e, $\max(\,|\,k_x-l_x\,|\,,\,|\,k_y-l_y\,|\,)$. Then the edge/edge reinforcement process between the points k and l has strength given by

$$c(k,e;l,e) = \cos(\alpha-\gamma)\,\cos(\beta-\gamma)/2^D \qquad (3.4)$$

To see the significance of this definition, a few simple examples are considered. In these examples, the arrows indicate the direction along the edge, with the dark side of the edge on the left. These examples showed that parallel and perpendicular edges have no effect on one another.

Case	α	β	γ	$\cos(\alpha - \gamma)\,\cos(\beta - \gamma)/2^D$
↑↑	90	90	0	0
↑↓	90	270	0	0
↑ ↑	90	90	90	$1/2^D$
↑ ↓	90	270	90	$-1/2^D$
↑ →	90	0	0	0
↑ ←	90	180	0	0

Table 1. Examples of edge / edge interaction [Ref. 8]

b. Edge/non-edge interaction

Besides the edge/edge interaction which occurs between the edge probabilities $p(k,e)$ and $p(l,e)$ there are also interactions involving the non-edge probabilities $p(l,n)$. The edge probability at pixel k is weakened by the non-edge probability at pixel l to the degree $c(k,e;l,n)$ defined by

$$c(k,e;l,n) = \min[0, -\cos(2\alpha-2\gamma)/2^D] \qquad (3.5)$$

14

In the examples, non-edge neighboring points collinear with the center edge points weaken them, whereas non-edge points alongside edge points have no effect on them.

Case	α	γ	$- \cos(2\alpha - 2\gamma) / 2^D$	$c(k, e; l, n)$
↑·	90	0	$1/2^D$	0
↓·	270	0	$1/2^D$	0
↑̣	90	270	$-1/2^D$	$-1/2^D$
̇↑	90	90	$-1/2^D$	$-1/2^D$

Table 2.   Examples of edge / non-edge interaction [Ref. 8]

c.   Non-edge/edge interaction

The non-edge probability at pixel k is affected by the edge probability at the neighbor pixel l to the degree $c(k,n;l,e)$ defined by

$$c(k,n;l,e) = \frac{(1 - \cos(2\beta - 2\gamma))}{2^{D+1}}$$ (3.6)

Case	α	β	γ	$(1 - \cos(2\alpha - 2\gamma)) / 2^D$
·↑	0	90	0	$1/2^D$
̇↑	0	90	90	0

Table 3.   Examples of non-edge / edge interaction [Ref. 8]

From the examples, neighbor edge points alongside the center point strengthen them, While edge points collinear with non-edge points have no effect on them.

d.   Non-edge/non-edge interaction

The non-edge probabilities at centered pixel k and the neighbor pixel l reinforce each other to the degree $c(k,n;l,n)$ defined by

$$c(k,n;l,n) = \frac{1}{2^D}$$ (3.7)

Non-edge probabilities are reinforced by the other nearby non-edge probabilities. $c(k,n;l,n)$ is directionally independent.

3. Combined reinforcement process

For each pixel k, the net effect of its neighboring pixel on its edge probability $p(k,e)$ and non-edge probability $p(k,n) = 1-p(k,e)$ is computed as follows

$$q^r(k,e) = \sum_l C_1 p^r(l,e)c(k,e;l,e)$$

$$+ \sum_l C_2 p^r(l,n)c(k,e;l,n) \tag{3.8}$$

$$q^r(k,n) = \sum_l C_3 p^r(l,e)c(k,n;l,e)$$

$$+ \sum_l C_4 p^r(l,n)c(k,n;l,n) \tag{3.9}$$

where C_1, C_2, C_3, C_4 are constants whose sum is equal one. The standard values of $C_1 = 0.866$, $C_2 = 0.124$, $C_3 = 0.005$, and $C_4 = 0.005$ was used in the paper by Bruce J.Schachter, et al [Ref.11]. The results of the iteration process are somewhat sensitive to the choice of the C's. For example, if C_1 is too large, the edge will thicken and will be extended into non-edge points; while if C_4 is too large, gaps will appear at weak spots in the edges and at sharp angles.

$$q'(k,e) = \frac{q^r(k,e)}{|q^r(k,e)| + |q^r(k,n)|} \tag{3.10}$$

$$q'(k,n) = \frac{q^r(k,n)}{|q^r(k,e)| + |q^r(k,n)|} \tag{3.11}$$

$$p'(k,e) = p^r(k,e)[1+q'(k,e)] \tag{3.12}$$

$$p'(k,n) = p^r(k,n)[1+q'(k,n)] \tag{3.13}$$

16

$$p^{r+1}(k,e) = \frac{p^r(k,e)}{p^r(k,e)+p^r(k,n)} \tag{3.14}$$

This process is then iterated by using $p^{r+1}(k,e)$ in place of $p^r(k,e)$, and by using $1-p^{r+1}(k,e)$ in place of $p^r(k,n)$. We also compute the estimated edge direction $\Delta'_x(k)$ and $\Delta'_y(k)$ at each point

$$\Delta'_x(k) = W p^r(l,e) \cos(\theta(k))$$

$$+ \sum_l p^r(l,e)c(k,e;l,e) \cos(\theta(l)) \tag{3.15}$$

$$\Delta'_y(k) = W p^r(l,e) \sin(\theta(k))$$

$$+ \sum_l p^r(l,e)c(k,e;l,e) \sin(\theta(l)) \tag{3.16}$$

$$\theta^{r+1}(k) = \tan^{-1}(\Delta'_y(x,y)/\Delta'_x(x,y)) \tag{3.17}$$

For large values of the constant W, θ^{r+1} is close to θ; while for small values, it is strongly influenced by the neighboring θ's.

B. LINE AND CURVE ENHANCEMENT(RXLN)

In section A, only two categories are associated with the image pixel, (i.e, edge and no.1-edge). In this section, more categories are considered in the RXLN relaxation procedures. RXLN is an acronym for "RelaXation method for LiNe detection."

This algorithm was devloped by Zucker, Hummel and Rosenfeld [Ref. 12]. The relaxation process is applied to the detection of smooth lines and curves in noisy images. Nine class labels associated with each image point will be considered. Eight classes indicate lines at various orientations and one indicates the no-line case. In the relaxation process, interaction takes place between the probabilities at neighboring points. This permits line segment with compatible orientations to strengthen one another. Similarily, no-line class is reinforced by neighboring no-line class and weakened by the neighboring oriented line classes.

17

1. Initial probabilities

At each point, probabilitic₃ are assigned to the nine classes which are lines in eight possible directions and a no-line class. The initial probability for each class is obtained by evaluatng a nonlinear line detector as shown in Figure 6 at every picture point along the eight orientations. Eight sets of pixels representing eight orientations are selected and shown in Figure 6. Each set of pixels are rearranged as the template shown in Figure 6 (b). When the condition in Figure 6 do not hold, the response is zero. When they do hold, the response is calculated by

$$R = (B+E+H) - \frac{1}{2}(A+D+G+C+F+I) \qquad (3.18)$$

At each pixel, the detector's response are computed for every orientation. Then only one orientation which has the maximum detector's response is selected as the orientation of the pixel. [Ref. 13]

<u>Template</u>   <u>Conditions</u>

A B C   A < B > C

D E F   D < E > F

G H I   G < H > I

(a)

```
B        B        B        B                        H        H

E        E        E    H E    H E B  H E      E        E

H        H    H                          B        B        B

(1)      (2)      (3)      (4)      (5)      (6)      (7)      (8)
```

(b)
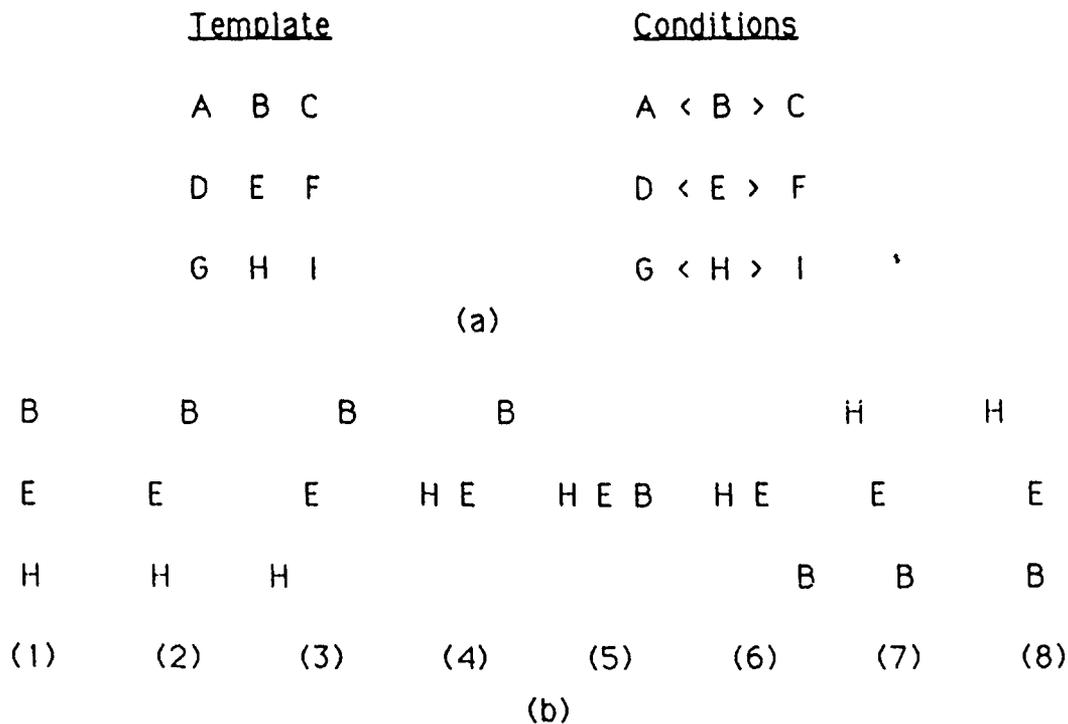
Figure 6.   **Nonlinear line detector.** (a) Nonlinear line detector for the vertical orientation. When the conditions do not hold, the response is zero; when they do hold, the response is calculated by (3.18). (b) Eight orientations of the detector.

To describe this process, let $d_s(k)$ denote the output of the nonlinear detector for orientations s at pixel k. Let $p(k,s)$ denote the probability that a pixel k belongs to class s. Then the initial probabilities can be obtained by scaling the detector's responses at each position by the maximum over the entire picture. The probability that pixel k has class s is expressed as follows

$$p(k,s) = \frac{\max_{s=1}^{8}(d_s(k))}{\max_{s=1}^{8}((\max_{k})d_s(k))} \times \frac{d_s(k)}{\sum_{s=1}^{8} d_s(k)} \tag{3.19}$$

$\max_{s=1}^{8}(d_s(k))$ means the one selected maximum value over all orientations. $\max_{k}(d_s(k))$ means the maximum value of oriented s over all image pixels, and the probability that pixel k has no-line is defined by

$$p(k,9) = 1 - \sum_{s=1}^{8} p(k,s) \tag{3.20}$$

2. Compatibility coefficients

To update the probabilities, the compatibility relation between a pixel and its neighbors must be specified. The compatibilities depend only on the relative orientations of the neighboring point. 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. If, on the other hand, two neighboring segments are oriented perpendicularly to one another, they will reduce support to each other. All other pairs of line segments are distributed between these two extremes. In this algorithm, there are three types of compatibility coefficients.

a. Compatibility coefficients between lines

Figure 7 shows the compatibility coefficients between the lines in five geometrical relationships. As standard values, from left to right: 1.0, 0.5, 0.05, -0.15, -0.25 are used. When straight line enhancement is desired, 1.0, 0.0, -0.1, -0.17, -0.22 are used. [Ref. 13]

b. Compatibility coefficients between line and no-lines

As its standard value, -0.1 is used. Decreasing the value to -0.5 increases the effect to fill the gap between line segments. [Ref. 13] The standard value is 0.25.
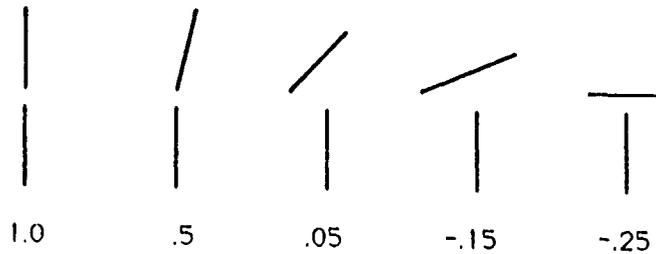
Figure 7. Compatibility weights between line labels

c. Compatibility coefficients between no-lines

Increasing the value up to 0.5 improve the noise effect but, may cause erasure of line labels. [Ref. 12]

For each pixel k, its neighbor pixel has 9x9 cases of interaction with the center pixel. If neighborhood size is 5x5. There are 5x5x9x9 cases of compatibility coefficients, $c(k,s;l,t)$.

$$\begin{vmatrix} c(k,1;1,1) & . & . & c(k,1;25,1) \\ . & . & . & . \\ . & . & . & . \\ c(k,9;1,9) & . & . & c(k,9;25,9) \end{vmatrix}$$

3. Updating probability

The updating process can be expressed in terms of compatibility functions. Let $q'(k,s)$ represents the increment applied to $p'(k,s)$, which is the probability that pixel k belongs to class s. The probabilities of each pixel in each class are computed as follows

$$q^r(k,s) = \sum_l \sum_t c(k,s;l,t)p^r(l,t) \tag{3.21}$$

where s represents nine classes of reference pixel and t represents nine classes of neighboring pixels and

$$p^{r+1}(k,s) = \frac{p^r(k,s)[1+q^r(k,s)]}{\sum_s p^r(k,s)(1+q^r(k,s))]} \tag{3.22}$$

C. LINE AND CURVE ENHANCEMENT(RXLA1)

This algorithm is delivered from the original idea developed by S.Peleg and A.Rosenfeld [Ref. 14]. RXLA1 is an acronymn for "RelaXation method for Line And curve." Basic concept using in this algorithm is the same as that used in the RXLN(see section B) except that the method for compatibility coefficient is different.

1. Compatibility coefficient

One possible interpretation of the compatibilies is in terms of statistical correlation. Correlation has properties as follows

    a. If class s and class t are compatible for pixel k and pixel l, $c(k,s;l,t) > 0$

    b. If class s and class t are incompatible for pixel k and pixel l, $c(k,s;l,t) < 0$

    c. If neither class is constrained by the other, $c(k,s;l,t) = 0$

    d. The magnitude of $c(k,s;l,t)$ represents the strength of the compatibility.

Estimates of the correlation coefficients derived from analyzing the initial probability are

$$c(k,s;l,t) = \frac{[p(k,s)-\bar{p}(k,s)][p(l,t)-\bar{p}(l,t)]}{\sigma(s)\sigma(t)} \tag{3.23}$$

$p(k,s)$ is the initial estimate of the probability of pixel k with class s, $\bar{p}(k,s)$ is the average of $p(k,s)$ for all pixels k, $\sigma(s)$ is the standard deviation of $p(k,s)$. If the neighborhood size is 5×5, $5 \times 5 \times 9 \times 9$ cases exist for each pixel as below

$$\begin{vmatrix} c(k,1;1,1) & . & . & c(k,1;25,1) \\ & . & . & . \\ & . & . & . \\ c(k,9;1,9) & . & . & c(k,9;25,9) \end{vmatrix}$$

Then, for every case and for all pixel k we have the summation as given by

$$
\begin{vmatrix} c(k,1;1,1) & . & . & c(k,1;25,1) \\ . & . & . & . \\ . & . & . & . \\ C(k,9;1,9) & . & . & c(k,9;25,9) \end{vmatrix} = \begin{vmatrix} c(k_1,1;1,1) & . & . & c(k_1,1,25,1) \\ . & . & . & . \\ . & . & . & . \\ c(k_1,9;1,9) & . & . & c(k_1,9;25,9) \end{vmatrix}
$$

$$
+ \begin{vmatrix} c(k_2,1;1,1) & . & . & c(k_2,1;25,1) \\ . & . & . & . \\ . & . & . & . \\ c(k_2,9;1,9) & . & . & c(k_2,9;25,9) \end{vmatrix}
$$

$$
+ \ldots
$$

Therefore

$$
c(k,s;l,t) = \frac{\sum_k [p(k,s) - \bar{p}(k,s)][p(l,t) - \bar{p}(l,t)]}{\sigma(s)\sigma(t)} \tag{3.24}
$$

In this algorithm, there is the effect of dominance, that is, when one class dominates the picture, its correlation coefficients with all other classes are high. Thus in the relaxation updating, the dominant class gets most of the support, and after a few iterations almost all the pixels have this class. The effect of dominance among classes can be alleviated by weighting the $c(k,s;l,t)$ by the probabilities that the corresponding classes do not occur. This greatly reduces the values of the coefficients involving dominant classes, but will only slightly reduce the values of the coefficients involving rare classes. The compatibility coefficients used in this algoritm are

$$
c(k,s;l,t) = [1 - \bar{p}(k,s)][1 - \bar{p}(l,t)] \times c(k,l,s,t) \tag{3.25}
$$

Two line and edge detection algorithms using relaxation technique were discussed in this chapter. These algorithms will be applied to the lofargram for line detection. Experimental results of these algorithms will be discussed in Chapter IV.

IV. EXPERIMENTAL RESULTS AND A MODIFIED RELAXATION APPROACH

In the experiments, four different relaxation algorithms are applied to lofargrams having different S/N. Both real and artificially created lofargrams were used as test images. As discussed previously, two algorithms RXLN and RXLA1 are line detection algorithms while the third algorithm RXEG is an edge detection algorithm. The purpose of the experiment is to test these three algorithms to see how they work for lofargrams. The secondary purpose of these experiments is to obtain thin lines for lofargram interpretation. Because the results of these algorithms were not satisfactory, improvement of the most suitable algorithms are suggested.

The results of the line detection algorithms RXLN and RXLA1 were not encouraging when the S/N is low. The reason will be explained later in this chapter. The results improve as the S/N increases. One effect is that the noise was removed very slowly in iterations of reinforcement. On the other hand, the edge detection algorithm, RXEG showed better results than the above two algorithms in the presence of low S/N. But the results were still not ideal. A double line was shown when the S/N was low because RXEG was the edge detection algorithm. Therefore, some modification is needed for line detection in the RXEG. This will be discussed in details later.

Two parts in the RXEG algorithm were modified. The first one is the initial magnitude of the pixel, since the outputs are greatly influenced by the initial magnitude of the input image. The other is the initial direction of the pixel. Because RXEG uses a differential type of edge detection operator, the initial direction in RXEG is not appropriate for the lofargram line detection. In RXEG the edge direction has a dark side which indicates the outside of the object and a bright side which indicates the inside of the object body. But the direction of a line simply indicates the potential direction of a track in a lofargram. Detail will be discussed in section C.

In a search for a better initial magnitude and direction, many other algorithms were tested. It assures that the magnitude and direction output from EGPR which is another line detection can be useful. The RXEG was modified and tested accordingly.

A. LINE DETECTION ALGORITHMS (RXLN, RXLA1)

Both RXLN and RXLA1 are line detection algorithms which use similar methods for initial probability and compatibility coefficient calculations as explained in Chapter

23

III. But the applications of these are very discouraging. Figure 8 shows a real lofargram as the input image and its initial magnitude calculated in RXLN. Figure 9 shows the results of two iterations of reinforcement. And Figure 10 shows an artificial lofargram with S/N of 6 db and its initial magnitude image. Figure 11 show the results of the four iterations of reinforcement. And Figure 12 show the results for 3 db and 6 db of S/N.

The initial magnitude of input image shows that the result is very noisy. In these algorithms, there is the effect of dominance. When one class dominates the picture, high value is assigned to the dominating class at each pixel and low values are assigned to the other classes. Thus, the dominant class gets the most support in the relaxation updating process. And after a few iterations almost all the pixels have the same class. Figure 8 shows that the noise dominates the picture when the lofargram has low S/N. Figure 10 (b) still shows that the noise dominates the picture. When the S/N is higher, the signal showed up weakly as shown in Figure 11. But noise still dominates the picture. Generally noise dominates the lofargram. Noise strongly dominates the lofargram when the S/N is low. Hence the RXLN and RXLA1 algorithms are not suitable for lofargram processing.

B. THE RESULTS OF THE EDGE DETECTION ALGORITHM RXEG

Figure 13 shows the initial magnitude of the RXEG of the input image. Figure 14 shows the results of the reinforcement applied to Figure 13. In this implementation, neighborhood size is 5x5, then the 24 neighboring pixels influenced the centered pixel. Figure 15 shows the artificial lofargram input image and its initial magnitude with 6db of S/N. The iteration scheme has considerable noise removal power as shown in the example of Figure 14. Figure 16 shows that the results of the artificial lofargram with higher S/N can yield a stronger response. A double line in Figure 14 (b) and Figure 16 indicate that the edge was detected from the input image. Because RXEG uses the edge detection operator, double lines are shown at lower iterations, The weaker line of the double lines disappeared gradually in the relaxation process with 4 successive iterations as shown in Figure 14. Figure 17 shows the results for 3db and 6db of S/N. A desirable example of edge detection is shown in Figure 18.

C. MODIFIED EDGE DETECTION ALGORITHM RXEG

The results of RXEG shows that this algorithm works as an edge detection algorithm. It can detect the edges in the lofargram, even though the S/N is very low. Double lines in Figure 15 (b) shows an example of the edge detection.
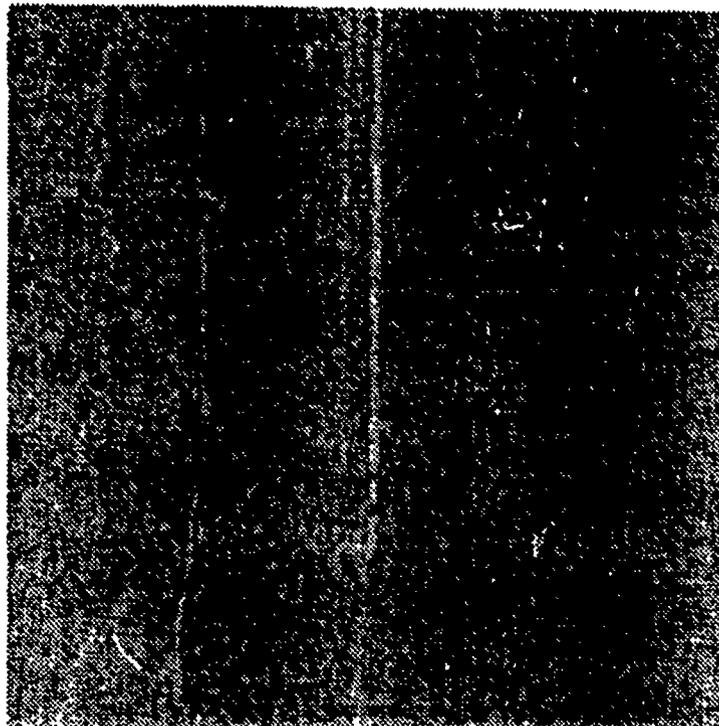
(a)
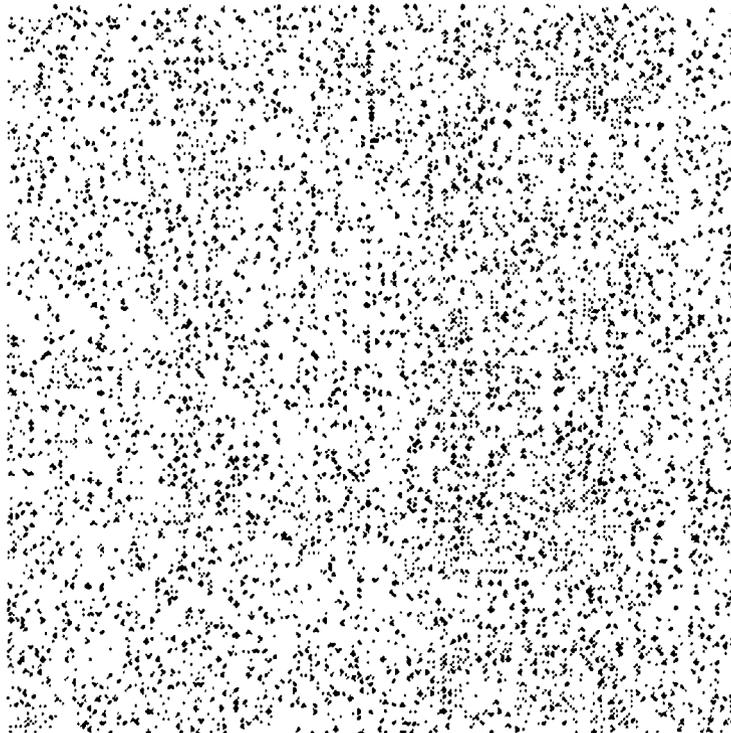
(b)

Figure 8.   A real lofargram input image and its initial magnitude from RXLN
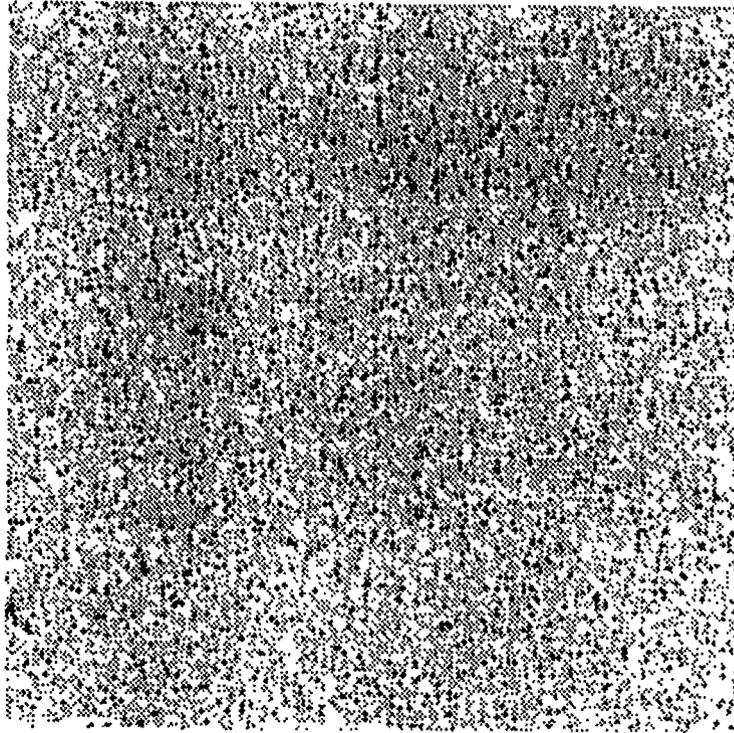
(a)
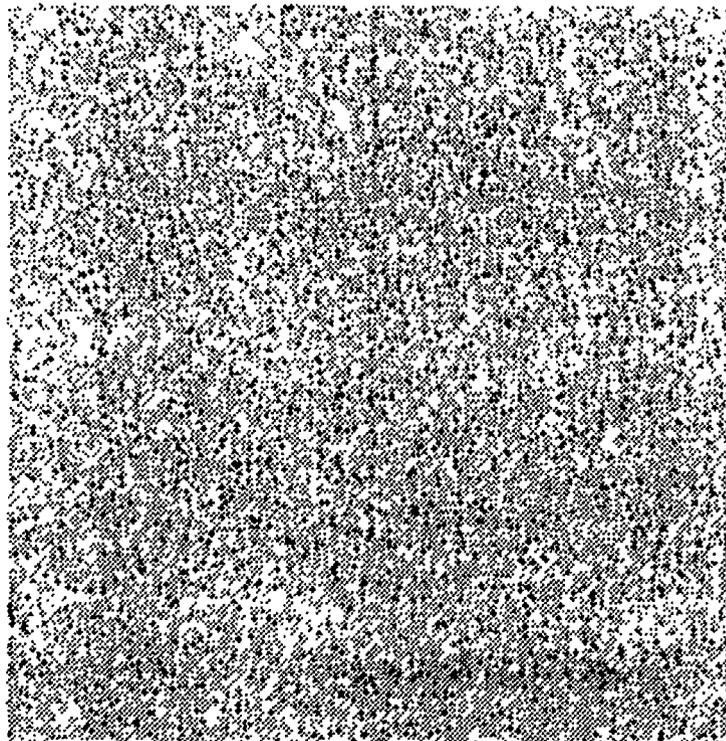
(b)

Figure 9. Two iterations(1,2) of reinforcement process in RXLN

(a)
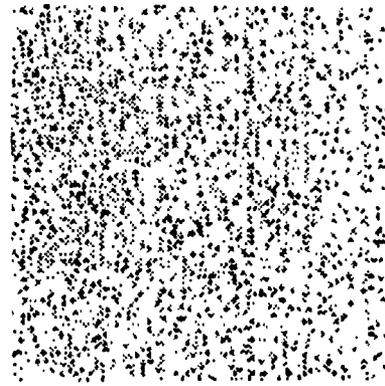
(b)

Figure 10.   Artificial lofargram of 6db and its initial magnitude from RXLN
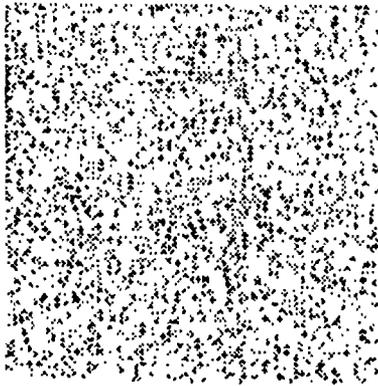
(a)
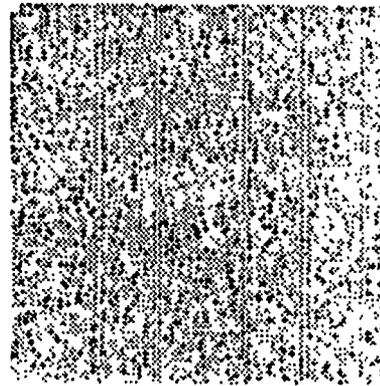
(b)
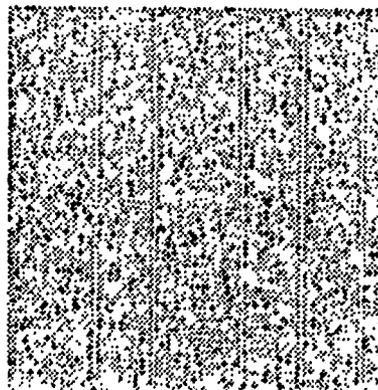
(c)
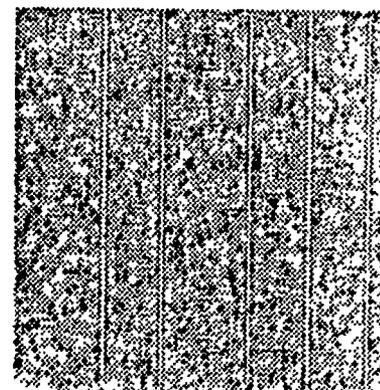
(d)

Figure 11.   Results of four iterations (1,2,3,4) of reinforcement from Fig 10
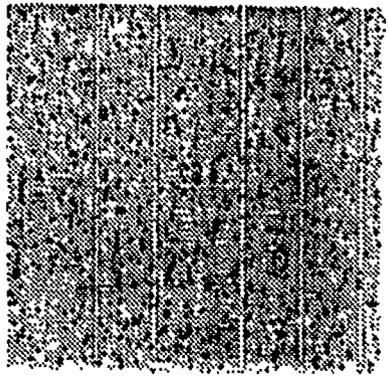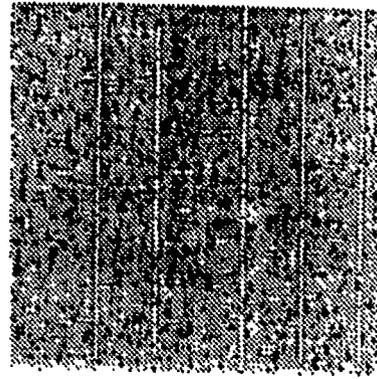
(a)　　　　　　　　　　　　　(b)

Figure 12.　Results of two lofargrams, (a) 3db (b) 6db. Results of the reinforcement with 5 iterations.

Figure 13.　Initial magnitude of Fig 8 (a) in RXEG. The Prewitt operator is used for initial probability.
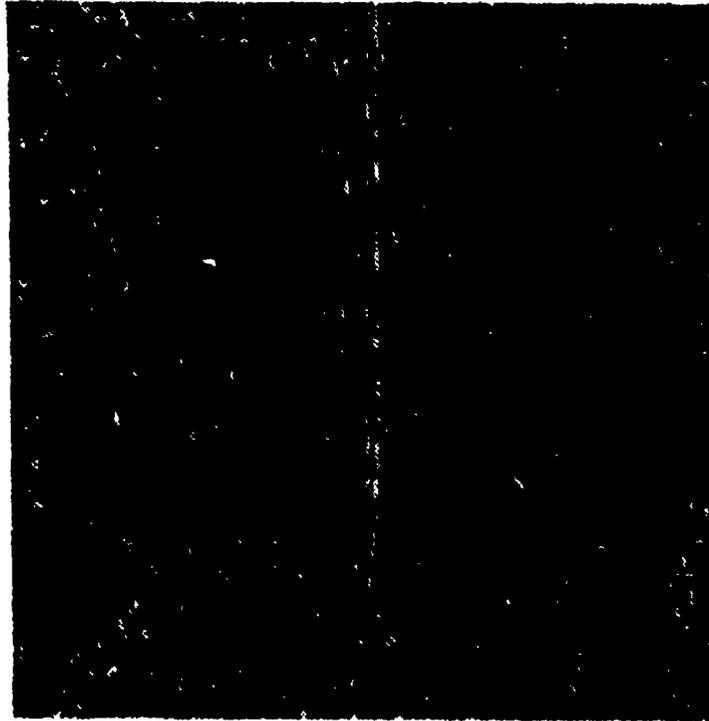
(a)
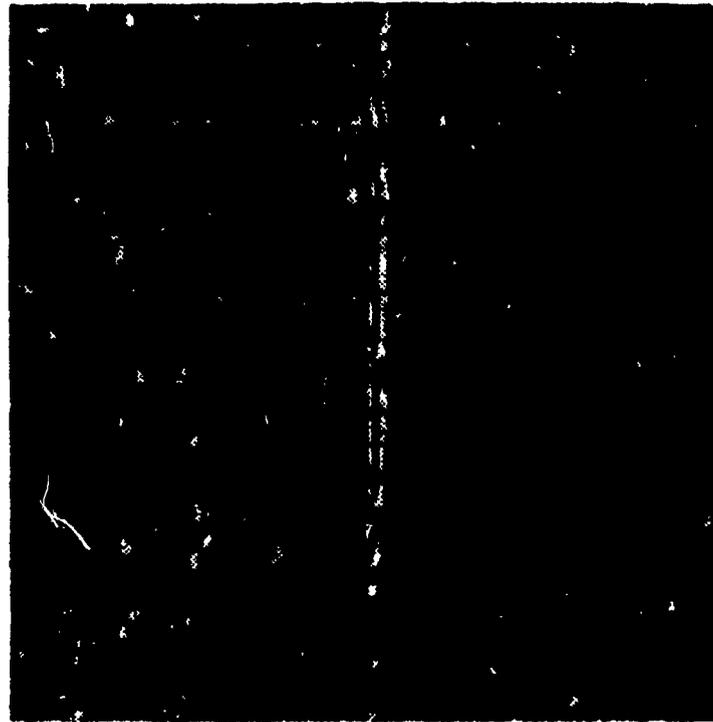
(b)

Figure 14.   Four iterations (1,2,3,4) of RXEG
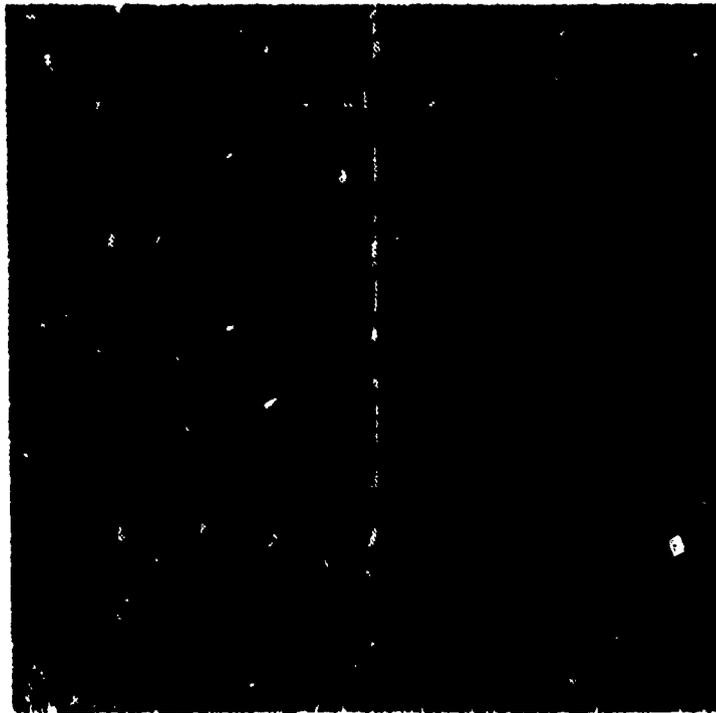
(c)
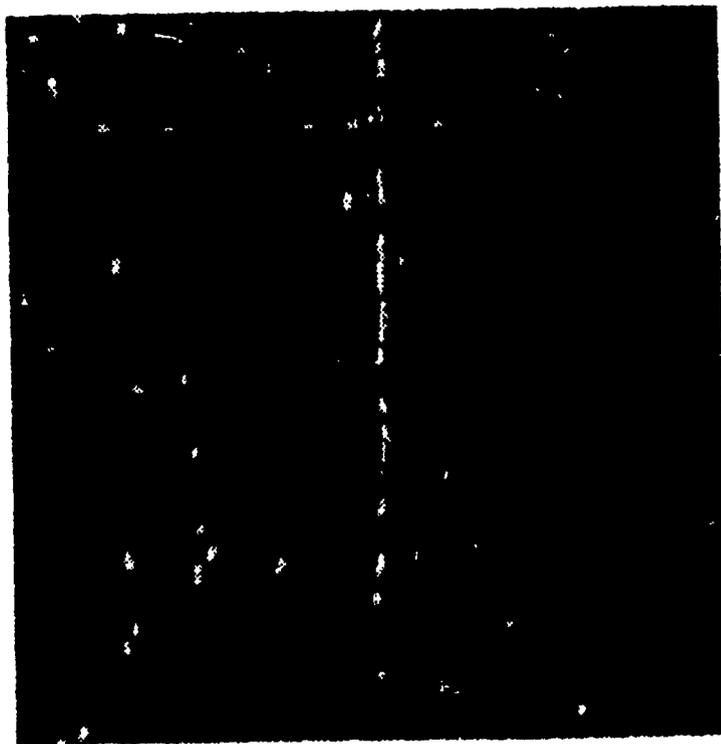
(d)

Figure 14.   Four iterations (1,2,3,4) of RXEG (continue)
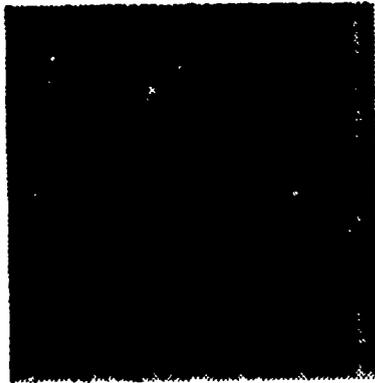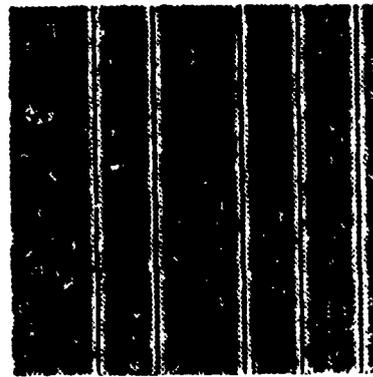
(a)   (b)

Figure 15.   Artificial lofargram with 6db S/N and its initial magnitude
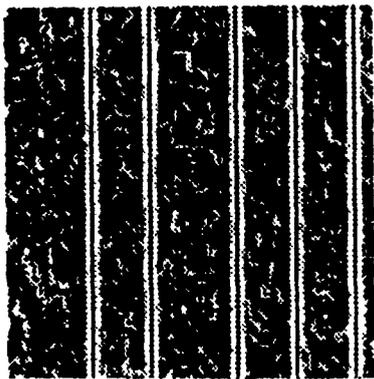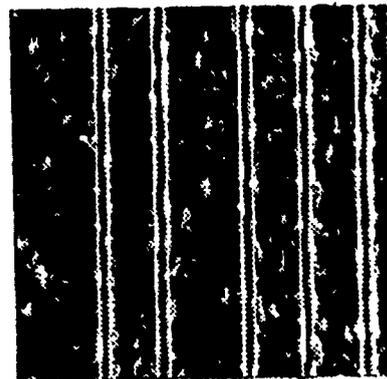
(a)   (b)
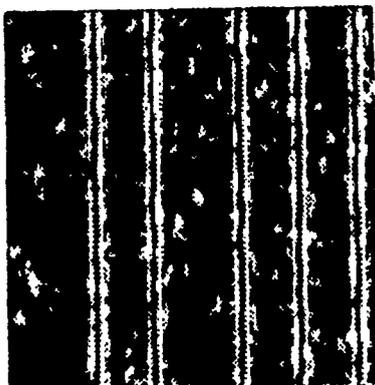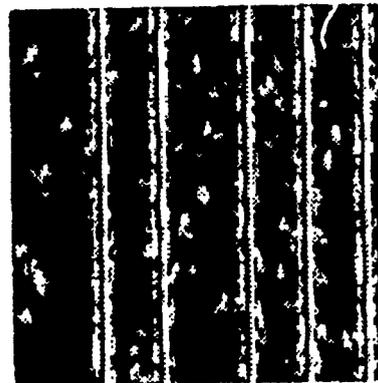
(c)   (d)

Figure 16.   Results of four iterations(1.2,3,4) of reinforcement from Fig 15
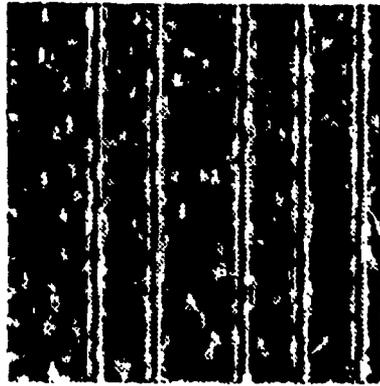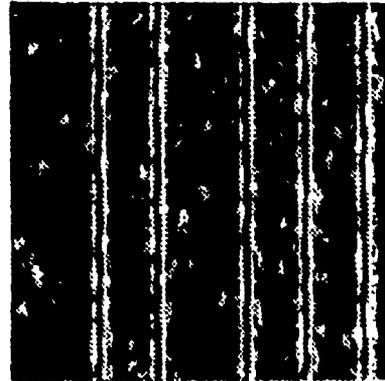
Figure 17.   Results of two lofargrams, (a) 3db (b) 6db.

Results of the reinforcement with 5 iterations. However, the purpose of lofargram processing is to detect the line of the signal for the interpretation of the lofargram. The desirable ideal example of the line detection is shown in Figure 19. To accomplish this, some changes in RXEG are required. Many algorithms were tested to generate the edge magnitude and line direction for the initial probability. The new magnitude and direction obtained from EGPR were adopted in the modified algorithm.
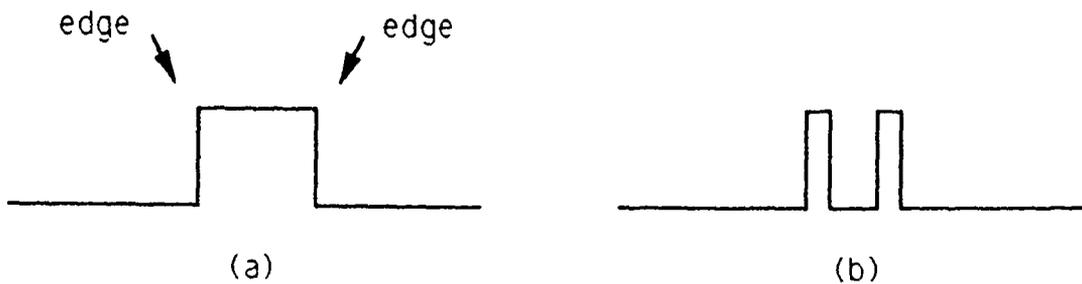
Figure 18.   Example of edge detection process (a) input (b) output

1. Edge magnitude output

The gray level of the edge enhanced magnitude of each pixel are processed by masks. Nine masks are formed around a pixel in the shape of pentagon, hexagon and rectangular as shown in Figure 20. The average values and the variances within the masks around a pixel are calculated. Then, the average value of the mask with minimum variance is assigned as the gray level of the center pixel.

32
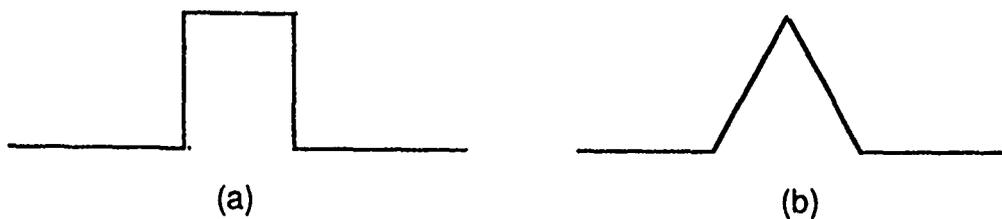
Figure 19. Example of line detection process (a) input (b) output
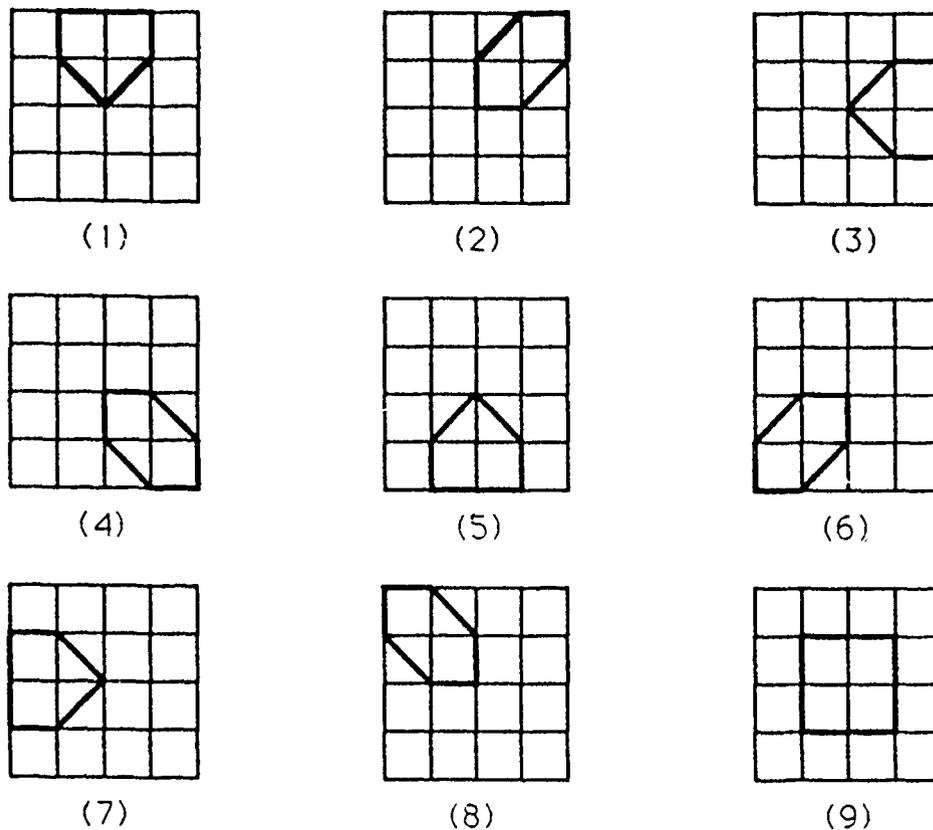
Figure 20. Nine masks from EGPR

2. Line direction output

The initial line direction of each pixel is determined from the direction of the mask with the minimum variance. Eight line directions can be associated with each image point. The directions are symmetric to the origin. Therefore, only four of them are considered as $0° = 180°$, $45° = 225°$, $90° = 270°$ and $135° = 315°$. The direction for an edge is the direction along the edge with the dark side of the edge on the left and the bright side on the right side. This means that the edge direction $0°$ and $180°$ are different di-

rections. However, the directions for line detection from masks in EGPR do not have dark side and bright side. The direction in EGPR simply implies the direction of a line without dark side and bright side. This means that the line direction 0˚ and 180˚ are the same direction as shown in Figure 21. The edge direction with bright on the right side are shown in Figure 22.
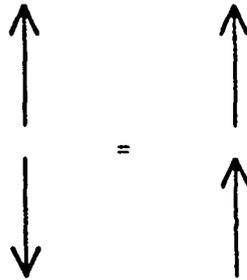
Figure 21.   An example of line direction
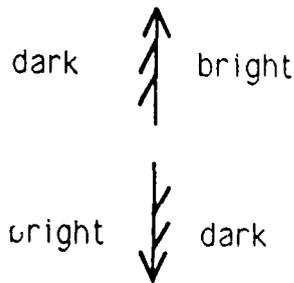
Figure 22.   An example of edge direction

The four directions generated from the EGPR are shown in Figure 13. But, start with the second iteration after the first relaxation process, infinite line directions are considered at each pixel.

3. Pixel and neighbor interaction

Edge detection in RXEG detects double edges of the signal line in the lofargram. It can detect the edges reliably when the S/N is high. But, the purpose here is to detect thin line as shown in Figure 19. For the thin line detection, it is necessary to construct a compatibility function with proper compatibility values for the interactions between different classes. Basically, the same compatibility function as that of the RXEG can be used for the interactions between lines. The only difference between the line direction

34
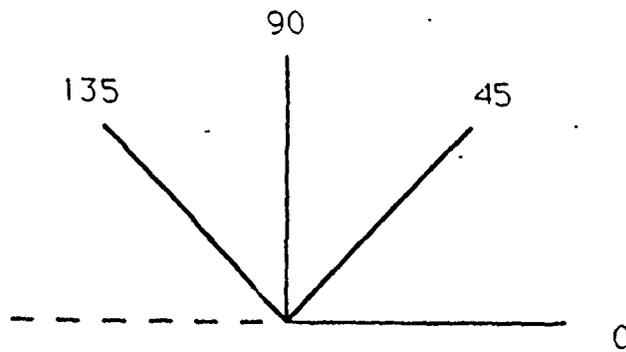
Figure 2.3.   Four possible line directions

and the edge direction is that the directionality of a line greater than 180˚ is considered as the same as its symmetric value through the origin. In other words, the compatibilities of lines between the center pixel and the neighboring pixels can be handled in the same way as in RXEG.

The center pixel k could have four kinds of interactions with its neighbor pixel l, i.e, line/line, line/no-line, no-line/line and no-line/no-line.

a.   Line/line interaction

Let k be the centered pixel position, and l the neighbor position. Let α be the line direction at k, β the line direction at the neighbor pixel l, γ be the direction of line joining k to l, D the chessboard distance from k to l, i.e, $\max(|k_x-l_x|,|k_y-l_y|)$. Then the line/line reinforcement process between k and l has strength given by

$$c(k,l;l,l) = \cos(\alpha-\gamma)\,\cos(\beta-\gamma)/2^D \qquad (4.1)$$

which is similar to eq (3.4). Here l represents the line class and n represents non-line class. To see the significance of this definition, a few simple examples are considered in Table 4.   Results changed with different compatibility value of parallel lines.   In the experiment, when a higher value of compatibility is chosen for parallel lines, background noise increases, while a lower value is chosen when the opposite situation takes place.

b.   Line/no-line interaction

The line probability at pixel k is weakened by the no-line probability at pixel l to the degree $c(k,l;l,n)$ defined by

$$c(k,l;l,n) = \min[0, -\cos(2\alpha-2\gamma)/2^D] \qquad (4.2)$$

which is similar to eq (3.5). A few examples are shown in Table 5.

35

c. No-line/line interaction

The no-line probability at pixel k is affected by the line probability at the neighbor pixel l to the degree $c(k,n;l,l)$ defined by

$$c(k,n;l,l) = \frac{(1 - \cos(2\beta - 2\gamma))}{2^{D+1}} \tag{4.3}$$

which is similar to eq (3.6). A few examples are shown in Table 6.

case	α	β	γ	$\cos(\alpha - \gamma)\cos(\beta - \gamma)/2^D$
		90	90	90	$1/2^D$
|—	90	0	0	0
			90	90	0	—

Table 4. Examples of line / line interaction

case	α	γ	$-\cos(2\alpha - 2\gamma)/2^D$	$c(k, l; l, n)$
	•	90	0	$1/2^D$	0
		90	90	$-1/2^D$	$-1/2^D$
		90	270	$-1/2^D$	$-1/2^D$

Table 5. Examples of line / no-line interaction

case	α	β	γ	$(1 - \cos(2\beta - 2\gamma))/2^D$
•		0	90	0	$1/2^D$
		0	90	90	0

Table 6. Examples of no-line / line interaction

36

d. No-line/no-line interaction

The no-line probabilities at the centered pixel k and that of the neighbor pixel l reinforce each other to the degree $c(k,n;l,n)$ defined by

$$c(k,n;l,n) = \frac{1}{2^D} \tag{4.4}$$

which is similar to eq (3.7)

4. Combined reinforcement process

For each pixel k, the net effect of its neighboring pixel on its probability $p(k,l)$ and no-line probability $p(k,n) = 1-p(k,l)$ is computed as follows

$$q^r(k,l) = \sum_l C_1 p^r(l,l) c(k,l;l,l)$$

$$+ \sum_l C_2 p^r(l,n) c(k,l;l,n) \tag{4.5}$$

$$q^r(k,n) = \sum_l C_3 p^r(l,l) c(k,n;l,l)$$

$$+ \sum_l C_4 p^r(l,n) c(k,n;l,n) \tag{4.6}$$

where C_1, C_2, C_3, C_4 are constants whose sum is taken to be one. The standard values needed here are $C_1 = 0.866$, $C_2 = 0.124$, $C_3 = 0.005$, and $C_4 = 0.005$. The results of the iteration process are somewhat sensitive to the choice of the C's. For example, if C_1 is too large, the line will thicken and will be extended into no-line points; while if C_4 is too large, gaps will appear at weak spots in the lines and at sharp angles.

$$q'(k,l) = \frac{q^r(k,l)}{|q^r(k,l)| + |q^r(k,n)|} \tag{4.7}$$

$$q'(k,n) = \frac{q^r(k,n)}{|q'(k,l)| + |q^r(k,n)|} \tag{4.8}$$

37

$$p'(k,l) = p^r(k,l)[1+q'(k,l)]$$ (4.9)

$$p'(k,n) = p^r(k,n)[1+q'(k,n)]$$ (4.10)

$$p^{r+1}(k,l) = \frac{p'(k,l)}{p'(k,l)+p'(k,n)}$$ (4.11)

This process is then iterated with $p^{r+1}(k,l)$ replacing $p^r(k,l)$, and $1-p^{r+1}(k,l)$ replacing $p^r(k,n)$. We also compute the estimated line direction at each point

$$\Delta'_x(k) = Wp^r(k,l) \cos(\theta(k))$$

$$+ \sum_l p'(l,l)c(k,l;l,l) \cos(\theta(l))$$ (4.12)

$$\Delta'_y(k) = Wp^r(k,l) \sin(\theta(k))$$

$$+ \sum_l p^r(l,l)c(k,l;l,l) \sin(\theta(l))$$ (4.13)

$$\theta^{r+1}(k) = \tan^{-1}(\Delta'_y(k)/\Delta'_x(k))$$ (4.14)

where W is a constant. For large values of the constant W, θ^{r+1} is close to θ; while for small values, it is strongly influenced by the neighboring θ's.

5. Experimental results

Figure 24 shows the initial magnitude of the input image of Figure 8 (a). Figure 25 shows the results for 3db and 6db of S/N. Figure 26 shows the results of four iterations of the reinforcement applied to Figure 24 using the modified RXEG. Figure 27 shows the artificial lofargram with 6db of S/N and its initial magnitude. Figure 28 shows the results of four iterations of reinforcement with 6db S/N artificial lofargram. Comparing Figure 13 and Figure 24 reveals that the initial magnitudes of the modified RXEG are improved by using the EGPR line detection. The initial magnitude of modified RXEG shows stronger signal line in Figure 24 when compared to Figure 13. However, the end results of reinforcement do not improve significantly. From Figure 28, it is obvious that a double line still exists and that the lines are thicker than those of the RXEG

algorithm. In this experiment, the purpose is to obtain single thin line from the lofargram. But the result is not fully satisfactory. A few things could be dis ussed with respect to the suspected reasons of the double lines and of the thicker line in the result. The initial magnitude as shown in Figure 27 (b) was caused by the Prewitt operator of the RXEG. In Figure 15 the result of RXEG shows strongly the splitting line. Therefore, the reinforcement process couldn't be the problem of causing double lines. Figure 29 shows the diagram of the modified RXEG. As for the possible cause of the thicker line, the modified initial magnitude may be the reason. Comparison of Fig 15 (b) and Fig 27 (b) reveals the improvement of the initial magnitude. The initial magnitude of the modified RXEG shows stronger and thicker lines. Consequently, the results of the reinforcement at a later stage applied to this initial magnitude show thicker lines.
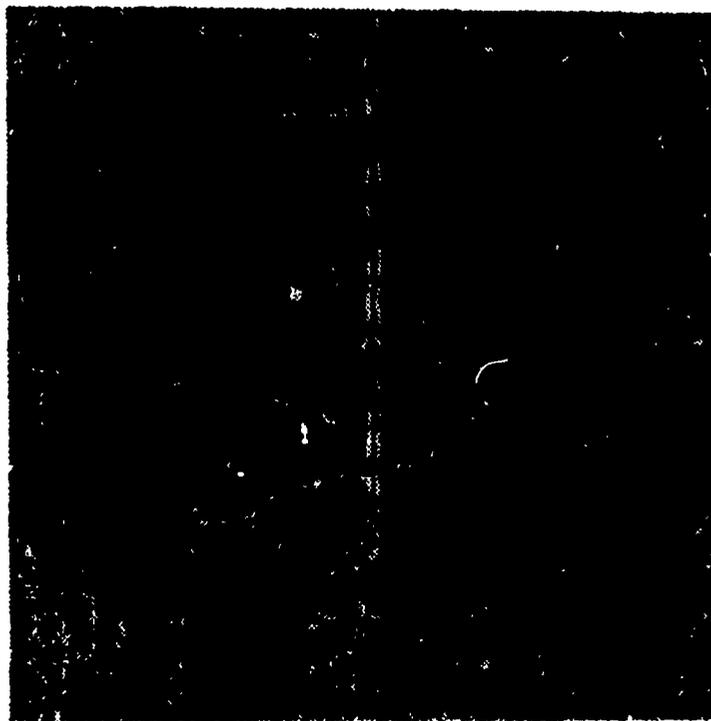
Figure 24.   Initial magnitude of Fig 15 (a) in modified RXEG

To solve the double line problem, it is necessary to find a method to yield the initial magnitude in single lines. Two different method were tested for this purpose. In the first method each pixel was divided by the maximum value over the entire picture as follow.

39

$$mag = \frac{p(k,ln)}{\max\limits_{k} p(k,ln)} \tag{4.15}$$

This method removed noise very quickly. The initial magnitudes at each pixel are calculated by dividing each pixel by the maximum values over the entire picture. Therefore, very small values are assigned to noise and the high values are assigned to the signal line when the S/N was high. On the other hand, when S/N is low, the high values are assigned to both the noise and the signal because the difference of gray level between the noise and the signal is small. As a result, this method is still not suitable to lofargram processing. In the second method, each pixel was divided by the maximum value of the local neighborhood as follow

$$mag = \frac{p(k,ln)}{\max\limits_{l} p(l,ln)} \tag{4.16}$$

Initial magnitude in this method depends on the local maximum value. In the experiments it is realized the initial magnitude could be white or dark entirely depending on the characteristic of the noise of the lofargram. This second method was not appropriate for the calculation of the initial magnitude before the relaxation process.

In this chapter various results of the algorithm studied were discussed. Thick line problem were discovered. Several attempts has been made to solve this problem.

(a)   (b)
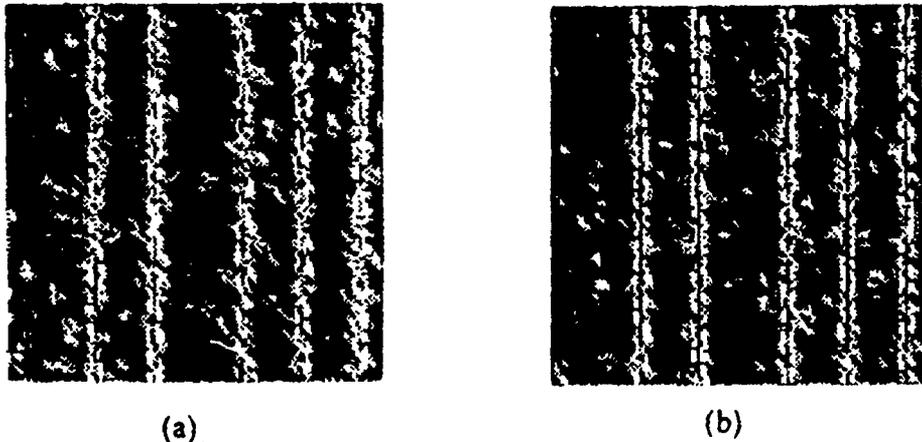
Figure 25.   Results of two lofargram in modified RXEG. (a) 3db (b) 6db
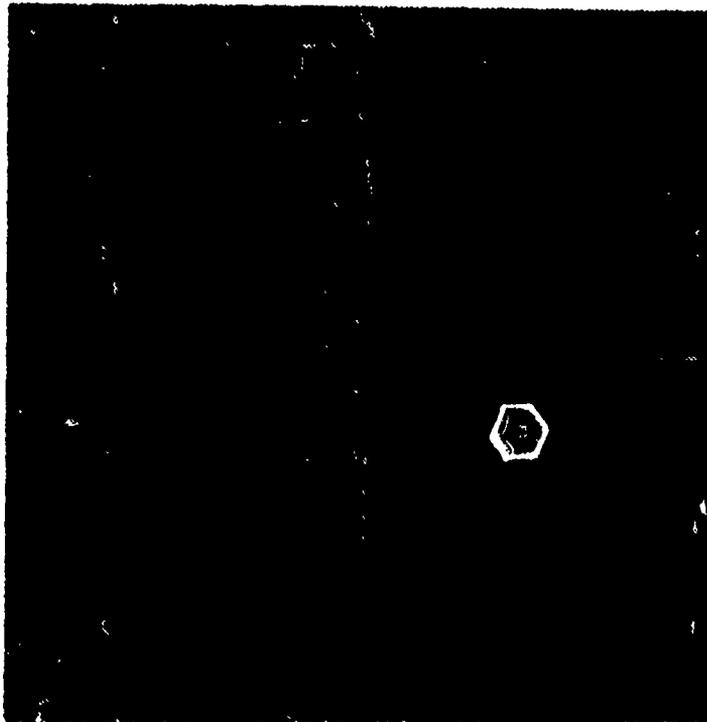
(a)
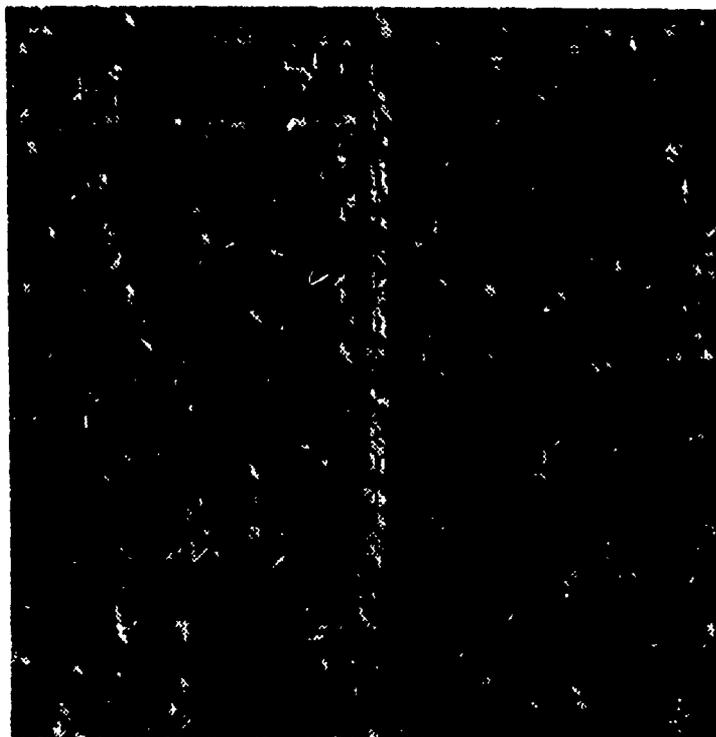
(b)

Figure 26. Four iterations (1,2,3,4) of modified RXEG

(c)
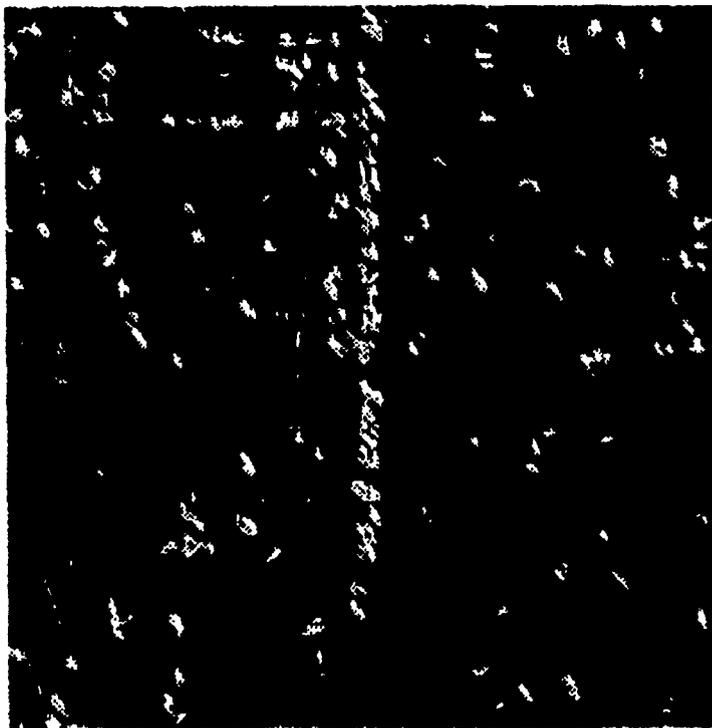
(d)

Figure 26.   Four iterations (1,2,3,4) of modified RXEG (continue)
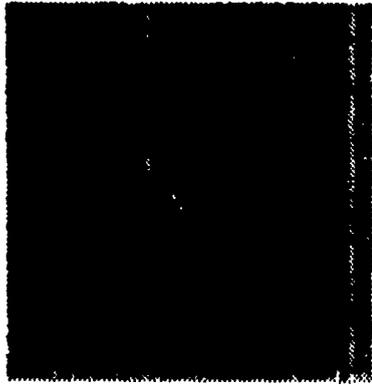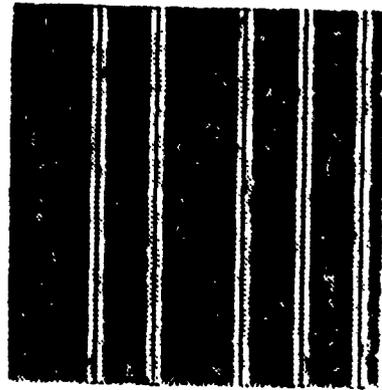
(a)   (b)

Figure 27.   Artificial lofargram with 6db S/N and its initial magnitude
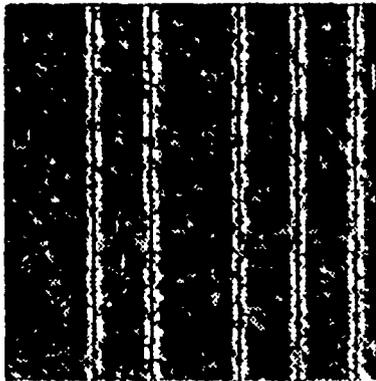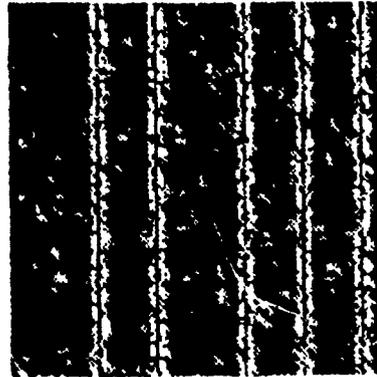
(a)   (b)
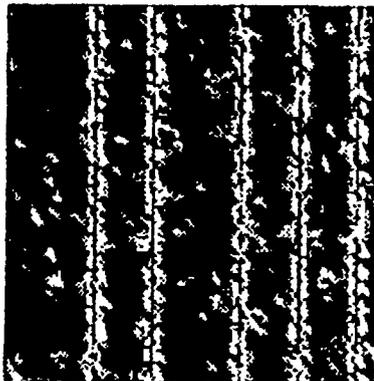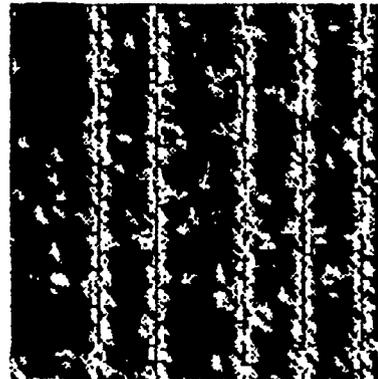
(c)   (d)

Figure 28.   Results of four iterations (1,2,3,4) from Fig 27 in modified RXEG

43
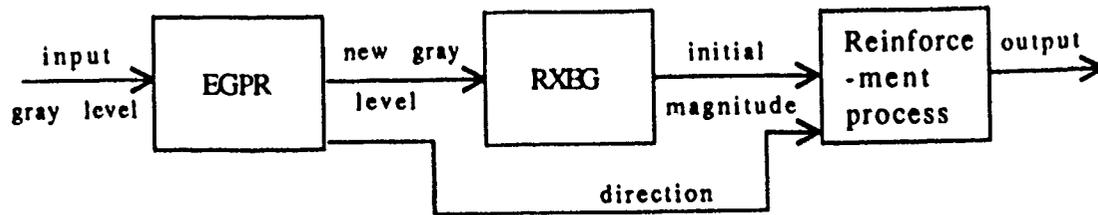
Figure 29. The procedure of modified RXEG

V. CONCLUSIONS AND RECOMMENDATION

The fact that each algorithm has its own peculiar characteristics in the lofargram processing was found in this study. These characteristics are very much dependent on the S/N of the input lofargram. Algorithms RXLN and RXLA1 using similar ideas can detect lines in the lofargram when the S/N is relatively high. But, the result were very discouraging when the S/N was low and noise smoothing was done slowly in the iterations of the reinforcement. Our main concern of the lofargram processing is to detect a thin line when S/N is low. Therefore, these algorithms are not suitable for lofargram process. On the other hand, the RXEG algorithm results in good lofargram processing. The only short coming is that it is an edge detection algorithm. It smoothes the noise and enhances the line components in the lofargram even at low S/N. The last algorithm studied in this thesis is the modified RXEG algorithm. Several method were tested to solve the double line problem. But, double line problem still exists in the modified algorithm. To solve this problem, it is necessary to search for an algorithm that can yield thin single line in the image of the initial magnitude.

APPENDIX. THE PROGRAM OF MODIFIED RXEG

```
      INTEGER*4 ISX/256/,ISY/256/,ITER/4/,ISNX/5/,ISNY/5/,ISW/1/
      INTEGER*4  IP(256,256),IPRB(256,256),JP(256,256)
      DIMENSION C(4),CTYPE(4,2),THETO(256,256)
      DIMENSION PRBE1(256,256),PRBE2(256,256),Q(8)
      DIMENSION THET1(256,256),THET2(256,256)
      BYTE    G(256)
      REAL LENGTH, W/3.0/,MAX,MIN
      OPEN(UNIT=1,NAME='BEAM00.BIN',TYPE='OLD',ACCESS='DIRECT',
     2        RECORDSIZE=64,MAXREC=256)
      OPEN(UNIT=2,NAME='TLOFAR20-4X5.DAT',TYPE='NEW',ACCESS='DIRECT',
     2        RECORDSIZE=64,MAXREC=256)


      DO 10 I=1,256
         READ(1'I)G
         DO 20 J=1,256

            IP(I,J)=G(J)
            IF(IP(I,J).LT.0) IP(I,J)=IP(I,J)+256
20       CONTINUE
10    CONTINUE
      CALL RXEG(IP,THET1,THET2,PRBE1,PRBE2,ISX,ISY,ISNX,ISNY,ITER,
     &            ISW,TEMP,JP,THETO)
      MAX=PRBE2(1,1)
      MIN=PRBE2(1,1)
      DO 30 I=1,256
         DO 40 J=1,256
            IF(MAX.LT.PRBE2(I,J)) MAX=PRBE2(I,J)
            IF(MIN.GT.PRBE2(I,J)) MIN=PRBE2(I,J)
40       CONTINUE
30    CONTINUE
      LENGTH=(MAX-MIN)
      DO 50 I=1,256
         DO 60 J=1,256
            IPRB(I,J)=JNINT(((PRBE2(I,J)-MIN)/LENGTH)*243.)
60       CONTINUE
50    CONTINUE
      DO 70 I=1,256
         DO 80 J=1,256
            IF(IPRB(I,J).GT.127) THEN
              G(J)=IPRB(I,J)-256
            ELSE
              G(J)=IPRB(I,J)
            ENDIF
80       CONTINUE
         WRITE(2'I)G
70    CONTINUE
      END
      SUBROUTINE RXEG(IP,THET1,THET2,PRBE1,PRBE2,ISX,ISY
     &                ,ISNX,ISNY,ITER,ISW,TEMP,JP,THETO)
C
C
CP    Edge reinforcement by relaxation method.
```

```
C
CS      CALL RXEG(IP,THET1,THET2,PRBE1,PRBE2,ISX,ISY
CS     ̄           ,ISNX,ISNY,ITER,ISW,TEMP)
C
CK      RELAXATION, EDGE
C
CA      IP(ISX,ISY)    :     Input image                           (I)
CA      THET1(ISX,ISY):     Gradient direction after ITER-1
CA                          iterations                            (W)
CA      THㄴT2(ISX,ISY):     Gradient direction after ITER iterations  (O)
CA      PRBE1(ISX,ISY):     Edge probability for each pixel after
CA                          ITER-1 iterations                     (W)
CA      PRBE2(ISX,ISY):     Edge probability for each pixel after
CA                          ITER-1 iterations                     (O)
CA      ISNX,ISNY     :     Size of neighborhood being considered
CA                          (odd number, for example: 5,5)        (I)
CA      ITER          :     Number of iterations                  (I)
CA      ISW           :     Switch (1 - standard, 2 - noise reduction
CA                          type                                  (I)
C
CN      Reference
CN @1!  B. J. Schachter, A. Lev, S  W. Zucker, and A. Rosenfeld, "An
CN      application of relaxation methods to edge reinforcement,"
CN      IEEE Trans. Syst., Man,  Cybern., vol. SMC-7, pp. 813-816,
CN      Nov. 1977.
C
CD       JULY 1979.  PROGRAMMED BY  K.SAKAUE
C
CM      (1) When computing the initial values for edge probabiſty and
CM          edge direction using an operator other than the Prewitt
CM          operator, rewrite lower-level subroutine RXEP. Any operators
CM          that ca obtain a differential (Dx,Dy) can be used.
CM      (2) The processing characteristic largely depends upon selection
CM          of coefficients C1, through C4 (See the reference.) The two
CM          standard kings of values have already been set in array CTYPE
CM          usㅣng data statements, and values can be changed with ISW.
CM          V. n using other values or changing values depending upon
CM          tᵤₑ number of iteration times, rewrite the program so a proper
CM          value is set to the argument C in lower-level subroutine RXEI.
CM          The following condition however, must be met.
CM             C(1)+C(2)+C(3)+C(4)=1
CM      (3) The coefficient W used when updating (THET1) is set in data
CM          statement to 3.0. The same value was also used in the referenc
CM          This value may be changed as neccessary.
C
        DIMENSION IP(ISX,ISY),JP(ISX,ISY)
        DIMENSION THET1(ISX,ISY),THET0(ISX,ISY)
        DIMENSION THET2(ISX,ISY)
        DIMENSION PRBE1(ISX,ISY)
        DIMENSION PRBE2(ISX,ISY),TEMP(256,256)
        DIMENSION CTYPE(4,2)
        DIMENSION C(4)
        DATA CTYPE/.866,.124,.005,.005
       &          ,.706,.176,.059,.059/
        DATA W/3.0/
C
CF      INITIAL EDGE VALUES AND INITIAL PROBABILITIES
C
```

```
      CALL EGPR(IP,JP,ISX,ISY,THET1,THET0)
      CALL RXEP(IP,THET1,PRBE1,ISX,ISY,TEMP)
C
CF    RELAXATION UPDATING PROCESS
C
      AMN=0.
      AMX=0.
      DO 9 I=1,4
        C(I)=CTYPE(I,ISW)
    9 CONTINUE
      DO 10 I=1,ITER
        IF(I.EQ.1)GO TO 11
        DO 12 IY=1,ISY
        DO 12 IX=1,ISX
          THET1(IX,IY)=THET2(IX,IY)
          PRBE1(IX,IY)=PRBE2(IX,IY)
   12   CONTINUE
   11 CONTINUE
      CALL RXEI(THET1,THET2,PRBE1,PRBE2,ISX,ISY,ISNX,ISNY,C,W)
   10 CONTINUE
      RETURN
      END




      SUBROUTINE EGPR(IP,JP,ISX,ISY,THET1,THET0)
C     Copyright (c) 1983 by AIST MITI(JAPAN)
C
CS    CALL EGPR(IP,JP,ISX,ISY)
C
CP    Edge preserving smoothing operation.
C
CK        EDGE PRESERVING SMOOTHING, SMOOTHING
C
CA    IP(ISX,ISY)    :  Input image array                        (I)
CA    JP(ISX,ISY)    :  Output image array                       (O)

M     The program is considerably long, but the main part consists
CM    of about 120 lines in the first half and the remainder is used
CM    as routines for edge processing.
C
      DIMENSION  IP(ISX,ISY),THET1(ISX,ISY),THET0(ISX,ISY)
      DIMENSION  JP(ISX,ISY)
      DIMENSION  A(9),  V(9),Q(9)
      DIMENSION  K(5,5),  L(5,5)
C
      EQUIVALENCE  (A(1),A1), (A(2),A2), (A(3),A3),
     &             (A(4),A4), (A(5),A5), (A(6),A6),
     &             (A(7),A7), (A(8),A8), (A(9),A9)
      EQUIVALENCE  (V(1),V1), (V(2),V2), (V(3),V3),
     &             (V(4),V4), (V(5),V5), (V(6),V6),
     &             (V(7),V7), (V(8),V8), (V(9),V9)
C
      EQUIVALENCE
     & (K11,K(1,1)),(K21,K(2,1)),(K31,K(3,1)),(K41,K(4,1)),(K51,K(5,1)),
     & (K12,K(1,2)),(K22,K(2,2)),(K32,K(3,2)),(K42,K(4,2)),(K52,K(5,2)),
     & (K13,K(1,3)),(K23,K(2,3)),(K33,K(3,3)),(K43,K(4,3)),(K53,K(5,3)),
```

```fortran
     &  (K14,K(1,4)),(K24,K(2,4)),(K34,K(3,4)),(K44,K(4,4)),(K54,K(5,4)),
     &  (K15,K(1,5)),(K25,K(2,5)),(K35,K(3,5)),(K45,K(4,5)),(K55,K(5,5))
        EQUIVALENCE
     &  (L11,L(1,1)),(L21,L(2,1)),(L31,L(3,1)),(L41,L(4,1)),(L51,L(5,1)),
     &  (L12,L(1,2)),(L22,L(2,2)),(L32,L(3,2)),(L42,L(4,2)),(L52,L(5,2)),
     &  (L13,L(1,3)),(L23,L(2,3)),(L33,L(3,3)),(L43,L(4,3)),(L53,L(5,3)),
     &  (L14,L(1,4)),(L24,L(2,4)),(L34,L(3,4)),(L44,L(4,4)),(L54,L(5,4)),
     &  (L15,L(1,5)),(L25,L(2,5)),(L35,L(3,5)),(L45,L(4,5)),(L55,L(5,5))
C
      AVE1(K1,K2,K3,K4,K5,K6,K7,K8,K9) =
     &         FLOAT(K1+K2+K3+K4+K5+K6+K7+K8+K9)/9.0
      AVE2(K1,K2,K3,K4,K5,K6,K7) =
     &         FLOAT(K1+K2+K3+K4+K5+K6+K7)/7.0
      VAR1(A1,L1,L2,L3,L4,L5,L6,L7,L8,L9) =
     &         7.0*(FLOAT(L1+L2+L3+L4+L5+L6+L7+L8+L9)-A1*A1*9.0)/9.0
      VAR2(A1,L1,L2,L3,L4,L5,L6,L7) =
     &         FLOAT(L1+L2+L3+L4+L5+L6+L7)-A1*A1*7.0
C
CF ***** Initialization
      ISX0 = ISX
      ISY0 = ISY
      ISX1 = ISX0-1
      ISY1 = ISY0-1
      ISX2 = ISX0-2
      ISY2 = ISY0-2
      ISX3 = ISX0-3
      ISY3 = ISY0-3
C
CF ***** Smoothing process for the region (3,3)--(ISX-2,ISY-2)
      DO 60 IY=3,ISY2
      IYM2 = IY-2
      IYP2 = IY+2
      DO 60 IX=3,ISX2
      IXM2 = IX-2
      IXP2 = IX+2
C
      KY = 0
      DO 20 JY=IYM2,IYP2
      KY = KY+1
      KX = 0
      DO 20 JX=IXM2,IXP2
      KX = KX+1
      IPD = IP(JX,JY)
      K(KX,KY) = IPD
      L(KX,KY) = IPD*IPD
   20 CONTINUE
C
CF *** Calculation of mean values and variances for nine masks
      A1 = AVE1(K22,K32,K42,K23,K33,K43,K24,K34,K44)
      A2 = AVE2(K21,K31,K41,K22,K32,K42,K33)
      A3 = AVE2(K41,K51,K32,K42,K52,K33,K43)
      A4 = AVE2(K42,K52,K33,K43,K53,K44,K54)
      A5 = AVE2(K33,K43,K34,K44,K54,K45,K55)
      A6 = AVE2(K33,K24,K34,K44,K25,K35,K45)
      A7 = AVE2(K23,K33,K14,K24,K34,K15,K25)
      A8 = AVE2(K12,K22,K13,K23,K33,K14,K24)
      A9 = AVE2(K11,K21,K12,K22,K32,K23,K33)
      V1 = VAR1(A1,L22,L32,L42,L23,L33,L43,L24,L34,L44)
```

```
                 V2 = VAR2(A2,L21,L31,L41,L22,L32,L42,L33)
                 V3 = VAR2(A3,L41,L51,L32,L42,L52,L33,L43)
                 V4 = VAR2(A4,L42,L52,L33,L43,L53,L44,L54)
                 V5 = VAR2(A5,L33,L43,L34,L44,L54,L45,L55)
                 V6 = VAR2(A6,L33,L24,L34,L44,L25,L35,L45)
                 V7 = VAR2(A7,L23,L33,L14,L24,L34,L15,L25)
                 V8 = VAR2(A8,L12,L22,L13,L23,L33,L14,L24)
                 V9 = VAR2(A9,L11,L21,L12,L22,L32,L23,L33)
CF *** SET THE DIRECTION FOR NINE MASKS
                 Q(2)=3.14
                 Q(3)=3.925
                 Q(4)=4.71
                 Q(5)=5.495
                 Q(6)=0.0
                 Q(7)=0.785
                 Q(8)=1.57
                 Q(9)=2.355
CF *** Mean gray value of the mask with minimum variance --> JP(IX,IY)
                 RMIN = V1
                 MI = 1
                 DO 40 I=2,9
                 IF (RMIN .LE. V(I)) GO TO 40
                 RMIN = V(I)
                 MI = I
          40 CONTINUE

                 IF(MI.EQ.6)THEN
                    THETO(IX,IY)=0.0
                    GO TO 45
                 ELSE IF(MI.EQ.7)THEN
                    THETO(IX,IY)=0.785
                    GO TO 45
                 ELSE IF(MI.EQ.8)THEN
                    THETO(IX,IY)=1.57
                    GO TO 45
                 ELSE IF(MI.EQ.9)THEN
                    THETO(IX,IY)=2.355
                    GO TO 45
                 ELSE IF(MI.EQ.2)THEN
                    THETO(IX,IY)=3.14
                    GO TO 45
                 ELSE IF(MI.EQ.3)THEN
                    THETO(IX,IY)=3.925
                    GO TO 45
                 ELSE IF(MI.EQ.4)THEN
                    THETO(IX,IY)=4.71
                    GO TO 45
                 ELSE IF(MI.EQ.5)THEN
                    THETO(IX,IY)=5.495
                    GO TO 45
                 ELSE IF(MI.EQ.1)THEN
                    GO TO 41
                 END IF

          41 CONTINUE
                 SMALL=V(2)
                 ASMALL=Q(2)
                 DO 42 I=3,9
```

```
          IF(SMALL.GT.V(I))THEN
             SMALL=V(I)

             ASMALL=Q(I)
          ENDIF
    42 CONTINUE
          THETO(IX,IY)=ASMALL
    45 CONTINUE
          JP(IX,IY) = A(MI)+0.5


    60 CONTINUE
C         .
CF ***** Operation for the four corners
C
C--------------------------------------------------------
C
          DO 80 IY=1,4
          DO 80 IX=1,4
          IPD = IP(IX,IY)
          K(IX,IY) = IPD
          L(IX,IY) = IPD*IPD
    80 CONTINUE
C
CF*****  JP(1,1)  *****
          JP(1,1) = AVE2(K11,K21,K12,K22,K32,K23,K33)+0.5
C
CF*****  JP(2,1)  *****
          A1 = AVE2(K21,K31,K22,K32,K42,K33,K43)
          A2 = AVE2(K21,K12,K22,K32,K13,K23,K33)
          V1 = VAR2(A1,L21,L31,L22,L32,L42,L33,L43)
          V2 = VAR2(A2,L21,L12,L22,L32,L13,L23,L33)
C
          RMIN = A1
          IF (V1 .GT. V2) RMIN = A2
          JP(2,1) = RMIN+0.5
C
CF*****  JP(1,2) *****
          A1 = AVE2(K21,K31,K12,K22,K32,K23,K33)
          A2 = AVE2(K12,K22,K13,K23,K33,K24,K34)
          V1 = VAR2(A1,L21,L31,L12,L22,L32,L23,L33)
          V2 = VAR2(A2,L12,L22,L13,L23,L33,L24,L34)
C
          RMIN = A1
          IF (V1 .GT. V2) RMIN = A2
          JP(1,2) = RMIN+0.5
C
CF*****  JP(2,2)  *****
          A1 = AVE1(K11,K21,K31,K12,K22,K32,K13,K23,K33)
          A2 = AVE2(K31,K41,K22,K32,K42,K33,K43)
          A3 = AVE2(K22,K32,K23,K33,K43,K34,K44)
          A4 = AVE2(K22,K13,K23,K33,K14,K24,K34)
          V1 = VAR1(A1,L11,L21,L31,L12,L22,L32,L13,L23,L33)
          V2 = VAR2(A2,L31,L41,L22,L32,L42,L33,L43)
          V3 = VAR2(A3,L22,L32,L23,L33,L43,L34,L44)
          V4 = VAR2(A4,L22,L13,L23,L33,L14,L24,L34)
C
          RMIN = V1
```

```
      MI = 1
      DO 100 I=2,4
      IF (RMIN .LE. V(I)) GO TO 100
      RMIN = V(I)
      MI = I
  100 CONTINUE
C
      JP(2,2) = A(MI)+0.5
C
C------------------------------------------------------------
C
      DO 120 IY=1,4
      KX = 0
      DO 120 IX=ISX3,ISX0
      KX = KX+1
      IPD = IP(IX,IY)
      K(KX,IY) = IPD
      L(KX,IY) = IPD*IPD
  120 CONTINUE
C
CF*****  JP(ISX-1,1)  *****
      A1 = AVE2(K31,K22,K32,K42,K23,K33,K43)
      A2 = AVE2(K21,K31,K12,K22,K32,K13,K23)
      V1 = VAR2(A1,L31,L22,L32,L42,L23,L33,L43)
      V2 = VAR2(A2,L21,L31,L12,L22,L32,L13,L23)
C
      RMIN = A1
      IF (V1 .GT. V2) RMIN = A2
      JP(ISX1,1) = RMIN+0.5
C
CF*****  JP(ISX,1)  *****
      JP(ISX0,1) = AVE2(K31,K41,K22,K32,K42,K23,K33)+0.5
C
CF*****  JP(ISX-1,2)  *****
      A1 = AVE1(K21,K31,K41,K22,K32,K42,K23,K33,K43)
      A2 = AVE2(K32,K23,K33,K43,K24,K34,K44)
      A3 = AVE2(K22,K32,K13,K23,K33,K14,K24)
      A4 = AVE2(K11,K21,K12,K22,K32,K13,K23)
      V1 = VAR1(A1,L21,L31,L41,L22,L32,L42,L23,L33,L43)
      V2 = VAR2(A2,L32,L23,L33,L43,L24,L34,L44)
      V3 = VAR2(A3,L22,L32,L13,L23,L33,L14,L24)
      V4 = VAR2(A4,L11,L21,L12,L22,L32,L13,L23)
C
      RMIN = V1
      MI = 1
      DO 140 I=2,4
      IF (RMIN .LE. V(I)) GO TO 140
      RMIN = V(I)
      MI = I
  140 CONTINUE
C
      JP(ISX1,2) = A(MI)+0.5
C
CF*****  JP(ISX,2)  *****
      A1 = AVE2(K32,K42,K23,K33,K43,K24,K34)
      A2 = AVE2(K21,K31,K22,K32,K42,K23,K33)
      V1 = VAR2(A1,L32,L42,L23,L33,L43,L24,L34)
      V2 = VAR2(A2,L21,L31,L22,L32,L42,L23,L33)
```

52

```
C
      RMIN = A1
      IF (V1 .GT. V2) RMIN = A2
      JP(ISX0,2) = RMIN+0.5
C
C--------------------------------------------------------------------
C
      KY = 0
      DO 160 IY=ISY3,ISY0
      KY = KY+1
      DO 160 IX=1,4
      IPD = IP(IX,IY)
      K(IX,KY) = IPD
      L(IX,KY) = IPD*IPD
  160 CONTINUE
C
CF*****  JP(1,ISY-1)  *****
      A1 = AVE2(K21,K31,K12,K22,K32,K13,K23)
      A2 = AVE2(K22,K32,K13,K23,K33,K24,K34)
      V1 = VAR2(A1,L21,L31,L12,L22,L32,L13,L23)
      V2 = VAR2(A2,L22,L32,L13,L23,L33,L24,L34)
C
      RMIN = A1
      IF (V1 .GT. V2) RMIN = A2
      JP(1,ISY1) = RMIN+0.5
C
CF*****  JP(2,ISY-1)  *****
      A1 = AVE1(K12,K22,K32,K13,K23,K33,K14,K24,K34)
      A2 = AVE2(K11,K21,K31,K12,K22,K32,K23)
      A3 = AVE2(K31,K41,K22,K32,K42,K23,K33)
      A4 = AVE2(K32,K42,K23,K33,K43,K34,K44)
      V1 = VAR1(A1,L12,L22,L32,L13,L23,L33,L14,L24,L34).
      V2 = VAR2(A2,L11,L21,L31,L12,L22,L32,L23)
      V3 = VAR2(A3,L31,L41,L22,L32,L42,L23,L33)
      V4 = VAR2(A4,L32,L42,L23,L33,L43,L34,L44)
C
      RMIN = V1
      MI = 1
      DO 180 I=2,4
      IF (RMIN .LE. V(I)) GO TO 180
      RMIN = V(I)
      MI = I
  180 CONTINUE
C
      JP(2,ISY1) = A(MI)+0.5
C
CF*****  JP(1,ISY)  *****
      JP(1,ISY) = AVE2(K22,K32,K13,K23,K33,K14,K24)+0.5
C
CF*****  JP(2,ISY)  *****
      A1 = AVE2(K12,K22,K32,K13,K23,K33,K24)
      A2 = AVE2(K32,K42,K23,K33,K43,K24,K34)
      V1 = VAR2(A1,L12,L22,L32,L13,L23,L33,L24)
      V2 = VAR2(A2,L32,L42,L23,L33,L43,L24,L34)
C
      RMIN = A1
      IF (V1 .GT. V2) RMIN = A2
      JP(2,ISY0) = RMIN+0.5
```

```
      KY - 0
      DO 200 IY-ISY3,ISY0
      KY - KY+1
      KX - 0
      DO 200 IX-ISX3,ISX0
      KX - KX+1
      IPD - IP(IX,IY)
      K(KX,KY) - IPD
      L(KX,KY) - IPD*IPD
  200 CONTINUE
C
CF***** JP(ISX-1,ISY-1) *****
      A1 - AVE1(K22,K32,K42,K23,K33,K43,K24,K34,K44)
      A2 - AVE2(K21,K31,K41,K22,K32,K42,K33)
      A3 - AVE2(K12,K22,K13,K23,K33,K14,K24)
      A4 - AVE2(K11,K21,K12,K22,K32,K23,K33)
      V1 - VAR1(A1,L22,L32,L42,L23,L33,L43,L24,L34,L44)
      V2 - VAR2(A2,L21,L31,L41,L22,L32,L42,L33)
      V3 - VAR2(A3,L12,L22,L13,L23,L33,L14,L24)
      V4 - VAR2(A4,L11,L21,L12,L22,L32,L23,L33)
C
      RMIN - V1
      MI - I
      DO 220 I-2,4
      IF (RMIN .LE. V(I)) GO TO 220
      RMIN - V(I)
      MI - I
  220 CONTINUE
C
      JP(ISX1,ISY1) - A(MI)+0.5
C
CF***** JP(ISX,ISY-1) *****
      A1 - AVE2(K22,K32,K23,K33,K43,K24,K34)
      A2 - AVE2(K21,K31,K22,K32,K42,K33,K43)
      V1 - VAR2(A1,L22,L32,L23,L33,L43,L24,L34)
      V2 - VAR2(A2,L21,L31,L22,L32,L42,L33,L43)
C
      RMIN - A1
      IF (V1 .GT. V2) RMIN - A2
      JP(ISX0,ISY1) - RMIN+0.5
C
CF***** JP(ISX-1,ISY) *****
      A1 - AVE2(K22,K32,K42,K23,K33,K43,K34)
      A2 - AVE2(K12,K22,K13,K23,K33,K24,K34)
      V1 - VAR2(A1,L22,L32,L42,L23,L33,L43,L34)
      V2 - VAR2(A2,L12,L22,L13,L23,L33,L24,L34)
C
      RMIN - A1
      IF (V1 .GT. V2) RMIN - A2
      JP(ISX1,ISY0) - RMIN+0.5
C
CF***** JP(ISX,ISY) *****
      JP(ISX0,ISY0) - AVE2(K22,K32,K23,K33,K43,K34,K44)+0.5
C
CF ***** Marginal operation
C
C------------------------------------------------------------------
C
```

```
          DO 360 IX=3,ISX2
          IXM2 = IX-2
          IXP2 = IX+2
C
C-------------------------------------------------------------
C
          DO 240 JY=1,4
          KX = 0
          DO 240 JX=IXM2,IXP2
          KX = KX+1
          IPD = IP(JX,JY)
          K(KX,JY) = IPD
          L(KX,JY) = IPD*IPD
    240 CONTINUE
C
CF*****  JP(IX,1)  *****
          A1 = AVE2(K31,K41,K32,K42,K52,K43,K53)
          A2 = AVE2(K31,K22,K32,K42,K23,K33,K43)
          A3 = AVE2(K21,K31,K12,K22,K32,K13,K23)
          V1 = VAR2(A1,L31,L41,L32,L42,L52,L43,L53)
          V2 = VAR2(A2,L31,L22,L32,L42,L23,L33,L43)
          V3 = VAR2(A3,L21,L31,L12,L22,L32,L13,L23)
C
          RMIN = V1
          MI = 1
          DO 260 I=2,3
          IF (RMIN .LE. V(I)) GO TO 260
          RMIN = V(I)
          MI = I
    260 CONTINUE
C
          JP(IX,1) = A(MI)+0.5
C
CF*****  JP(IX,2)  *****
          A1 = AVE1(K21,K31,K41,K22,K32,K42,K23,K33,K43)
          A2 = AVE2(K41,K51,K32,K42,K52,K43,K53)
          A3 = AVE2(K32,K42,K33,K43,K53,K44,K54)
          A4 = AVE2(K32,K23,K33,K43,K24,K34,K44)
          A5 = AVE2(K22,K32,K13,K23,K33,K14,K24)
          A6 = AVE2(K11,K21,K12,K22,K32,K13,K23)
          V1 = VAR1(A1,L21,L31,L41,L22,L32,L42,L23,L33,L43)
          V2 = VAR2(A2,L41,L51,L32,L42,L52,L43,L53)
          V3 = VAR2(A3,L32,L42,L33,L43,L53,L44,L54)
          V4 = VAR2(A4,L32,L23,L33,L43,L24,L34,L44)
          V5 = VAR2(A5,L22,L32,L13,L23,L33,L14,L24)
          V6 = VAR2(A6,L11,L21,L12,L22,L32,L13,L23)
C
          RMIN = V1
          MI = 1
          DO 280 I=2,6
          IF (RMIN .LE. V(I)) GO TO 280
          RMIN = V(I)
          MI = I
    280 CONTINUE
C
          JP(IX,2) = A(MI)+0.5
C
C-------------------------------------------------------------
```

```
C
      KY - 0
      DO 300 JY=ISY3,ISY0
      KY - KY+1
      KX - 0
      DO 300 JX=IXM2,IXP2
      KX - KX+1
      IPD - IP(JX,JY)
      K(KX,KY) - IPD
      L(KX,KY) - IPD*IPD
  300 CONTINUE
C
CF*****  JP(IX,ISY-1)  *****
      A1 - AVE1(K22,K32,K42,K23,K33,K43,K24,K34,K44)
      A2 - AVE2(K42,K52,K33,K43,K53,K44,K54)
      A3 - AVE2(K41,K51,K32,K42,K52,K33,K43)
      A4 - AVE2(K21,K31,K41,K22,K32,K42,K33)
      A5 - AVE2(K11,K21,K12,K22,K32,K23,K33)
      A6 - AVE2(K12,K22,K13,K23,K33,K14,K24)
      V1 - VAR1(A1,L22,L32,L42,L23,L33,L43,L24,L34,L44)
      V2 - VAR2(A2,L42,L52,L33,L43,L53,L44,L54)
      V3 - VAR2(A3,L41,L51,L32,L42,L52,L33,L43)
      V4 - VAR2(A4,L21,L31,L41,L22,L32,L42,L33)
      V5 - VAR2(A5,L11,L21,L12,L22,L32,L23,L33)
      V6 - VAR2(A6,L12,L22,L13,L23,L33,L14,L24)
C
      RMIN - V1
      MI - 1
      DO 320 I=2,6
      IF (RMIN .LE. V(I)) GO TO 320
      RMIN - V(I)
      MI - I
  320 CONTINUE
C
      JP(IX,ISY1) - A(MI)+0.5
C
CF*****  JP(IX,ISY)  *****
      A1 - AVE2(K42,K52,K33,K43,K53,K34,K44)
      A2 - AVE2(K22,K32,K42,K23,K33,K43,K34)
      A3 - AVE2(K12,K22,K13,K23,K33,K24,K34)
      V1 - VAR2(A1,L42,L52,L33,L43,L53,L34,L44)
      V2 - VAR2(A2,L22,L32,L42,L23,L33,L43,L34)
      V3 - VAR2(A3,L12,L22,L13,L23,L33,L24,L34)
C
      RMIN - V1
      MI - 1
      DO 340 I=2,3
      IF (RMIN .LE. V(I)) GO TO 340
      RMIN - V(I)
      MI - I
  340 CONTINUE
C
      JP(IX,ISY0) - A(MI)+0.5
C
  360 CONTINUE
C
C-----------------------------------------------------------------
C
```

```fortran
      DO 500 IY-3,ISY2
      IYM2 - IY-2
      IYP2 - IY+2
C
C-----------------------------------------------------
C
      KY - 0
      DO 380 JY-IYM2,IYP2
      KY - KY+1
      DO 380 JX-1,4
      IPD - IP(JX,JY)
      K(JX,KY) - IPD
      L(JX,KY) - IPD*IPD
  380 CONTINUE
C
CF*****  JP(1,IY)  *****
      A1 - AVE2(K21,K31,K12,K22,K32,K13,K23)
      A2 - AVE2(K22,K32,K13,K23,K33,K24,K34)
      A3 - AVE2(K13,K23,K14,K24,K34,K25,K35)
      V1 - VAR2(A1,L21,L31,L12,L22,L32,L13,L23)
      V2 - VAR2(A2,L22,L32,L13,L23,L33,L24,L34)
      V3 - VAR2(A3,L13,L23,L14,L24,L34,L25,L35)
C
      RMIN - V1
      MI - 1
      DO 400 I-2,3
      IF (RMIN .LE. V(I)) GO TO 400
      RMIN - V(I)
      MI - I
  400 CONTINUE
C
      JP(1,IY) - A(MI)+0.5
C
CF*****  JP(2,IY)  *****
      A1 - AVE1(K12,K22,K32,K13,K23,K33,K14,K24,K34)
      A2 - AVE2(K11,K21,K31,K12,K22,K32,K23)
      A3 - AVE2(K31,K41,K22,K32,K42,K23,K33)
      A4 - AVE2(K32,K42,K23,K33,K43,K34,K44)
      A5 - AVE2(K23,K33,K24,K34,K44,K35,K45)
      A6 - AVE2(K23,K14,K24,K34,K15,K25,K35)
      V1 - VAR1(A1,L12,L22,L32,L13,L23,L33,L14,L24,L34)
      V2 - VAR2(A2,L11,L21,L31,L12,L22,L32,L23)
      V3 - VAR2(A3,L31,L41,L22,L32,L42,L23,L33)
      V4 - VAR2(A4,L32,L42,L23,L33,L43,L34,L44)
      V5 - VAR2(A5,L23,L33,L24,L34,L44,L35,L45)
      V6 - VAR2(A6,L23,L14,L24,L34,L15,L25,L35)
C
      RMIN - V1
      MI - 1
      DO 420 I-2,6
      IF (RMIN .LE. V(I)) GO TO 420
      RMIN - V(I)
      MI - I
  420 CONTINUE
C
      JP(2,IY) - A(MI)+0.5
C
C-----------------------------------------------------
C
```

57

```
C
      KY = 0
      DO 440 JY=IYM2,IYP2
      KY = KY+1
      KX = 0
      DO 440 JX=ISX3,ISX0
      KX = KX+1
      IPD = IP(JX,JY)
      K(KX,KY) = IPD
      L(KX,KY) = IPD*IPD
  440 CONTINUE
C
CF*****  JP(ISX-1,IY)  *****
      A1 = AVE1(K22,K32,K42,K23,K33,K43,K24,K34,K44)
      A2 = AVE2(K21,K31,K41,K22,K32,K42,K33)
      A3 = AVE2(K11,K21,K12,K22,K32,K23,K33)
      A4 = AVE2(K12,K22,K13,K23,K33,K14,K24)
      A5 = AVE2(K23,K33,K14,K24,K34,K15,K25)
      A6 = AVE2(K33,K24,K34,K44,K25,K35,K45)
      V1 = VAR1(A1,L22,L32,L42,L23,L33,L43,L24,L34,L44)
      V2 = VAR2(A2,L21,L31,L41,L22,L32,L42,L33)
      V3 = VAR2(A3,L11,L21,L12,L22,L32,L23,L33)
      V4 = VAR2(A4,L12,L22,L13,L23,L33,L14,L24)
      V5 = VAR2(A5,L23,L33,L14,L24,L34,L15,L25)
      V6 = VAR2(A6,L33,L24,L34,L44,L25,L35,L45)
C
      RMIN = V1
      MI = 1
      DO 460 I=2,6
      IF (RMIN .LE. V(I)) GO TO 460
      RMIN = V(I)
      MI = I
  460 CONTINUE
C
      JP(ISX1,IY) = A(MI)+0.5
C
CF*****  JP(ISX,IY)  *****
      A1 = AVE2(K21,K31,K22,K32,K42,K33,K43)
      A2 = AVE2(K22,K32,K23,K33,K43,K24,K34)
      A3 = AVE2(K33,K43,K24,K34,K44,K25,K35)
      V1 = VAR2(A1,L21,L31,L22,L32,L42,L33,L43)
      V2 = VAR2(A2,L22,L32,L23,L33,L43,L24,L34)
      V3 = VAR2(A3,L33,L43,L24,L34,L44,L25,L35)
C
      RMIN = V1
      MI = 1
      DO 480 I=2,3
      IF (RMIN .LE. V(I)) GO TO 480
      RMIN = V(I)
      MI = I
  480 CONTINUE
C
      JP(ISX0,IY) = A(MI)+0.5
C
  500 CONTINUE
      DO 600 I=1,256
      DO 600 J=1,256
        IP(I,J)=JP(I,J)
  600 CONTINUE
```

```
           DO 601 IX=1,2
           DO 601 IY=1,256
           THET0(IX,IY)=0.0
   601 CONTINUE
           DO 602 IX=1,256
           DO 602 IY=1,2
           THET0(IX,IY)=0.0
   602 CONTINUE
           DO 603 IX=255,256
           DO 603 IY=1,256
           THET0(IX,IY)=0.0
   603 CONTINUE
           DO 604 IX=1,256
           DO 604 IY=255,256
           THET0(IX,IY)=0.0
   604 CONTINUE
           DO 605 IX=1,256
           DO 605 IY=1,256
           THET1(IX,IY)=THET0(IX,IY)
   605 CONTINUE
           DO 606 IX=1,256
           DO 606 IY=1,256
           IF(THET1(IX,IY).GE.3.14)THEN
               THET1(IX,IY)=THET1(IX,IY)-3.14
           ELSE
               THET1(IX,IY)=THET1(IX,IY)
           ENDIF
   606 CONTINUE
C
CF ***** RETURN
       RETURN
       END
       SUBROUTINE RXEP(IP,THET1,PRBE1,ISX,ISY,TEMP)
C
C      Copyright (c) 1983 by AIST MITI(JAPAN)
C
CP     Computes initial parameter and initial probability for edge
CP     reinforcement by relaxation method (Lower-level routine
CP     for subroutine RXEG).
C
CS     CALL RXEP(IP,THET1,PRBE1,ISX,ISY)
C
CK     RELAXATION, EDGE
C
CA     IP(ISX,ISY)   :      Input image
CA     THET1(ISX,ISY):      Initial value of gradient direction for
CA                          each pixel
CA     PRBE1(ISX,ISY):      Initial value of edge probability for
CA                          each pixel
C
CN         Reference
CN @1!  B. J. Schachter, A. Lev, S. W. Zucker, and A. Rosenfeld, "An
CN     application of relaxation methods to edge reinforcement,"
CN     IEEE Trans. Syst., Man, Cybern., vol. SMC-7, pp. 813-816,
CN     Nov. 1977.
C
CD         JULY 1979.  PROGRAMMED BY  K.SAKAUE
C
CM     A small real number -1E30 is set in variable AMX as an initial
```

```
CM      value. When using a computer with different word lengths, care
CM      should be taken.
C
        DIMENSION IP(ISX,ISY),TEMP(ISX,ISY)
        DIMENSION THET1(ISX,ISY)
        DIMENSION PRBE1(ISX,ISY)
        DIMENSION NE(3,3)
C
CF      INITIAL EDGE VALUES
CF      BY PREWITT OPERATOR (DIFFERENTIAL TYPE)
C
        AMX=-1E30
        DO 10 IY=1,ISY
        DO 10 IX=1,ISX
          DO 11 J=1,3
          DO 11 I=1,3
             J1=IY+J-2
             I1=IX+I-2
             IF(J1.LT.1.OR.J1.GT.ISY)GO TO 12
             IF(I1.LT.1.OR.I1.GT.ISX)GO TO 12
             NE(I,J)=IP(I1,J1)
   11     CONTINUE
          DX=FLOAT(NE(3,1)+NE(3,2)+NE(3,3)-NE(1,1)-NE(1,2)-NE(1,3))
          DY=FLOAT(NE(1,3)+NE(2,3)+NE(3,3)-NE(1,1)-NE(2,1)-NE(3,1))
          AMG=SQRT(DX**2+DY**2)
          IF(AMG.GT.AMX)AMX=AMG
          PRBE1(IX,IY)=AMG
          IF(DX.LE.0E0.AND.DY.LE.0E0)DX=0.0001

          GO TO 10
   12     CONTINUE
          PRBE1(IX,IY)=0.001

   10 CONTINUE
        DO 20 IY=1,ISY
        DO 20 IX=1,ISX
          PRBE1(IX,IY)=PRBE1(IX,IY)/AMX
          PRBE1(IX,IY)=AMAX1(.01,AMIN1(.9,PRBE1(IX,IY)))
          TEMP(IX,IY)=PRBE1(IX,IY)
   20 CONTINUE
        RETURN
        END
        SUBROUTINE RXEI(THET1,THET2,PRBE1,PRBE2,ISX,ISY,ISNX,ISNY,C,W)
C
C       Copyright (c) 1983 by AIST MITI(JAPAN)
C
CS      CALL RXEI(THET1,THET2,PRBE1,PRBE2,ISX,ISY,ISNX,ISNY,C,W)
C
CP      Updates edge parameter and edge probability for edge
CP      reinforcement by relaxation method.
CP      (Lower-level routine for subroutine RXEG)
C
CK      RELAXATION, EDGE
C
CA      THET1(ISX,ISY):    Gradient direction (before updating)
CA      THET2(ISX,ISY):    Gradient direction (after updating)
CA      PRBE1(ISX,ISY):    Edge probabilities (before updating)
CA      PRBE2(ISX,ISY):    Edge probabilities (after updating)
```

```
CA      ISNX,ISNY        :    Size of neighborhood being considered
CA                            (limited to odd number three or higher)
CA      C(4)             :    Parameter C1 through C4 (See the
CA                            reference.)
CA                            .866,.124,.005,.005 (standard)
CA                            .706,.176,.059,.059 (noise cleaning)
CA      W                :    Parameter W (See the reference.)
CA                            3.0 for standard use.
C
        DIMENSION THET1(ISX,ISY)
        DIMENSION THET2(ISX,ISY)
        DIMENSION PRBE1(ISX,ISY)
        DIMENSION PRBE2(ISX,ISY)
        DIMENSION C(4)
C
        DATA  HPAI/1.570796327/
C
C
        ISFTX=ISNX/2+1
        ISFTY=ISNY/2+1
        DO 10 IY=1,ISY
        DO 10 IX=1,ISX
          ALPHA=THET1(IX,IY)
          PXY=PRBE1(IX,IY)
          Q1=0.0
          Q2=0.0
          DHX=W*PXY*COS(ALPHA)
          DHY=W*PXY*SIN(ALPHA)
          DO 11 J=1,ISNY
          DO 11 I=1,ISNX
            IU=IX+I-ISFTX
            IV=IY+J-ISFTY
            IF(IV.EQ.IY.AND.IU.EQ.IX)GO TO 11
            IF(IV.LT.1.OR.IV.GT.ISY)GO TO 11
            IF(IU.LT.1.OR.IU.GT.ISX)GO TO 11
            IDY=IV-IY
            IDX=IU-IX
            JD=IABS(IDY)
            ID=IABS(IDX)
            LD=MAX0(ID,JD)
C
CF          LD : CHESSBOARD DISTANCE FROM (IX,IY) TO (IU,IV)
C
            ALD=FLOAT(LD)
            D2=1.0/2.0**ALD
            BETA=THET1(IU,IV)
            DY=FLOAT(IDY)
            DX=FLOAT(IDX)
            IF(IDX.EQ.0.AND.IDY.EQ.0)DX=.00001
            GAMMA=ATAN2(DY,DX)
            PUV=PRBE1(IU,IV)
C
CF          COMPATIBILITY COEFFICIENTS
CF          EDGE/EDGE INTERACTION
C
            IF(ALPHA.EQ.BETA.AND.BETA.EQ.GAMMA)THEN
              REE=1.0

            ELSE IF(ALPHA.EQ.BETA.AND.GAMMA.NE.BETA)THEN
```

```fortran
              REE=0.5
            ELSE
              REE=COS(ALPHA-GAMMA)*COS(BETA-GAMMA)*D2
            ENDIF
C
CF          INTERACTION WITH NONEDGES
C
            AG2=2.0*(ALPHA-GAMMA)
            AG2=-COS(AG2)*D2
            REN=AMIN1(0.0,AG2)
            BG2=2.0*(BETA-GAMMA)
            RNE=(1.0-COS(BG2))*D2*0.5
            RNN=D2
C
CF          COMBINED REINFORCEMENT PROCESS
C
            Q1=Q1+C(1)*PUV*REE+C(2)*(1.0-PUV)*REN
            Q2=Q2+C(3)*PUV*RNE+C(4)*(1.0-PUV)*RNN
            PREE=PUV*REE
            DHX=DHX+PREE*COS(BETA)
            DHY=DHY+PREE*SIN(BETA)
      11    CONTINUE
            QQ=ABS(Q1)+ABS(Q2)
            Q1=Q1/QQ
            Q2=Q2/QQ
            P1D=PXY*(1.0+Q1)
            P2D=(1.0-PXY)*(1.0+Q2)
            PNEW=P1D/(P1D+P2D)
            IF(DHX.LE.0E0.AND.DHY.LE.0E0)DHY=0.00001
            THNEW=ATAN2(DHY,DHX)
            PRBE2(IX,IY)=PNEW
            THET2(IX,IY)=THNEW
      10 CONTINUE
         RETURN
         END
```

LIST OF REFERENCES

1. Robert. J. Urick., *Principles of Underwater Sound* , pp. 343, Mcgraw Hill, Inc., 1983

2. William, C. Knight, "Digital Signal Processing for Sonar," IEEE Trans pp. 1489, 1980.

3. L.G.Roberts, "Machine perception of three dimensional solids," in *Optical Electro-optical processing of information*, Academic Press, pp. 75-149, 1970.

4. "Edge and line detection," in SPIDER users manual, Joint System Design Corp, Japan.

5. L. G. Roberts, "Machine perception of three dimensional solids," in *Optical Electro-optical processing of information*, MIT press, pp. 159-197, 1965.

6. J. M. S. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, Academic press, pp. 75-149, 1970.

7. R. O. Duba, P. E. Hart, Pattern classification and Scene analysis (Wiley, New York 1973).

8. D. Waltz, in *Applied Computation Theory*, ed. by T. Yeh (Prentice Hall, New York 1976) pp. 468-529.

9. S. W. Zucker, Proc. 3rd Intern. Joint Conf. Pattern Recognition (Coronado, Calif. Nov. 8-11, 1976) pp. 852-861.

10. D. H. Ballard, C. M. Brown, *Computer vision*, pp. 408-420 (Prentice-Hall, Inc. 1982).

11. B.J.Schachter, A.Lev, S.W.Zucker, and A.Rosenfeld, "An application of relaxation methods to edge reinforcement,"*IEEE Trans., Sys., man, Cybern.,* vol. SMC-7, pp 813-816.

12. S. W. Zucker, R. A. Hummel, and A. Rosenfeld, "An application of relaxation labeling to line and curve enhancement," IEEE Trans. Compt., vol. C-26, pp. 394-493, April 1977.

13. A.Rosenfeld, R. B. Thomas, and Y. H. Lee, "Edge and curve detection for texture discrimination," in *Picture Processing and Psychopictorics,* B. S. Lapkin and A. Rosenfeld, Eds., Academic Press, New York, 1970.

14. S. Peleg and A. Rosenfeld, "Determining compatibility coefficients for curve enhancement relaxation process." IEEE Trans. Syst., Man, Cybern., vol. SMC-8, pp. 548-554, July 1978.

INITIAL DISTRIBUTION LIST