

A SONAR IMAGING SYSTEM

by

MICHAEL K. A. SMAYRA

M.Sc., The University of British Columbia, 1992

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE**

in

THE FACULTY OF GRADUATE STUDIES

Department of Electrical Engineering

Vancouver, British Columbia

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

April 1992

Copyright © Michael Smayra, 1992

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of ELECTRICAL ENGINEERING

The University of British Columbia
Vancouver, Canada

Date 30 APRIL 1992

Abstract

In this thesis, a sonar mine detection system is presented. The three stages of the system, image enhancement, edge detection and data compression are analyzed. For each stage, different approaches are studied and the appropriate algorithm is chosen.

For image enhancement, a cascaded morphological enhancement algorithm is developed. Our algorithm is shown to preserve high-detail regions, including edges surrounding features of concern (suspect mines) and their corresponding shadows. This is accomplished without introducing any blurring, which is an inherent side-effect of conventional enhancement algorithms.

For edge detection, amongst various methods, we have found the two dimensional morphological edge detector, the Alpha-Trimmed Morphological (ATM) filter, to be immune to noise, a major problem encountered in sonar imagery, and consequently this filter has outperformed the other conventional edge detectors.

At the compression stage, three algorithms are examined. The Vector Quantization (VQ) method, using a new codebook generation algorithm, which we have developed, is found to yield very high quality reconstructed images. The decompressed images have their high-detail regions well preserved, and furthermore the low-detail regions are minimally distorted, to the extent that the reconstructed images are virtually indistinguishable from the original images. With the new compression algorithm, a compression ratio of 1:10, or equivalently 0.8 bits per pixel

(bpp) is achieved. Furthermore, the processing time for codebook generation is greatly shortened.

Contents

Abstract	ii
List of Figures	vii
List of Tables	xi
Acknowledgment	xii
1 Introduction	1
2 Side-Scan Sonar System	5
3 Image Enhancement	11
3.1 Conventional Smoothing Algorithms	15
3.1.1 Spatial Averaging	15
3.1.2 Order Statistics Filtering	17
3.1.3 Out-Range Noise Cleaning	20
3.2 Cascaded Morphological Smoothing Filters	21
3.2.1 Principles of Mathematical Morphology	21
3.2.2 The Enhancement Filter	27
3.3 Sonar Image Enhancement Results	30
4 Edge Detection	33
4.1 Introduction	33
4.1.1 Zero-Crossing of the Laplacian of Gaussian	34
4.1.2 Step Function Edge Detector	36
4.1.3 Histogram Thresholding	37

4.1.4	Multidimensional Morphological Edge Detection	38
4.1.5	Other Edge Detectors	39
4.2	Multidimensional Morphological Edge Detector	39
4.3	Results of Edge Detection	45
4.3.1	Edge Detection Success Measurement	45
4.3.2	Results Using Different Pre-Processing Enhancement filters .	45
4.3.3	Results	47
5	Image Compression	51
5.1	History and Definitions	51
5.1.1	Predictive Coding	52
5.1.2	Transform Coding	52
5.1.3	Block-Based Coding	53
5.2	Block Truncation Coding	54
5.3	JPEG Algorithm	55
5.3.1	JPEG Encoding	56
5.3.2	JPEG Decoding	58
5.3.3	Experimental Results	59
5.4	Vector Quantization	63
5.4.1	Introduction to Vector Quantization	63
5.5	Summary of Compression Results	70

6	Conclusions	81
6.1	Original Contribution	81
6.2	Suggestions for Future Work	82
A	REFERENCES	83

List of Figures

Figure 1	Sidescan sonar apparatus.	6
Figure 2	Emitted sonar beam geometry.	7
Figure 3	Original sonar image	11
Figure 4	A three-dimensional view of fig. 3.	13
Figure 5	Fourier transform of fig. 3.	14
Figure 6	Fourier transform of a noise-free image.	14
Figure 7	Effect of averaging on an edge.	16
Figure 8	Output image of a mean filter on the sonar image of Fig. 3.	17
Figure 9	Output of an order statistics filter using a 3x3 window, and the 10th percentile.	18
Figure 10	Output of an order statistics filter using a 3x3 window, and the 50th percentile (Median filter).	19
Figure 11	Output of an order statistics filter using a 5x5 window, and the 50th percentile (Median filter).	19
Figure 12	Out —range noise cleaning, using a high threshold value.	20
Figure 13	Dilation of object A by structuring element B (from [6]). .	23
Figure 14	Erosion of object A by structuring element B (from [6]). .	24
Figure 15	Morphological opening (from [6]).	26

Figure 16	Morphological closing (from [6]).	27
Figure 17	Block diagram of the cascaded morphological enhancement filter.	28
Figure 18	Output of cascaded filter with a horizontal bar as structuring element.	29
Figure 19	Output of cascaded filter with a vertical bar as structuring element	29
Figure 20	Output of cascaded filter using a square window as a structuring element	30
Figure 21	Output of cascaded filter using a circular window as a structuring element.	30
Figure 22	Output of a zero-crossing edge detector using a narrow standard deviation range.	35
Figure 23	Output of a zero-crossing edge detector using a wide standard deviation range.	35
Figure 24	Output of a step function edge detector searching for intensity steps larger than 20.	36
Figure 25	Histogram thresholding	38
Figure 26	Sensitivity of histogram thresholding: threshold changed by 0.1% of that of FIG. 25.	38
Figure 27	Block diagram of the "primitive" morphological edge detector.	40

Figure 28	Block diagram of the ATM edge detector:	42
Figure 29	ATM Edge Detection, with $\alpha = 1$ and preceded by a morphological enhancement step.	43
Figure 30	ATM Edge Detection without the enhancement step. . .	44
Figure 31	ATM Edge Detection after spatial averaging.	46
Figure 32	ATM Edge Detection after out-range noise removal. . .	46
Figure 33	ATM Edge Detection after median filtering.	47
Figure 34	Parallel array shift architecture (from [9]).	48
Figure 35	Reconstructed sonar image after BTC encoding and decoding	55
Figure 36	Block diagram of the JPEG encoder	56
Figure 37	Zig-Zag scanning used to get long sequences of zeros .	57
Figure 38	Block diagram of the JPEG decoder	58
Figure 39	The zoomed original image (Zooming factor = 2)	73
Figure 40	The zoomed reconstructed image after JPEG compression, with 5:1 compression ratio.	74
Figure 41	The zoomed reconstructed image after VQ compression, with 10:1 compression ratio.	75
Figure 42	The zoomed reconstructed image after JPEG compression with quality factor of 30 (10:1 compression)	76

Figure 43	The zoomed reconstructed image after VQ compression (10:1 compression)	77
Figure 44	Difference image after JPEG compression	78
Figure 45	Difference image after VQ compression	79

List of Tables

Table 1	Tabulated results of edge detection.	49
Table 2	Visual “measurement” using a quality factor of 75, for images 1 and 2.	60
Table 3	Visual “measurement” using a quality factor of 75, for images 3 and 4.	61
Table 4	Quantitative comparison of JPEG compression algorithm.	62
Table 5	Results of VQ method for images 1 and 2.	67
Table 6	Results of VQ method for images 3 and 4.	68
Table 7	Quantitative comparison of modified VQ compression algorithm.	69

Acknowledgment

I wish to express my sincere gratitude to Professor R. Ward for her advice, guidance, and for many useful comments and suggestions throughout the duration of this thesis.

I wish to dedicate this work to my parents Albert and Libuse and to my sister Mona who have been providing me with so much love and inspiration.

1 Introduction

In recent years, many countries have expressed interest in developing sonar systems that would make their harbors and coastal waters safer, in case of emergencies such as wars. Sharing this concern, Canada established a multi-billion dollar Mine Sweeping Counter Measure project (MSCM)¹ for the development of such a system. This system will require the development of many complicated sub-systems, such as sonar sensors and remotely controlled tow-ships. This thesis is concerned with a small but vital part of that wider project: image analysis and storage. Typically, an operator will be monitoring the bottom of the ocean by analyzing sonar images that will be displayed in front of him or her on a screen. Since this will be done on a real-time basis, the operator will see the sea bed scrolling on the screen, while the operating ship will be scanning the area. Once these images are displayed, it is required that they are then compressed and stored on tapes for future use.

Specifically, this thesis examines the three major components of such a system, namely:

1. Image Enhancement.
2. Edge Detection.
3. Image compression and decompression.

¹ The Canadian government refers to this program as the Maritime Coastal Defense Vessel program.

The organization of this thesis is as follows. Chapter 2 provides an overview of the MSCM system with reference to its components and the way in which they interact. This chapter does not provide any new results or analysis, but introduces readers to the subject and establishes a logical framework within which subsequent chapters are placed.

One unfortunate characteristic of sonar images is their noisiness, and one of the challenges to overcome in the MSCM system is to identify and locate features (suspect mines) in the presence of a very noisy background. For this reason, before applying intelligent analysis tools such as edge detectors, a preliminary step is required. The purpose of this step is to enhance the image by suppressing noise and sharpening the edges that delimit features paving the way for easy edge detection. Chapter 3 is devoted to this pre-processing step, to examining traditional enhancement algorithms. Many of these traditional algorithms use some sort of averaging in order to suppress noise. Unfortunately, this has the side-effect of blurring since high spatial frequencies are lost or weakened. Although the averaging operation is used in a wide spectrum of applications, it is found to be inappropriate for sonar images since it causes a blurring that removes small features, an operation that is unacceptable for military standards (no suspect mine can be overlooked).

Morphological enhancement filters have been used in “very noisy images” applications, and they lead to promising results [15, 41, 43, 59]. We develop the Cascaded Morphological filter which is shown to preserve the required features

and at the same time to suppress the noise. Since the purpose of such enhancement filters is to aid in the edge detection process, the “best” enhancement filter is chosen after applying edge detectors to the enhanced images.

The next step in the display process is to detect edges by overlaying the edges on top of the original image. This edge detection operation would make the detection task easier for the operator, since the features will be highlighted. In Chapter 4, we apply different edge detectors to the enhanced images and then we compare the results. It is shown that the ATM edge detector outperforms all the other industry-used detectors. This filter is not recent, and in fact its roots go back to 1960 when Mitsubishi used it to locate defects in integrated circuits boards. Since then, a few research papers which investigate the usefulness of this detector on a wide range of imagery have appeared in the literature. For sonar imagery, the good performance of the ATM detector is mainly attributed to its immunity to noise. Because most edge detectors are based on finding intensity steps or gradients, it is evident that these algorithms will fail in the sonar images applications. This chapter ends with a tabulated results of the different investigated algorithms.

Another important step in this system is to compress the sonar images and archive them on tape. A high compression ratio is desirable since this would cut down the cost of tape required to store data that represent sea beds near harbors and coastal waters, which amounts to very vast areas. In Chapter 5 we compare three compression methods, Block Truncation Coding (BTC), Joint

Photographic Experts Group (JPEG) algorithm, and Vector Quantization (VQ). Excellent feature preservation and very low bit rates are achieved by using an improved VQ algorithm, with a new codebook generation method which we developed. In addition to the high image quality obtained after reconstruction, the processing time of the codebook generation method is significantly shortened, making the VQ our recommended compression/decompression algorithm.

Finally, we reserve Chapter 6 for concluding remarks, the original contributions of this thesis, and some ideas for future research.

2 Side-Scan Sonar System

A brief overview of the MSCM is provided here to introduce the system as a whole, and to frame the different chapters of this thesis.

The Canadian marine forces consider certain areas prone to mine attacks. Harbors must be secure at all times, especially in an emergency situation like a war, since they provide a vital link to the movement of warships and submarines, as well as to commercial ships providing food supplies and other essential commodities. Other areas considered strategically important are the shallow coastal waters where mines could potentially sabotage the safe passage of ships.

The mine tracking and counter measure operation would function as follows: a ship would tow a towfish, which would have two transducers, one on the port side and one on the starboard (left and right sides of a ship). Each transducer would emit pulses of sonar energy which would travel through the water until they are reflected back to the towfish by objects in the water and by the sea bed.

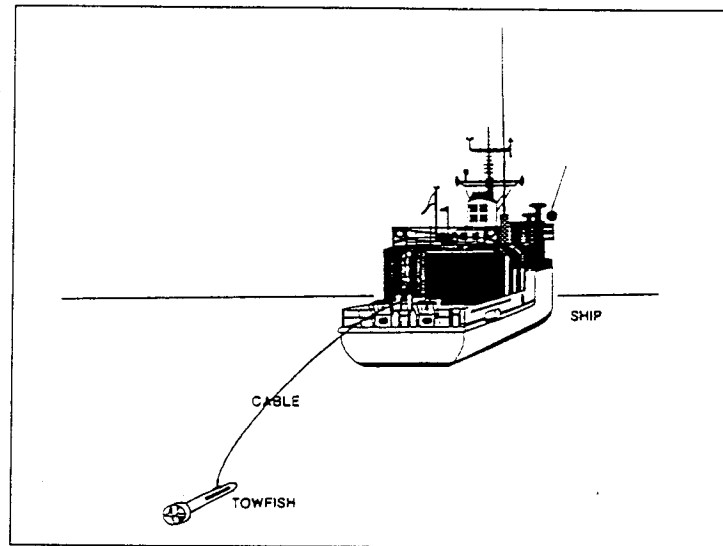


Figure 1 Sidescan sonar apparatus.

The sonar pulses cover a long and narrow strip of the ocean floor as illustrated in Figure 2:

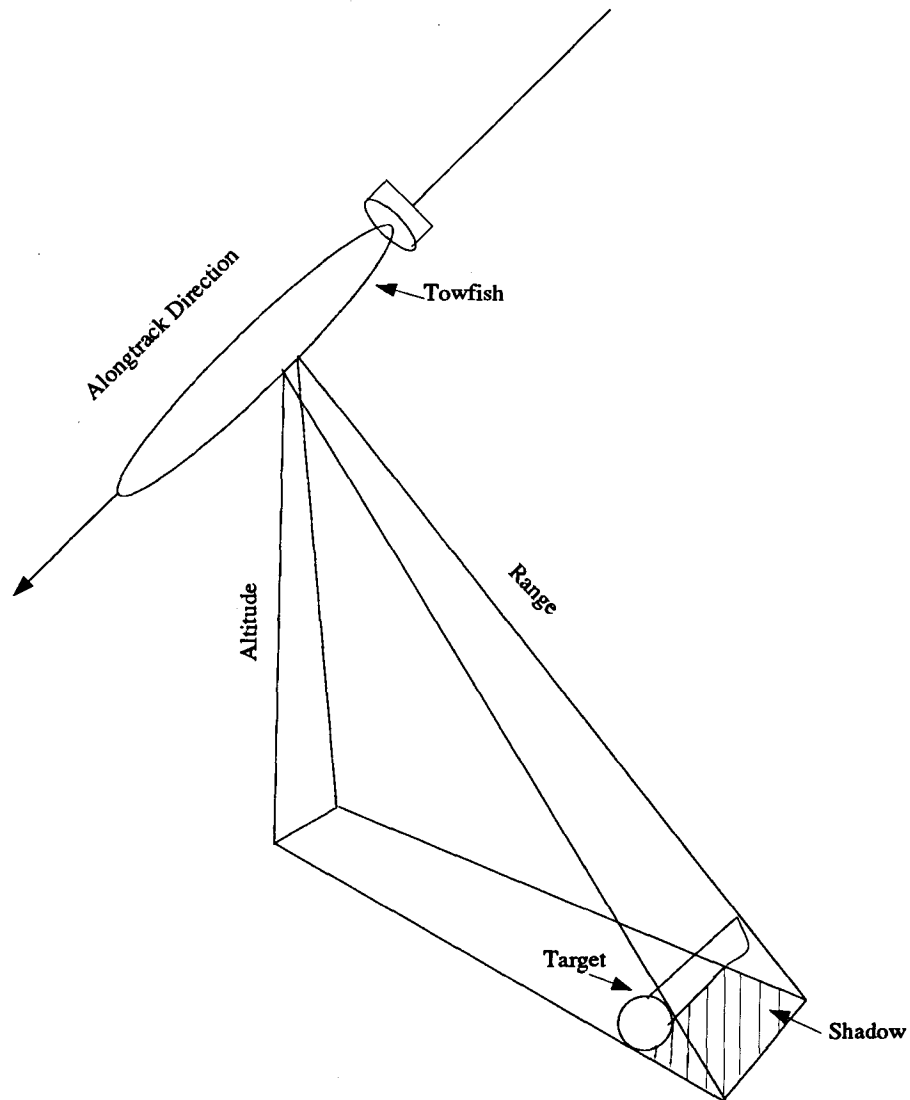


Figure 2 Emitted sonar beam geometry.

The wavefront, travelling in circular waves, reaches the sea bed directly beneath the towfish first, and then hits it further and further away from the towfish. Due to propagation loss in water, the sound energy is rapidly attenuated as it travels greater distances. The echoes measured by the transducers describe the shape and composition of the sea bed. The reflected received pulses are then transformed into images. In these images, high intensities represent objects that are at higher altitudes than their surroundings, and low intensities represent areas that are obscured by rocks, mines, and other objects that may be lying on the sea bed.

As the towfish is pulled by the ship, or is remotely controlled by it, an operator positioned in the ship would be monitoring the sea floor, looking for features or suspect mines. In this thesis, we will use the terms feature and suspect mine interchangeably, since they both refer to the same object. Monitoring these images could potentially be a slow and time-consuming process. In order to aid the operator in this task, an analysis tool is provided. This tool detects edges and highlights them to the operator. If the operator suspects the presence of a mine he or she would be able to view, on a different screen, images of the sea bed taken on an earlier mission of the same region, and would be able to make a visual comparison to determine whether or not that object was previously present, or if it might be a newly deposited mine. All images scanned on a particular day are then compressed and saved on tapes. These tapes are re-displayed on the next mission of the same region, as explained earlier. For this purpose, the

decompression speed must work on a real-time basis.

The images used in this thesis are 256 by 256 pixels in size, and each pixel represents an area of approximately 100 centimeters square. The intensities of the pixels contained in these images range in values from 0 (black) to 255 (white), although these extreme values are seldom found, i.e. pixel values are not normalized.

3 Image Enhancement

The main objective of image smoothing is to accentuate and sharpen features contained in an image. In the case of sonar imagery, any suspect mine is considered a feature. Figure 3 depicts a sonar image that will be used throughout this thesis, since it contains three features, two of which are obvious, and the third may be noticed by its shadow only (upper left quadrant of image). Each feature is recognized by its locally higher intensity, and by its shadow, recognized by locally lower intensities and adjacency to the feature. Depending on the position of the suspect mine, the locally higher intensities may not be obvious, making the task of detection even harder.

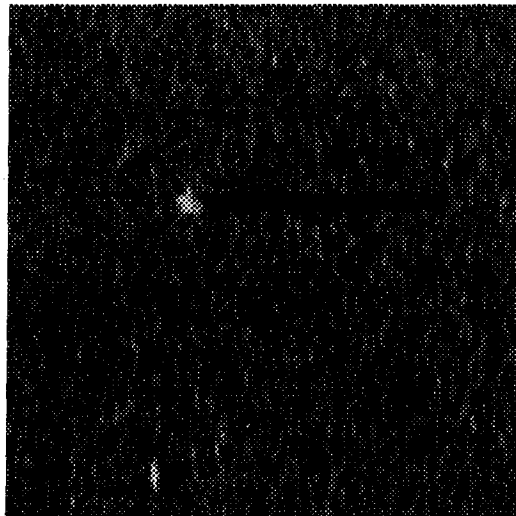


Figure 3 Original sonar image

As mentioned earlier, sonar images are inherently very noisy. The range of the noise intensity varies from high values comparable to those of the feature pixels to low values comparable to those of the shadow pixels. By trying to attenuate the noise, there will be some feature and shadow information loss. To illustrate the high levels of noise in sonar images, Figure 4 shows a three-dimensional meshed view of the original image, and Figure 5 depicts its Fourier transform. From the meshed picture, it is obvious that the greatest difficulty in the smoothing process will be to attenuate the noise without attenuating the already degraded suspect mines. It is sometimes beneficial to study an image in the frequency domain, but from Figure 5 it is evident that the outcome of such a study would be limited. The central spike in the middle of the image reflects the DC component, the neighboring region represents low spatial frequencies present in the image, and all the other impulses represent spatial frequencies of edges mainly caused by noise. A noise-free image would contain the DC component, and very limited number of high spatial frequencies (impulses). Figure 6 illustrates the Fourier transform of a smooth image (image of Lenna from [55]). It can be noticed that it contains a high DC component and some low spatial frequencies, but very limited number of high frequencies.

We first applied conventional smoothing methods to our sonar images and then applied edge detectors to them. Since the results were not satisfactory, we then developed our own smoothing filter built on the image morphology approach. At this point, we would like to clarify the requirements imposed on such a detection

system:

1. No suspect mines can be overlooked.
2. The number of false suspect mines (background pixels of high intensities) should be minimized.

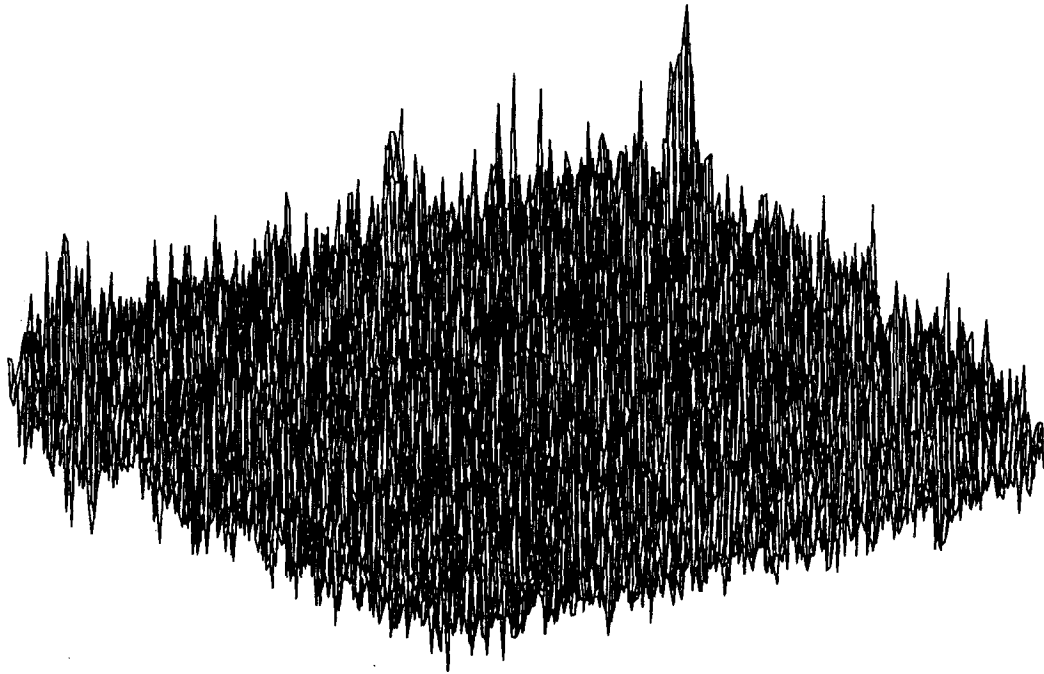


Figure 4 A three-dimensional view of fig. 3.

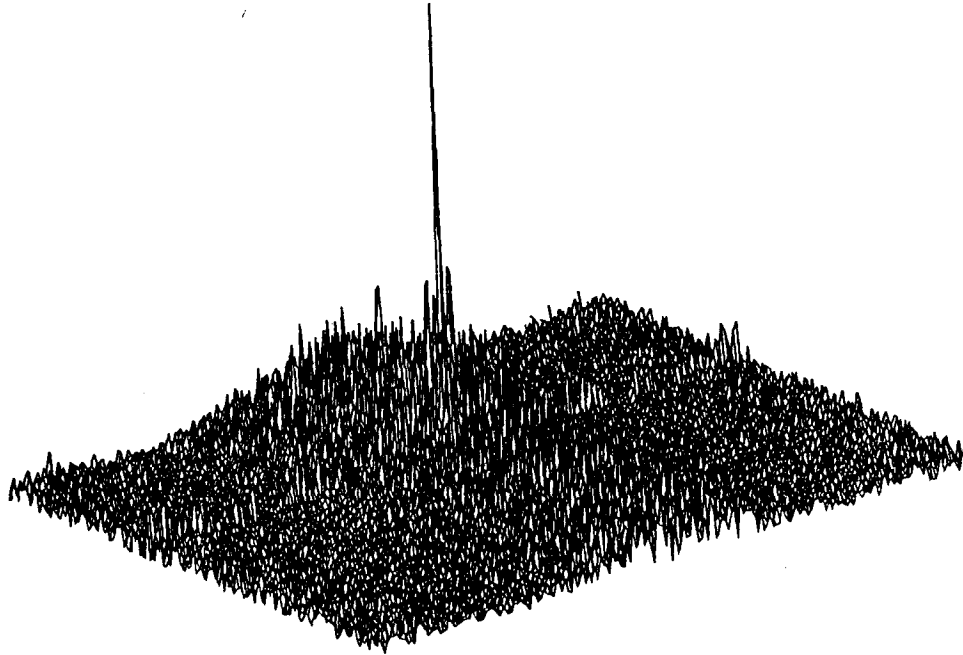


Figure 5 Fourier transform of fig. 3.

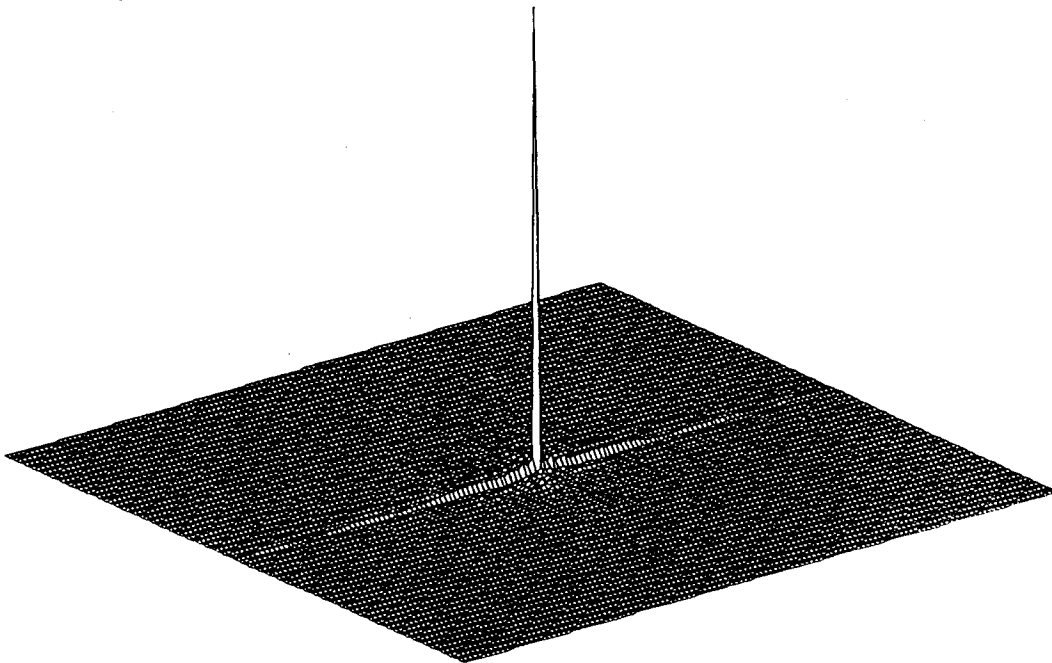


Figure 6 Fourier transform of a noise-free image.

1 Conventional Smoothing Algorithms

There are several studies and algorithms that aim at the attenuation of noise. For our study, we first considered the following three conventional techniques:

1. Spatial Averaging [2][3][4].
2. Order Statistics Filtering [1][3].
3. Out Range Noise Removal [2].

Because of the high intensity levels of noise in sonar images, it is almost impossible to discriminate noise pixels (background pixels of high intensities) from feature pixels, or even background (low intensity) pixels. What identifies a feature is an aggregation of high intensity pixels, whereas noise pixels tend to be more isolated. Therefore, the techniques that we considered do not discriminate noise pixels; rather, they process all pixels in a unified fashion. It should be evident that the smoothing process must be moderate due to the stringent condition imposed on the detection scheme: not to miss any suspect mine.

Spatial Averaging

One of the simplest ways to smooth images is to use a spatial averaging filter. This image enhancement method replaces each pixel in the image by a weighted average of its neighborhood pixels:

$$y(m, n) = 1/W \sum_{k \in W} \sum_{l \in W} a(k, l) x(m - k, n - l)$$

where x and y are the original and processed images, respectively, and W is a scanning window that traverses the original image. Although this smoothing

filter tends to attenuate the noise, it also smooths the edges surrounding features and their respective shadows, an operation that is highly undesirable. Figure 7 illustrates the effect of averaging on an edge, which translates into blurring, noticeable by the human eye.

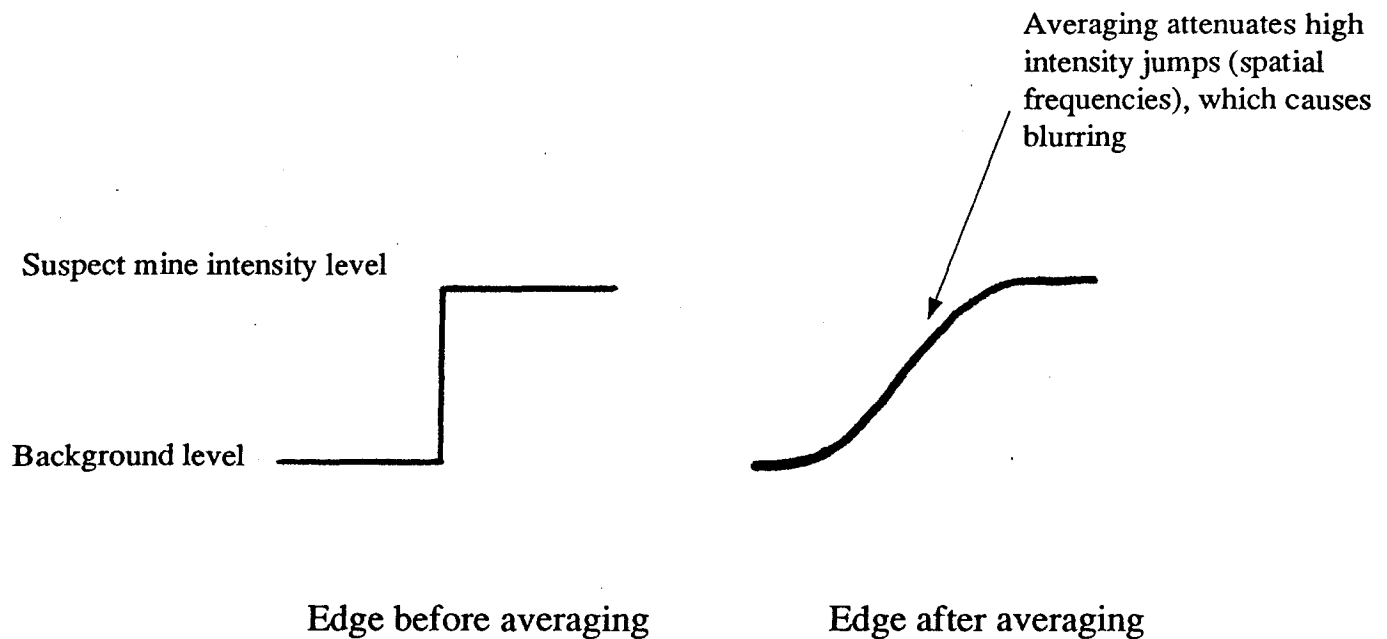


Figure 7 Effect of averaging on an edge.

Figure 8 shows the output image of a mean filter, using a 3x3 scanning window.

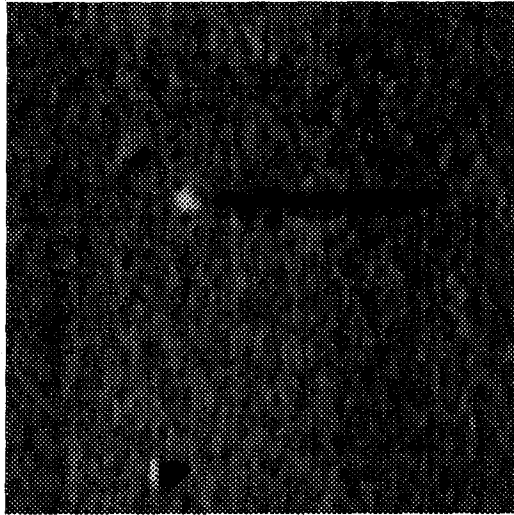


Figure 8 Output image of a mean filter on the sonar image of Fig. 3.

The main advantages of this linear filter are its simplicity and its effectiveness in noise suppression, but the fact that feature pixels are attenuated shows that spatial averaging is unsuitable for sonar images.

Order Statistics Filtering

Order statistics filters are a class of non-linear translation-invariant filters that are very popular in digital image processing, because they can be applied to different types of imagery, ranging from medical images to meteorological images. One other attractive feature of this class of filters is that they are easily implemented, and they suppress impulsive noise, while preserving the edges of the signal [1]. As in the previous averaging filter, a scanning window is positioned at each pixel in the image, and the central pixel is replaced by a percentile (in a 3x3 scanning window, the 5th ordered pixel is the fiftieth percentile), after having ordered the

pixels in the window in increasing intensities. A special case of an order statistics filter is the median filter, where the central pixel in a window is replaced by the 50th percentile of the ordered pixels. Figures 9, 10 and 11 represent samples of the filtered images. Using a low percentile caused high intensities (suspect mines have high intensities) to be filtered out, and consequently loss of information. On the other extreme, using high percentiles caused low-intensity pixels (shadows belong to this set) to be removed. Compromising between these two extremes, a median filter (50th percentile) produced blurred suspect mines. As shown earlier, blurring is caused by the disappearance of high spatial frequencies, which also translates into the attenuation of suspect mines pixels.

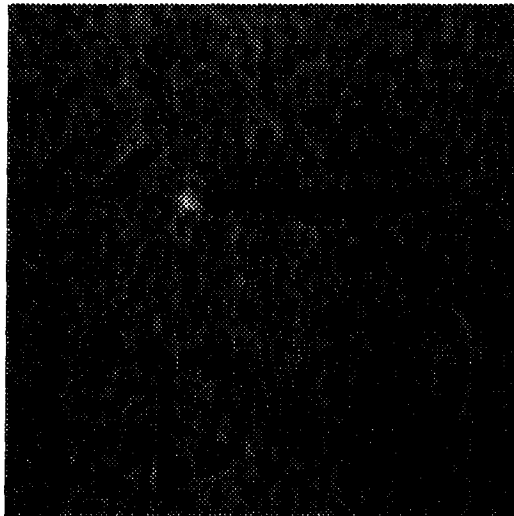


Figure 9 Output of an order statistics filter using a 3x3 window, and the 10th percentile.

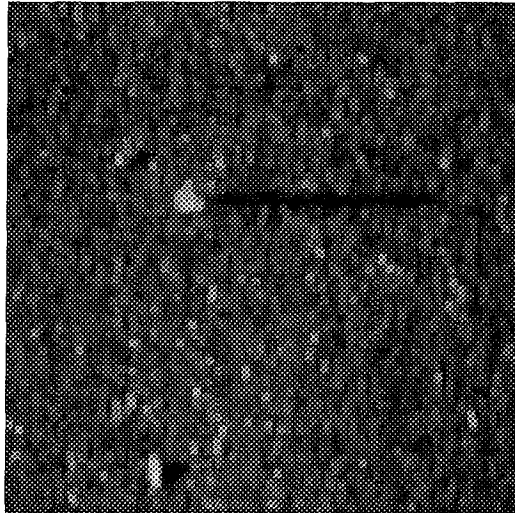


Figure 10 Output of an order statistics filter using a 3x3 window, and the 50th percentile (Median filter).

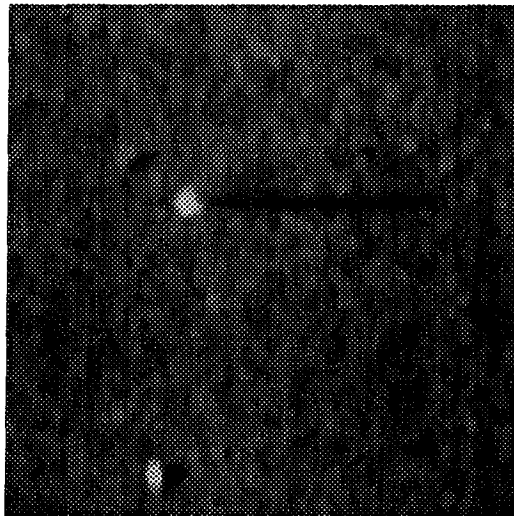


Figure 11 Output of an order statistics filter using a 5x5 window, and the 50th percentile (Median filter).

Out-Range Noise Cleaning

Unlike the conventional methods described above, this technique attempts to identify noise pixels [2], and is therefore more selective. Each pixel is sequentially examined, and if its intensity is greater than the average of its neighboring pixels by some threshold, the pixel is defined as a noise pixel, and consequently it is replaced by the neighboring average. At first glance this technique seemed promising, although experimental results showed otherwise. Figure 12 depicts a resulting image of this noise removal method.

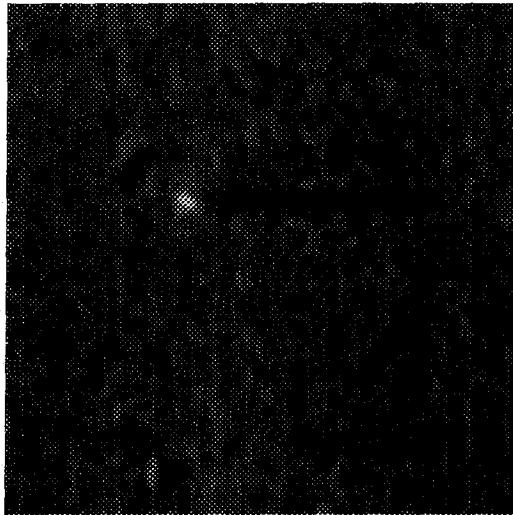


Figure 12 Out —range noise cleaning, using a high threshold value.

Furthermore, choosing a constant threshold value is an undesirable operation, since the intensities contained in a sonar image may change from region to region, depending on the composition of the sea bed. Sand seabeds have higher reflectiveness ratios than mud seabeds.

2 Cascaded Morphological Smoothing Filters

The morphological filtering of images was initiated by Matheron and Serra in the late 1960s, at the School of Mines in Paris, at Fontainebleau. Their early work culminated in the book *Image Analysis and Mathematical Morphology* [5], in which the set-theoretical methodology for image analysis was first introduced. The objective of mathematical morphology is the qualitative description of geometrical structures. Its foundations lie in the disciplines of integral geometry and geometrical probability, also drawing upon harmonic analysis and algebraic topology.

Mathematical morphology has been applied to a wide range of image analysis problems including smoothing and edge detection. Some of these applications include range imagery feature extraction [59], geological sample analysis [5], and circuit board inspection [58]. Morphological operators are suited for very local, pixel-oriented operation such as noise removal. Several papers have been dedicated to noise removal using morphological operations [6–8]. The encouraging results described in those papers prompted us to pursue this smoothing method. A short overview of mathematical morphology is first provided below for readers who have not previously encountered this subject, and the ideas presented are essential for the comprehension of this section and the ATM edge detector section. Easy to read and morphologically related introductions can be found in [6, 9–15].

Principles of Mathematical Morphology

There are two fundamental morphological operations upon which the entire filtering process is founded: dilation and erosion. For ease of understanding, we

will consider binary images (features have a value of 1, and background is 0). Let A be a binary object and B be a mask, which mathematically is referred to as a structuring element. Let us define Minkowski addition and subtraction:

Let A and B be two sets. The Minkowski addition is the following set-theoretical operation:

$$A \oplus B = \bigcup_{b \in B} A + b$$

$A \oplus B$ is constructed by translating A by each element of B and then taking the union of all the resulting translates.

Similarly, Minkowski subtraction is defined as the following operation:

$$A \ominus B = \bigcap_{b \in B} A + b$$

In this operation, A is translated by every element of B and then the intersection is taken. Based on the Minkowski addition and subtraction, we can define the most basic morphological operations, namely dilation and erosion.

$$D : Z^2 \rightarrow Z^2 \text{ where } Z \text{ is the integers' set.}$$

$$D(A, B) = A \oplus B$$

and

$$E : Z^2 \rightarrow Z^2$$

$$E(A, B) = A \ominus B$$

Geometrically speaking, Dilation and Erosion can be interpreted as follows:

1. **Dilation:** Take the original object A , translate A by each element of B and then take the union of all the resulting translates.
2. **Erosion:** Take the original object A , translate A by each element of B and then take the intersection of all the resulting translates.

Perhaps, the best way to illustrate these mathematical operations is with the following figures:

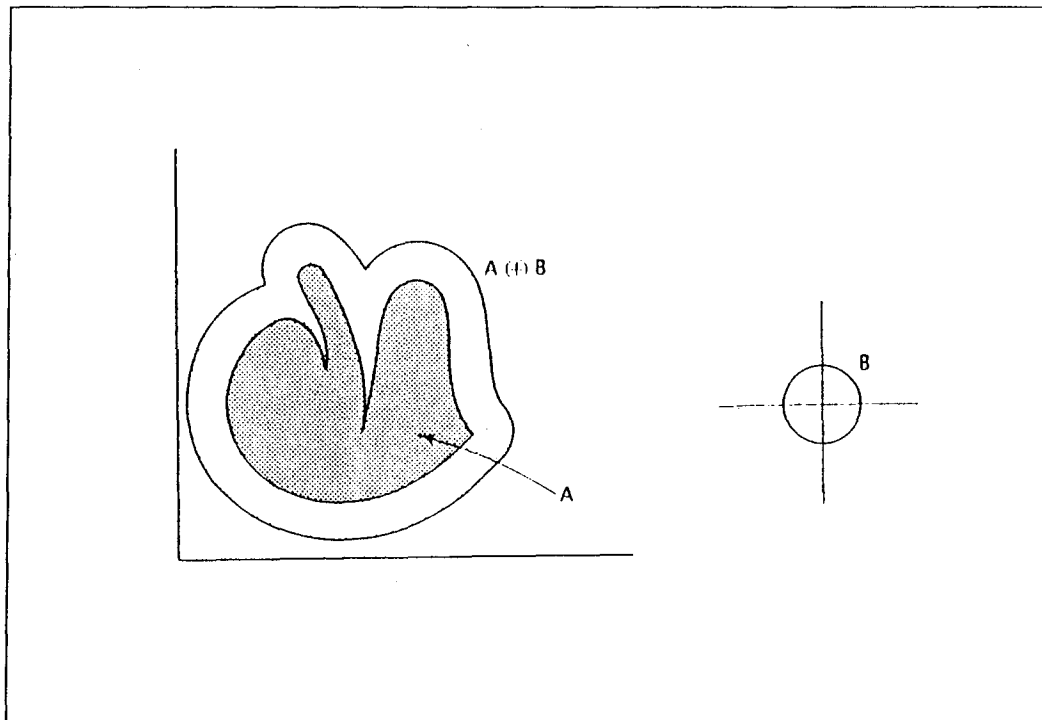


Figure 13 Dilation of object A by structuring element B (from [6]).

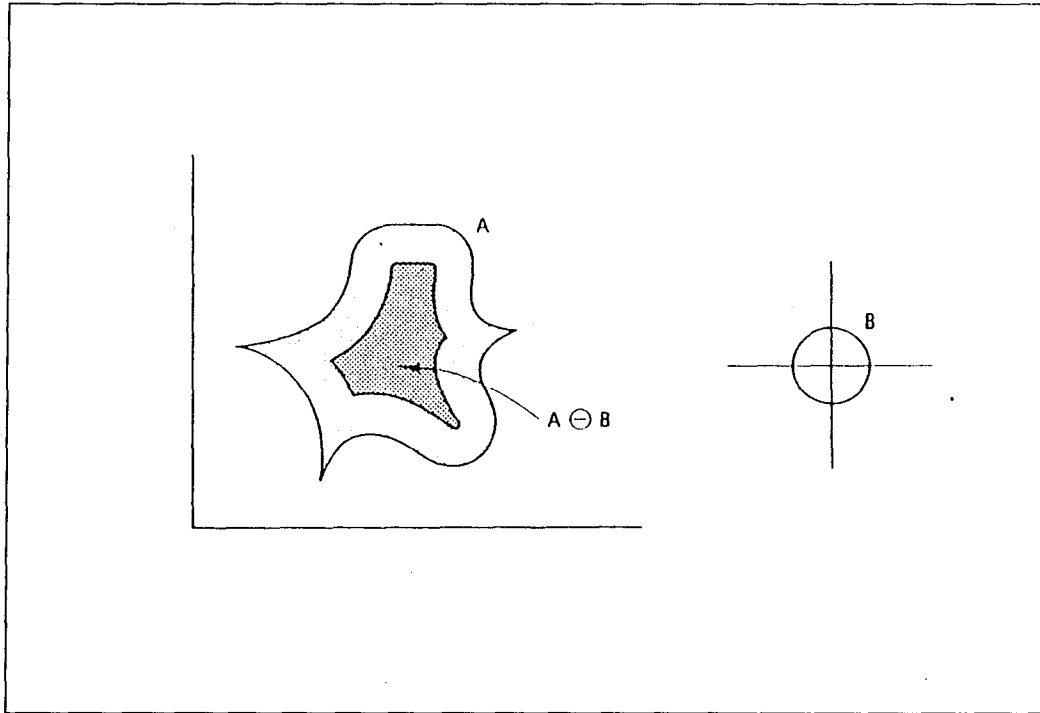


Figure 14 Erosion of object A by structuring element B (from [6]).

For completeness, a summary of Minkowski's properties are listed [6], which resemble the usual Euclidean properties:

1. **Commutativity:** $A \oplus B = B \oplus A$
2. **Associativity:** $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
3. **Translation invariance:** $A \oplus (B + x) = (A \oplus B) + x, x \in R^2$
4. **Duality:** $A \oplus B = [A^c \ominus B]^c$
5. If $A_1 \subset A_2$, then $D(A_1, B) \subset D(A_2, B)$

For grey scale operations, the “Union” and “Intersection” operations are replaced by “maximum” and “minimum” operations [15]. Also, in the grey scale case, the object A represents the whole image. Based on the two primitive morphological operations, there are two additional operations, namely closing and opening, which are defined as follows:

$$O(A, B) = [A \ominus (-B)] \oplus B$$

where $-B$ is simply B rotated 180° around the origin, and

$$C(A, B) = [A \oplus (-B)] \ominus B$$

Employing erosion and dilation terminology, we have

$$O(A, B) = D[E(A, B), B]$$

and

$$C(A, B) = E[D(A, -B), -B]$$

The Opening and Closing are illustrated in Figures 15 and 16 respectively.

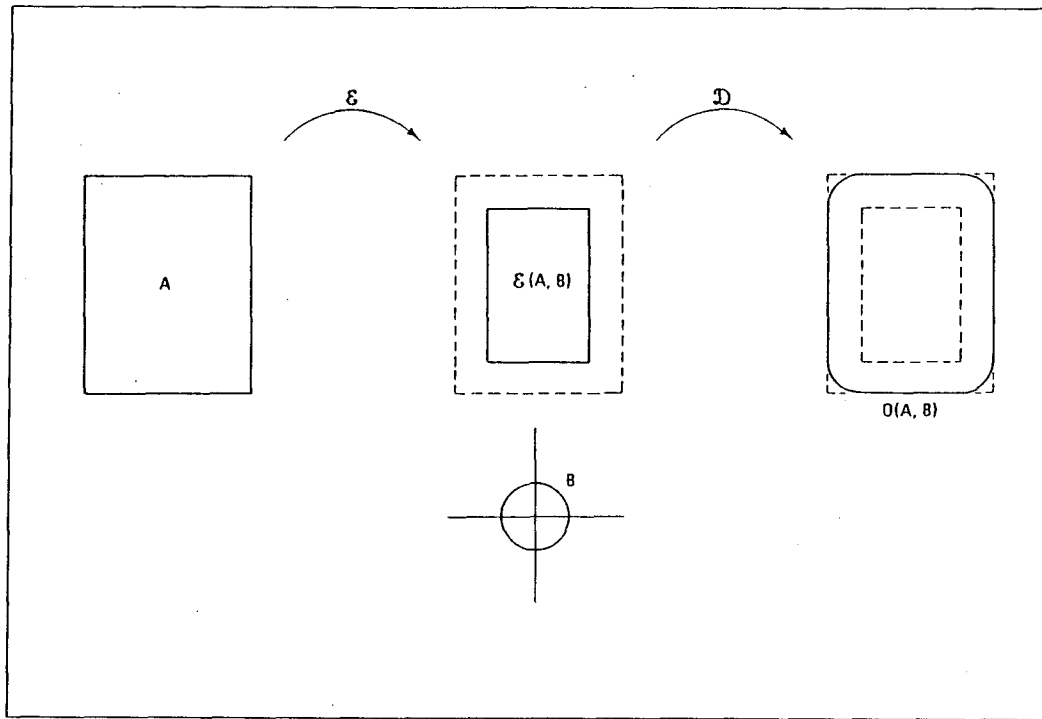


Figure 15 Morphological opening (from [6]).

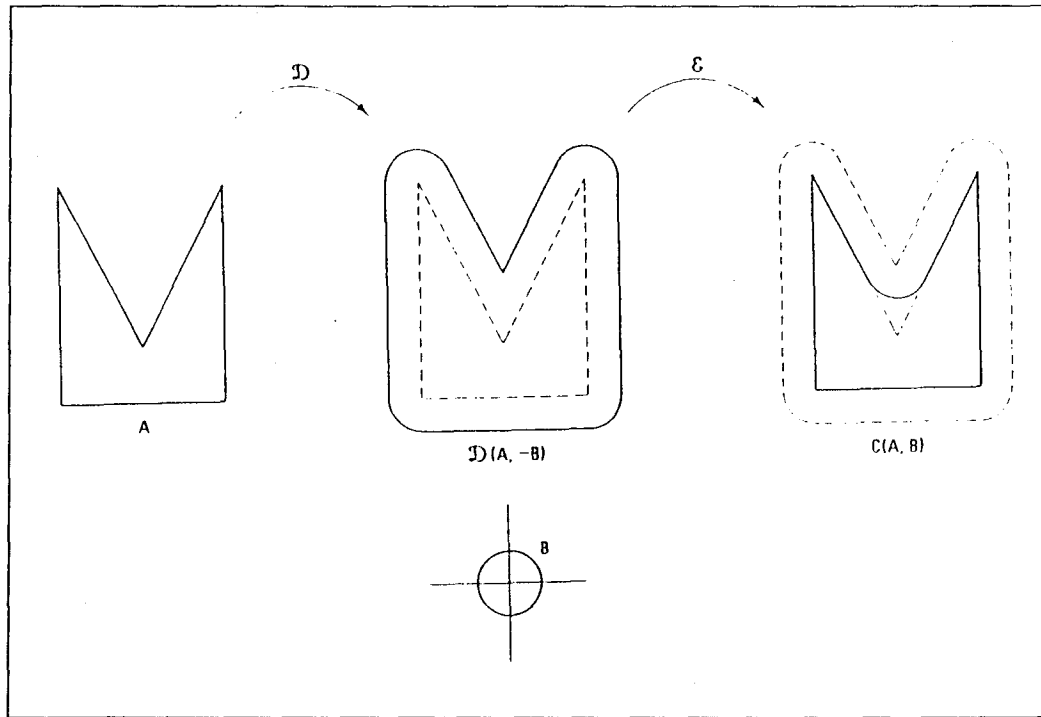
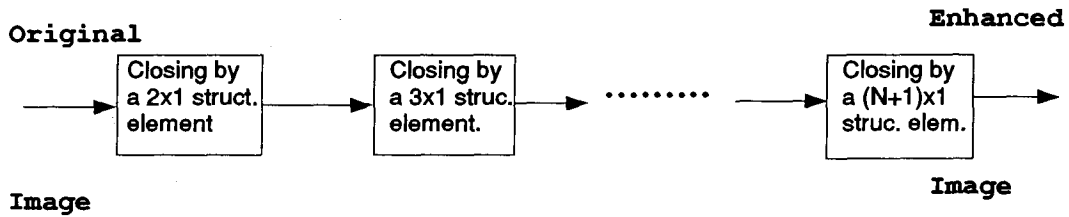


Figure 16 Morphological closing (from [6]).

The Enhancement Filter

We applied morphological opening and closing to the original image, using three structuring elements: a 3x3 window, 3x1 window, and a 1x3 window. We noticed that applying these filters gradually removed the noise, without greatly affecting the features that are contained in these images. Encouraged by these results, we used a cascaded morphological filter, with a block diagram as shown in Figure 17. The structuring element in each cascaded filter has the same

shape but different dimensions. The larger the structuring element, the more



N cascaded Morphological Filters

Figure 17 Block diagram of the cascaded morphological enhancement filter.

information loss occurs (Minkowski properties). But, also the larger the structuring element, the more noise that is removed, of course at the expense of removing features as well. Therefore there is a trade-off between noise removal and feature preservation. We noticed that using no more than 3 cascaded filters lead to satisfactory result. Figures 18, 19, 20, and 21 show the enhanced images, using the cascaded filters of order 3, with a horizontal bar (2x1 followed by 3x1, followed by 4x1 structuring element), a vertical bar (1x2, 1x3, then 1x4), a square window (2x2, 3x3, then 4x4), and a circular window (radius=2, radius=3, then radius=4) structuring elements. Two-dimensional and Large structuring elements cause higher attenuation of noise and also higher attenuation of features than one-dimensional and small structuring elements. Since we want to remove noise in a moderate fashion, a one-dimensional window (bar) would achieve our goal. Since features and shadows are usually longer than wider (particularly true for

shadows due to the geometry), information loss caused by horizontal structuring elements would have a lesser effect than by vertical bars.

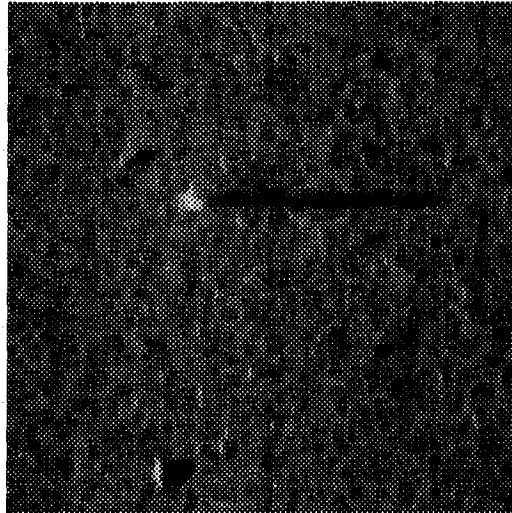


Figure 18 Output of cascaded filter with a horizontal bar as structuring element.

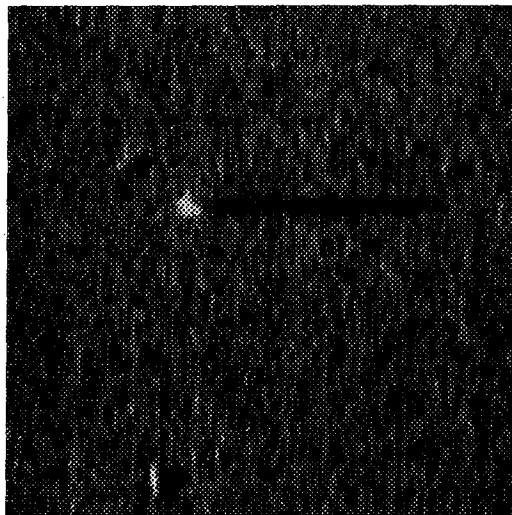


Figure 19 Output of cascaded filter with a vertical bar as structuring element

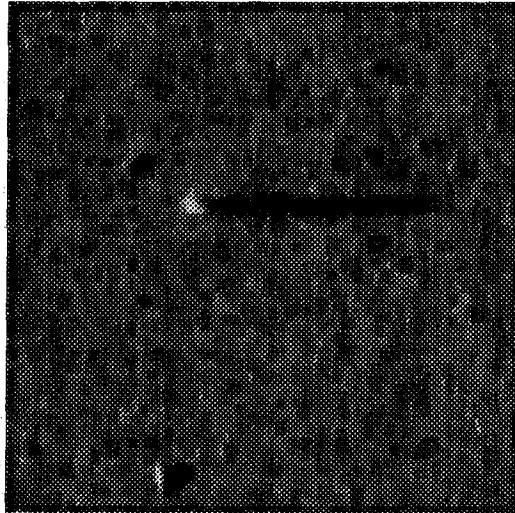


Figure 20 Output of cascaded filter using a square window as a structuring element

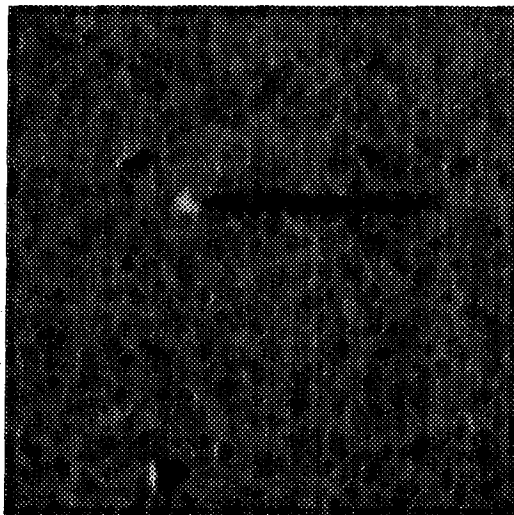


Figure 21 Output of cascaded filter using a circular window as a structuring element.

3 Sonar Image Enhancement Results

From an enhancement point of view, all filters that we considered attenuated noise. In addition, they all attenuated feature and shadow pixels. However,

some attenuated these informative pixels more than others. The goal of the enhancement step is to remove noise while preserving features as much as possible. This preservation of features is needed by edge detectors since they base their search on edges that surround features.

4 Edge Detection

1 Introduction

After having preprocessed the sonar images with a smoothing filter, the task of locating features, or equivalently suspect mines, is tackled. Much research has been done on this subject [18–40]. Edge Detection can be defined as the process that locates the boundary between two adjacent regions of an image, which differ in some characteristics. In this thesis, this translates into edges that surround high-intensity objects (suspect mines) and adjacent low-intensity regions (shadows). No single Edge Detector (ED) algorithm will successfully work on all types of images. Each class of imagery including optical, biomedical, meteorological, and sonar, has different characteristics that require specialized algorithms for locating edges contained in those images. This task of finding edges is more complex, and sometimes impossible, if noise is present. Since sonar images are noisy images, we investigated edge detection research papers concerned with overcoming the noise problem. In addition, there are different kinds of noise, such as white Gaussian, impulsive, additive, and multiplicative. Therefore, we were very selective in finding an ED that may suit our set of sonar imagery.

Suk and Hong [17] described a new edge extraction technique specifically developed for noisy images. A couple of years later, another paper was presented by Feehs and Arce[41], which also dealt with immunity to noise.

At the beginning of our research, we felt that only few edge detectors may be suitable to the sonar set, and out of 4 promising methods briefly described below, only one was identified successful for the sonar images application.

Zero-Crossing of the Laplacian of Gaussian

This technique is introduced by Marr and Hildreth [30]. It is based on combining zero-crossings from the output of a Laplacian of Gaussian filters for several spatial frequency channels. The Gaussian filter is first used for smoothing the image, and is followed by the Laplacian which is a nondirectional second derivative 2D operator. Since the Gaussian is a low-pass filter, the smoothing step of this method should suppress noise and edges indiscriminately. The outcome is a blurred image, which is what we tried to avoid in the previous chapter, since this information loss is non-recoverable.

Mathematically, the Laplacian of Gaussian operator is defined as:

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^4} \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

where σ is the standard deviation of the Gaussian.

Although this approach proved to be successful in many applications, and many improvements have been appended to this ED [24, 31], in the sonar images case it failed. The next figures show the output of this edge detector using a narrow and wide standard deviation ranges for the Gaussian processing.

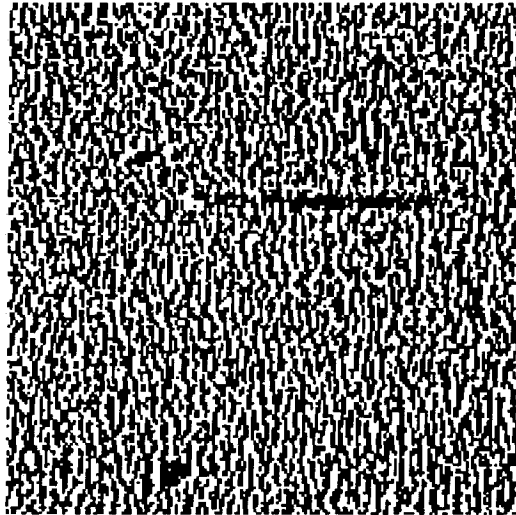


Figure 22 Output of a zero-crossing edge detector using a narrow standard deviation range.

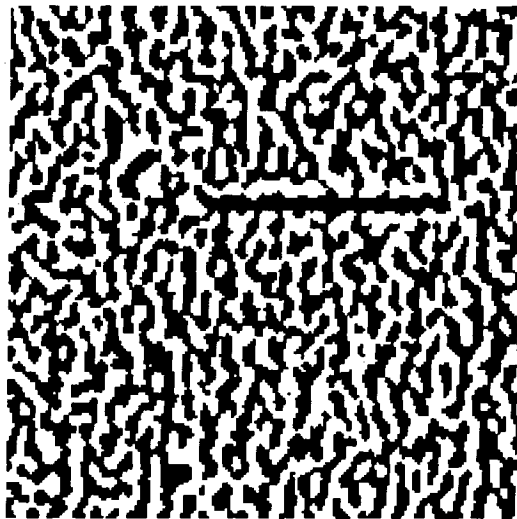


Figure 23 Output of a zero-crossing edge detector using a wide standard deviation range.

The failure is attributed to the high levels of noise, i.e. the comparable intensities of features and noise pixels on the one hand, and shadows and noise

pixels on the other. The space constant of this filter σ was made very large in order to ignore small edges, caused by noise pixels. But even with large σ , the resulting image did not meet our preset specifications (see FIG. 23).

Step Function Edge Detector

As its name suggests, the step function edge detector search is oriented towards finding an intensity step. After implementation, we realized that this technique is highly inappropriate for sonar images. The three dimensional view of the image (fig. 4) shows how many step functions it contains. The task of labeling the step functions that correspond to the edges that surround features and shadows is impossible. Figure 24 illustrates how inappropriate is this method.

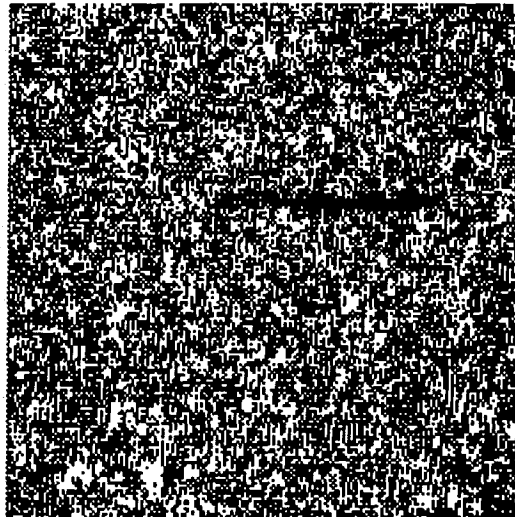


Figure 24 Output of a step function edge detector searching for intensity steps larger than 20.

Histogram Thresholding

Easy in concept and easy to implement, this method bases its search for edges on thresholding. It does not specifically look for edges, rather it looks for features and shadows. Therefore, the Histogram Thresholding is not a true edge detector, rather it is a feature locator. Nevertheless, in this thesis it was classified as an edge detector since it leads to the same outcome: locating suspect mines.

The histogram of the original image is built, and two values are extracted from it: a lower threshold, chosen from the low-intensity end of the histogram, and an upper intensity threshold is chosen from the high-intensity end. The filtering proceeds as follows: if the intensity of any given pixel is either lower than the lower threshold, or higher than the upper threshold, then the pixel is unchanged, otherwise a value of 128 (grey value) is assigned for background distinction.

After thresholding the image, many noise pixels were filtered through, but some isolated ones remained. In order to remove these isolated noise pixels, a morphological opening was performed. Figure 25 illustrates the output of this filter. It appears that the drawback of this technique is its use of thresholding values, an operation that is not recommended, since the choice of threshold values is highly dependent on the image under study. In this case however, both upper and lower thresholds were extracted from the histogram of the image, which is operator-independent. Just to illustrate the sensitivity of thresholding, Figure 26 shows the output of a histogram thresholder, with threshold values that differ by 60 (0.1 % deviation) pixels from the ones used in Figure 25.

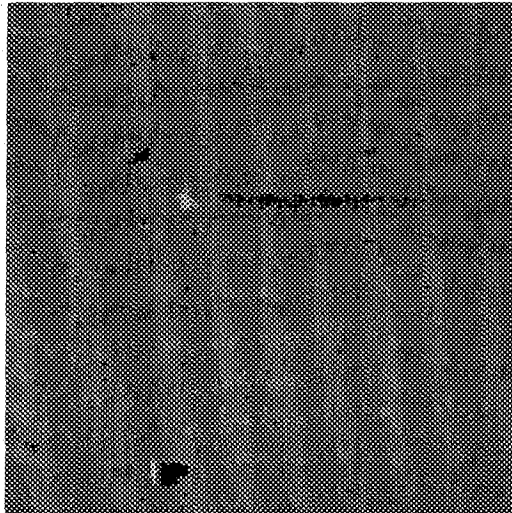


Figure 25 Histogram thresholding

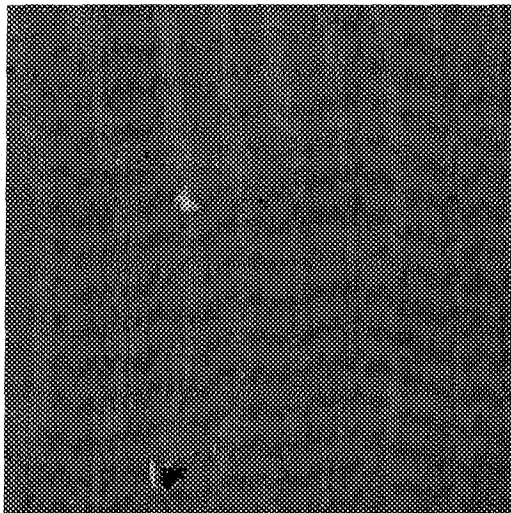


Figure 26 Sensitivity of histogram thresholding: threshold changed by 0.1% of that of FIG. 25.

Multidimensional Morphological Edge Detection

For noisy images, this class of edge detectors outperformed most mask and differentiation based edge detectors [7]. The reason for this technique's

success stems from the fact that it is based on analyzing the geometric structures contained in images. Lee, Haralick, and Shapiro [8] introduced a more primitive morphological edge detector, which showed considerably better results at low signal to noise ratios than any of the previously developed edge detectors. We found that this approach gave us encouraging results, better than any other edge detector that we have considered. Section 4.2 is dedicated to the analysis of this ED.

Other Edge Detectors

Other edge detection work has concentrated on finding different smoothing filters to deal with noise [17,18], or on optimizing positional accuracy by using directional operators [26, 27].

Chen, Huertos and Medioni [26, 37] have developed an iterative approach for the removal of juxtaposed edges. A fairly recent and sophisticated scheme was published by Canning, Kim, and Rosenfeld [40], which consists of two passes, one for the search of i-pixel ridges, and the other for “double checking” purposes. Other sophisticated edge detectors were suggested but, as we mentioned earlier, our judgement was that they were inappropriate for sonar imagery.

2 Multidimensional Morphological Edge Detector

As mentioned in the previous section, we found the multidimensional morphological edge detector to be suitable for sonar imagery. A more primitive type of edge detector of the same category was designed and tested by Lee, Haralick

and Shapiro [8] in 1986. In their paper, they applied this ED to noisy images, but not comparable to the amount of noise present in sonar images. This ED was efficient and robust, and more importantly immune to noise (small amounts of noise). It consisted of two major processing stages. The first step was to blur the image, using a blurring filter, which had a side effect of attenuating the noise present in these images. It also caused feature loss, which was tolerated in [8], since the noise level was small. The second step consisted of morphological operations. Figure 27 illustrates these steps:

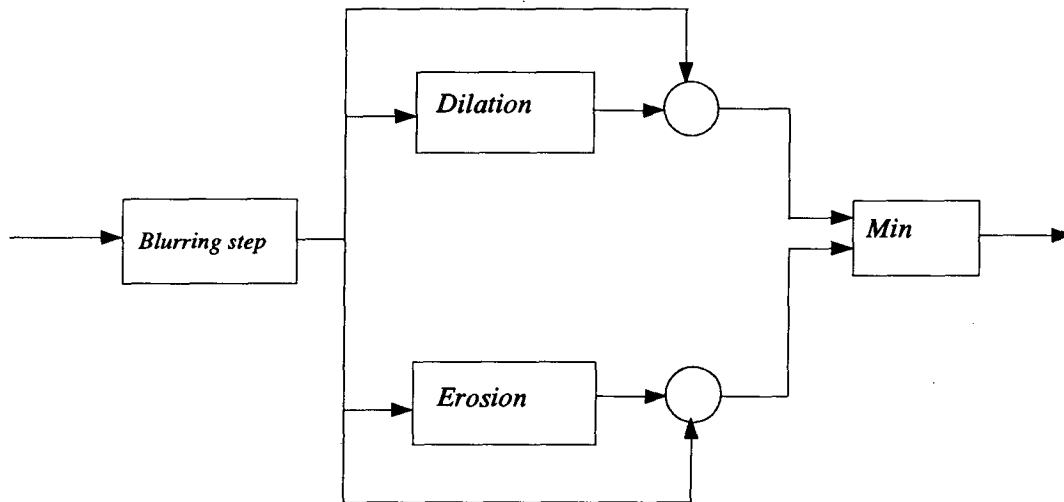
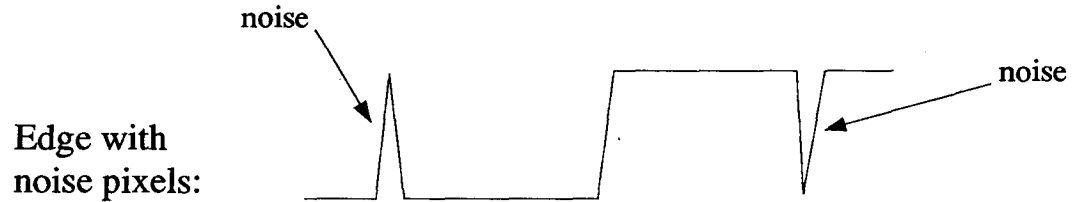


Figure 27 Block diagram of the “primitive” morphological edge detector.

A similar edge detector was later introduced by Feehs and Arce [7], the Alpha-Trimmed Morphological Edge Detector. In this filter, the image is first preprocessed through an Alpha Trimmed operator [42], and the detection is done with

more sophisticated cascaded morphological filters. Perhaps the best way to illustrate the idea behind this edge detector is by an example: (for simplicity, consider one line of the image with the following intensities)



<i>Original:</i>	3 3 9 3 3 3 3 3 9 9 9 9 9 3 9 9
<i>Blurred:</i>	3 5 5 5 3 3 3 5 7 9 9 9 7 7 7 9
<i>Dilation:</i>	5 5 5 5 5 3 5 7 9 9 9 9 9 7 9 9
<i>Erosion:</i>	3 3 5 3 3 3 3 3 5 7 9 7 7 7 7 7
<i>Closing:</i>	3 5 5 5 3 3 3 5 7 9 9 9 7 7 7 9
<i>opening:</i>	3 5 5 5 3 3 3 5 7 9 9 9 7 7 7 9
<i>lower branch:</i>	2 0 0 0 2 0 2 2 2 0 0 0 2 0 2 0
<i>upper branch:</i>	0 2 0 2 0 0 0 2 2 2 0 0 0 0 0 2
<i>Output of ED:</i>	0 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0



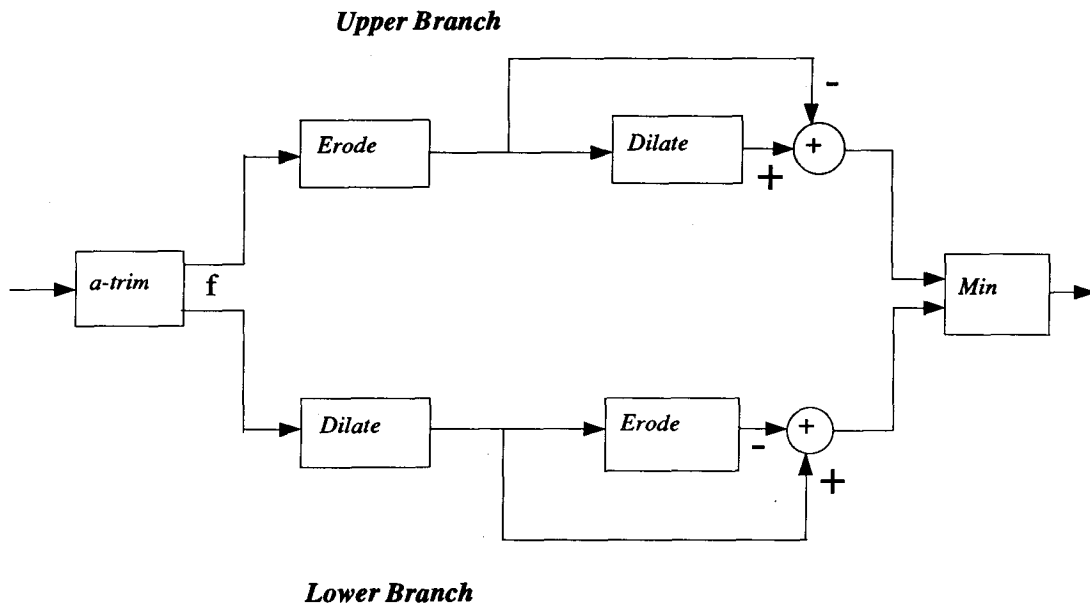


Figure 28 Block diagram of the ATM edge detector:

The results of this ED are shown in Figure 29. The appropriate value of α was easily determined. Relatively large values of α would cause the α -trimmed filter to behave like a spatial averaging filter. And as it is shown in the conclusive remarks in this chapter, spatial averaging followed by ATM edge detection performed poorly. Choosing small values of α gave much better results (FIG. 29). Specifically, a value of $\alpha = 1$ gave best results.

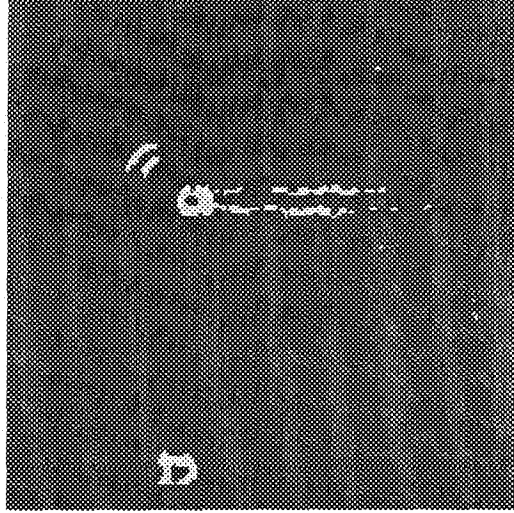


Figure 29 ATM Edge Detection, with $\alpha = 1$ and preceded by a morphological enhancement step.

Mathematically, the ATM edge detector can be defined as follows: let the input and output images to this ED be denoted as $f_i(z) \subset Z^2$, and $f_o(z) \subset Z^2$ respectively. Moreover, let $f(z) \subset Z^2$, denote the alpha-trimmed mean blur of the input image and be defined as

$$f(z) = \sum_{l=1+\alpha}^{k-\alpha} \frac{f_{(l)}(z)}{k - 2\alpha}$$

where $k = (2n + 1)^2$ is the number of samples spanned by the basis function B, n the order of the filter, and $f_{(l)}(z)$ is the l th smallest sample of $f_i(z)$ spanned by B. In simpler terminology, $f_{(l)}(z)$ is the l th ordered pixel in a scanning window of dimension k (for example, if the scanning window is 3x3, then $k=3 \times 3=9$). The ATM edge detector output is defined as

$$f_o(z) = \min \{ [f_B(z) - (f \ominus B)(z)], [(f \oplus B)(z) - f^B(z)] \}$$

where $(f \oplus B)(z)$ and $(f \ominus B)(z)$ are respectively, the **dilation and erosion** of image $f(z)$ with the structuring element B . f_B is erosion followed by a dilation operation, and f^B is dilation followed by an erosion operation.

A more elaborate analysis of morphological edge operators can be found in [43]. Just to illustrate the need for a smoothing step before applying an ED, we have included the output of the ATM ED, applied to the original image in Figure 3. Figure 30 shows the result of that operation, where it can be seen that many false edges (edges caused by noise pixels) have been detected. These results violate the specifications of this system.

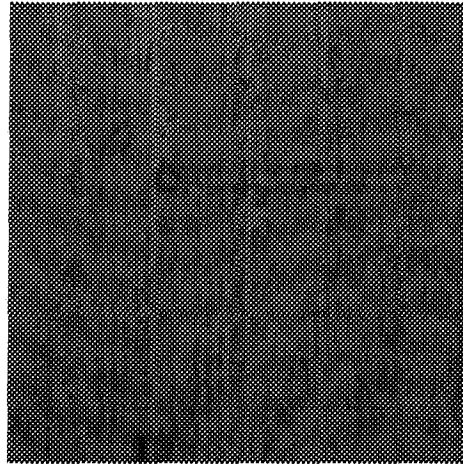


Figure 30 ATM Edge Detection without the enhancement step.

3 Results of Edge Detection

Edge Detection Success Measurement

Our measurement of success for edge detection filters was based on the detection requirements introduced earlier. These are:

1. No suspect mines can be overlooked.
2. The number of false suspect mines (background pixels of high intensities) should be minimized.

A feature was considered successfully extracted if at least 2/3 of the perimeter of a feature was detected. We also considered a continuity judgement. The continuity constraint meant that a feature must not have large holes (relative to the extracted perimeter segments).

Results Using Different Pre-Processing Enhancement filters

Since all edge detectors, with the exception of the ATM edge detector, failed to meet the requirements of this system with and without the enhancement step, we chose to compare the different smoothing filters using the ATM edge detector only. Figures 31, 32 and 33 show the results of the ATM edge detector after spatial averaging, out-range noise removal, and median filtering, respectively. It can be easily concluded that these pre-processing filters attenuated feature and

shadow edges more than the cascaded morphological filter. This edge attenuation caused the ATM edge detector to fail to meet the requirements.

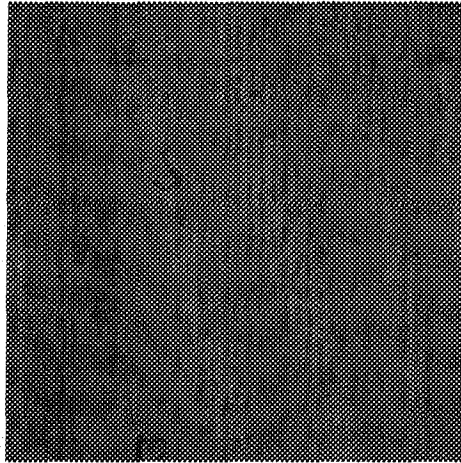


Figure 31 ATM Edge Detection after spatial averaging.

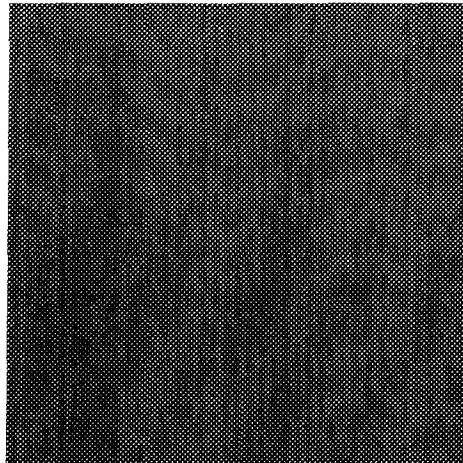


Figure 32 ATM Edge Detection after out-range noise removal.

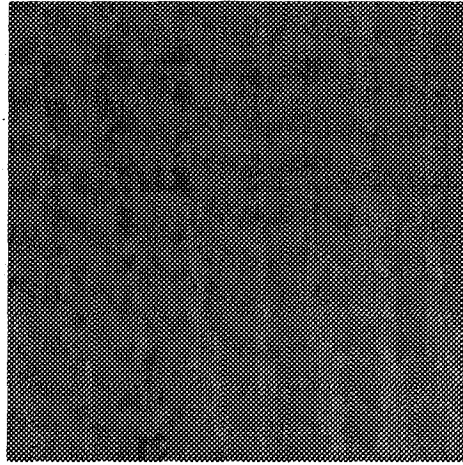


Figure 33 ATM Edge Detection after median filtering.

Results

In preparing for this stage of our research, we studied many technical papers describing “new” edge detectors that work well in the presence of noise, however we found no papers that dealt with images as noisy as those that we were considering.

After extensive testing, it was found that the ATM edge detector extracted all the edges that surround features and their corresponding shadows. All detection operations were performed on the enhanced images that were discussed in Chapter 3. Without a smoothing step, all edge detectors including the ATM edge detector performed poorly since they introduced false suspect mines (by false suspect mines, we mean background pixels that have high intensities). Therefore, the preprocessing enhancement step is mandatory. Obviously, the success of this

edge detector is contributed to its immunity to high levels of noise. Another factor that is crucial in this filter's success is the choice of the structuring element. In [9, 10, 16], it is shown that the structuring element must belong to a subset of eroded features, a product of skeletization, or in simple terms, must be "similar in shape" to the feature. This filter has another important advantage over conventional filters: it is efficient. It could be implemented in a real-time image processing operation, which is required by the Mine Counter Measure System (cf. Chapter 2). Two efficient implementations of morphological filters are described in [9] and [11]. The practical method for performing morphological transformations takes advantage of the parallel shifting array architecture as depicted in Figure 34.

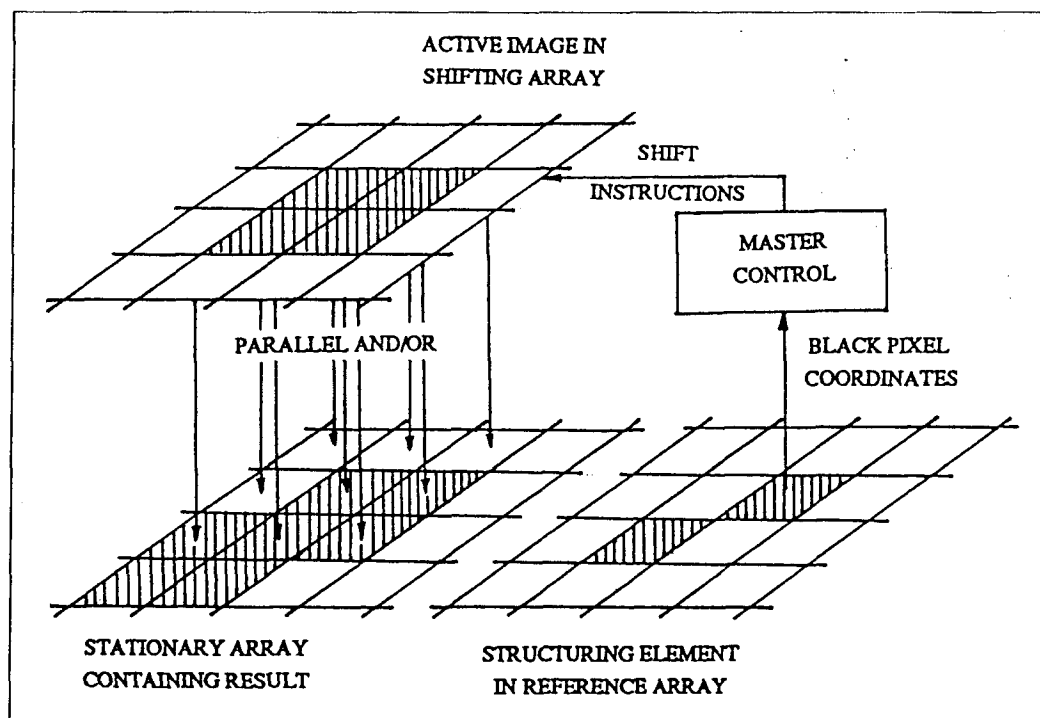


Figure 34 Parallel array shift architecture (from [9]).

The following table summarizes the results of four edge detectors, two of which showed good results, while the other two were incompatible with images as noisy as sonar images.

Edge Detectors	Advantages	Disadvantages
<i>Laplacian of Gaussian</i>		<ol style="list-style-type: none"> 1. Too many false edges 2. Slow edge detection (20 seconds)
<i>Step Function</i>		<ol style="list-style-type: none"> 1. Too many false edges
<i>Histogram Thresholding</i>	<ol style="list-style-type: none"> 1. Fast algorithm (2 seconds) 	<ol style="list-style-type: none"> 1. The value of the threshold is picture-dependent 2. Extremely sensitive to deviations from the "right" threshold
<i>Multidimensional Morphological Edge Detector</i>	<ol style="list-style-type: none"> 1. Detected all feature edges (with our pre-processing enhancement step) 2. Fast algorithm (3 seconds) 	

Table 1 Tabulated results of edge detection.

5 Image Compression

1 History and Definitions

Motivation for the development of data compression algorithms stems from continuing demand for cost-effective and higher capacity data storage. Since one of the goals of the MSCM is to store a library of images that would cover the major Canadian ports and coastal waters, it is mandatory to compress the images in order to render that goal feasible. In addition, this goal would prove to be very costly if some sort of data compression is not employed.

Although the principle of data compression is not new, there has been world-wide renewed interest, triggered by the recent deployment of Integrated Services Digital Network (ISDN), and the need to have a global image compression standard. The field of data compression can be partitioned into two fields: lossless and lossy coding. Even though lossless compression is used in applications that cannot tolerate information loss, the achieved compression ratio is small, in the order of 2 to 1. Such a compression ratio is too small for this application, and information loss can be tolerated as long as *no features are lost after reconstruction*.

Lossy compression algorithms achieve much higher compression ratios, obviously at the expense of image degradation after reconstruction. Lossy compression techniques fall into three categories:

1. Predictive Coding.
2. Transform Coding.
3. Block-based Coding.

Predictive Coding

In a predictive image coding system, the value of each pixel is predicted based on the history of previously processed pixels. The estimate is then subtracted from the actual pixel value, and the difference is quantized. In the reconstruction process, a similar prediction is done, which should produce the same error obtained at the encoder, and the quantized difference is added to the prediction, leading to an estimated value very close to the original pixel value.

Transform Coding

This method is simple in concept and yields good compression ratios. It takes advantage of the high energy concentration in the lower coefficients of the image representation in the transformed domain. This coding technique has become one of the most widely-used technique in the industry. The International Consultative Committee for Telephone and Telegraph (CCITT), the international body in charge of the standardization of image compression and transmission (ISDN application) has adopted a compression technique that is based on transform coding [44, 45]. Recently, the Joint Photographic Experts Group (JPEG), has submitted a proposal towards establishing the first international digital image compression standard for

still greyscale and colored images [46]. The proposed algorithm suggests using the discrete cosine transform as one of its major processing steps.

Block-Based Coding

Block Truncation Coding (BTC) and Vector Quantization (VQ) are the main two coding techniques that fall under this category. In these methods, the image is subdivided into non-overlapping squares, which are sometimes referred to as blocks or vectors. Instead of processing each pixel individually, each block is considered as one entity. BTC is a relatively old technique which seems to have reached its maturity, in terms of research [47–50]. Nonetheless, its simplicity and ease of implementation make it a very attractive coding/decoding scheme.

More recently, a more interesting and still a highly researched topic, is the block-based scheme: Vector Quantization. This scheme maps a sequence of blocks, mathematically referred to as vectors, into a digital sequence.

In the above paragraphs, we have very briefly described the different compression schemes that have been in use or are still being studied by the research community. Since our goal is to compress sonar images, we opted for studying the following compression methods:

1. BTC

2. JPEG

3. VQ

2 Block Truncation Coding

The BTC is an image coding method, introduced in 1979 by Delp and Mitchell [47]. As mentioned earlier, the idea behind this coding method is simple: each image is divided into 4x4 non-overlapping blocks of pixels, and each block is coded individually. The mean and the variance of each block are stored along with 16 bits, each bit taking the value of 0 or 1, depending on whether each pixel is less or greater than the mean in that block. Only 4 bytes are needed to represent the block (mean of block, variance of block, and 16 bits as explained above), instead of 16 bytes, assuming the values of the pixels range from 0 to 255. Therefore, each smooth block leads to a compression ratio of (2+2)/16, or 4 to 1.

At the reconstruction stage, if the block was smooth, then each pixel that had a value higher than the mean in that block is assigned the value:

$$B = \bar{x} + \sigma \sqrt{(16 - q)/q}$$

and each pixel that had a value below the mean of that block is assigned the value:

$$A = \bar{x} - \sigma \sqrt{q/(16 - q)}$$

Nasiopoulos, Ward and Morse [50] devised an adaptive BTC algorithm which preserves edges: The algorithm calculates a smoothness measure (the variance of the block), and if the block is not smooth enough (a comparison of the variance

and a user-selected threshold is performed), then all 16 pixels are stored without modification. If the block is smooth, then BTC is used.

Applying this scheme to sonar images, we realized that it is highly inadequate to our application. The achieved compression ratio was of the order of 2 to 1, because only few blocks were smooth, due to the presence of noise. In addition to the small compression ratio, the quality of the reconstructed image was very poor, as illustrated in Figure 35.

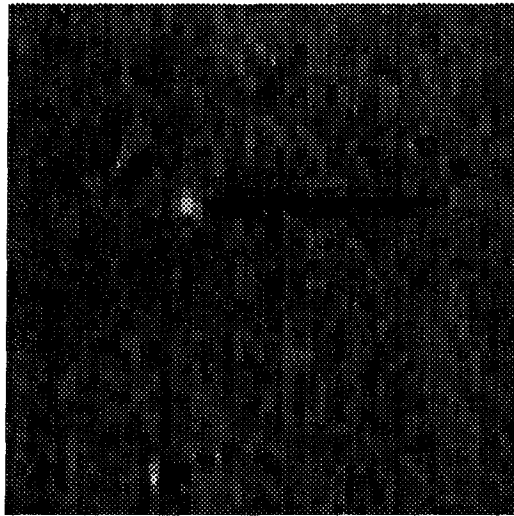


Figure 35 Reconstructed sonar image after BTC encoding and decoding

3 JPEG Algorithm

As mentioned in an earlier section, JPEG is a transform coding scheme, and furthermore a proposed still image compression standard that has been submitted to CCITT. Even though this proposed standard has not yet been approved, the industry has already implemented chips that would execute the compression and decompression JPEG algorithms [51]. In this thesis, we used the most recent

software release [version 2.3] of this algorithm, which was generously made available to the public by JPEG. We did not modify any parts of the algorithm, instead, we used it to compare results using the VQ method as modified by us to suit sonar imagery. For completeness, we will present a brief overview of the different processing steps of the JPEG coding/decoding algorithm. Figures 36 and 38 depict the block diagrams of the encoder and the decoder, respectively.

JPEG Encoding

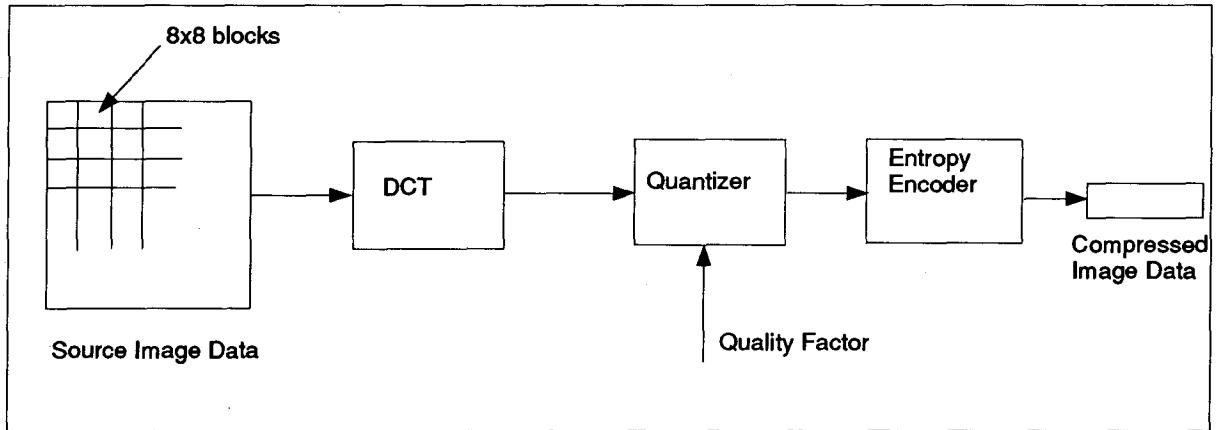


Figure 36 Block diagram of the JPEG encoder

In the encoding step, the image is first divided into 8x8 non-overlapping blocks, and the Discrete Fourier Transform is performed on each block, using the following equation:

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

where $C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$ and $f(x, y)$ is the pixel value.

The output of the DCT is a matrix of coefficients that represent the energy in 2D frequency domain. Generally, most of the energy is concentrated at the upper left corner of that matrix (in other words, the weights assigned to the low harmonics are significantly higher than those of the high harmonics). If this matrix is scanned in a zig-zag manner, by starting at the upper left corner and ending at the bottom right corner, as shown in Figure 37, then the resultant sequence will contain long strips of zeros, especially towards the end of the sequence. At this

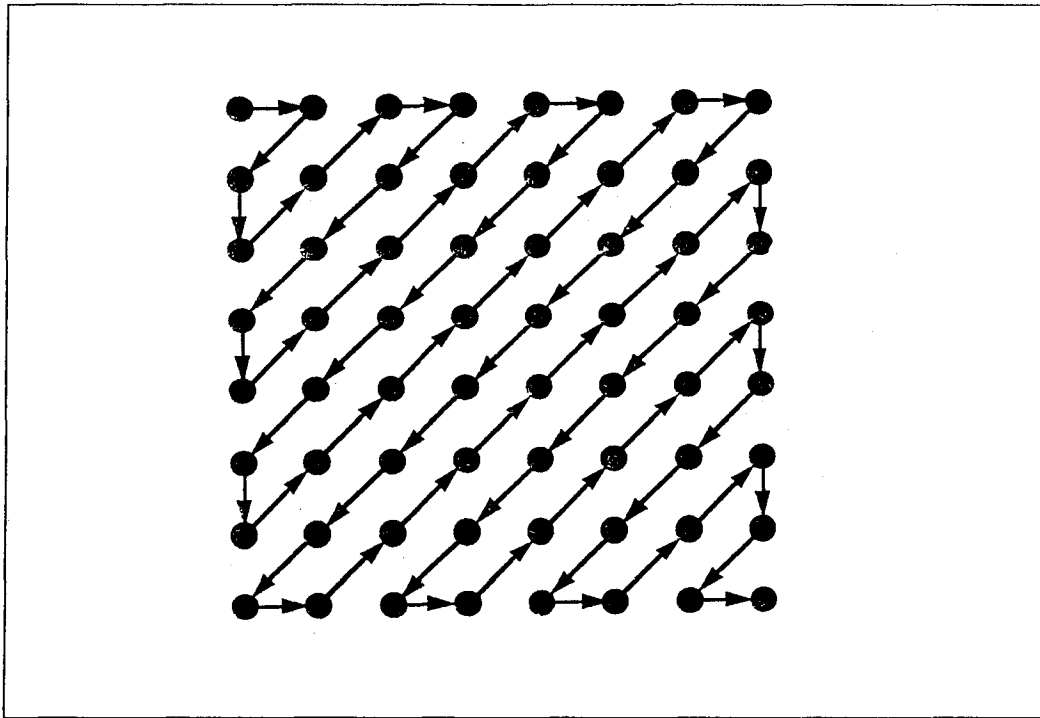


Figure 37 Zig-Zag scanning used to get long sequences of zeros

point, a quantization step is performed in order to achieve further compression

by representing the DCT coefficients with less precision, which should cause a slight degradation in the image.

The quantized resultant sequence is then represented by the non-zero terms and the number of consecutive zero terms. To increase further the compression ratio, lossless entropy coding (Huffman for example) is then used to encode the number of zero terms in every sequence of zeros. The released encoding algorithm uses two user-specified parameters. The first parameter specifies the quantization step, which translates into a quality factor ranging in values from 0 to 100, where a value of 100 represents minimal compression. The other parameter specifies which of the two available entropy coding methods —Huffman or arithmetic entropy coding— is to be used.

JPEG Decoding

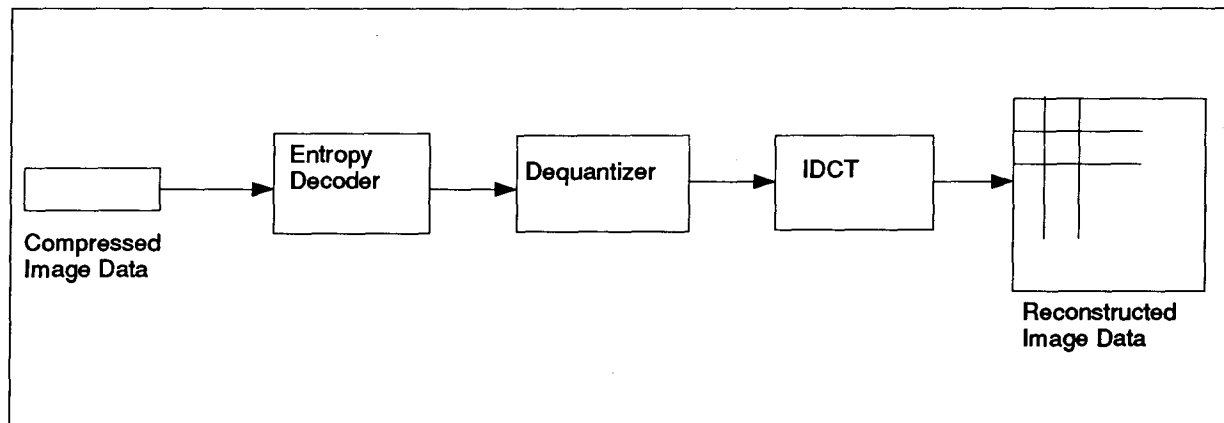


Figure 38 Block diagram of the JPEG decoder

Similarly to the encoding step, except in reverse order, entropy decoding is first performed, followed by a dequantization step, followed by the Inverse DCT:

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

where $C(u)$ and $C(v)$ are as before.

Experimental Results

As discussed before, there are two parameters that control the encoder: the quality factor parameter related to the compression ratio required, and the entropy coding algorithm parameter. We noticed that using arithmetic coding leads to lower compression ratios than Huffman coding, without introducing any visual degradation to the reconstructed images. For the compression ratio parameter, we decided upon a value that caused the reconstructed image to be minimally distorted, and therefore used a quality factor of 75. As subjective as it may seem, our main concern was to preserve *all* features. Apart from this visual “measurement”, we used the Root Mean Squared (RMS) error as a measurement of success. Also, we took into consideration the performance of the encoding and decoding stages. Tables 2 and 3 illustrate the visual measurement: the left-hand side column contains the original images, whereas the right-hand side column contains the reconstructed images, using a quality factor of 75. For a higher compression ratio, the degradation of the reconstructed image was evident, and consequently we decided not to include those results in this thesis, except for comparison purposes. Table 4 summarizes the measure of success of each image (4 images in total), along with some performance measurements.

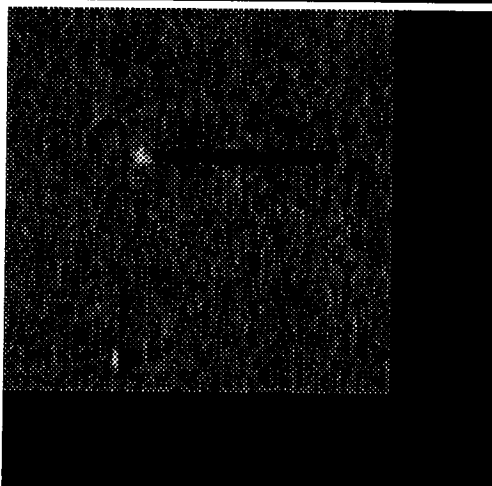
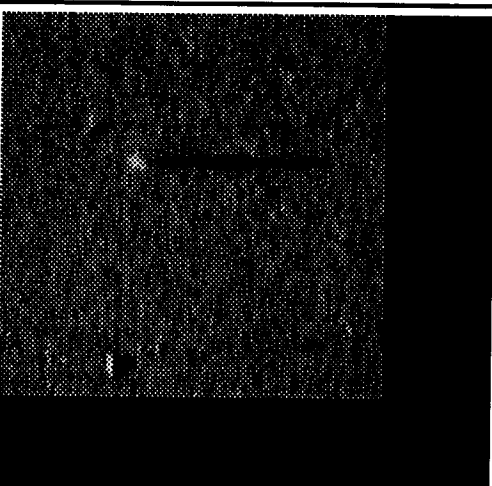
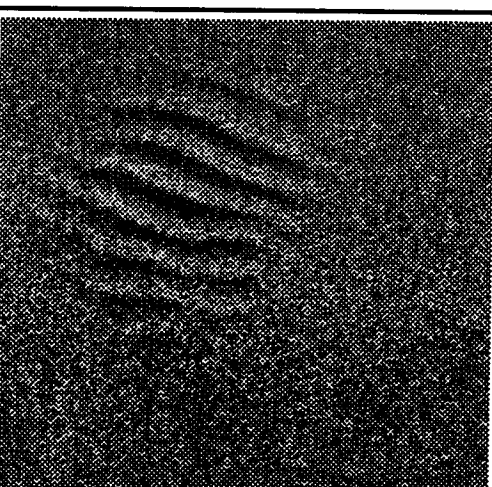
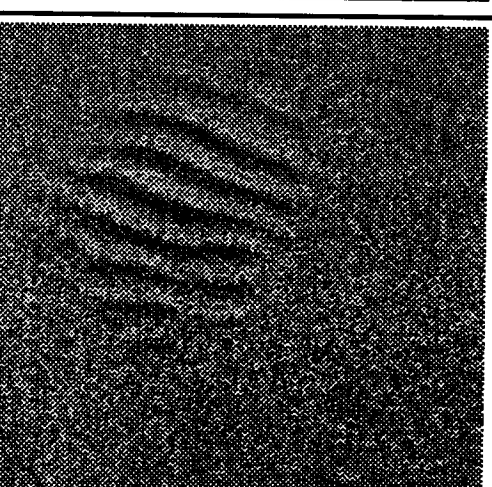
<i>Original Sonar Images</i>	<i>Reconstructed Images, using JPEG Encoding/Decoding Algorithm</i>
	
	

Table 2 Visual "measurement" using a quality factor of 75, for images 1 and 2.

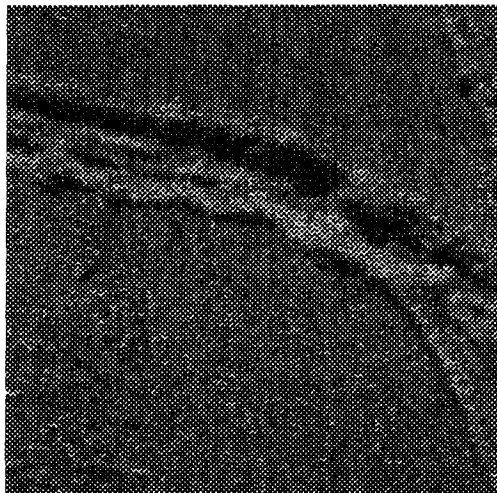

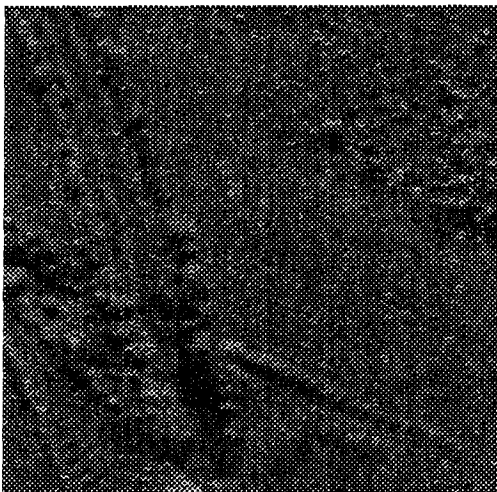

<i>Original Sonar Images</i>	<i>Reconstructed Images, using JPEG Encoding/Decoding Algorithm</i>
	
	

Table 3 Visual “measurement” using a quality factor of 75, for images 3 and 4.

<i>Original Sonar Images</i>	<i>RMS * error between original image, and recon- structed image</i>	<i>Compression ratio</i>	<i>Bits per Pixel representation</i>	<i>Encoding time</i>	<i>Decoding time</i>
<i>Image 1</i>	5.66	5.0:1	1.59 bpp	5 seconds	11 sec.
<i>Image 2</i>	9.11	3.3:1	2.37 bpp	5 seconds	12 sec.
<i>Image 3</i>	7.13	5.2:1	1.54 bpp	5 seconds	10 sec.
<i>Image 4</i>	7.23	5.0:1	1.59 bpp	5 seconds	12 sec.

Table 4 Quantitative comparison of JPEG compression algorithm.

These tests were carried out on a Sun 4/60 computer, with the following specifications:

1. 1.8 Specification Marks (SM)
2. 12 Million Instructions per Second (MIPS)
3. 1.9 Mega FLOat OPERations (MFLOPS)

* Maximum RMS Error is 256.

4 Vector Quantization

Introduction to Vector Quantization

Originally used for speech coding, the vector quantization method now dominates research in image compression. Many papers have appeared on this subject, some of them aiming at improving the quality of the reconstructed images, and others aiming at making the encoding step faster and more efficient. We found the vector quantization attractive for two reasons: the image reconstruction is faster than any other reconstruction algorithm, and the compression ratio that can be achieved by this scheme is larger. The drawback is the slow speed in encoding.

In this thesis, we are introducing a new codebook generation algorithm. This algorithm, when used by VQ, results in large compression ratios, larger than those obtained by JPEG method. Furthermore, features and their corresponding shadows are better preserved, and the blurring effect produced after JPEG reconstruction is absent in this method. In addition, information theory implies one can always obtain better performance by coding vectors, as is the case in VQ, instead of scalars [52], as is the case in JPEG.

Mathematically, the image is subdivided into 4x4 blocks, or vectors. Let each vector be represented by $\bar{x} = (x_0, x_1, \dots, x_{15})$. In the case of the sonar image that we have been analyzing, the image is 256x256 pixels, and therefore it consists of 64x64 vectors. An alphabet, or codebook, is a set of vectors that *represent* the whole image. By choosing a codebook that has less vectors than

the image itself, data compression is achieved. Consequently, encoding is simply a mapping between each image vector \bar{x} and one vector \bar{y} of the codebook. The Hamming² distances between each image vector and all codebook vectors are calculated, and the codebook vector which is closest to the image vector is chosen to represent the image vector. In other words, many image vectors will map to one codebook vector.

There are many methods for building codebooks [52], the best known being the one suggested by Linde, Buzo, and Gray (LBG algorithm) in 1980 [53]. This algorithm is set as follows:

1. *Choose a set of codebook vectors from the image vectors set. In this initial set, choose those vectors so that they are evenly distributed, and no two vectors have adjacent rows nor columns.*
2. *Assign the image vectors to cells, where the number of cells equals the number of codebook vectors required. Use Hamming distance as the criterion for finding which image vector belongs to which cell. It is eventually desired to have each vector in the codebook as the centroid of each cell.*
3. *A measure of error on this choice of codebook vectors is performed. If the error is too large, then the centroid of each cell is calculated, and the new centroids are used as a new set of codebook vectors.*
4. *This operation of calculating the centroid of each cell is performed until the measure of error is small (a user-specified threshold is used for comparison).*

² Other algebraic distances may be used as well.

It can be seen from this layout that the set of codebook vectors will be an optimal set to represent the whole image. This optimality is based on the Hamming distance. This optimality has a drawback, especially in the case of sonar images, since it treats all vectors alike, i.e. without processing feature and shadow vectors differently from the other (background) vectors. For sonar images, it is preferable to compress the feature and shadow areas minimally and the background maximally. The resulting compression ratio of this method was very encouraging, in the order of 8 to 1. However, the reconstructed image had all of its features distorted.

This prompted us to try a new way of generating codebooks, which would process feature vectors in a different way than a background vector. This new algorithm, which we called the MAS algorithm, is simple in concept, easy to implement, leads to better compression ratios than those achieved by the LBG algorithm, preserves features and their shadows, and is faster than the LBG algorithm by orders of magnitude (it takes 3 seconds to create a codebook using the MAS algorithm, whereas it takes about 3 minutes by using the LBG algorithm). The outline of the MAS algorithm is as follows:

1. *Calculate the mean intensities in all image vectors.*
2. *Order all vectors according to their increasing mean values.*
3. *Choose the codebook vectors from this order list as follows:*

- a. *Choose the first 30³ vectors, the vectors with the lowest mean values in the ordered list. These vectors should represent all the shadows contained in the image.*
- b. *Choose the last 30 vectors, the vectors with the highest mean values in the ordered list. These vectors should represent all features contained in the image.*
- c. *Choose the rest of the codebook vectors from the ordered list, so that these vectors are evenly distributed in that list. These vectors should represent the background surrounding features and their corresponding shadows.*

Assuming that the original image is 320x320 pixels, and therefore 80x80 vectors, and if the codebook chosen consists of 256 vectors (30 vectors with lowest mean values, 30 vectors with highest mean values, and 196 vectors chosen evenly from the ordered vectors set), then the achieved compression ratio would be 10:1.

$$CR = \frac{80 * 80(\text{vectors}) * 16(\text{bytes/vector})}{256(\text{codebook vectors}) * 16(\text{bytes/vectors}) + [80 * 80(\text{lookup addresses})]} = 9.76$$

We used the same measure of success criteria to judge the reconstructed images using VQ, as we did for the JPEG method. The images are presented in two columns. The left column contains the original sonar images, and the right column contains the reconstructed images using the vector quantization method.

³ Choosing any value that is greater than 20 would lead to same results. A value of 30 was chosen to increase its sensitivity on a particular threshold.

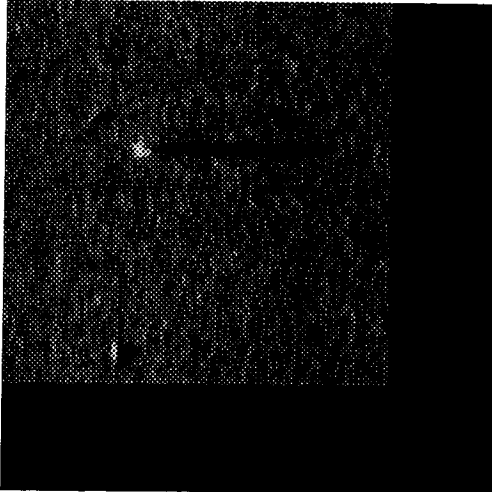
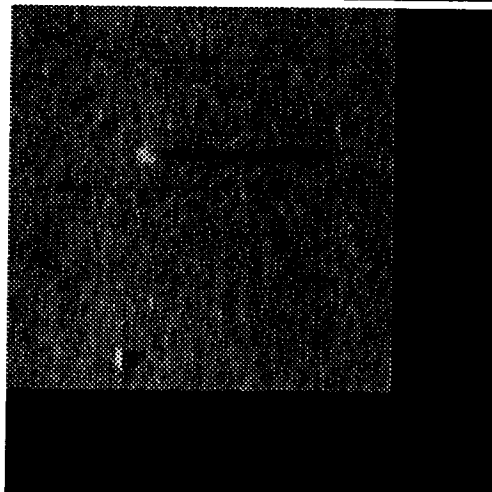
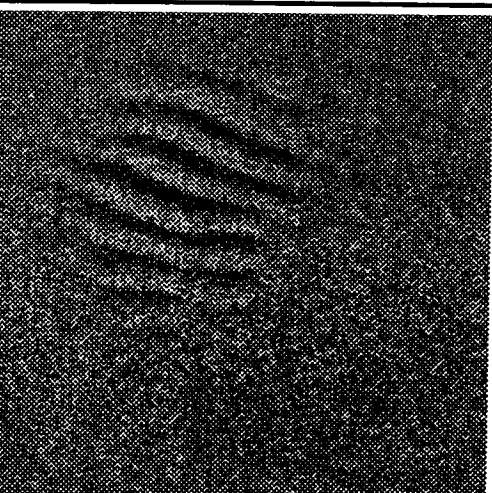
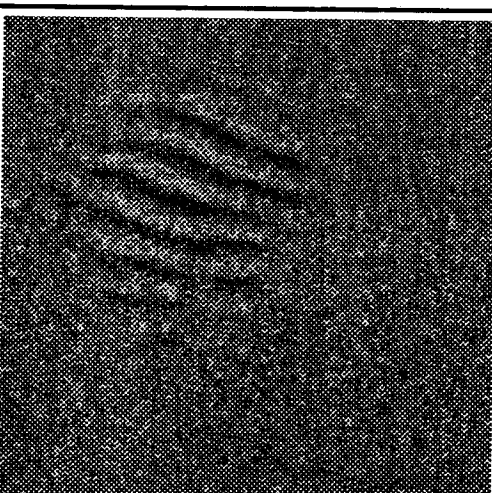
<i>Original Images</i>	<i>Reconstructed Images using VQ</i>
	
	

Table 5 Results of VQ method for images 1 and 2.



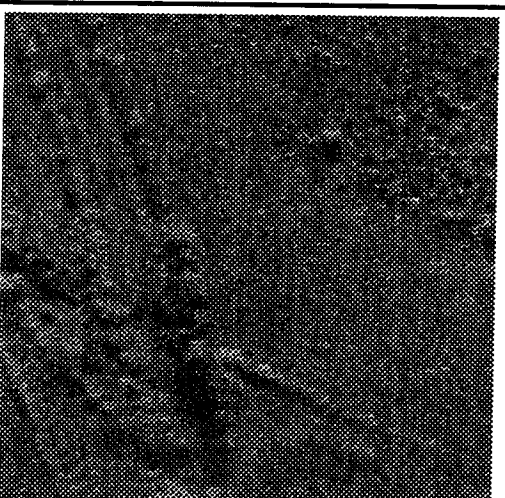
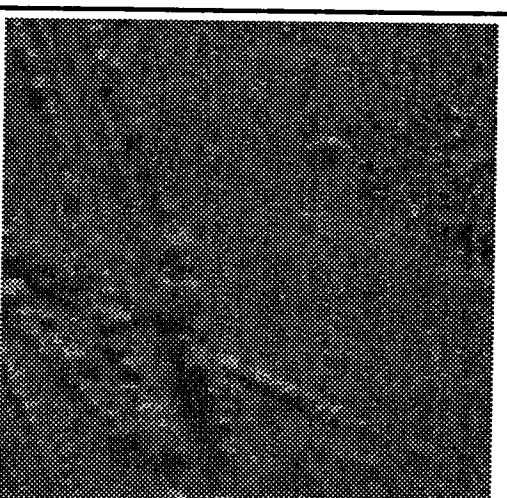
<i>Original Images</i>	<i>Reconstructed Images using VQ</i>
	
	

Table 6 Results of VQ method for images 3 and 4.

<i>Original Sonar Images</i>	<i>RMS error between original and reconstructed</i>	<i>Compression Ratio</i>	<i>Bits per pixels</i>	<i>Coding time in seconds</i>	<i>Decoding time in seconds</i>
<i>Image 1</i>	8.96	10:1	0.8	54	1
<i>Image 2</i>	21.72	10:1	0.8	48	1
<i>Image 3</i>	13.26	10:1	0.8	50	1
<i>Image 4</i>	12.89	10:1	0.8	51	1

Table 7 Quantitative comparison of modified VQ compression algorithm.

5 Summary of Compression Results

The first method we tried, the adaptive BTC, gave very poor results. Much more encouraging results were achieved by using the JPEG encoding/decoding algorithm. Depending on a user-specified quality factor, the compression ratio ranged from 5:1 to about 3:1. The variation of compression ratios, with the quality factor set at a constant value of 75, was caused by the texture difference of the images considered. The last compression algorithm that we considered was based on Vector Quantization, with a new codebook generation introduced. This codebook generation gave good quality reconstructed images.

It is difficult to be objective when comparing the JPEG and the VQ methods. Both methods have their advantages and disadvantages. We do not believe that one method outperforms the other in the absolute sense. The Mean Squared Error is not a very good measure for comparison, since the VQ method tries to compress the sonar images, without compressing the features contained in that image. Therefore, the features are much better preserved in the reconstructed image after a VQ compression, whereas the background is not so well preserved. The MSE of the reconstructed features is almost zero in the case of VQ and is much larger in the JPEG case. It would be safe to say that the VQ is tailored to sonar images, whereas the JPEG is in fact a generalized compression/decompression algorithm that fits its international standardization status very well.

Figures 39, 40, and 41 illustrate our claim. At this point, it is noteworthy to point out that Figure 40 was reconstructed from a 5:1 compressed image, whereas Figure 41 was reconstructed from a 10:1 compressed image.

In order to compare JPEG and VQ more objectively such that both images have the same compression ratios, the JPEG method was used with a quality factor of 30, resulting in a 10:1 JPEG compression ratio. Figure 41 illustrates the reconstructed image after JPEG compression with a quality factor of 30. Now, Comparing Figures 42 and 43 shows the obvious advantage of the VQ compression method. In the latter case, both the compression ratio and the reconstructed image quality make it a more favorable compression method. Reconstruction after JPEG compression cuts off high spatial frequencies, causing the resulting images to be blurred. This blurring effect is accompanied with a loss of edges that surround suspect mines and could potentially remove small features altogether, an event that is unacceptable for such a sophisticated system.

Another useful measure of reconstructed image quality is the difference image. In a difference image, the reconstructed image is subtracted from the original image, and a value of 128 is added to the difference in order to highlight the discrepancies. Figures 44 and 45 illustrate the difference images of a JPEG (FIG. 40) and VQ (FIG. 41). In Figure 44, it is seen that the JPEG algorithm treats all regions equally. The degradation of feature and shadow regions is equivalent to the degradation of the background regions.

For the VQ, it can be seen that feature and shadow areas are much better

preserved than those in the JPEG reconstructed images. In those regions, the value of the difference is zero, as desired. This proves that the VQ compression method meets the requirements in that feature and shadow regions are 100% preserved.

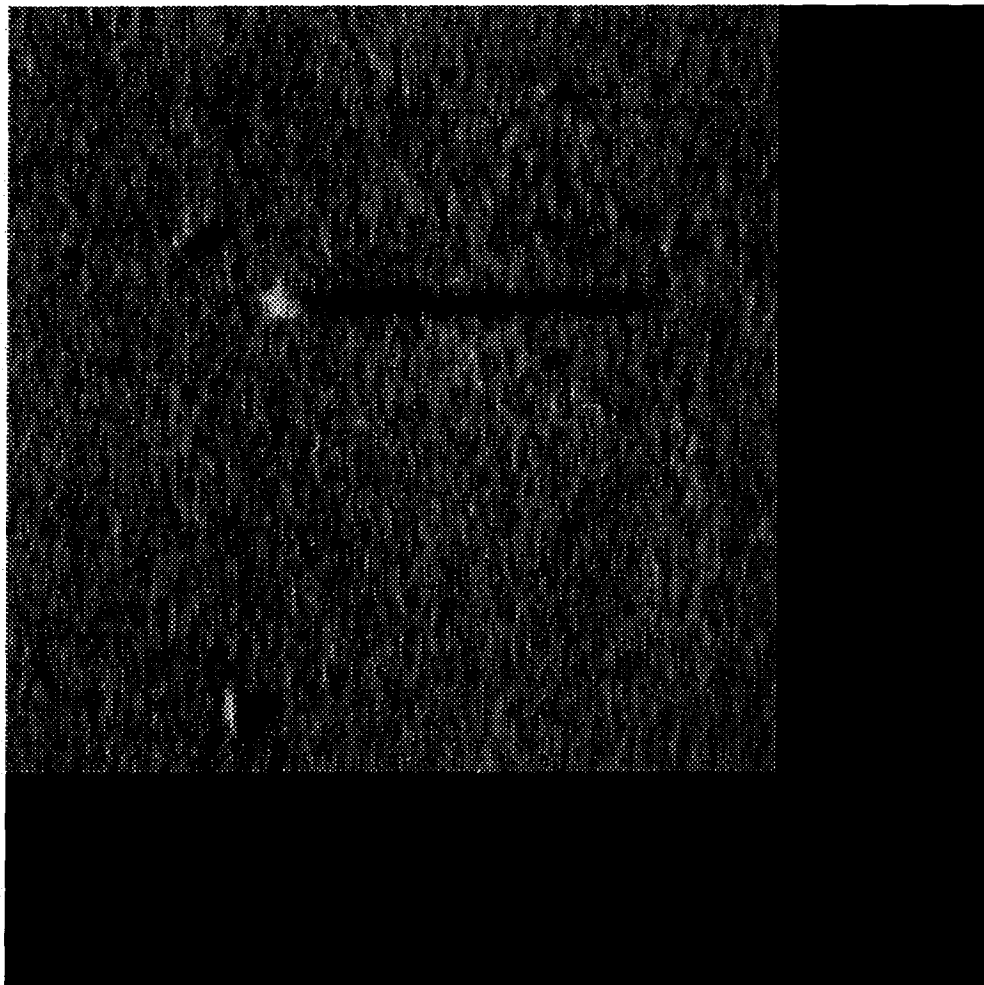


Figure 39 The zoomed original image (Zooming factor = 2)

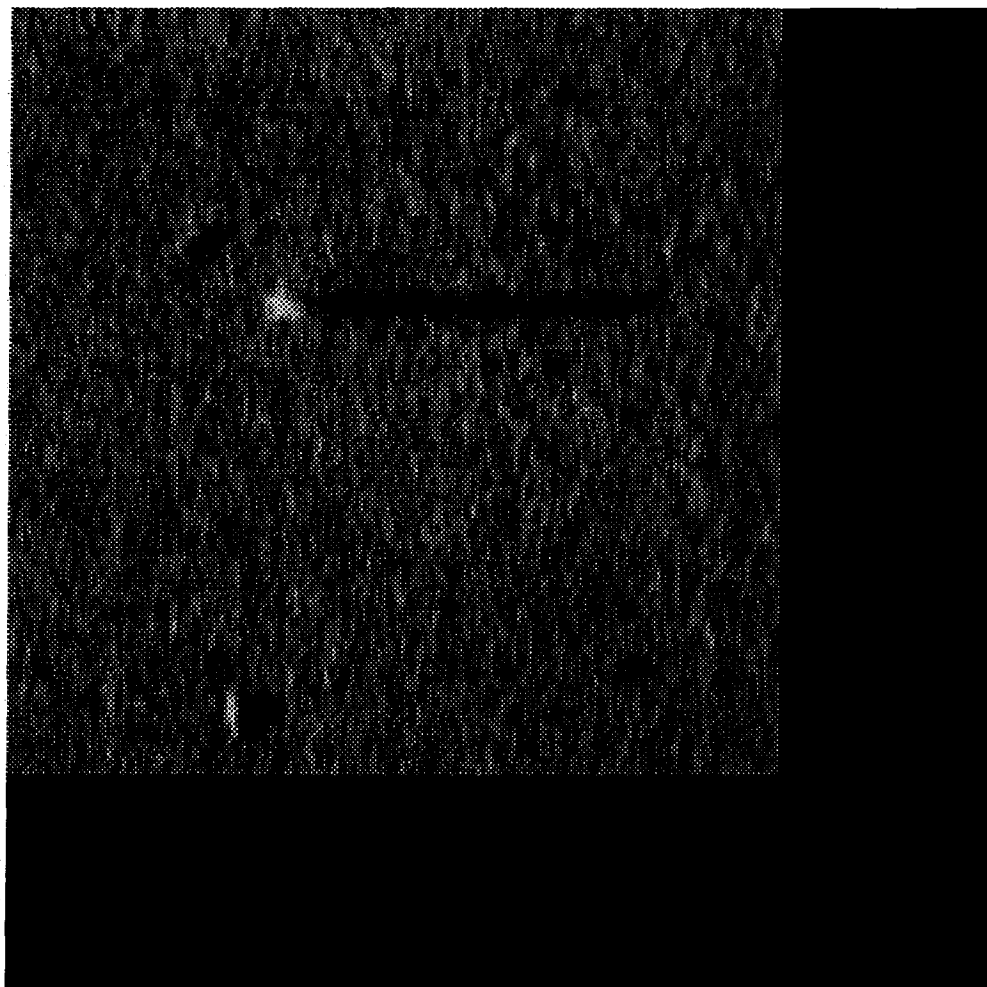


Figure 40 The zoomed reconstructed image after JPEG compression, with 5:1 compression ratio.

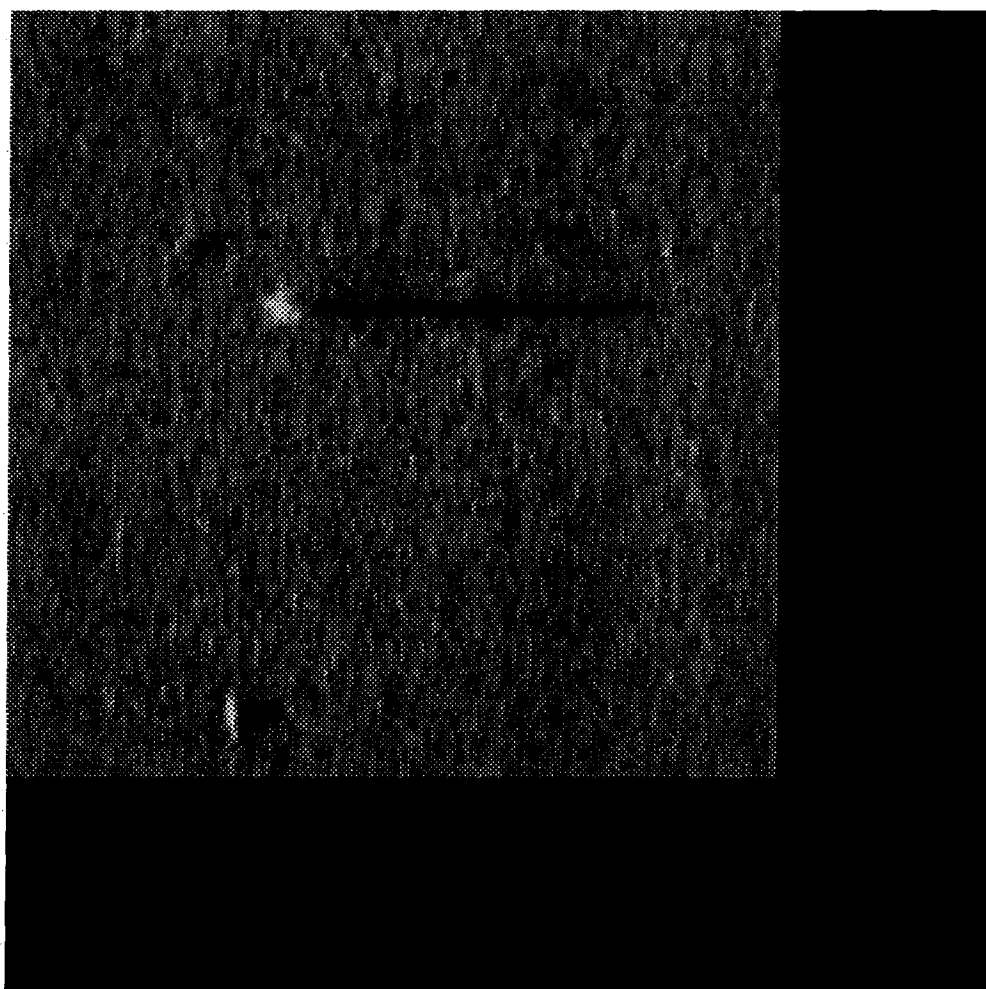


Figure 41 The zoomed reconstructed image after VQ compression, with 10:1 compression ratio.

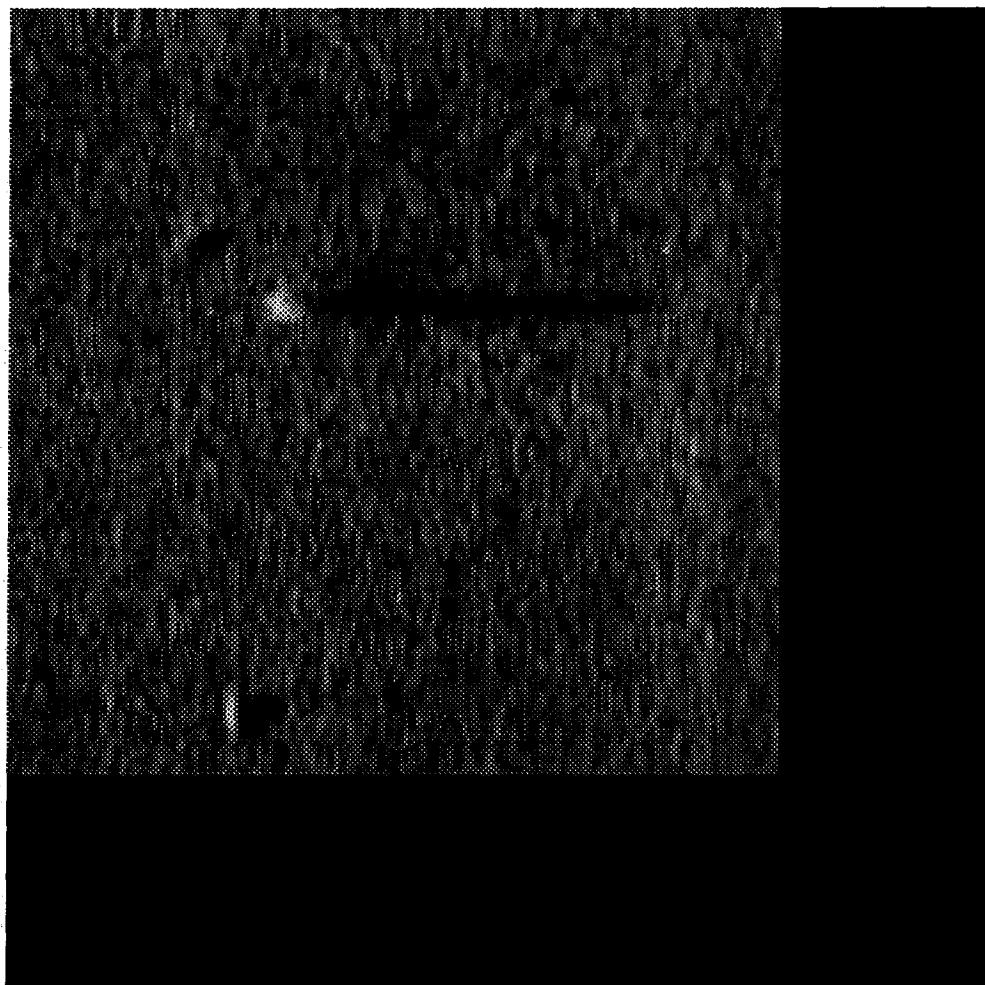


Figure 42 The zoomed reconstructed image after JPEG
compression with quality factor of 30 (10:1 compression)

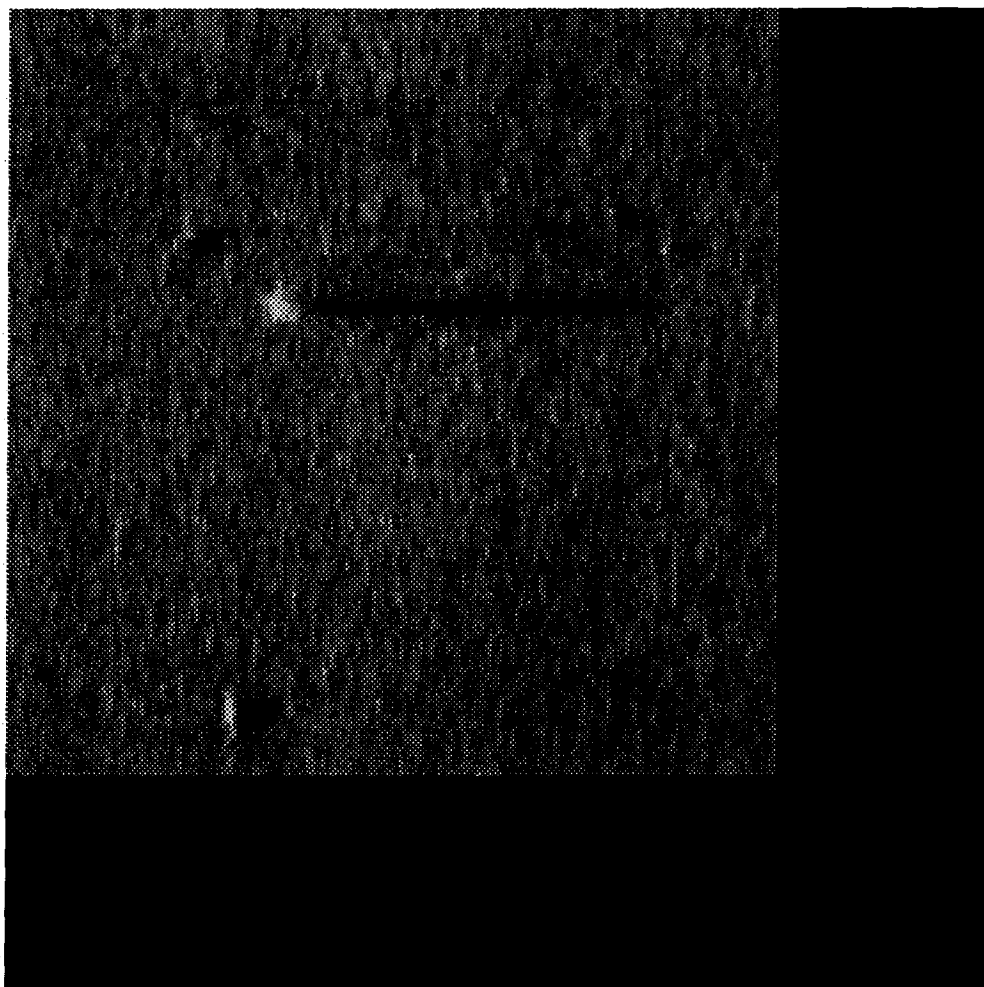


Figure 43 The zoomed reconstructed image after VQ compression (10:1 compression)

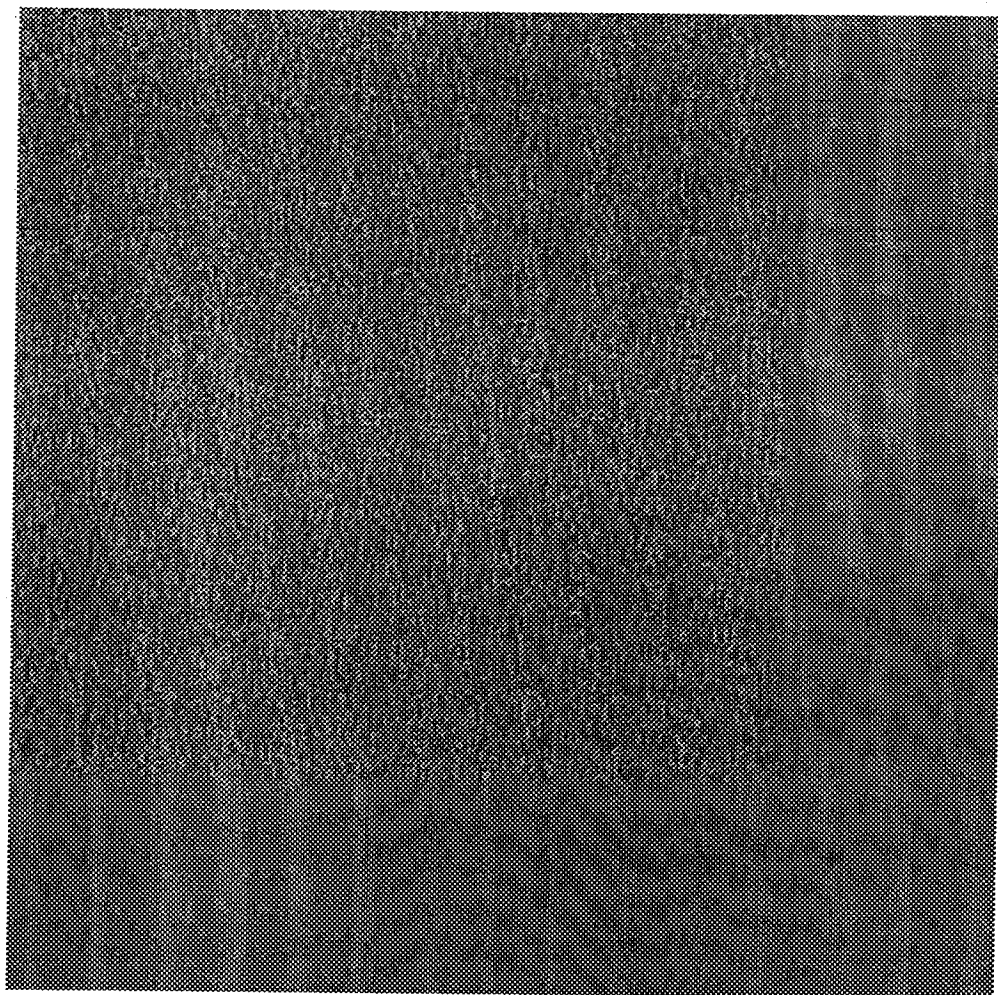


Figure 44 Difference image after JPEG compression

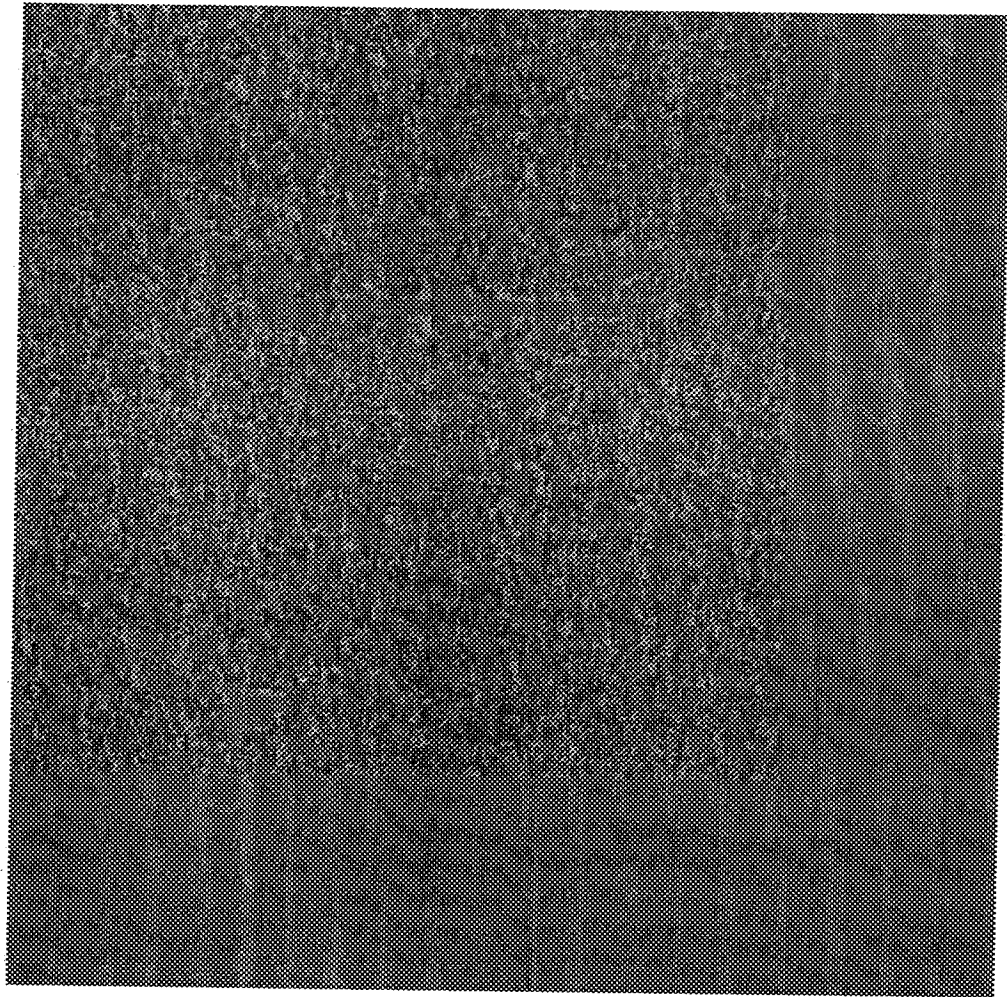


Figure 45 Difference image after VQ compression

6 Conclusions

1 Original Contribution

For the pre-processing step of image enhancement, we developed the Morphological Cascaded Smoothing Filter which was found to enhance the sonar imagery by suppressing noise and sharpening edges that surround features and shadows. This enhancement was done gradually, in order to preserve feature pixels.

Even though no original work was presented in the edge detection stage, an ATM edge detector was found to suit sonar imagery. The other three edge detectors that were considered had major drawbacks, making them unsuitable for sonar images. The success of this edge detector was contributed to its immunity to high levels of noise and to the smoothing effect caused by the Morphological Cascaded filter.

For the final stage, image compression and storage, a new codebook generation algorithm was introduced in association with the VQ coding. The VQ with our codebook led to a very low bit rate of 0.8 bpp, where features and shadows were very well preserved after reconstruction. Results of this compression method were compared to an internationally proposed compression algorithm, the JPEG algorithm, and it was shown that *in the case of sonar imagery* the VQ coding algorithm gives better reconstructed image feature quality and lower bit rates.

2 Suggestions for Future Work

The next logical step would be to investigate the performance of the proposed compression algorithm by using an efficient methodology, such as quad-trees [60]. Recent research work in Vector Quantization has addressed different ways of making this compression/decompression methodology more efficient [54, 55, 57].

Having investigated the performance of this algorithm, a hardware implementation would constitute a very interesting and challenging research topic.

Another suggested topic would be to investigate the reusability of codebooks in order to achieve higher compression ratios. The calculation of the compression ratio took into consideration the codebook size. It may not be necessary to store a codebook for each image, a simplification that would lead to a 16:1 compression ratio. Adjacent images may be represented by one codebook, preventing the unnecessary storage of a separate codebook for each image.

A REFERENCES

1. P.A Maragos, *A Unified Theory of Translation Invariant Systems with Applications to Morphological Analysis and Coding of Images*, Ph.D. Thesis, Georgia Institute of Technology, July 1985.
2. W.K. Pratt, *Digital Image Processing*, J. Wiley and Sons, New York, 1978.
3. A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
4. A Rosenfeld and A.C. Kak, *Digital Picture Processing*, Academic Press, New York, 1976.
5. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.
6. C.R Giardina and E.R Dougherty, *Morphological Methods in Image and Signal Processing*, Prentice Hall, New Jersey, 1988.
7. R.J. Feehs and G.R Arce, *Multidimensional Edge Detection*, SPIE Vol. 845, 1987, pp. 285–292.
8. S.J. Lee, R.M. Haralick, and G. Shapiro, *Morphologic Edge Detection*, Machine Vision International, Ann Arbor, Michigan 84104, Aug. 1986.
9. J.F. Bronskill, *Multidimensional Shape Description and Recognition Using Mathematical Morphology*, Master Thesis, University of Toronto, July 1986.
10. Z. Zhou, *Morphological Skeleton Representation and Shape Recognition*, Mater Thesis, University of Toronto, July 1988.

11. M. Jelavic and T. Whalen, *Morphology Changes the Shape of Machine Vision*, The Electronic System Design Magazine, November 1987, pp 67–75.
12. P. Maragos and R. Schafer, *Morphological Systems for Multidimensional Signal Processing*, Proceedings of the IEEE, Vol. 78, No. 4, April 1990.
13. T.R. Esselman and J.G. Verly, *Feature Extraction from Range Imagery Using Mathematical Morphology*, SPIE Vol. 845, Visual Communications and Image Processing II, 1987, pp 233–240.
14. S.R. Sternberg, *Biomedical Image Processing*, IEEE Computer, Jan. 1983, pp 22–34.
15. S.R. Sternberg, *Grayscale Morphology*, Computer Vision, Graphics Image Processing, Vol. 35, 1986, pp 333–355.
16. M.K. Smayra, *The Effect of Structuring Elements on Shape Description and Recognition*, Bachelor Thesis, University of Toronto, April 1989.
17. M. Suk and S. Hong, *An Edge Extraction Technique for Noisy Images*, Computer Vision, Graphics and Image Processing, 25, 1984, pp 24–45.
18. Mashua and Gilbert, *Finding Edges in Noisy Scenes*,
19. V. Nalwa and T. Binford, *On Detecting Edges*, IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, Nov. 1986, pp 699–714.
20. W.H. Lunscher and M.P. Beddoes, *Optimal Edge Detector Evaluation*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-16, No 2, March 1986, pp 304–312.

21. G.F. McLean and M.E. Jernigan, *Hierarchical Edge Detection*, Computer Vision, Graphics, and Image Processing 44, 1988, pp 350–366.
22. A. Rosenfeld, *The Max Roberts Operator is a Hueckel-Type Edge Detector*, IEEE PAMI, Vol. 3, No 1, Jan. 1981, pp 101–111.
23. Suk and T. Cho, *Object Detection Algorithm Based on Region-Adjacency Graph*, Proceedings of the IEEE, Vol. 72, No 7, July 1984, pp 985–986.
24. W. Grimson and E.C. Hildreth, *Comments on "Digital Step Edges from Zero Crossings of the Second Directional Derivatives"*, IEEE PAMI, Vol. 7, No 1, Jan. 1985, pp 121–199.
25. A.N. Venetsanopoulos, *Edge Detectors based on Nonlinear Filters*, IEEE PAMI-8, No 4, July 1986.
26. J.S. Chen and G. Medioni, *Detection, Localization, and Estimation of Edges*, IEEE PAMI, Vol. 11, No. 2, Feb. 1989, pp 191–198.
27. J. Canny, *A computational Approach to Edge Detection*, IEEE PAMI, Vol. 8, No 6, Nov 1986, pp 679–699.
28. V. Torre and T. Poggio, *On Edge Detection*, IEEE PAMI, Vol. 8, No 2, March 1986, pp 147–163.
29. E. Hildreth, *Edge Detection*, The Encyclopedia of Artificial Intelligence, J. Wiley and Sons, New York, 1987.
30. D. Marr and E. Hildreth, *Theory of Edge Detection*, Technical Report AI Memo 518, MIT AI Lab, April 1979, pp 187–217.

31. R.M. Haralick, *Digital Step Edges from Zero Crossing of Second Directional Derivatives*, IEEE PAMI, Vol. 6, No 1, Jan. 1984, pp 58–68.
32. R. Nevatia, *Linear Feature Extraction*, Technical Report of Image Processing Institute, University of Southern California, pp 73–76.
33. K. Sijmons, *Computer-assisted Detection of Linear Features from Digital Remote Sensing Data*, ITC Journal, Vol. 1, 1987.
34. A. Bovik, *On Detecting Edges in Speckle Imagery*, IEEE Transactions on ASSP, Vol. 36, No 10, Oct. 1988, pp 1618–1627.
35. Y. Leclerc and P. Fua, *Finding Object Boundaries Using Guided Gradient Ascent*, Technical Report AI centre, SRI International, 1984.
36. V. Nalwa, *On Detecting Edges*, Technical Report AI Lab, Stanford University, pp 157–164.
37. A. Huertas and G. Medioni, *Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks*, IEEE PAMI, Vol. 8, No 5, Sep. 1986, pp 651–663.
38. W.H.H.J Lunscher, *The Asymptotic Optimal Frequency Domain Filter for Edge Detection*, IEEE PAMI, Vol. 5, No 6, Nov. 1983, 678–680.
39. R. Nevatia and K. Babu, *Linear Feature Extraction and Description*, Computer Graphics and Image Processing 13, 1980, pp. 257–269.
40. J. Canning, J. Kim, and A. Rosenfeld, *Symbolic Pixel Labeling for Curvilinear Feature Detection*, Image Understanding Workshop, Feb. 1987, pp 242–256.

41. S. Zucker and R. Hummel, *A Three-Dimensional Edge Operator*, IEEE PAMI, Vol. 3, No 3, May 1981, pp 324–331.
42. Bednar and Watt, *Alpha-Trimmed Means and Their Relationship to Median Filters*, IEEE Transactions on ASSP, Vol 32, No 1, Feb. 1984, pp 145–153.
43. R. Stevenson and G. Arce, *Theoretical Analysis of Morphological Filters*, 24th Annual Allerton Conference on Communications, Control, and Computing, Oct. 1986.
44. M.L. Liou, *Visual Telephony as an ISDN Application*, IEEE Communications Magazine, Feb. 1990, pp 30–38.
45. A. Wong, C. Chen, D.L. Le Gall, F. Jeng, and K. Uz, *A video Coding Algorithm for Transmission and Storage Applications*, IEEE Communications Magazine, Nov. 1990, pp 24–31.
46. G.K. Wallace, *The JPEG Still Picture Compression Standard*, Communications of the ACM, Vol. 34, No 4, April 1991, pp 30–45.
47. E.J. Delp and O.R. Mitchell, *Image Compression Using Block Truncation Coding*, IEEE Transactions on Communications, Vol. 27, No 9, Sep. 1979, pp 1335–1342.
48. D.R. Halverson and N.C. Griswold, *A Generalized Block Truncation Coding Algorithm for Image Compression*, IEEE Transactions on ASSP, Vol. 32, No 3, June 1984, pp 664–668.
49. D.R. Halverson, *On The Implementation of Block Truncation Coding Algorithm*, IEEE Transactions on Communications, Vol. 30, No 11, Nov. 1982.

50. P. Nasiopoulos, R. Ward and D. Morse, *Adaptive Compression Coding*, IEEE Transactions on Communications, Vol. 39, No 8, Aug. 1991, pp 1245–1254.
51. M. Leonard, *IC Executes Still-Picture Compression Algorithms*, Electronic Design, May 1991, pp 49–53.
52. R.M. Gray, *Vector Quantization*, IEEE ASSP Magazine, Vol. 1, No 2, April 1984, pp 4–29.
53. Y. Linde, A. Buzo, and R.M. Gray, *An Algorithm for Vector Quantizer Design*, IEEE Transactions on Communications, Vol. 28, No 1, Jan. 1980, pp 84–95.
54. D.J. Vaisey and A. Gersho, *Variable Block-Size Image Coding*, Proc. ICASSP, 1987.
55. Y. Ho and A. Gersho, *Variable-Rate Multi-Stage Vector Quantization For Image Coding*, Proc. ICASSP, 1988, pp 1156–1159.
56. Z. Zhou and A.N. Venetsanopoulos, *Directional Decomposition of Morphological Pyramid Representations for Low Bit-Rate Image Coding*, Conference on Digital Signal Processing, 1991, pp 404–409.
57. P. Yu and A.N. Venetsanopoulos, *Hierarchical Multirate Vector Quantization for Image Coding*, Canadian Conference on Electrical and Computer Engineering, Sep. 1991.
58. L. Sutro and J. Lerman, *Robot Vision*, internal report, CSD Laboratory, Cambridge, Massachusetts, April 1973.
59. J. Verly and T Esselman. *Some Applications of Mathematical Morphology to Range Imagery*, Electronic Imaging'88, Boston, Oct.1988.

60. H. Samet, *The Quad Tree and Related Hierarchical Data Structures*, ACM Computing Surveys, Vol.16, No 2, June 1984, pp 187–260.