

AD-A248 389



TECHNICAL REPORT EL-92-17

(2)

US Army Corps
of Engineers

ADAPTIVE COMPRESSION OF MULTISENSOR IMAGE DATA

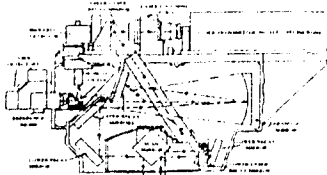
by

Ernesto R. Cespedes

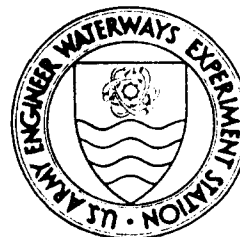
Environmental Laboratory

DEPARTMENT OF THE ARMY

Waterways Experiment Station, Corps of Engineers
3909 Halls Ferry Road, Vicksburg, Mississippi 39180-6199



DTIC
ELECTE
APR 10 1992
S B D



March 1992

Final Report

Approved For Public Release; Distribution Is Unlimited

92-09069



Prepared for DEPARTMENT OF THE ARMY
Assistant Secretary of the Army (R&D)
Washington, DC 20315

Under Work Unit A91D-LR-007

92 4 08 040



Destroy this report when no longer needed. Do not return it
to the originator.

The findings in this report are not to be construed as an
official Department of the Army position unless so
designated by other authorized documents.

The contents of this report are not to be used for
advertising, publication, or promotional purposes.
Citation of trade names does not constitute an
official endorsement or approval of the use
of such commercial products.

Preparation of this report was sponsored by the Department of the Army
In-House Laboratory Independent Research Program. Neither the
Department of the Army, nor the U.S. Army Corps of Engineers, nor
their employees makes any warranty, expressed or implied, or
assumes any legal liability or responsibility for the accuracy,
completeness, or usefulness of any information, apparatus,
product, or process disclosed, or represents that its use
would not infringe privately owned rights.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 1992		3. REPORT TYPE AND DATES COVERED Final report
4. TITLE AND SUBTITLE Adaptive Compression of Multisensor Image Data			5. FUNDING NUMBERS A91D-LR-007 (ILIR)	
6. AUTHOR(S) Ernesto R. Cespedes				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Waterways Experiment Station, Environmental Laboratory, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199			8. PERFORMING ORGANIZATION REPORT NUMBER Technical Report EL-92-17	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Corps of Engineers, Assistant Secretary of the Army (R&D), Washington, DC 20315			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The objective of this study is to develop and test a methodology for the design of multisensor image compression systems. Data collected from an airborne active (laser) and passive (thermal infrared) imaging system are analyzed, and an efficient adaptive transform coding method is developed to exploit the characteristics of the source. A number of adaptive techniques that compensate for the nonstationary nature of the source are evaluated, and a novel model-based bit allocation and quantization strategy is adopted. The compression/reconstruction scheme is implemented in a massively parallel processor, and a large number of multisensor images are processed in order to define the rate-distortion performance of the scheme. The utility of the reconstructed imagery is also evaluated by examining the performance of multichannel target detection algorithms as a function of bit rate.				
14. SUBJECT TERMS Active/passive scanner Data compression			15. NUMBER OF PAGES 204	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT
20. LIMITATION OF ABSTRACT				

PREFACE

The study reported herein was performed by personnel of the U.S. Army Engineer Waterways Experiment Station (WES) under the In-House Laboratory Independent Research (ILIR) program, Work Unit A91D-LR-007, entitled "Adaptive Compression of Multisensor Image Data."

The work was performed and this report prepared by Dr. Ernesto R. Cespedes, Acting Chief, Battlefield Environment Group, WES. Results of the study were also published as a dissertation for Mississippi State University under the direction of Dr. John K. Owens of the Electrical Engineering Department.

The study was conducted under the general supervision of Dr. Victor C. Barber, Acting Chief, Environmental Systems Division, and Dr. John Harrison, Chief, Environmental Laboratory.

Dr. Robert W. Whalin was Director of WES. COL Leonard G. Hassell, EN, was Commander and Deputy Director.

This report should be cited as follows:

Cespedes, Ernesto R. 1992. Adaptive compression of multisensor image data. Technical Report EL-92-17. Vicksburg, MS: U.S. Army Engineer Waterways Experiment Station.



i/11

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ACKNOWLEDGMENTS

I wish to thank the members of my committee for their participation and active encouragement during my extended period of graduate work. Special thanks go to my major professor, Dr. John K. Owens, for his technical advice and for guiding me through the many obstacles of a Ph.D. program. I am indebted to Dr. Robert J. Moorhead for his interest in this work and his helpful suggestions.

I would like to thank my colleagues at the U.S. Army Engineer Waterways Experiment Station (WES), especially Dr. Daniel Cress and the late Dr. Roger Sumner, whose dedication to scientific research, wisdom, and wit inspired me to pursue this course of study, and Mr. Ricky Goodson, whose expertise and advice on high-performance computing topics were invaluable.

I am grateful to my friends and family members whose constant inquiries on the progress of this dissertation hastened its completion.

This work was partly funded by WES under the In-House Laboratory Independent Research Program. The funding support and the use of the excellent computing facilities are greatly appreciated.

CONTENTS

	Page
PREFACE	i
ACKNOWLEDGMENTS	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER	
I. INTRODUCTION	1
Background	1
Dissertation Outline	3
II. FUNDAMENTALS OF IMAGE DATA COMPRESSION	6
Preliminary Concepts and Definitions	6
Image Source Modeling Considerations	10
Lossless Compression Techniques	12
Lossy Compression Techniques	13
Predictive Methods	14
Transform Methods	16
Other Compression Methods	21
III. MULTISENSOR IMAGING SYSTEM DESCRIPTION	25
Overview	25
Sensor Description	30
Physics of Polarization Imaging	33
Image Procesing Equipment	40

CHAPTER	Page
IV. ANALYSIS OF MULTISENSOR IMAGERY	42
Introduction	42
Interchannel Correlation Analysis	43
Spatial (Data) Domain Analysis	52
Spatial Frequency (Spectral) Domain Analysis	55
Source Model Selection	66
Source Model Verification	73
V. DEVELOPMENT OF MULTISENSOR IMAGE COMPRESSION SCHEME	84
Introduction	84
Selection of Transform Type	84
Bit Allocation Strategy	89
Quantization of Transform Coefficients	90
Adaptivity to Nonstationary Statistics	98
Adaptive Transform Coding Algorithm Description	106
VI. IMPLEMENTATION AND TESTING	110
Introduction	110
DCT Implementation	110
Adaptivity Implementation	116
VII. ANALYSIS OF RESULTS	123
Introduction	123
Rate-Distortion Performance	123
Utility of Reconstructed Imagery	136
VIII. CONCLUSIONS AND RECOMMENDATIONS	141
Results	141
Future Work	143
APPENDIX	
A. REMOVAL OF COHERENT LASER POWER VARIATIONS . .	146
C. IMAGE COMPRESSION ALGORITHM SOURCE CODE	164
REFERENCES	187

LIST OF TABLES

Table

1. Interchannel Correlation Coefficients	45
2. Statistics of P and C Imagery	49
3. Image Statistics After Rotating by Equation (4.3)	51
4. Distribution of DCT Coefficient Variances - P' Channel	75
5. Distribution of DCT Coefficient Variances - C' Channel	76
6. Distribution of DCT Coefficient Variances - Thermal Channel	77
7. Mean Mismatch Ratios	78
8. Bit Allocation Matrix - P' Channel	81
9. Bit Allocation Matrix - C' Channel	82
10. Bit Allocation Matrix - Thermal Channel	83
11. W Matrices for Three Types of Transforms	87
12. Transform Gain For Three Types of Transforms	88

LIST OF FIGURES

Figure	Page
1. DPCM Structure	15
2. Transform Coding Structure	17
3. Typical Application of Multisensor Imagery	28
4. Cutaway View of Multisensor Imaging System	31
5. Possible Backscattering Paths	34
6. Multiple Scattering Within a Valley of a Rough Surface	35
7. Multiple Scattering of Linearly Polarized Light	37
8. Example of Multisensor Image Data	39
9. Block Diagram of Image Processing Hardware	41
10. Plot of P and C Image Data	46
11. Principal Components Transformation	48
12. P' Channel Imagery	58
13. C' Channel Imagery	59
14. Thermal Imagery	60
15. Two Dimensional Power Spectrum of P' Imagery	61
16. Two Dimensional Power Spectrum of C' Imagery	62
17. Two Dimensional Power Spectrum of Thermal Imagery	63

18. Mandril Imagery (Red Channel)	64
19. Two Dimensional Power Spectrum of Mandril Imagery	65
20. Reconstructed P' Imagery Using Laplacian Non-Uniform Quantizers	95
21. Reconstructed P' Imagery Using Gaussian Non-Uniform Quantizers	96
22. Partially Adaptive Compression and Reconstruction of P' Imagery.	101
23. Partially Adaptive Compression and Reconstruction of C' Imagery	102
24. Partially Adaptive Compression and Reconstruction of Thermal Imagery . .	103
25. Partially Adaptive Compression and Reconstruction of Mandril Imagery . . .	104
26. Compression Stage of Adaptive Transform Coding Algorithm	108
27. Reconstruction Stage of Adaptive Transform Coding Algorithm	109
28. Block Diagram of Hou's Algorithm (Decimation-in-Frequency)	114
29. Recursive Computation of DCT for $N = 8$	115
30. P' Channel Image Coded at 0.8 Bits Per Pixel	120
31. C' Channel Image Coded at 0.8 Bits Per Pixel	121
32. Thermal Image Coded at 0.8 Bits Per Pixel	122
33. Rate Distortion Performance for P Channel Data	125
34. Rate Distortion Performance for C Channel Data	126
35. Rate Distortion Performance for Thermal Channel Data	126
36. Original P Channel Imagery	127
37. Original C Channel Imagery	128
38. Original Thermal Channel Imagery	129
39. P Channel Imagery Compressed at 0.55 Bits Per Pixel	130

40. C Channel Imagery Compressed at 0.55 Bits Per Pixel	131
41. Thermal Channel Imagery Compressed at 0.55 Bits Per Pixel	132
42. P Channel Imagery Compressed at 0.80 Bits Per Pixel	133
43. C Channel Imagery Compressed at 0.80 Bits Per Pixel	134
44. Thermal Channel Imagery Compressed at 0.80 Bits Per Pixel	135
45. Target Cueing Performance - Original Imagery	137
46. Target Cueing Performance - Image Coded at 0.50 Bits Per Pixel	138
47. Small Targets in Background Clutter - Original Thermal Imagery	139
48. Small Targets in Background Clutter - Thermal Imagery at 0.80 Bits Per Pixel	140
49. Unprocessed Thermal Infrared Channel Imagery	147
50. Unprocessed P Channel Imagery	148
51. Unprocessed C Channel Imagery	149
52. Laser Power Monitor Channel	151
53. Two Dimensional Power Spectrum of Figure 49	152
54. Two Dimensional Power Spectrum of Figure 50	153
55. Two Dimensional Power Spectrum of Figure 51	154
56. Two Dimensional Power Spectrum of Figure 52	155
57. Plot of Single Line of Noisy P Channel Imagery	157
58. Plot of Single Line of Noisy C Channel Imagery	157
59. Plot of Single Line of Laser Power Channel	158
60. Plot of P and Laser Power Channels	158

61. Plot of C and Laser Power Channels	159
62. Magnitude of the FFT of Noisy Section of P Channel	159
63. Flowchart of Image Correction Operations	160
64. Data from Figure 57 After Processing to Remove Noise	162
65. Processed P Channel Imagery	163

CHAPTER I

INTRODUCTION

Background

Airborne imaging systems that incorporate very high resolution, multisensing technologies are presently under development for a variety of military and civilian applications. Operational versions of these systems will be capable of collecting extremely large volumes of data at very high rates and will place great demands on transmission and storage resources. Therefore, processing designed to reduce the raw bit rates produced by these sensors prior to transmission or storage will become a most important part of these multisensor systems. The purpose of the research for this dissertation is to develop and test adaptive techniques that can be used to compress image data collected from airborne multisensor systems and which will result in acceptable levels of distortion of the reconstructed imagery.

The problem of compressing image data for transmission over bandwidth-limited channels has generated a very active area of research in recent years. Most of the work, however, has addressed standard imaging applications such as video teleconferencing, high definition television (HDTV), facsimile transmission, and satellite remote sensing [1-7]. The applications of the multisensor imaging systems considered in this study are quite different from these, and normally require discriminating very small objects located in highly cluttered backgrounds (e.g.

detection of minefields from airborne platforms). These types of applications have led to the development of imaging systems that can collect multiple channels of image data having very high spatial resolution. These systems are required to collect data at all times of day or night, and they normally operate in the thermal infrared region and/or incorporate an active (laser) illumination source with polarization-sensitive imaging capabilities. This type of high resolution, multichannel active/passive imagery did not exist until very recently, and has not been analyzed for the purpose of performing compression. Due to the substantial differences between these image data sources and those considered in the standard image compression literature, the applicability of existing source models and image compression schemes must be investigated. It has been observed that a compression scheme that has been developed and optimized for a given type of imaging source will, in general, not be optimized for a different source, and in fact may perform quite poorly. For example, when techniques based on the CCITT¹ international digital facsimile data compression standard are applied to digital halftone images, as much as a 50% data expansion can occur due to the fact that the two sources have significantly different statistics [8]. In order to ensure high performance compression of multisensor image data, this research has emphasized the development and analysis of new mathematical source models and compression techniques that account for the unique characteristics and applications of multisensor

¹ Comité Consultatif International de Téléphonie et Télégraphie.

imaging systems.

Dissertation Outline

The organization of the remainder of this dissertation is as follows:

Chapter 2 provides an overview of current image compression techniques with emphasis on those that take into account the nonstationary nature of real images. These adaptive techniques, though resulting in higher implementation complexity, have better performance characteristics (in a rate-distortion sense) than techniques based on average statistics that have been obtained from a limited set of images. Image modeling is discussed, and various performance measures, including the rate-distortion function, are presented.

Chapter 3 describes the typical applications of multisensor image data, and includes a description of the airborne multisensor imaging system that was used to collect the images used in this study. The polarization phenomenology exploited by this sensor, together with the reflectance and the thermal infrared sensing capabilities are described.

Chapter 4 consists of in-depth analyses of a large number of images from three different sensors (two polarization-sensitive active laser channels and a passive thermal-infrared channel). Results of these analyses are used to develop and verify a mathematical source model of the multisensor. A novel method of removing coherent laser power variations from the active imagery is developed (Appendix A), and its effect on bandwidth reduction is presented. A coordinate transformation

matrix is developed for use in removing inter-channel redundancies and to concentrate a major portion of the information content on one channel for efficient coding. An inverse transformation is then used to obtain the original individual channels during reconstruction.

Chapter 5 describes the development of a highly efficient compression scheme. Selection of the compression system components are evaluated using the adopted model, and an algorithm suitable for real-time implementation is formulated. The compression scheme relies on block transform coding that approximates the optimal Karhunen-Loeve transform, but which can be efficiently implemented using fast algorithms. Adaptivity to the nonstationarities of the imagery is provided by subdividing the images into small blocks and employing a bit allocation strategy that relies on a novel model-based scheme for estimating the distribution of transform coefficients in each block. Blocks with very high information content are identified and coded using a different adaptive bit allocation strategy.

Chapter 6 describes the mapping of the algorithm on a single-instruction multiple-data (SIMD) processing system that consists of a two-dimensional array of processors. Program listings of the algorithms coded in FORTRAN Plus (an extension of FORTRAN that facilitates parallel operations) are included in Appendix B. A large number of multisensor images collected at various times of day and in different environments are used in the evaluation of the compression scheme.

Chapter 7 presents an analysis of the reconstructed images, and documents the performance of the proposed scheme. The effects of distortion levels for various bit rates are evaluated by processing the reconstructed imagery with target detection algorithms, and analyzing the resulting false alarm and missed target performance.

Finally, chapter 8 summarizes the main results of this study, and points out areas requiring further work.

CHAPTER II

FUNDAMENTALS OF IMAGE DATA COMPRESSION

Preliminary Concepts and Definitions

In general, the goal of digital data compression is to minimize the number of symbols required to transmit or store the information content of a given source while at the same time limiting the distortion of the reconstructed signal. In order to quantify this goal, definitions of information content and distortion are required. A generally accepted definition of the information content conveyed by the occurrence of a specific source symbol k is

$$i_k = \log_2 \frac{1}{p_k} = -\log_2 p_k \quad \text{bits} \quad (2.1)$$

where p_k is the probability of the k -th source symbol occurring. This equation embodies a fundamental concept of information theory that the occurrence of an unlikely event (low p_k) carries more information than the occurrence of a likely one. In this study we are dealing with images that are composed of 8 bits per pixel (picture element) so that k , which can range from 0 to 255, corresponds to the luminance or gray level of a given pixel, and p_k is the probability that the source generates a particular gray level value.

If we assume a discrete memoryless source, that is, one that generates statistically independent symbols, then the average information content of the source

is given by its single symbol or zero order entropy H defined by:

$$H = -\sum_{k=1}^L p_k \log_2 p_k \quad \text{bits per symbol} \quad (2.2)$$

where L is the total number of symbols (255 in the case of 8 bit pixels) and where it is assumed that $0 \log_2 0 = 0$.

For sources that do not generate statistically independent symbols, conditional or multisymbol probabilities are used in the equations above, and the resulting higher order entropies can be significantly lower than those given by the single symbol entropy. In the case of multidimensional sources such as images, the large number of possible dependencies between symbols make accurate calculation of higher order entropies very difficult. In these cases it is normally assumed that the source can be modeled by a finite memory or low order Markov model where the number of possible interdependencies are limited to a small neighborhood of each pixel.

Entropy measures are used to estimate the minimum number of bits necessary to represent a given image without any loss of information. This concept is known as Shannon's noiseless coding theorem for discrete sources [29] which states that a source having entropy H can be coded without distortion by using $H + \epsilon$ bits per symbol (bit rate), where ϵ is an arbitrarily small positive quantity. This theorem is useful for predicting the compressibility of a given source, and for evaluating the performance of specific compression techniques. The significance of this theorem lies

in the fact that it provides a lower bound of any information preserving compression scheme and guarantees that a coding scheme that approximates that bound exists. Unfortunately, the theorem does not provide any insight or methodology for developing such a scheme. Compression schemes that approach the lower bound given by the Shannon theorem make up a class of information-preserving coding techniques known as lossless or entropy coding methods which are described later in this chapter.

There are also techniques for coding image data at bit rates lower than the entropy of the source. These techniques result in distortion due to some loss of information and are therefore classified as lossy coding methods. These lossy techniques are also covered in subsequent sections of this chapter.

To quantify the performance of lossy compression methods, a definition of distortion and a functional relationship between bit rate and level of distortion are needed. There have been a number of attempts to develop measures of image distortion that match the image quality determined subjectively by human analysis. Models of the human visual system (HVS) are being developed and coding schemes that distribute the coding error or distortions over the less sensitive portions of the HVS have been developed [3]; however, a robust measure of distortion that relates coding error to visible image quality has not yet been formulated. For the image applications considered in this study, this is not a serious drawback. In most cases, the multisensor image data is used for computer-based automatic target detection or

cueing, and as such, it is subjected to various filtering, thresholding and clustering operations used in pattern recognition that are affected by signal-to-noise considerations and not by HVS parameters. Therefore peak signal to (reconstruction) noise ratio (PSNR) and mean square error (MSE) will be used as the distortion measurement criteria in this study. If we let $x(i,j)$, where $1 \leq i \leq M$ and $1 \leq j \leq N$, be the gray level value of each pixel in a given $M \times N$ image, and after compression, error-free transmission and reconstruction, we obtain a distorted version of $x(i,j)$ given by $\hat{x}(i,j)$, then the error measures used in this study are defined as

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x(i,j) - \hat{x}(i,j))^2 \quad (2.3)$$

and

$$PSNR = 10 \log_2 \frac{(255)^2}{MSE} \quad (2.4)$$

Having defined our error metrics, we need to also define the functional relationship between the average number of bits per pixel (bit rate) used to code a given source and the amount of resulting distortion. This relationship, called the rate-distortion function $R(D)$, was first described by Shannon [29] for simple memoryless sources. Berger [30] and Gray [31] have expanded the theory and developed methods for computing $R(D)$ for more realistic sources. In general the rate-distortion function of a source is found by minimizing the average information subject to a fidelity

criterion; therefore, it indicates the minimum amount of information required to reconstruct the source output with a specified maximum level of distortion. Azadegan [8] has computed $R(D)$ bounds for some of the more popular compression techniques for image sources that can be modeled by 2-dimensional (2-D) separable first order Gauss-Markov models. Empirical $R(D)$ calculations for our selected compression scheme and distortion measure are presented in Chapter 7.

Image Source Modeling Considerations

As the reader may have observed, most of the concepts discussed in the previous section are prefaced by assumptions about the characteristics of the image source. It is appropriate that we now consider more closely the characteristics of these sources before proceeding to the review of compression techniques since their performances are heavily dependent on the source model adopted.

A characteristic common to all image sources of interest is that they are spatially nonstationary. Even over relatively short distances, the mean, variance, and the autocorrelation (or covariance) statistics can vary significantly. As a result, coding techniques that are based on statistics calculated from an ensemble of images (or from large areas of one image under the assumption of ergodicity) will be less than optimal. Such techniques tend to assign too many bits to areas of the imagery that have little detail (low variance) while under-coding areas that have higher levels of detail.

Various techniques have been proposed to deal with the problem of space-varying

mean and autocorrelation function. Hunt [28] proposed the use of nonstationary statistical image models to transform the original image into a stationary one that can be efficiently compressed by non-adaptive techniques. This process required hybrid digital/optical computing to carry out the computationally intensive spatial warping operations. More practical approaches involve partitioning the image into blocks that are sufficiently small to allow each block to be considered the realization of one of a finite number of random processes. This type of model forms the basis for the adaptive transform coding methods presented later in this chapter.

Another technique has been to model the image source as a Markov or autoregressive (AR) process with space-varying parameters. The most widely used is the 2 dimensional (2-D), separable, first order Gauss-Markov model in which the value of the present pixel $x(i,j)$ is causally dependent on three of its nearest neighbors as given by

$$x(i,j) = \rho_v x(i-1,j) + \rho_h x(i,j-1) - \rho_v \rho_h x(i-1,j-1) + w(i,j) \quad (2.5)$$

where ρ_h and ρ_v are the horizontal and vertical correlation coefficients which can be allowed to be space varying and $w(i,j)$ is a 2-D, zero-mean sequence of independent identically distributed (i.i.d.) Gaussian random variables with variance given by

$$\sigma_w^2 = \sigma_x^2 (1 - \rho_h^2) (1 - \rho_v^2) \quad (2.6)$$

where σ_x^2 is the variance of the resulting sequence $\{x(i,j)\}$. This model has been found to provide a good approximation to a large class of real world images, and can

be used to calculate the $R(D)$ function explicitly. This relationship provides the ultimate performance limitations of any encoding scheme that operates on sources that can be modeled by Equation (2.5) . Some of the adaptive predictive coding methods discussed later in this chapter are based on this model.

Lossless Compression Techniques

Lossless coding methods are used in applications where perfect reconstruction of the image data is required such as for storage and retrieval of medical images. A characteristic that makes these techniques undesirable for most imaging applications is that they do not produce significant bit rate reduction since most images of natural scenes have single symbol entropies in the range of 6 to 7.5 bits per pixel. A more common application of noiseless coding is at the output of a lossy scheme in order to remove any remaining redundancy and further reduce the bit rate. Since our interest is in the latter case, we will consider only those lossless techniques referred to as entropy or variable length coding and we will ignore other techniques such as bit plane encoding and lossless predictive coding that are used for coding complete images.

Entropy or variable-length coding achieves compression by exploiting the fact that the symbols (or group of symbols) generated by a source have unequally distributed p_k . Techniques such as Shannon-Fano coding or the more popular Huffman coding reduce bit rates by assigning the shortest code words to the symbols that occur most often (high p_k) and reserve the longest code words for those that are

very unlikely to occur. The average bit rate reduction is dependent on the unequal probability distribution of the original data sequence. Huffman coding can be made adaptive to changing image statistics by means of a two-pass technique where the data sequence is buffered, an algorithm calculates single or multi-symbol statistics and generates the codebook during the first pass, and the data is encoded during the second pass. Unlike Huffman coding of stationary sequences, this adaptive (or dynamic) Huffman coding technique requires that the codebook be included with the coded data to allow for reconstruction.

Though Huffman coding has long been considered optimal, a recently developed technique called arithmetic coding results in greater compression, is faster for adaptive models, and does not require blocking symbols together. Huffman coding is only optimal if all the symbol probabilities are integral powers of $1/2$. Since this is not normally the case, Huffman coding can require up to one bit per symbol higher than the source entropy. Arithmetic coding, on the other hand, does not place restrictions on the symbol probabilities and actually achieves the theoretical Shannon entropy bound [32,33]. A tutorial on arithmetic coding theory was presented by Langdon in [34].

Lossy Compression Techniques

The techniques considered in this section are classed as lossy schemes due to the fact that by coding at rates below the Shannon entropy bound, they constitute many-to-one mappings and cannot, in general, result in perfect reconstruction.

However, they can achieve very high compression performance, on the order of less than 1 bit per pixel, while maintaining acceptable levels of distortion. All of the techniques that follow rely on the fact that most image data of interest have a high degree of correlation between neighboring pixels.

Lossy image compression methods can be divided into two main classes, predictive methods and transform methods. In addition, there are a number of methods such as hybrid methods and vector quantization methods that do not fit into these two major classes. A variety of adaptation techniques have been employed to allow these methods to deal with the nonstationary characteristics of image data. The purpose of the remainder of this chapter is to provide an overview of those lossy coding techniques that are applicable to the goal of achieving high performance compression of multisensor imagery.

Predictive Methods

Predictive methods have been widely employed for compressing 1-D signals such as speech. Because of their low implementation complexity, they have also formed the basis for a large number of digital image compression schemes. Predictive methods are based on the assumption that the source generates symbols that are highly correlated and therefore contain considerable redundancy. The goal of predictive techniques is to separate the redundant or predictable part of the source, which carries essentially no information, from the innovative or information-conveying part. The most popular predictive scheme is differential pulse code modulation (DPCM) shown

in Figure 1.

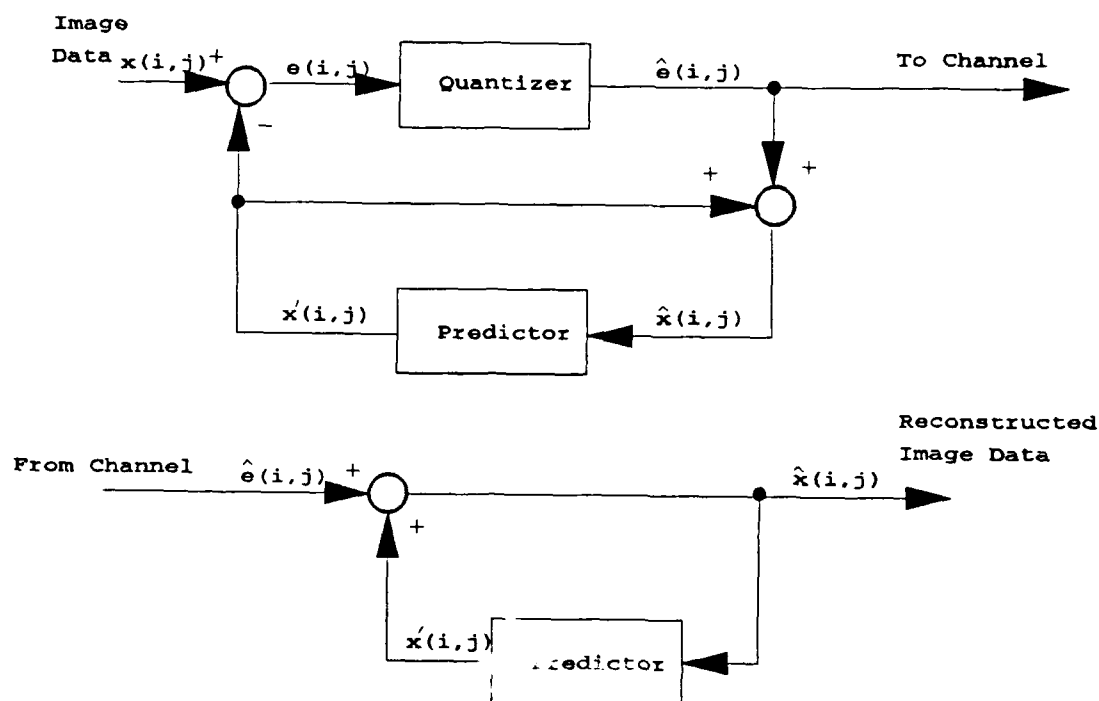


Figure 1. DPCM Structure

The predictor makes use of the correlation between samples to predict the value of subsequent samples. In this case, the predictor is a linear combination of past pixel values and is used to estimate the value of the present pixel. The estimated value $x'(i,j)$ is subtracted from the actual value and this difference $e(i,j)$ is quantized. Data compression is the result of the fact that the difference $e(i,j)$ has much lower variance than the original data, the correlation or redundancy is significantly reduced, and the quantized result $\hat{e}(i,j)$ can be efficiently coded by entropy methods. The reconstruction of the imagery is performed by using a predictor identical to the one

used in the compression stage. In the case of image data, a 2-D predictor based on the causal first order Gauss-Markov model given in Equation (2.5) is often used. The predictor coefficients are then matched to the modeled or computed correlation coefficients.

DPCM schemes can be made adaptive by changing either the predictor coefficients or the quantizer characteristics or both. Two types of adaptation have been employed in practical schemes, the difference being whether they use the source samples or the reconstructed values to effect the adaptation. The first method is called forward adaptation and requires the sending of side information which adds transmission overhead and necessitates the use of synchronization strategies. The second method is called backward adaptation and since the adaptation is based on information available to both the receiver and the transmitter, no transmission of side information is required [4]. For this reason, backward adaptive DPCM methods are preferred in high-performance applications.

One of the main disadvantages of DPCM coding of image data is its poor distortion performance when edges are encountered. Since images normally contain a large number of edges, DPCM has not been as widely accepted for image coding applications as it has been for speech coding.

Transform Methods

Transform coding methods are among the most efficient techniques for compressing image data. As in predictive methods, the purpose of transform coding

is to remove the redundancy between pixels. The difference is that the predictive methods operate in the spatial domain while transform methods map the spatial data into a different domain where the information is concentrated into a small number of uncorrelated coefficients that can be more efficiently coded.

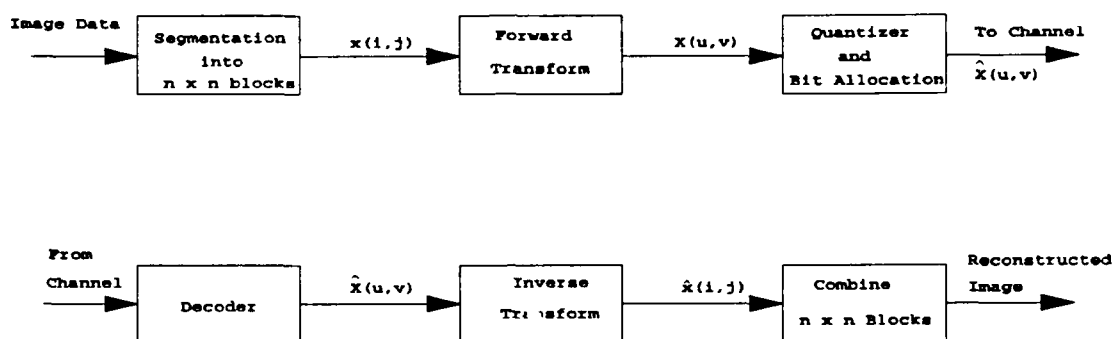


Figure 2. Transform Coding Structure

In general, the technique for transform coding shown in Figure 2 entails partitioning the image field into non-overlapping blocks which are then mapped via an orthogonal transformation into a frequency-like domain. Selection of the transform is based on its image energy (variance) or information compaction capabilities as well as the existence of "fast" algorithms for implementation. Usually block sizes that are powers of two are selected in order to employ fast Fourier transform (FFT) type algorithms, and they can range from 4 x 4 to as large as 128 x 128 pixels. Due to

the nonstationary nature of image data, the use of small blocks is normally preferred since local stationarity assumptions usually hold for small neighborhoods, such as 8×8 blocks. In addition, small block transforms have significantly lower implementation complexity than larger ones. However, small block sizes have the disadvantage that significant amounts of interblock redundancies remain after coding, and the resulting bit rates are higher than for larger blocks.

Four different methods have been described in the literature for adapting the transform coding scheme of Figure 2 to the variations in the image statistics:

1. Modify the transform basis functions or change the transform used based on the block statistics. Clarke [10] claims that performance differences between practical transform types are small enough that significant bit reductions would not result from this form of adaptation.
2. Vary the block sizes based on local image statistics; that is, partition the imagery into non-overlapping blocks whose sizes depend on local activity measures, and then apply suitably sized transforms to each block. Dinstein et al [37] describe a variable block size technique that relies on complex classification and clustering techniques to partition the image into nine different square or rectangular block sizes. The advantage of this technique is that selection of large blocks for homogeneous image areas results in lower bit rates while the use of small blocks in high energy regions preserve the texture or small details. A disadvantage of this method, other than computational

complexity, is its requirement for manually pre-specifying the number of bits to be allocated to each block size, and the overhead required to transmit the block partitioning and clustering maps.

3. Adapt the bit allocation based on block statistics. Clarke [10] suggests that progress in this area has the best payoff potential of all adaptive transform coding techniques. Chen and Smith [26] and others have suggested partitioning the imagery into fixed block sizes that are then sorted into four equi-populated classes based on block activity measures. Each class would then have its own fixed bit allocation strategy. Overhead requirements consist of four bit allocation maps (run length or Huffman encoded), and an additional two bits per block for identification of block class assignment. Clarke [10] suggests that better results are obtained by having three equi-populated classes and one highly unlikely, very high activity (high detail region) class.
4. Adapt the quantizer levels. In this adaptive scheme the bit allocation is kept constant but the quantizer levels are adjusted according to changes in the statistics of the transform domain samples [2]. This approach has not been as popular as the adaptive bit allocation technique.

The theoretically optimal transform that results in minimum MSE for the number of coefficients retained, and which also produces totally uncorrelated coefficients having minimum entropy, is the Karhunen-Loeve transform (KLT) [4]. Unfortunately, the basis vectors of the KLT are signal dependent and no generalized

fast algorithms are available. The difficulty in implementing the KLT is that the data covariance matrix must be estimated for each block and the eigenvectors of the covariance matrix must be calculated before the matrix of basis vectors which diagonalize the covariance matrix are found. Since the basis vectors are different for each data covariance matrix, they must be coded and transmitted (stored) along with the coded coefficients in order to reconstruct the image at the receiver.

Implementation complexity and the large overhead required to transmit or store the basis vectors make KLT methods impractical for real-time image compression applications.

A large number of transforms have been proposed as substitutes for the KLT. These range from easily implementable Walsh-Hadamard and Haar transforms which have rectangular basis vectors, to FFT methods which have complex valued sinusoidal basis vectors. For those sources that have a high degree of inter-pixel correlations, the two dimensional discrete cosine transform (DCT) has been found to perform almost identically to the KLT [11].

One of the problems associated with transform coding is the "blocking effect" caused by the fact that the imagery is first partitioned into blocks which are then transformed and encoded independently, and thus any distortion within a block tends to be discontinuous across block boundaries. As the bit rates are lowered, the distortion increases, and the block boundaries become highly visible in the reconstructed imagery and would be unacceptable in automatic target recognition

(ATR) algorithms since they would be enhanced by the edge and shape detection operators normally employed in these systems. Several methods have been proposed for reducing "blocking effect" artifacts. Reeve and Lim [36] proposed low-pass filtering the block boundary pixels of the reconstructed imagery; while this method does not increase bit rate, it does blur the image data across block boundaries which is not desirable if objects of interest are small. Reeve and Lim also suggested dividing the image into overlapping blocks prior to transform coding. The drawback of this method is the increased bit rate required to transmit redundant data. Hinman et al [39] proposed a short space Fourier Transform technique which is intrinsically free from blocking effects; the shortcoming of this scheme is that this transform suffers from ringing around edges. Malvar and Staelin [40] suggested the lapped orthogonal transform (LOT) which has the same benefits of overlapping blocks but without increasing bit rates. A possible disadvantage of this scheme is that the LOT does not have a D.C. basis vector so that coding a homogeneous image area requires the use of high order (high frequency) components. Rose et al [23] suggested an alternate DCT/DST transform which removes blocking effects at the expense of considerable increase in implementation complexity.

Other Compression Methods

There are a number of methods for compressing image data that do not fall into the types described in the preceding sections. The more popular of these other methods include hybrid compression, vector quantization, and subband coding

methods.

Hybrid methods basically consist of a combination of predictive and transform schemes in an attempt to exploit the advantages of each. The hybrid method often preferred consists of dividing the image into small blocks, transform coding the individual blocks using a 2-D transform, and using DPCM to code the resulting coefficients. The DPCM prediction for each coefficient in a block would be based on the corresponding coefficient of the horizontally preceding block. Hybrid methods for coding applications such as video telephony and teleconferencing have also been proposed. Since this study does not deal with multiframe or time sequential images, techniques that depend on temporal (interframe) redundancy will be ignored.

Vector quantization (VQ) consists of decomposing the image data into vectors of equal length which are compared to a set of vectors stored in a codebook. Each image vector is matched with a particular codevector by means of clustering techniques using some minimum distance criterion. The address of the selected codevector is then transmitted to the receiver which uses this address to fetch a codevector from a codebook identical to the one in the compression stage. The efficiency of this technique depends on designing a codebook that is representative of all the possible image vector combinations. This is not a trivial design problem since, even for small image blocks such as 4 x 4 pixels (i.e. vectors of length 16), there will theoretically be 256^{16} possible image vectors. It is possible to design a small codebook that approximates the vast number of possible image vectors with acceptable

overall distortion if an adequate set of training images is available [41].

Unfortunately, the imaging sensors considered in this study must operate in a wide range of environments and such a training set may be difficult to assemble.

Subband coding (SBC) is similar to transform coding in that the image data is decomposed into frequency components that can be more efficiently encoded than the original image. The difference is that, while transform coding uses a set of basis functions (e.g. sampled cosines) to decompose the image, SBC uses a set of bandpass filters to decompose the image into a set of subimages, each of which contains a limited range of spatial frequencies. The major components of an SBC scheme consists of the following:

1. A bank of analysis filters which decompose and downsample the original image data.
2. A bank of coders (normally DPCM or VQ) to efficiently compress the downsampled data.
3. A corresponding bank of decoders at the receiver.
4. A bank of synthesis filters to upsample and reconstruct the subimages which are then added together to form the reconstructed image.

In order to prevent distortions resulting from aliasing introduced during the downsampling operations, quadrature mirror filters (QMF), which cancel out any aliasing, are normally used [4]. An advantage of this technique is that, unlike transform coding at low bit rates, blocking artifacts are not a problem with SBC since

the entire image is decomposed without partitioning into blocks. Additionally, SBC schemes normally have lower implementation complexity than transform coders.

A technique that has recently received considerable publicity is the use of wavelets for data compression. This technique lies between transform coding methods (specifically the short space Fourier Transform technique), and subband coding methods since it involves application of an orthogonal transform (the Wavelet Transform) to decompose the original data into a multiresolution domain. References [56] and [57] provide an excellent overview of this emerging field, and an extensive list of references is included in [56].

The level of interest generated by Wavelet Theory is comparable to that generated by fractals just a few years ago, and its proponents speculate that wavelet techniques will result in tremendous performance improvements and will make Fourier-based techniques obsolete; however, practical applications are still very limited. It is expected that the major contributions of Wavelet Theory will be in unifying the variety of techniques used in the field of nonstationary signal analysis. At the present time, it may be premature to expect that drastic improvements in image data compression will result from simple application of wavelets.

CHAPTER III

MULTISENSOR IMAGING SYSTEM DESCRIPTION

Overview

One factor common to most military sensing applications is the need to discriminate objects (targets) that have been deliberately designed to blend with the backgrounds in which they operate. Examples of concealment techniques range from the simple application of camouflage paint schemes designed to degrade visual identification, to complex applications of stealth technology that deny detection by acoustic, optical, and radar sensing techniques. In this study we are concerned with image data collected with optical sensors designed to detect targets that have been camouflaged to visually blend with natural backgrounds.

Systems designed to perform these detection tasks normally combine or fuse information from several sensors in order to increase their capabilities to discriminate, detect, and classify objects of interest in varying background conditions. This multisensor concept exploits the fact that the spectral radiance of an object is dependent on many parameters so concealment throughout the entire spectrum of optical sensors is impossible. Thus, separation of the objects of interest from backgrounds can be effected by means of a suite of sensors that are tuned to collect a number of independent features. For example, if we consider sensors operating in the

optical region (0.2 to 1,000 μm), then the spectral radiance L of an object at a particular coordinate location (i,j) on the scene is a combination of an emission and a reflectance component, and is given by

$$L(i,j,\lambda,p) = (1-r(i,j,\lambda,p)) M(\lambda) + r(i,j,\lambda,p) l(i,j,\lambda,p) \quad (3.1)$$

where $r(i,j,\lambda,p)$ is the spectral reflectance of the object, $l(i,j,\lambda,p)$ is the spectral irradiance or incident illumination of the object, $M(\lambda)$ is the spectral radiant emittance of a blackbody source, λ is the wavelength, and p is the polarization. L is also dependent on the angle of illumination and on the viewing angle which, in the case of the sensors used for this study, are small enough (within 20° of normal incidence) to be negligible.

For wavelengths in the visible through the near-infrared region (0.4 to 2 μm), the emitted component of L is negligible and the reflected component dominates, while for wavelengths in the thermal-infrared region (8 to 14 μm), the emitted energy is the dominant component. The polarization dependence of L is a function of surface roughness (as discussed later in this chapter). Therefore, a multisensor system that is capable of measuring the radiance of a scene at two well separated wavelength bands, λ_1 and λ_2 , and which is also capable of measuring the polarization state p_i of the reflected energy, will provide three basically independent sources of information about each point on the scene. Each (i,j) coordinate or point imaged by such a sensor would consist of a three dimensional vector composed of the three measured quantities

such that

$$\vec{x}(i,j) = \begin{bmatrix} L_{\lambda_1} \\ L_{\lambda_2} \\ L_{p_1} \end{bmatrix} \quad (3.2)$$

We can also visualize this vector as a point in three dimensional feature space defined by three orthogonal basis vectors corresponding to each of the sensed L quantities.

Therefore, if we would like to separate an object based on *a priori* radiance information, we could process the image data to search for all pixels that have values such that

$$\left\| \begin{bmatrix} \dot{L}_{\lambda_1} - \sigma_1 \\ \dot{L}_{\lambda_2} - \sigma_2 \\ \dot{L}_{p_1} - \sigma_3 \end{bmatrix} \right\| \leq \left\| \vec{x} \right\| \leq \left\| \begin{bmatrix} \dot{L}_{\lambda_1} + \sigma_1 \\ \dot{L}_{\lambda_2} + \sigma_2 \\ \dot{L}_{p_1} + \sigma_3 \end{bmatrix} \right\| \quad (3.3)$$

where the L' values correspond to the means of our *a priori* measured signatures, σ_1 , σ_2 , and σ_3 are threshold ranges based on the standard deviation of our *a priori* measurements, and $\| \cdot \|$ is any suitable norm. Even fairly wide threshold ranges result in a significant reduction in the number of pixels in the scene that could still make up our objects of interest. If we add additional sensors that are tuned to different parameters of L , such as a different wavelength or the polarization characteristics, or even energy outside of the optical region such as millimeter wave

radar, then we increase the number of elements or features of the $x(i,j)$ vector.

Further separation of our object of interest from the background can then be achieved by application of additional thresholds and by exploiting shape and spatial distribution of the objects in the scene.

An example of the operations performed on multisensor imagery for a typical target detection application is shown in Figure 3. This type of processing will be used in later chapters to determine the allowable distortion bounds, and to evaluate the effects of the distortions produced by the compression and reconstruction process.

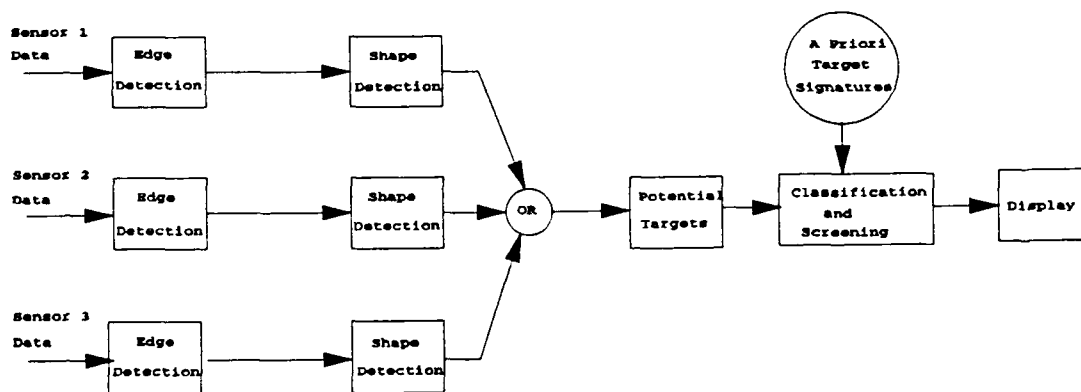


Figure 3. Typical Application of Multisensor Imagery

In this example, each image channel is processed to locate and enhance any edges. This is accomplished by convolving each image channel with a 2-D gradient such as the Sobel operators followed by thresholding operations to create binary images that contains all of the edges in each of the channels. The edge images are

then convolved with 2-D filters designed to pass edges that correspond to objects that are within the range of sizes and shapes of potential targets. The three resulting images are then combined by means of a logical "OR" operation to form a combined potential target image that contains all of the pixel locations of interest. The significance of the processing up to this point is that targets can be discriminated as long as there is adequate target to background contrast in at least one image channel. The original three gray level values of the potential target pixel locations are then used for the follow-on processing which consists of classification and screening. The classification is accomplished by means of a clustering operation that groups the pixels using Euclidian distance measures [12]. This step can use *a priori* target signatures to perform the classification, in which case the pixels that are closest to the target signatures are cued (e.g. color coded) on one of the image channels, and presented to a human analyst for confirmation. In the case that *a priori* signatures are not available, an unsupervised clustering can be performed in order to group all pixels that have very similar signatures, and screenings based on number of objects or on their spatial distributions can be performed [19]. In this example the sensors are assumed to have identical field of view and footprint on the ground, and the three images are assumed to be perfectly registered. If these assumptions were not true, then geometric correction techniques [5] would have to be applied prior to performing the operations shown in Figure 3.

In this study it is assumed that the image data is to be used for detection of a

variety of target types. That is, it is assumed that the same imagery could be processed a number of times by means of different algorithms, some of which may require interactive threshold setting, filter selection, and human interpretation to detect the presence of various types of objects. Otherwise, if the imagery were to be used for a single type of object detection, the image data compression problem could be significantly simplified by performing all of the target detection operations at the sensor, and transmitting only the target locations. The flexibility resulting from having the individual sensor images for analysis on the ground justifies the effort required to efficiently compress the significant amounts of multisensor data.

Sensor Description

The imagery used in this study was collected using a helicopter-mounted, multisensor line scanner developed by the U.S. Army Corps of Engineers for the purpose of conducting surface minefield research. The scanner is configured to sense three independent, optically aligned radiation quantities. The concept consists of transmitting a beam of linearly polarized laser energy and sensing the reflected electromagnetic components parallel and perpendicular to the transmitted polarization. The relative magnitudes of these two components are dependent on the depolarization and directional reflectance properties of the surface. In addition to these two properties, the system also measures the thermal energy emitted by the same surface area.

The multisensor scanner, shown in detail in Figure 4, is based on a diode

array, side pumped Neodymium: Yttrium-Lithium-Fluoride (Nd:YLF) laser providing 1.2 watts of preferentially polarized energy at a wavelength of $1.053\ \mu\text{m}$. A half wave plate rotates the polarization vector from a nominally vertical orientation to a nominally horizontal orientation, and a calcite polarizer provides a linearly polarized output beam. A beam expander focuses the laser and two relay mirrors redirect the laser beam to the scan mirror. The four sided scan mirror rotates at 5250 rpm to yield 350 scan lines per second with a 40° field of view, while the aircraft operates at altitudes ranging from 100 to 400 ft above ground level (AGL) and 30 to 120 miles per hour groundspeed. The resulting ground resolutions range from approximately 1.5 to 6 inches with a 1:1 pixel aspect ratio.

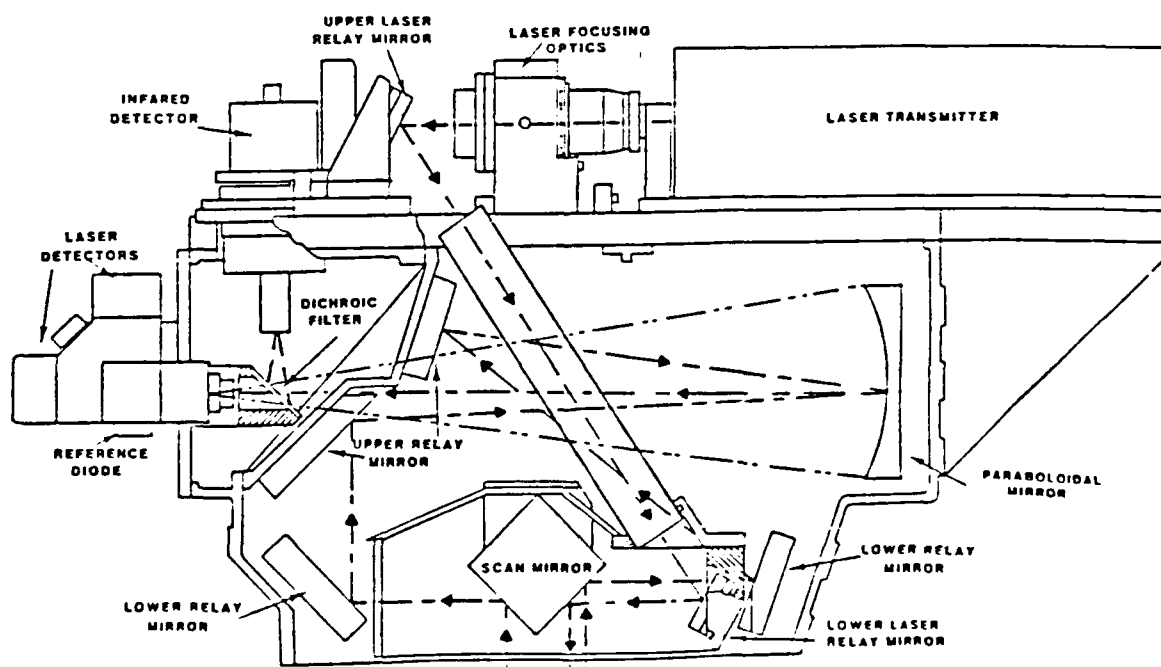


Figure 4. Cutaway View of Multisensor Imaging System

A portion of the backscattered energy, as well as passive thermal infrared energy emitted by the scene, are collected by two faces of the scan mirror and redirected along two receiver paths to a paraboloidal mirror for focusing. A dichroic filter redirects passive thermal energy to a Mercury Cadmium Telluride (MCT), liquid nitrogen cooled detector. The laser return is transmitted through the dichroic filter and collimated before entering a beam splitting polarizer that separates the two polarization components which are then focused onto two avalanche photodiode detectors. In addition, a third detector (not shown) is used to record the variations of the laser energy output.

Data from the sensors are input to an analog pre-processor that combines them into six separate channels, and provides anti-aliasing filtering with cutoff frequency 440 KHz and 24 dB per octave attenuation. The individual channels are then routed to six digitizers each operating at 1.05 MHz sampling rate with 11 bit resolution. The six channels of digital image data consist of the following: (1) a parallel (P) channel consisting of the laser return that has the same polarization as the transmitted beam, (2) a cross (C) channel consisting of the laser return with polarization perpendicular to the transmitted beam, (3) a polarization channel that corresponds to the digitized ratio of the difference and the sum of the P and C channels, (4) a reflectance channel that consists of the sum of the P and C channels, (5) a laser power channel that records the variations of the power output of the laser, and (6) a thermal-IR channel that consists of the thermal emittance of the scene in the 8.5 to 14

micron band. Channels 1, 2, and 6 contain all of the independent image information collected by the system, and the most significant 8 bits of these three channels will comprise the data used in the compression/reconstruction tests conducted in this study.

Physics of Polarization Imaging

Thermal infrared technology is a fairly mature field, and a variety of imaging systems that sense emitted energy in the 3 to 15 micron wavelength region have been commercially available for a number of years. In contrast, polarization sensitive imaging systems are a recent development, and the physics underlying their performance is not well understood. The purpose of this section is to present a mostly qualitative explanation of the phenomenology of polarization imaging and its application in remote sensing of man-made objects. References to theoretical and quantitative methods are also presented should the reader be interested in a more rigorous treatment of this subject.

The interest in polarization imaging is driven by the premise that man-made objects tend to support the requirements for a polarization signature while natural backgrounds do not. This premise has been based largely on surface roughness and geometry arguments rather than on rigorous theoretical work. In addition, a considerable number of experiments have been documented which provide a qualitative explanation of the scattering of coherent light from random rough surfaces [42 - 45].

Figure 5 depicts a simple model of a rough surface illuminated by a laser source, and the three mechanisms by which the incident energy can be backscattered towards the source. The material surface is modeled as a statistically large distribution of specularly reflecting planar microfacets [45]. In the first case (path A), the laser energy strikes a single planar microfacet and is specularly reflected back towards the source where the state of polarization can be determined by two sensors properly oriented to measure the two independent polarization components.

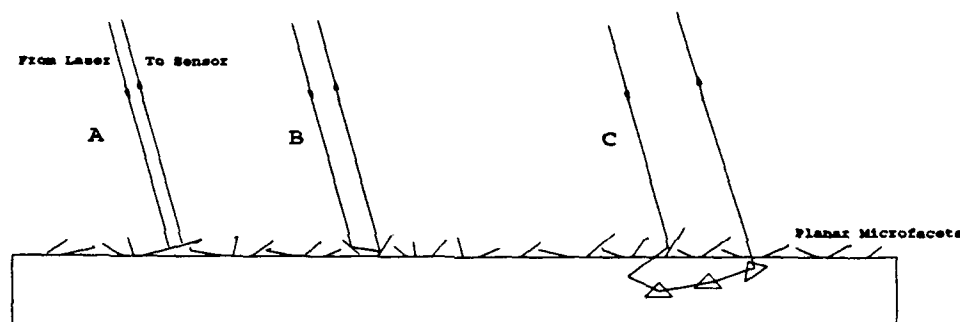


Figure 5. Possible Backscattering Paths

In path B, the incident light undergoes multiple specular reflections off the planar microfacets before traveling back to the sensors. In path C, the incident energy penetrates into the surface of the material before being reflected back out. Paths A and B correspond to surface scattering, and path C to volume scattering effects. If the incident laser illumination is linearly polarized, then the energy reflected in A will be in the same direction as the incident energy. Path B, on the other hand, will result in a rotation of the polarization direction (depolarized). Path C

will result in diffuse, unpolarized reflection.

Assuming that surface roughness is related to the number and sizes of the microfacets, then the premise in polarization sensing is that smooth surfaces such as metallic or painted objects have predominantly path A reflections and that rougher surfaces such as soils and vegetation support the multiple and volume scattering paths depicted by B and C.

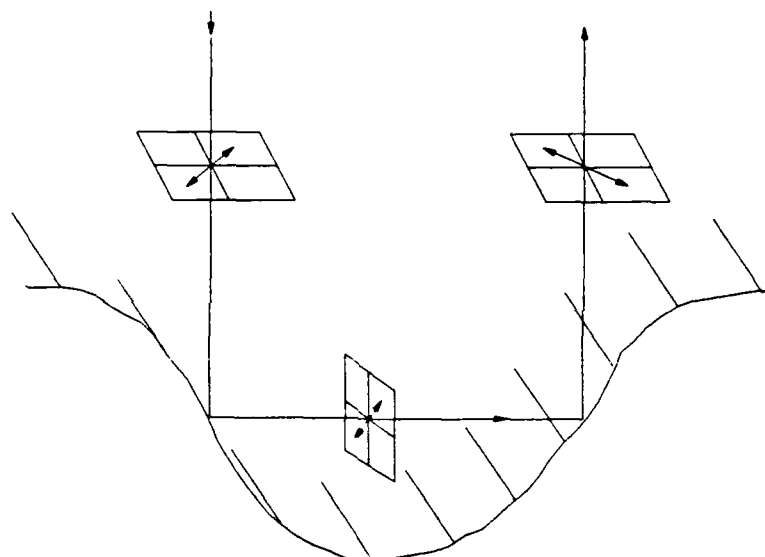


Figure 6. Multiple Scattering Within a Valley of a Rough Surface

Mendez and O'Donnell [43] conducted experiments that better illustrate the effects of multiple reflections on the state of polarization. In these experiments, a photoresist surface was etched and gold-plated to produce a surface profile that approximated a Gaussian random process. This surface was then illuminated by linearly polarized laser light. Figure 6 shows the geometry of a multiple (two)

reflection path within a surface valley. In this illustration the linearly polarized electric field orientation is shown as a double arrow in the plane perpendicular to the direction of propagation. It is assumed that this electric vector strikes the surface of the material at an angle with respect to the locally flat surface. If the material is assumed to be a perfect conductor, then no electric field can exist inside the surface, and application of the Ewald-Oseen extinction theorem [47] requires that the incident field be exactly canceled by sources along the surface of the material. These sources in turn produce the intermediate wave that propagates to the other side of the valley with the polarization orientation rotated as shown in Figure 6. At the second reflection point (also assumed to be at an angle), the polarization orientation is again rotated so that the energy reflected towards the source has undergone a rotation within the plane normal to the direction of propagation of the incident energy.

Figure 7 illustrates four of the infinite number of possible two-reflection paths from a material that is modeled as a perfect conductor and which has a rough surface consisting of random circular valleys. In these diagrams, E represents the polarization orientation of the electric field, k represents the direction of propagation (wave vector), and the circle represents a plan view of a valley or dimple on the surface of the illuminated target. The surface is assumed to lie along the plane of the paper, and the incident illumination is normal to this plane and linearly polarized in the vertical direction. In Figures 7(a) and 7(c), the polarization of the backscattered energy is in the same orientation (parallel polarized) as that of the incident

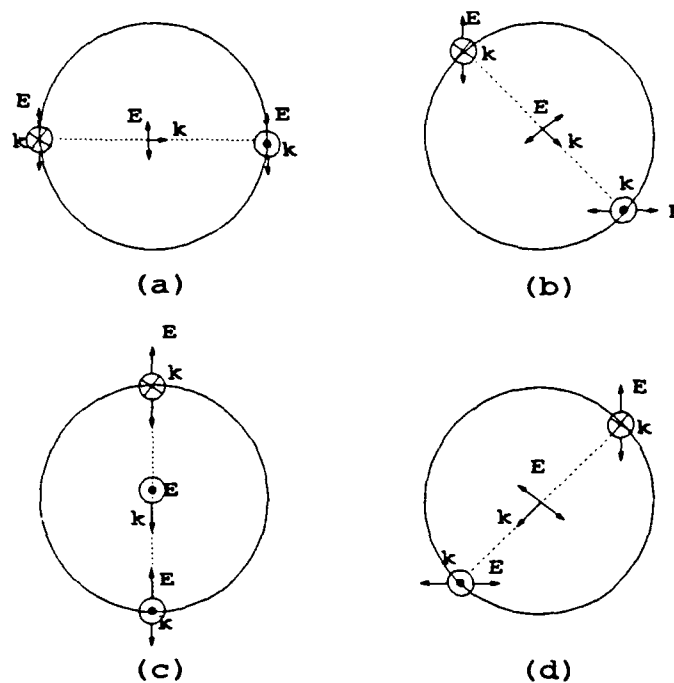


Figure 7. Multiple Scattering of Linearly Polarized Light

illumination. In Figures 7(b) and 7(d), however, the incident wave strikes the wall of the valley at a 45° angle between the locally flat surface normal and the polarization orientation, and the resulting backscattered energy has a polarization orientation orthogonal (or cross polarized) to that of the incident wave. If the laser illuminates a sizable area of the surface, as is the case with the sensor system considered in this study, then the backscattered energy will consist of the superposition of a large number of contributions from the paths shown in Figure 7. The polarization state of the backscattered energy will then depend on the relative intensities of the parallel (P) and cross (C) polarized components measured by the two sensors. A generally

accepted definition of the change in polarization state that linearly polarized energy undergoes after being reflected from a rough surface is given by

$$D = \frac{(P - C)}{(P + C)} \times 100 \% \quad (3.4)$$

From this definition, and the previous discussion, we can conclude that energy backscattered from smooth surfaces will consist of predominantly P orientations and will therefore have high values of D (near 100%). In contrast, rougher surfaces will support multiple reflections that result in a higher C component, and which in turn results in lower values of D .

In order to illustrate the practical advantages of using polarization imaging for man-made target detection, an example of actual imagery is presented in Figure 8. It consists of 710 by 1024 pixel areas of the Thermal (a), Reflectance (b), and Polarization (c) channels collected over a test minefield placed in a desert background. The targets are readily visible as bright objects in the Polarization image, and have been automatically cued (encircled by red squares) by a simple program that uses shape and polarization information to separate the targets from the background. By comparison, the Thermal channel shows very little target to background contrast, and the amount of clutter makes it very difficult to extract the targets even by careful photo-interpretation techniques.

The preceding discussion represents a greatly simplified overview of polarization sensing. An in-depth treatment would require consideration of real

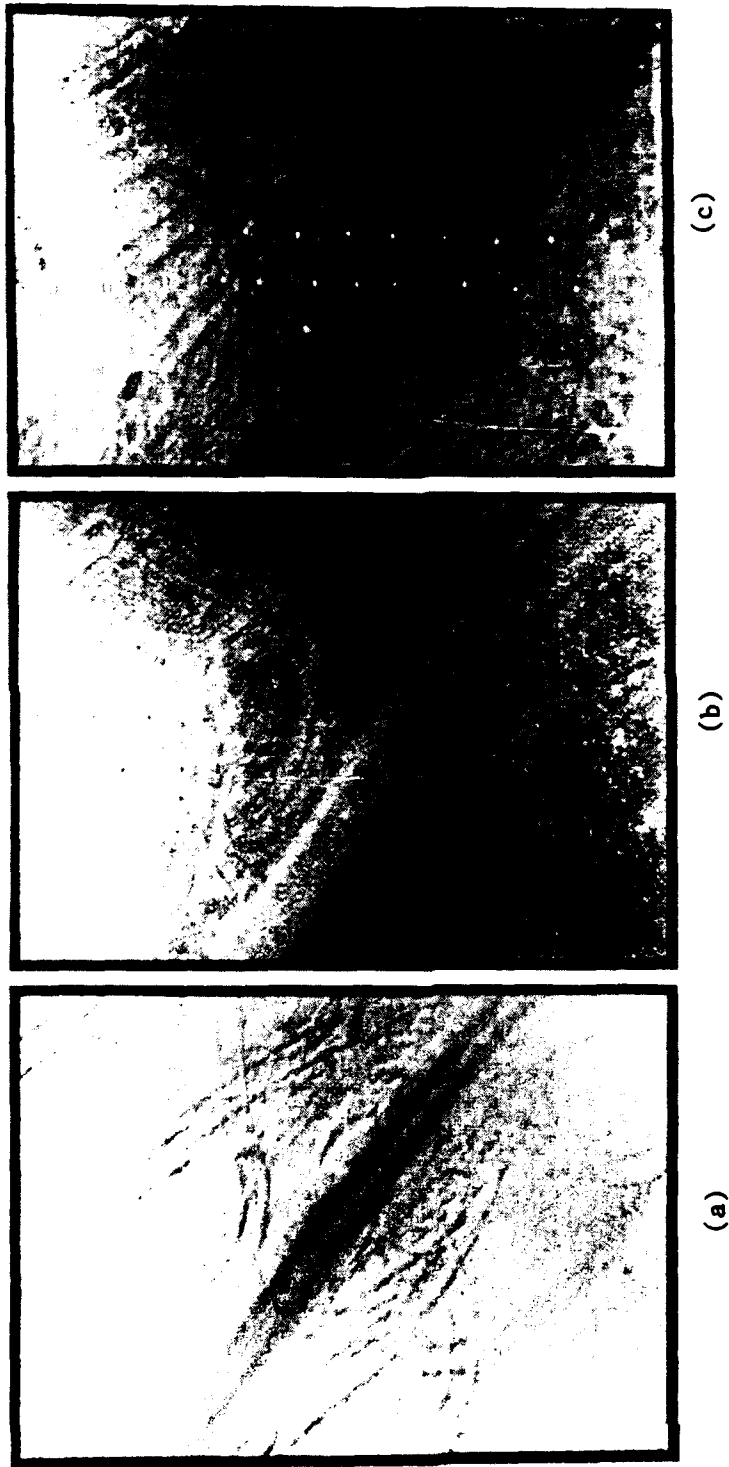


Figure 8. Example of Multisensor Image Data (a) Thermal Channel, (b) Reflectance Channel, and (c) Polarization Channel

materials that have complex reflectivities, and derivation of analytical solution of the electromagnetic boundary conditions present on realistic rough surfaces. For a more complete study of the scattering of electromagnetic waves from randomly rough surfaces, references [48] and [49] are recommended. It should be pointed out, however, that a generalized theory which explains results of actual polarization backscattering experiments has yet to be formulated.

Image Processing Equipment

The data from the multisensor imaging system were stored, processed, and displayed by means of the equipment shown in Figure 9. The digital data are stored on a Honeywell VLDS helical scanning magnetic tape recorder capable of storing 5.1 GBytes of data on one VHS tape cartridge. This recorder was used in this study to feed the multichannel digital data into a massively parallel processor at the same rate (3.3 MBytes/second) as the data were collected in order to conduct real-time compression tests. The real-time processing system consists of a 64 by 64 array of processors (Active Memory Technology's Distributed Array of Processors Model DAP-610) operating synchronously at 10 MHz, and capable of 40 Giga Operations per second at a maximum input/output data rate of 100 MBytes per second. Additional analyses were conducted off-line by means of an International Imaging Systems IVAS image processor which uses a Concurrent Computers MC-6450 multiprocessor as its host. The hardcopies of the imagery included in this study were produced on a Toyo TPG-4300 color video printer and by photographic devices.

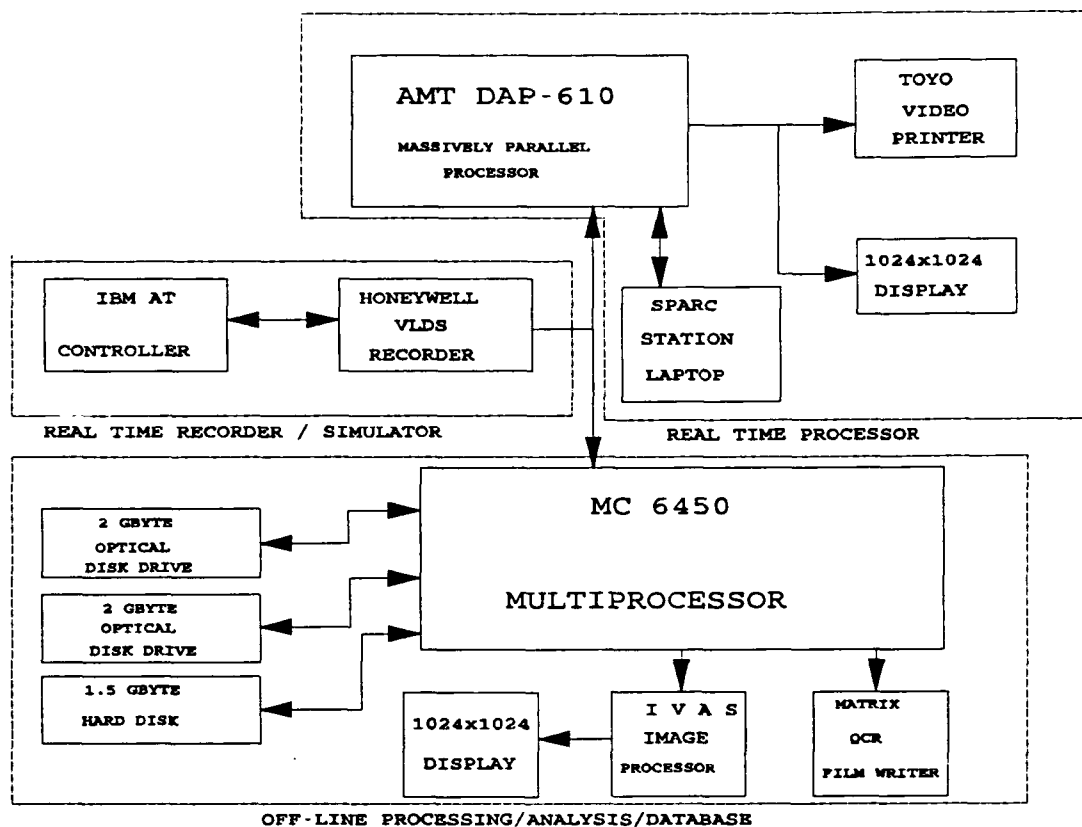


Figure 9. Block Diagram of Image Processing Hardware

CHAPTER IV

ANALYSIS OF MULTISENSOR IMAGERY

Introduction

The purpose of this chapter is to present a systematic analysis of three image channels (P, C, and Thermal) of the multisensor system. The goal of this analysis is to develop a methodology for exploiting the source characteristics, and for determining the range of parameters that affect the compressibility of the multisensor image data. A careful attempt has been made to prevent this study from becoming an ad-hoc attempt at developing a compression algorithm that is optimized for a limited number of specific images. It is envisioned that the techniques developed will be applicable to future generation multisensor systems whose operational characteristics (e.g. spatial resolution and wavelengths) may vary from those considered here.

The analysis consists of initially defining the interchannel correlations and developing decorrelating techniques that take into account the physics and characteristics of the image sources. The decorrelated channels are then analyzed in order to develop mathematical models that will be used in the next chapter for the development and evaluation of the adaptive image compression algorithms. The

analysis of the individual channels is divided into spatial (or data) domain, and spatial frequency (or spectral) domain characterization of the multisensor data. In the spatial domain analysis, individual channel's entropy, dynamic range, interpixel correlation coefficients, probability density, and covariance functions are examined. Spatial frequency domain analysis is also included in order to evaluate the applicability of various transform-based compression methods.

It should be noted that the analyses of the P and the C image channels described in this chapter were performed after any required pre-processing to remove coherent laser power noise (Appendix A).

Interchannel Correlation Analysis

A large percentage of published multichannel image compression schemes tend to ignore the significant correlations present between channels. The primary reason is that implementation of standard techniques to remove these correlations result in a considerable increase in system complexity. For example, the optimal decorrelating technique, the Karhunen-Loeve (KL) transform, requires estimating the interchannel covariance matrix, solving its characteristic equation to find the eigenvalues, and then solving for the corresponding eigenvectors. These eigenvectors are then used to form linear combinations of the original channels that result in totally decorrelated channels which can then be efficiently coded for transmission. An inverse transform operation is also required at the receiver in order to reconstruct the original channels. The computational expense of this technique has led to some simplified schemes that

circumvent the difficulties of implementing the KL transform. For example, when compressing three-channel red, green, blue (RGB) imagery, a fixed transformation to the luminance, inphase, quadrature (YIQ) space is made so the varying sensitivities of the HVS can be exploited. The coding scheme involves allocating more bits to the Y channel since it contains most of the visible detail, and significantly fewer bits to the I and Q channels that contain the chrominance information where the HVS has reduced bandwidth [9]. In the majority of other multichannel schemes, the individual channels are coded independently. This results in decidedly suboptimal coding if there are significant interchannel correlations. In this section, we will employ statistical methods using MSE rather than HVS criteria to decorrelate the channels of the multisensor system.

A large number of 3-band images, ranging in size from 710 x 1024 to 710 x 20,000 pixels per band were analyzed. These images were collected at various times of day, background environments, and resolutions. Table 1 shows that the correlations between the P and C channels were consistently around 0.9 while the correlations between the thermal and the P and C channels varied significantly with the time of day, but were consistently below 0.30.

The conclusion that can be drawn from this analysis is that there is a considerable amount of redundancy in the P and C channels that should be removed for efficient compression. In addition, it can be seen that there is little to be gained from attempting to remove the redundancies between the active and the thermal

channels since their correlations are quite low and are very time of day and background type dependent. This characteristic of the imagery is to be expected, since the thermal channel is sensing a totally different phenomenon (emitted IR energy) than the polarization sensitive reflectance measured by the P and C channels. The remainder of this section will be devoted to decorrelating the P and C channels only.

TABLE 1
INTERCHANNEL CORRELATION COEFFICIENTS

BACKGROUND	TIME	AVERAGE GROUND RES.	AVERAGE	CORREL.	COEFF.
			P/C	P/Th	C/Th
Tall Grass, Standing Water	1800	2"	0.937	-0.172	-0.186
Tall Grass, Standing Water	0800	1.8"	0.914	0.179	0.183
Short Grass, Plowed Fields	1350	2"	0.864	0.256	-0.50
Short Grass, Plowed Fields	0010	2"	0.830	-0.297	+0.276
Short Grass, Plowed Fields	1430	2"	0.963	-0.149	-0.240
Bare Soil/Sand	1600	3"	0.938	-0.021	-0.066

Several characteristics of the sensor and the image data can be used to define a simple but robust decorrelating technique for the P and C channels. First of all, the vast majority of the imagery consists of natural backgrounds that have predictably

stable ranges of polarization. Additionally, the sensing method is such that P is always greater than or equal to C so that the allowable values of C are quite restricted once P is known. This is illustrated in Figure 10 which is a representative plot of the P & C values for each pixel in a small section of actual imagery that contains targets in grass background.

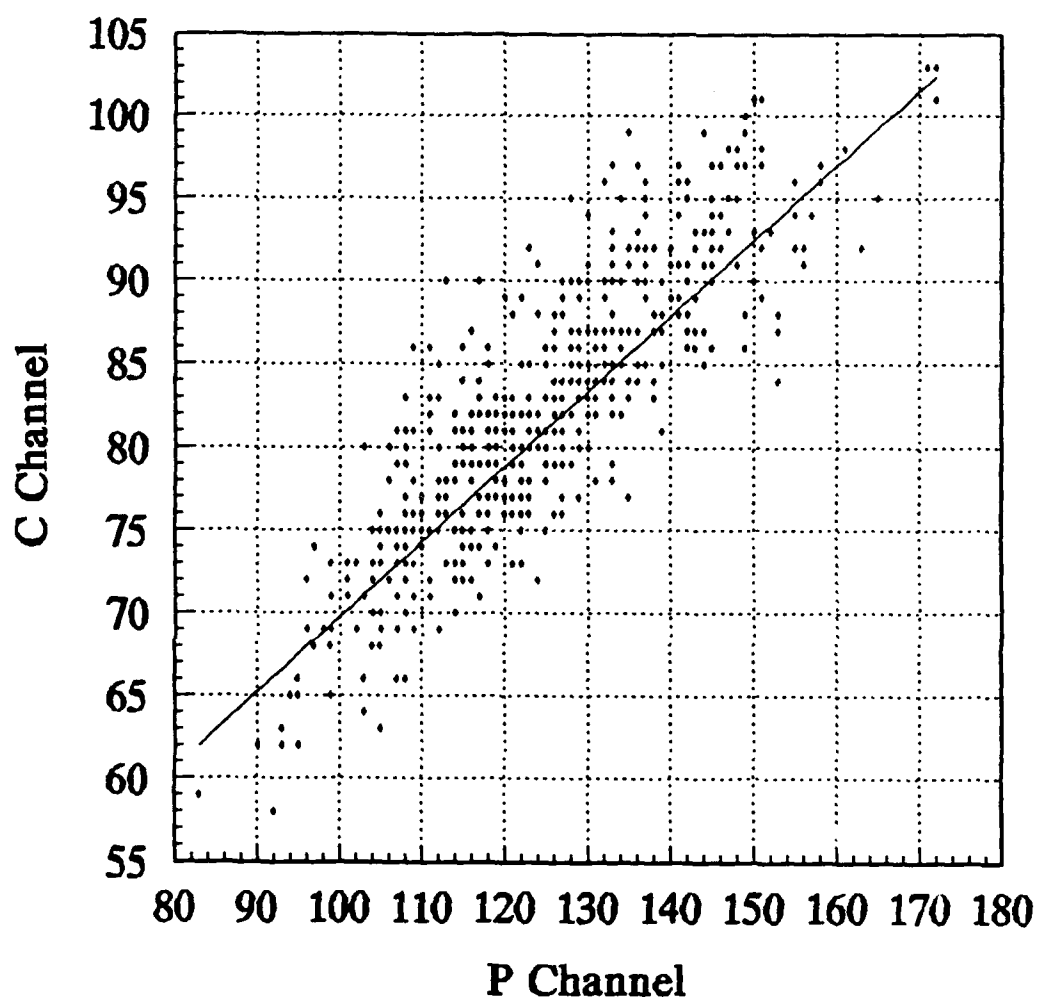


Figure 10. Plot of P and C Image Data

Figure 10 illustrates the fact that the image data tend to cluster tightly around a

line (linear regression line) that defines a constant relationship between P and C . This characteristic results from the fact that a very large percentage of the imagery consists of natural background areas which tend to have limited ranges of polarization. Even over target areas such as a surface-laid minefield, background pixels make up over 99% of all the data in each 1024-line frame of imagery. In order to decorrelate the two channels, a rotation of the principal components (P and C) can be performed such that the maximum amount of variance is concentrated on one of the components (i.e. image channels).

Analysis of very large data sets obtained during airborne and ground-based-polarization field experiments [48], indicate that the backgrounds have polarizations that range from 0.10 for bare soil or sand backgrounds to 0.30 for very dry, dense vegetation. Normal, healthy vegetation such as grass and crop fields have polarizations of approximately 0.20. Assuming that the average polarization of a given background is d , then from the definition of polarization,

$$E [D] = E \left[\frac{P-C}{P+C} \right] = d \quad (4.1)$$

where $0 \leq d < 1$ and E is the expectation operator.

From equation 4.1 and Figure 11, we can calculate the angle ϕ as

$$\phi = \tan^{-1} \left[\frac{1-d}{1+d} \right] \quad (4.2)$$

And the transformation matrix required to rotate the P and C components to the P' and C' components is given by

$$\begin{bmatrix} P' \\ C' \end{bmatrix} = [A] \begin{bmatrix} P \\ C \end{bmatrix} \quad (4.3)$$

$$\text{where } [A] = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

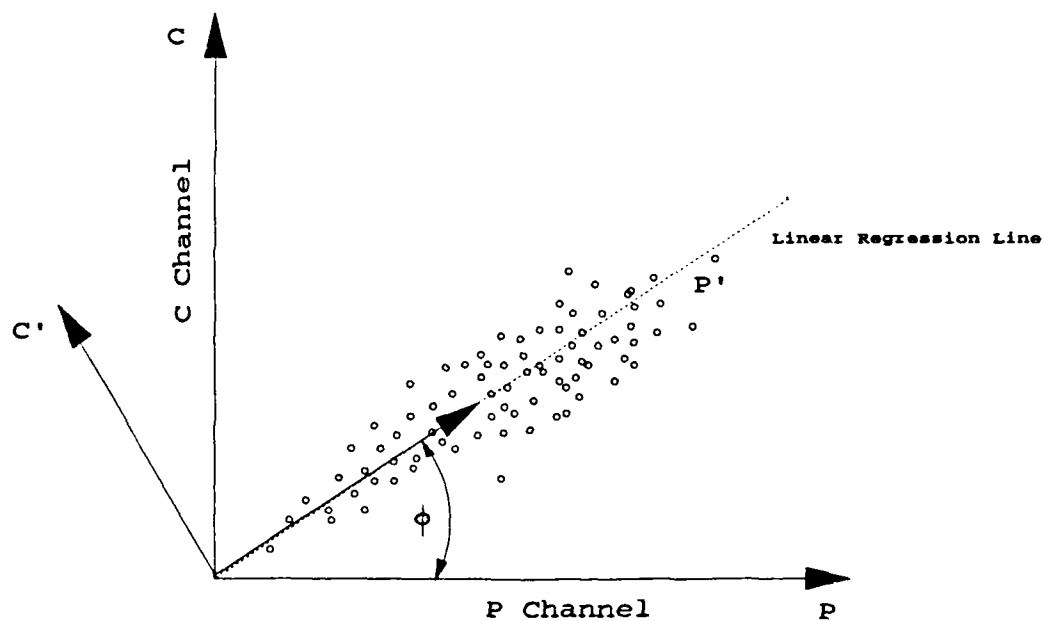


Figure 11. Principal Components Transformation

The $[A]$ matrix corresponds to the discrete KL transform if the angle ϕ defines the least mean squares fit to the data. Since we are using the mean of the

polarization (d) to define ϕ , it is prudent to examine the effect of approximating the optimal KL transform by this simple technique.

A large data file consisting of 14,500 lines (710 pixels per line) of imagery collected over a grassy, wet background was analyzed, and the statistics shown on Table 2 were computed. The KL matrix was formed by column ordering the eigenvectors corresponding to the eigenvalues of the covariance matrix.

TABLE 2
STATISTICS OF P AND C IMAGERY

<i>Mean of Channel P</i> = 55.388
<i>Mean of Channel C</i> = 38.091
<i>Covariance Matrix</i> = $\begin{bmatrix} 203.894 & 127.226 \\ 127.226 & 97.052 \end{bmatrix}$
<i>Correlation Matrix</i> = $\begin{bmatrix} 1.000 & 0.906 \\ 0.906 & 1.000 \end{bmatrix}$
<i>Eigenvalues of Cov. Matrix</i> = 0.959 , 0.041
<i>KL Matrix</i> = $[A] = \begin{bmatrix} 0.833 & 0.554 \\ -0.554 & 0.833 \end{bmatrix}$

Using the average values for P and C, we can calculate $d=0.1850$ and from

equation 4.2, $\phi = 34.519^\circ$. Therefore the computed [A] matrix is

$$[A] = \begin{bmatrix} 0.8239 & 0.5666 \\ -0.5666 & 0.8239 \end{bmatrix}$$

It should be noted that a slightly more accurate approximation of the KL transform matrix can be obtained by calculating the average of the polarization channel (or polarization computed from P and C) directly, but the added storage and calculations required are not justifiable.

The approximation given by equation 4.3 does not require calculation of covariances, eigenvalues or eigenvectors, yet the resulting transformation matrix is very close to the optimal KL transform. The approximation of equation 4.3 was used to rotate the original P and C data, and the computed statistics of these new images are given on Table 3. Comparison of the correlation matrices before and after transformation show that the interchannel correlation is reduced from 0.906 to 0.086. In order to compute the decorrelating efficiency of the approximation we use the formula

$$\eta = \left[1 - \frac{\sum Y}{\sum X} \right] \times 100\% \quad (4.4)$$

where $\sum X$ = sum of the absolute values of the off-diagonal terms of the original

data covariance matrix and $\sum Y$ is the sum of the off-diagonal terms of the transformed data covariance matrix [59]. In the case of the KL transform, the data covariance matrix is diagonalized so that $\sum Y = 0$ and $\eta = 100\%$. Using the data from Tables 2 and 3, the calculated efficiency of the approximate transformation is 96.84%.

TABLE 3
IMAGE STATISTICS AFTER ROTATING BY EQUATION (4.3)

$$\text{Covariance Matrix} = \begin{bmatrix} 90.703 & 2.011 \\ 2.011 & 5.668 \end{bmatrix}$$

$$\text{Correlation Matrix} = \begin{bmatrix} 1.000 & 0.089 \\ 0.089 & 1.000 \end{bmatrix}$$

$$\text{Eigenvalues of Covariance Matrix} = 0.942, 0.058$$

$$\text{New KL Matrix} = \begin{bmatrix} 1.000 & 0.024 \\ -0.024 & 1.000 \end{bmatrix}$$

For the remainder of this study, unless otherwise stated, the analyses and processing will be performed on the rotated P and C image data, and these rotated

channels will be denoted as P' and C'.

Spatial (Data) Domain Analysis

The previous section presented a technique that will be included in the compression algorithm developed in the next chapter. This technique is effective in removing interchannel redundancies prior to coding. In this section we seek to define the amount of intrachannel redundancy that can be removed from the three individual channels (P', C', and Thermal). In the process, those parameters that will be required for the development of an image model will be identified.

One important image property that is useful in estimating the amount of redundancy in a given image is its first order, or single symbol, entropy which was previously defined in equation 2.1. While the actual application of this parameter is very limited, it is nevertheless useful in estimating the compressibility of the image data. Before rotating the P and C channels, the first order entropies of a number of large image files (approximately 1 MBytes each) were calculated. The calculated values ranged from 5.1 bits per pixel to 6.1 bits per pixel, whereas the original data contained 8 bits per pixel. Huffman coding of these image produced a best case of 35.8% compression and a worst case of 23.6%. While the compression ratios achieved by Huffman coding are not high, comparison with documented analyses of standard images of natural scenes [9] whose entropies range from 6 to 7.5 bits per pixel indicate that the multisensor imagery used in this

study has greater redundancy and should therefore be more compressible. Higher-order entropies that determine the redundancy present between groups of pixels are significantly more difficult to compute since the number of possible combinations of groups of pixels increase rapidly. Clarke [10] has speculated that the results obtained using higher order entropies do not justify the computational effort required. In addition, since such calculations are too computationally intensive to be of use in real-time implementations, they are not considered in this study.

A more useful set of parameters are the interpixel correlation coefficients, in particular, the single step horizontal and vertical correlation coefficients ρ_h and ρ_v . Considering a single line of image data as a 1-D sequence $x(i)$ for $i=1$ to N , then the single step horizontal correlation coefficient is defined as

$$\rho_h = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} (x(i) - \bar{x})(x(i+1) - \bar{x})}{\frac{1}{N} \sum_{i=1}^N (x(i) - \bar{x})^2} \quad (4.5)$$

That is, ρ_h is the ratio of the one-step autocovariance of a horizontal line of image data to the zero-step autocovariance of the same line. Performing the calculation of equation 4.5 down a column of image results in the vertical correlation coefficient ρ_v . Application of equation 4.5 to all rows or columns of an image and averaging the results provides a usable estimate of these parameters. At this point it should be noted that the nonstationarities of actual image data have been

neglected. In this case we have assumed at least wide sense stationarity since we are assuming constant mean ($\bar{x}$) for each line or column and an autocovariance that is a function of shift (step) and independent of spatial location. Regardless of these simplifying assumptions, the parameters thus obtained will prove to be useful in estimating the compressibility of the data, selecting the type of transform, and developing an image model. The nonstationarities of the actual imagery will be accounted for by other means described in the next chapter.

Again, a large number of images (P', C', and Thermal) were processed to determine the range of correlation coefficients. The lowest correlation coefficient found was $\rho_v = 0.58$ in one of the P' images, while most of the Thermal images had correlation coefficients in the range of 0.85 to 0.95. Due to the imaging sensor characteristics, it was observed the vertical correlation coefficient was affected by the ground speed variations of the helicopter platform. The magnitude of ρ_v was found to be inversely proportional to groundspeed, which is as expected since low groundspeed results in stretching (or replication of lines of imagery) which would tend to increase ρ_v , while too high a groundspeed results in a faster change of background than normal (lower ρ_v). Since it is expected that an operational sensor would incorporate automatic velocity to height (V/H) correction, only images that were visually confirmed to have approximately the right aspect ratio were selected. It should be emphasized, however, that even scenes with

proper aspect ratios did not, in general, exhibit isotropic ($\rho_h = \rho_v$) behavior.

The range of correlation coefficients magnitudes (≥ 0.5) provides an indication that transform coding methods are applicable for incorporation in the data compression scheme. The actual transform type will be selected by examining the image model developed later in this chapter.

There are a number of other spatial parameters that are of interest, such as probability density function (pdf) and amplitude distributions (histograms), but since our scheme will incorporate transform coding, these parameters are less important than their counterparts in the transform domain. In subsequent sections we will consider the pdf and distributions of the transform coefficients.

Spatial Frequency (Spectral) Domain Analysis

In view of the fact that the compression scheme developed in this study is based on a 2-D transform, which is basically a spatial frequency domain process, it is appropriate that we examine the properties of the multisensor imagery in this domain.

In this section, some representative samples of multisensor imagery are presented in both the data and the frequency domain in order to illustrate a number of properties of the imaging source. In addition, spatial frequency (or spectral) analysis of the image data was performed to estimate the compressibility of the source and the applicability of transform-based compression schemes. Finally, spectral analysis was used to identify and remove image distortions caused by noise

in the laser imaging system (Appendix A).

The 2-D power spectra shown in this section were calculated by computing the square of the magnitude of the 2-D FFT of 512 x 512 pixel areas of image data. No data windowing or smoothing was used since these techniques are not normally used in transform coding applications. The computed magnitudes were mapped into the range of 0 to 255 (8 bits) and displayed as images where the lighter tones correspond to the higher values. The horizontal and vertical axis have been normalized by the sampling frequency (1.05 MHz horizontal and 350 Hz vertical) and range from -0.5 to +0.5 of the sampling frequency. The zero frequency component has been moved to the center of the image by using the technique described in [5]. As was the case in the spatial domain analysis, we must assume that the image data is stationary in order to apply transform techniques.

Examples of P' , C' and thermal imagery are shown in Figures 12, 13 and 14 respectively. The corresponding 2-D spectral domain representations are shown on Figures 15, 16 and 17. For comparison purposes, a single channel (Red) of RGB imagery is shown in Figure 18 and its corresponding spectrum in Figure 19.

Two important properties that are common to line scanning sensors in general and to thermal-IR sensors in particular, can be observed in these images: (1) the effects of striping caused by the scanning mechanism which result in high amplitude (bright) components along the vertical axis and (2) the fact that thermal-

IR scenes, which appear as smooth, slightly out-of-focus images, have most of their power concentrated in the low frequency components.

Other properties of the imagery, which are unique to this particular sensor, are the vertical stripes that are caused by the high frequency laser power fluctuations (see Appendix A), and the fact that the C' channel has a very low-pass spectrum. This property of the C' imagery is largely due to the fact that the rotation operation applied to the P and C channels removes a large portion of the variance from the C' data.

Comparison of the multisensor image spectra with that of the mandril (Figure 19) indicate that the former should be much easier to compress. Since standard transform-based compression methods, such as JPEG's DCT [9] technique, have successfully compressed the mandril imagery, it should be expected that these techniques would be effective for our purposes. It should be noted, however that JPEG's algorithm exploits the nonlinearities of the HVS, which we are not able to do in this case since MSE is our fidelity criterion, not subjective image quality.

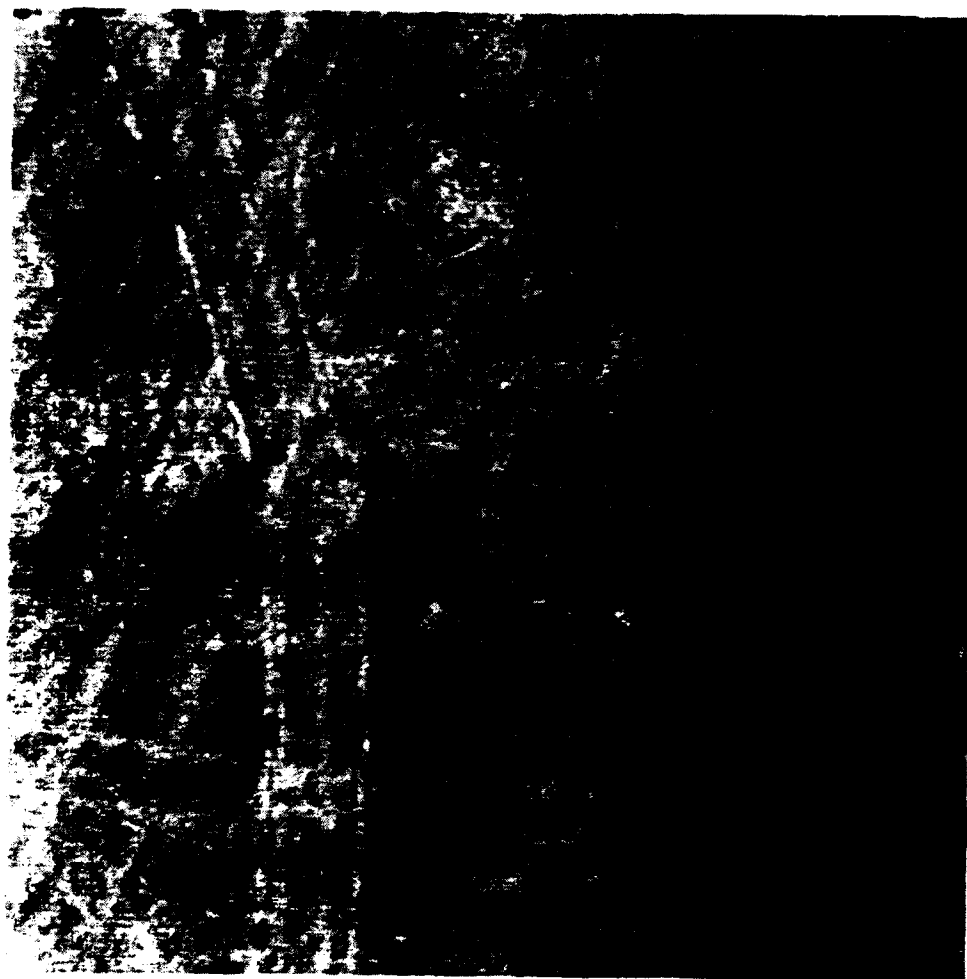


Figure 12. P' Channel Imagery

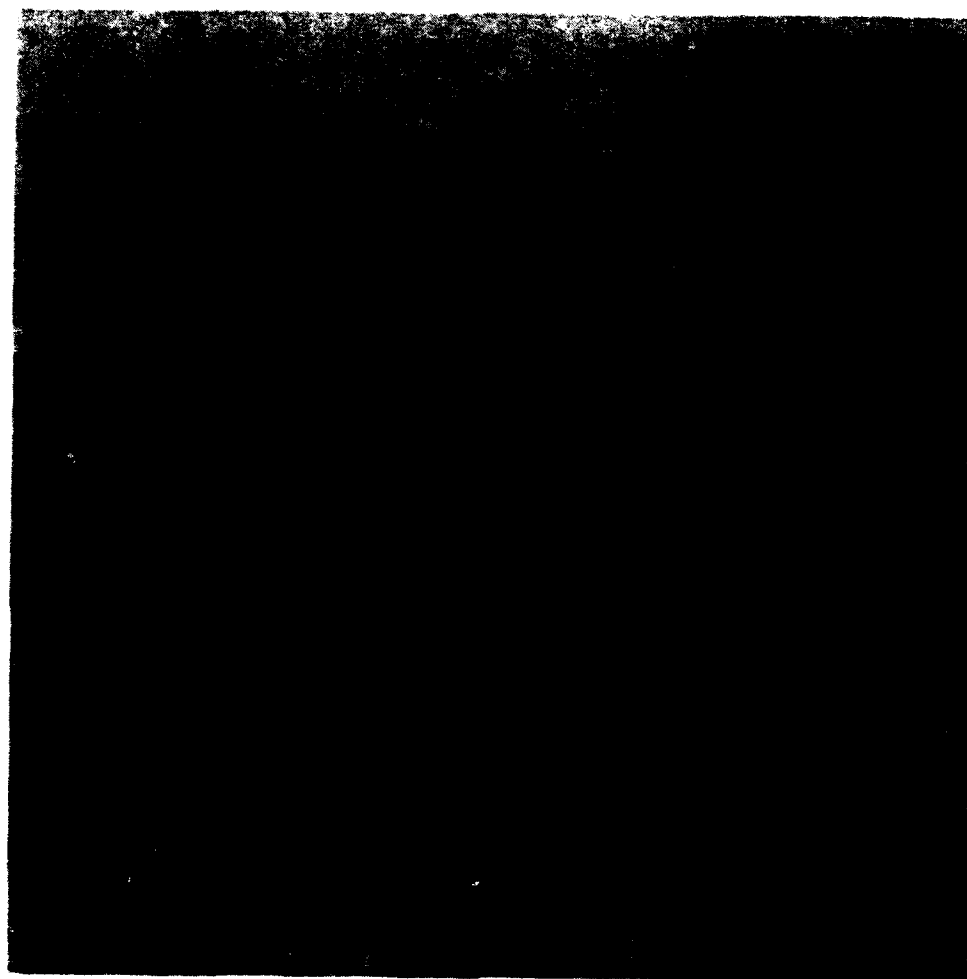


Figure 13. C' Channel Imagery

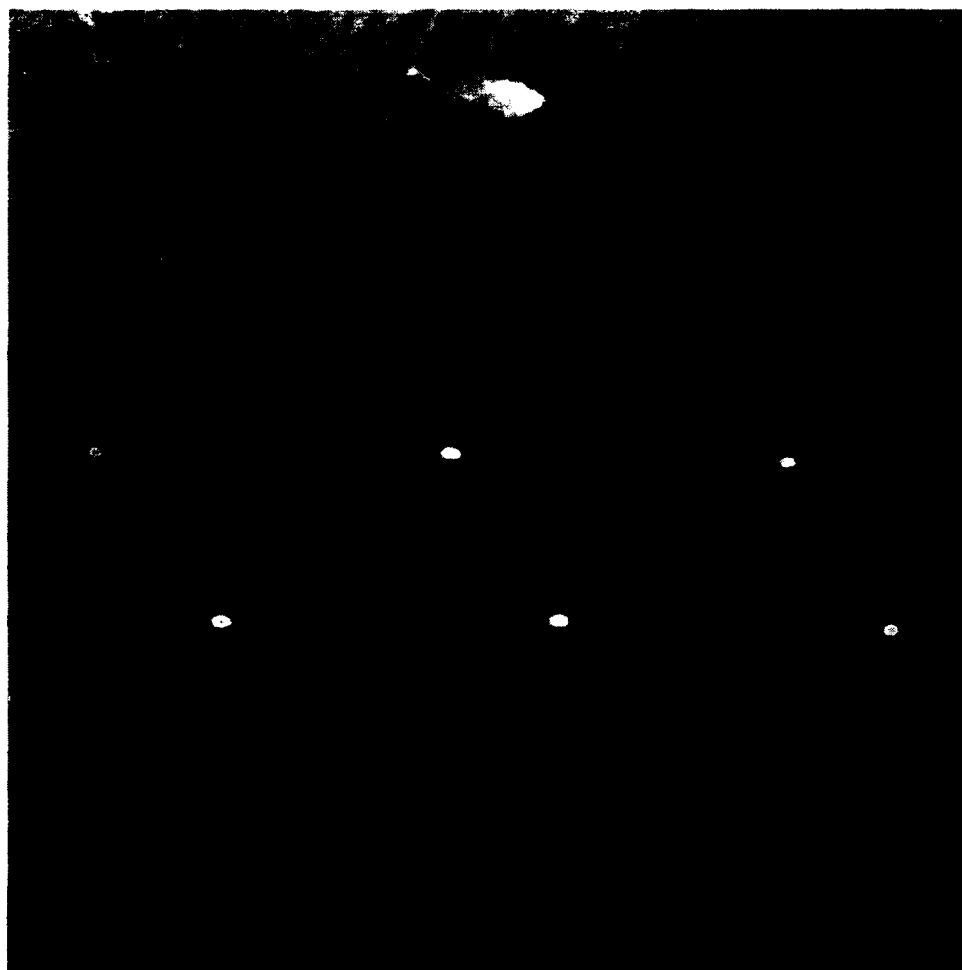


Figure 14. Thermal Imagery

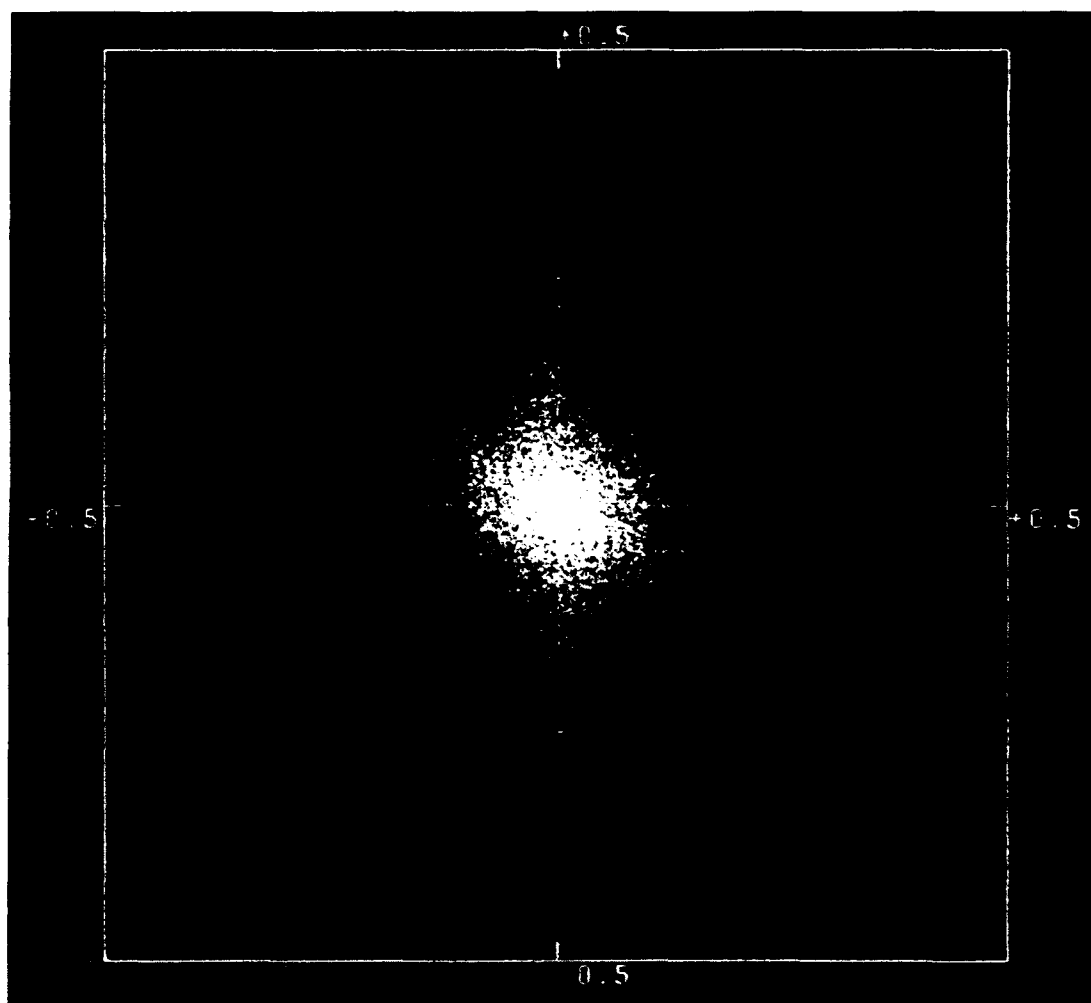


Figure 15. Two Dimensional Power Spectrum of P' Imagery

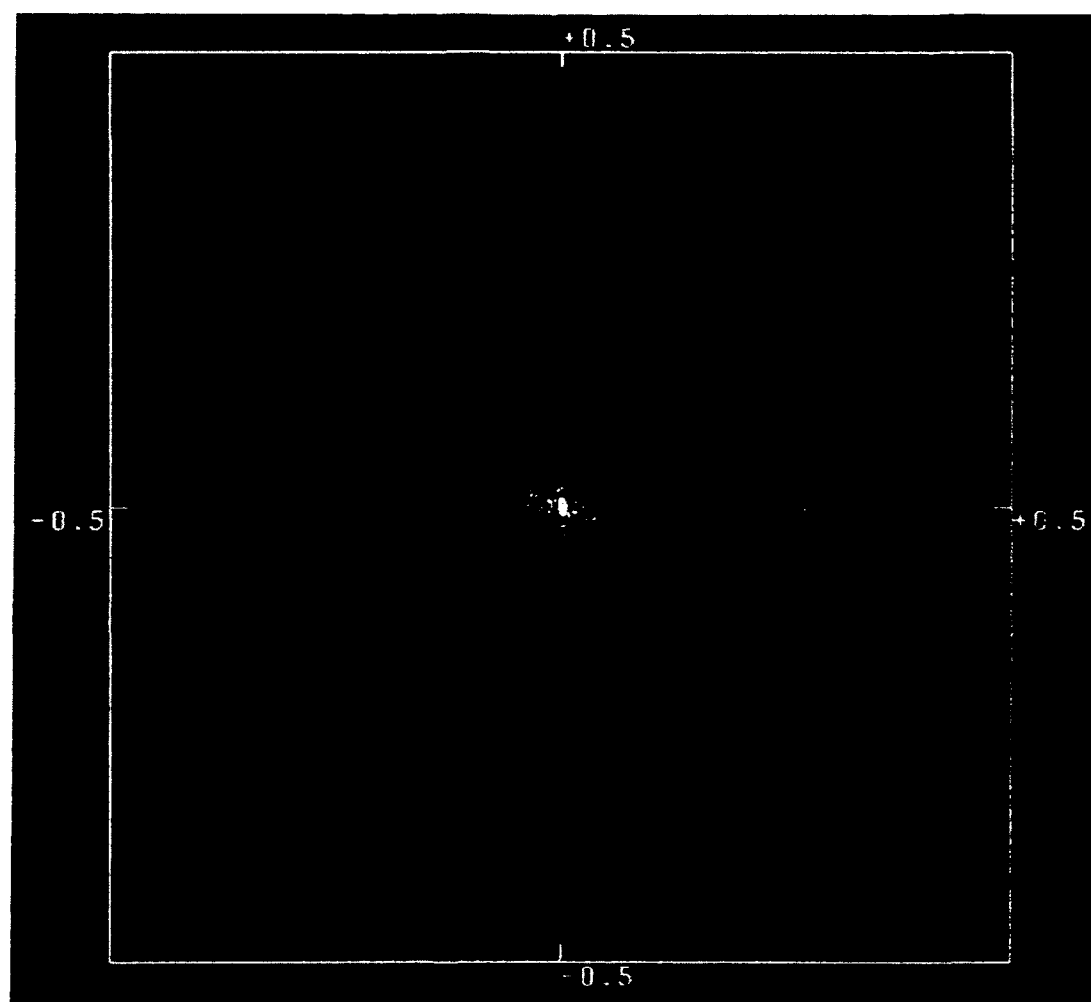


Figure 16. Two Dimensional Power Spectrum of C' Imagery

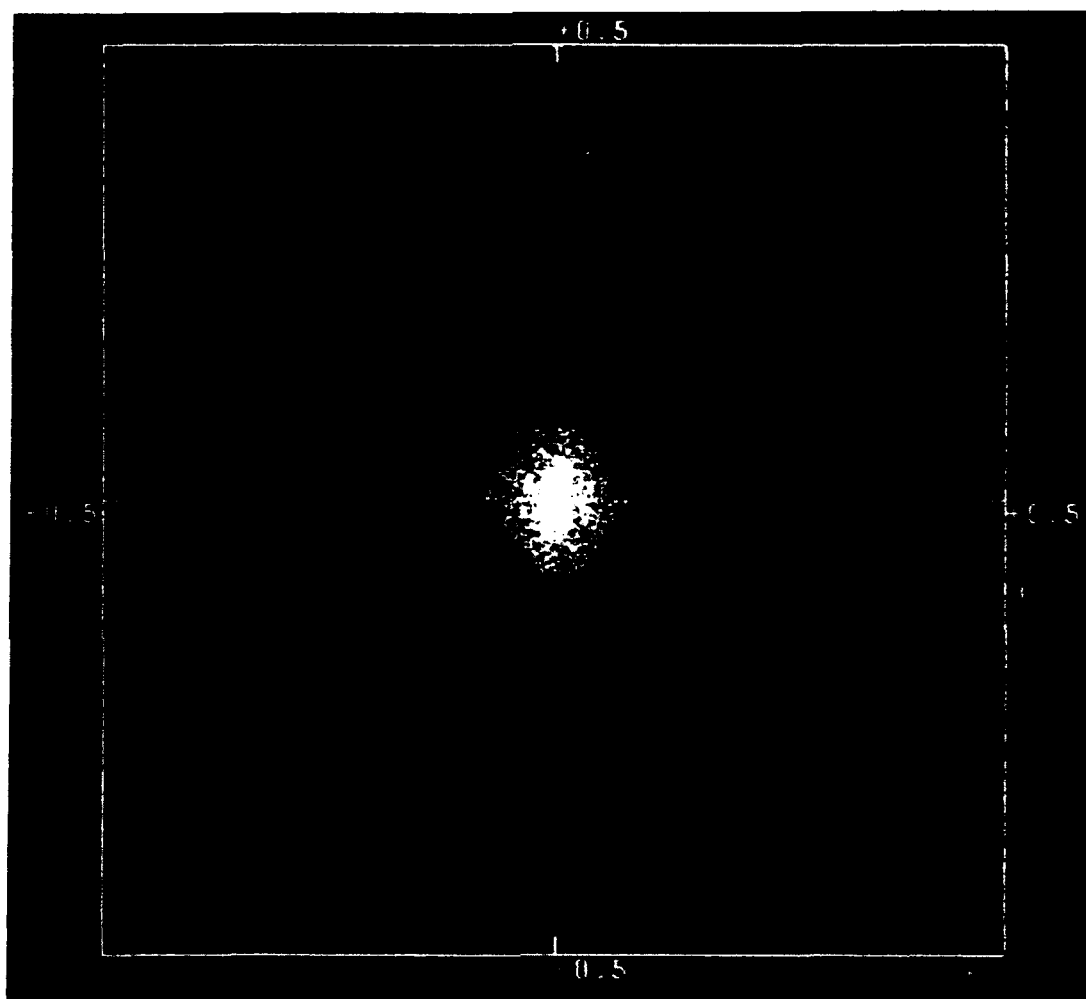


Figure 17. Two Dimensional Power Spectrum of Thermal Imagery



Figure 18. Mandril Imagery (Red Channel)

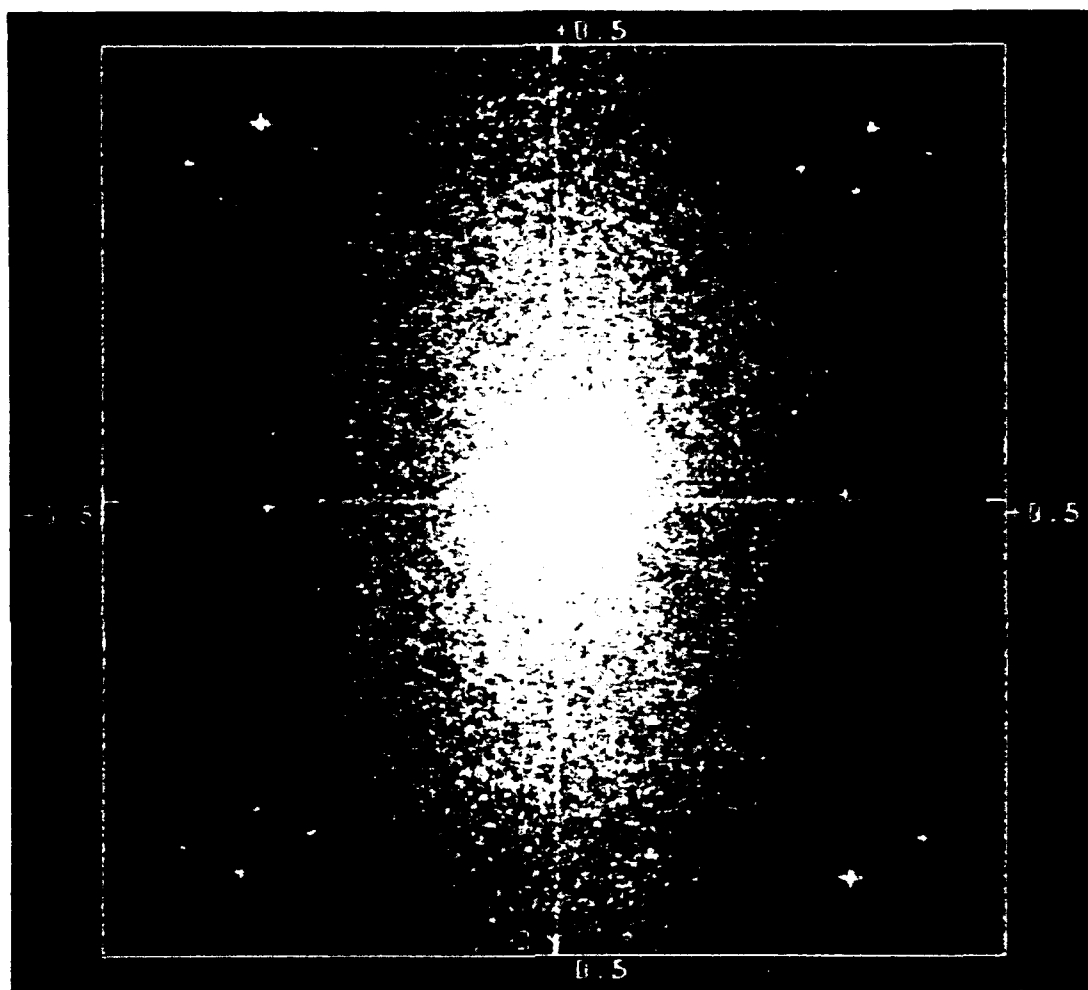


Figure 19. Two Dimensional Power Spectrum of Mandril Imagery

Source Model Selection

The goal of digital image compression can be formulated in a general signal processing framework as the problem of estimating and extracting the useful information from a signal and discarding the non-essential portions. As such, a general approach for attaining this goal consists of three steps [20]:

1. Specification of a criterion function by which the efficiency of various candidate techniques can be evaluated.
2. Development and selection of an adequate model for the source of signal data.
3. Development and implementation of an algorithm.

Step 1 was accomplished earlier when MSE was selected as our distortion metric, and distortion as a function of rate as our efficiency criterion. In this section we focus on step 2, and the model developed in this section will be required for accomplishing step 3 in later chapters.

The importance of selecting an adequate source model is based on the fact that it will be used to determine the following critical components of the adaptive transform coding system:

- a) The choice of transform
- b) The bit allocation strategy
- c) The design of coefficient quantizers
- d) The method of adapting the scheme to nonstationarities of the

actual image data

- e) The type and amount of overhead information that must be transmitted in order to reconstruct the imagery at the receiver.

As mentioned in previous chapters, a number of image source models have been proposed in the literature [11] and these have given rise to a variety of DPCM, transform, and hybrid coding schemes. Since transform coding has been demonstrated to be superior to other coding techniques, this section will focus on image source models that are directly applicable to transform methods. As stated previously, transform coding is used to decorrelate the source data so that the transformed data has the major portion of the variance (energy) concentrated in a small number of coefficients that can in turn be coded with fewer bits. The desired model, therefore, should be such that it can be used to determine the efficiency of different transforms to "pack" the variance into few coefficients, and to predict the distribution of variances so that bits can be optimally allocated to higher energy coefficients. Correlation (or covariance) models are ideally suited for this task since the spatial correlation function can be readily used to determine the variances of the coefficients of any 2-D transform [11]. In this section we will show how the covariance model provides a straightforward link between the spatial domain statistics (i.e. correlation coefficients) and the transform or spectral domain statistics (i.e. distribution of coefficient variances).

In image modeling, the two most widely used correlation models are the

2-D separable and the isotropic models. In the separable model, the 2-D image data is decomposed into independent horizontal and vertical 1-D processes with correlation functions that are assumed to be exponential, that is,

$$\rho_h(x) = e^{-\alpha x} \quad (4.6)$$

and

$$\rho_v(y) = e^{-\beta y} \quad (4.7)$$

where α can be estimated from equations 4.5 and 4.6 by setting $x = 1$, and β can be likewise computed. The separable 2-D correlation model results from combining equations 4.6 and 4.7 into

$$\rho(x,y) = e^{-\alpha x} e^{-\beta y} = \exp[-(\alpha x + \beta y)] \quad (4.8)$$

Natarajan and Ahmed [59] showed that this is a poor model for image data sources. This is due to the fact that the model correlation falls off much more rapidly with increasing diagonal distance than the actual data. In an attempt to correct this problem, Natarajan and Ahmed proposed a nonseparable model which, because it assumed that $\rho_h = \rho_v$, is called the isotropic model and it is given by

$$\rho(x,y) = \exp[-\alpha(x^2 + y^2)^{\frac{1}{2}}] \quad (4.9)$$

Mauersberger [54] showed that this model overcompensates for the shortcomings of the separable model and therefore fails to decrease as rapidly as the actual data. Mauersberger, therefore, proposed what he called a generalized correlation model defined as

$$\rho(x,y) = \exp\left[-\left[(\alpha x^{r_1})^h + (\beta y^{r_2})^h\right]^{\frac{1}{h}}\right] \quad (4.10)$$

where α and β are found, as before, from equations 4.5, 4.6, and 4.7. The parameters r_1 , r_2 and h must be estimated by solving a constrained optimization problem using actual image data. It can be observed that both the separable and the isotropic models can be derived from the generalized model by proper selection of the parameters.

In [54] Mauersberger estimated the parameters of the generalized correlation model by assuming that $\alpha = \rho$, and then using a set of fourteen test images obtained from a wide variety of sources. Akansu and Haddad [55] performed the parameter optimization with the same test set as Mauersberger, but without constraining the horizontal and vertical correlation coefficients (i.e α and β) to be equal. Their optimized parameter values are $r_1 = 1.137$, $r_2 = 1.09$, and $h = \sqrt{2}$. Clarke [10] also determined that $h = \sqrt{2}$ was optimal for a different test image set. Since in this study we are interested in methodology, rather than in optimizing a scheme for a particular (and unique) sensor, we will use these

parameters as a starting point. The interested reader can apply the iterative algorithm described in [60] in order to optimize the model for a particular source.

Having selected a correlation model for our study, it is now necessary to define the process required to transition to the transform domain where the image data compression will be effected. The process is best illustrated by considering the 1-D case, where our data is assumed to be a stationary random sequence defined in vector form as

$$\mathbf{x} = [x(1), x(2), \dots, x(N)]^T \quad (4.11)$$

If this sequence is transformed by means of a unitary matrix $[A]$ (i.e. $A^{-1} = A^T$) then the transformed data is given by the vector of coefficients $\mathbf{X}$ such that

$$\mathbf{X} = [A]\mathbf{x} \quad \text{where} \quad \mathbf{X} = [X(1), X(2), \dots, X(N)]^T \quad (4.12)$$

Each component of $\mathbf{X}$ is found from equation 4.12 to be

$$X(i) = \sum_{m=1}^N a(i,m)x(m) \quad \text{for } i = 1, 2, \dots, N \quad (4.13)$$

Without loss of generality, we can assume that our data sequence $\mathbf{x}$ has zero mean, and this will result in a covariance function equal to the correlation function. Under this assumption, the expected value of each of the coefficients will be zero,

and their variances will be

$$\begin{aligned}
 \sigma^2(i) &= E[(X(i))^2] \\
 &= E\left[\sum_m^N \sum_n^N a(i,m) a(i,n) x(m) x(n)\right] \\
 &= \sum_m^N \sum_n^N a(i,m) a(i,n) R(m-n)
 \end{aligned} \tag{4.14}$$

where $R(m-n)$ represents the autocorrelation function which is defined as

$$R(m) = E[x(m+n)x(n)] \tag{4.15}$$

In [55] and [61] Haddad and Akansu derive a new expression for equation 4.14

$$\sigma^2(i) = \sum_{k=0}^{N-k} w(i,k+1) R(k) \tag{4.16}$$

where

$$w(i,k+1) = \begin{cases} g(i,k) & \text{for } k = 0 \\ 2g(i,k) & \text{for } k = 1, 2, \dots, N-1 \end{cases} \tag{4.17}$$

and

$$g(i,k) = \sum_{j=1}^{N-k} a(i,j) a(i,j+k) \tag{4.18}$$

Equations 4.14, 4.16, and 4.17 can then be combined in vector/matrix form as

$$\sigma^2 = [W] R \quad (4.19)$$

In equation 4.19, the W matrix provides the link between the signal's autocorrelation function in the spatial domain, and the distribution of coefficient variances in the spectral or transform domain. In [55] and [61] Haddad and Akansu computed the W matrix for some of the more commonly used transforms, the DCT and the Walsh-Hadamard transform (WHT), and also for the less popular Modified Hermite transform (MHT). It should be noted, however, that both references have numerous typographical errors in the tabulation of all three W matrices. A tabulation of the correct W matrices for these three transforms is presented in Chapter 5 (Table 11).

Akansu and Haddad extended the 1-D case to include the nonseparable 2-D correlation function $R(m,n)$, and derived the link between the spatial and the transform domains as

$$[V] = [W][R][W]^T \quad (4.20)$$

where

$$[V] = [\sigma^2(i,j)] \quad (4.21)$$

and

$$[R] = [R(m,n)] \quad \text{for } 0 \leq m, n \leq N-1 \quad (4.22)$$

Equation 4.20 represents a closed form expression that separates the transformation from the correlation or covariance model so that, for a given model, the effect of selecting a specific transform can be studied. This will be done in the following chapter after we have validated the image model.

At this point we need a method for scaling the model so that it can be applied in equation 4.20, and for subsequent bit allocation and coefficient normalization and quantization calculations. Since the model provides an estimate of the normalized covariance, it can be readily re-scaled to the actual covariances by multiplying by the overall data variance which for an $N \times N$ image is given by

$$\sigma_x^2 = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N (x(m,n) - \bar{x})^2 \quad (4.23)$$

where $\bar{x}$ is the mean of the image data values.

Thus our adopted model for the remainder of this study is

$$R(x,y) = \sigma_x^2 \exp \left[- \left[(\alpha x^{1.137})^{\sqrt{2}} + (\beta y^{1.09})^{\sqrt{2}} \right]^{\frac{1}{\sqrt{2}}} \right] \quad (4.24)$$

Source Model Verification

A number of 512x512 sections of multisensor images were analyzed for the purpose of verifying the coefficient variance estimates derived from the generalized covariance model given in equation 4.24. The procedure used was to divide each image into 4096 blocks of 8x8 pixels. These blocks were then individually

transformed via a fast DCT algorithm (described in Chapter 6), and the actual variances of the 64 individual DCT coefficients were calculated over the 4096 values. The actual variances were then compared with those obtained from the generalized covariance model. The model parameters were estimated by computing global averages of the horizontal and vertical single-step correlation coefficients (equation 4.5) and of the overall data variance (equation 4.23). These parameters were then used to calculate the 8x8 matrix of covariances (equation 4.24). Lastly, these covariances were then used to calculate the DCT coefficient variances by means of equation 4.20.

The actual and modelled DCT coefficient variances for 3 channels of typical imagery are shown in Tables 4, 5, and 6. In these tables, the top left hand value of each 8x8 array corresponds to the DC coefficient, and the bottom right hand value corresponds to the highest 2-D frequency coefficient. Inspection of these results indicate that the model provides a reasonably accurate approximation to the actual coefficient variances. The model is quite accurate in estimating the variances of the coefficients corresponding to the horizontal frequencies of the P' and C' channels, but is considerably poorer in predicting the variances of the vertical frequency coefficients. Conversely, the model is quite accurate in predicting the variances of the vertical frequency components of the Thermal channel, and has larger deviations for the horizontal frequencies. It should be pointed out that the model parameters, other than the correlation coefficients, were not optimized to the actual image data;

TABLE 4
DISTRIBUTION OF DCT COEFFICIENT VARIANCES - P' CHANNEL

Actual Variances

10478.9	1064.67	357.99	115.57	55.44	26.33	14.36	8.18
1013.0	353.11	186.89	93.84	45.09	25.55	13.49	7.88
426.27	201.59	134.15	77.64	40.60	22.74	13.27	7.79
231.93	130.67	98.57	59.52	36.92	20.54	12.74	7.59
189.91	108.36	76.11	50.69	31.90	20.18	11.22	7.25
150.66	95.27	70.02	45.04	30.23	18.79	11.55	7.59
138.85	84.55	64.44	43.20	28.07	17.88	11.39	6.97
131.89	75.98	63.10	41.09	26.48	17.28	10.75	6.71

Modelled Variances

6747.07	1226.6	321.7	111.2	58.76	37.40	26.13	22.40
2972.12	669.75	208.1	79.98	44.04	28.86	21.17	18.19
1368.33	383.5	141.6	61.04	35.26	23.89	18.17	15.73
628.37	218.84	96.30	47.17	28.90	20.36	15.99	14.00
350.50	140.87	70.66	38.47	24.90	18.17	14.62	12.92
225.78	100.81	55.75	32.94	22.32	16.76	13.73	12.24
164.68	79.42	47.09	29.51	20.69	15.86	13.16	11.80
137.03	68.97	42.59	27.64	19.79	15.37	12.84	11.57

TABLE 5
DISTRIBUTION OF DCT COEFFICIENT VARIANCES - C' CHANNEL

Actual Variances

285.0	77.55	30.34	12.89	6.38	3.53	2.13	1.40
88.68	39.26	19.86	9.91	5.48	3.23	2.02	1.37
44.95	22.51	13.56	8.15	4.83	3.01	1.95	1.30
23.05	15.56	9.76	6.54	4.24	2.76	1.76	1.28
15.74	10.94	7.74	5.56	3.74	2.39	1.61	1.21
11.57	8.39	6.505	4.92	3.41	2.39	1.63	1.23
9.88	7.96	6.399	4.45	3.26	2.26	1.60	1.14
9.02	6.85	5.71	4.27	2.95	2.13	1.56	1.06

Modelled Variances

197.1	76.73	31.04	12.7	6.75	4.27	3.10	2.60
117.0	49.8	22.06	9.79	5.44	3.56	2.65	2.24
67.4	31.81	15.55	7.58	4.44	3.01	2.30	1.98
36.1	19.08	10.46	5.69	3.57	2.53	2.00	1.74
21.2	12.27	7.39	4.42	2.96	2.19	1.78	1.58
14.1	8.68	5.62	3.63	2.56	1.96	1.63	1.46
10.4	6.76	4.61	3.14	2.30	1.82	1.54	1.39
8.74	5.82	4.09	2.88	2.17	1.74	1.48	1.35

TABLE 6

DISTRIBUTION OF DCT COEFFICIENT VARIANCES - THERMAL CHANNEL

Actual Variances

2439.96	260.9	75.54	21.36	9.12	4.03	1.49	1.33
413.79	92.63	33.15	13.50	6.78	3.19	1.36	.729
135.93	46.04	21.69	12.39	5.71	3.32	1.38	.572
42.02	21.23	14.62	8.95	4.96	3.35	1.53	.569
32.37	14.13	10.27	6.88	4.49	3.01	1.54	.539
21.18	10.33	8.57	6.53	4.27	3.04	1.50	.538
13.71	9.62	7.85	6.51	4.06	2.99	1.48	.526
16.35	8.58	7.33	5.67	4.07	2.83	1.62	.529

Modelled Variances

1851.4	332.85	87.07	29.78	15.44	9.64	6.60	5.61
577.75	139.29	46.01	18.23	10.01	6.51	4.76	4.06
213.91	67.26	27.44	12.62	7.41	5.05	3.87	3.34
87.07	34.29	16.91	9.02	5.73	4.12	3.28	2.88
47.03	20.91	11.65	6.94	4.73	3.56	2.92	2.60
29.75	14.45	8.80	5.69	4.09	3.19	2.68	2.42
21.33	11.12	7.21	4.94	3.69	2.96	2.53	2.31
17.75	9.55	6.41	4.55	3.48	2.83	2.44	2.24

therefore, some mismatch in the variance predictions is to be expected. The remainder of this section is devoted to quantifying the effects of the model mismatch on the variance predictions, and on the resulting bit allocation.

A quantitative measure of the accuracy of the coefficient variances predicted by the model is the mean mismatch ratio defined by Mauersberger [54] as

$$f = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \max \left(\frac{\sigma_{i,j}^2}{\bar{\sigma}_{i,j}^2}, \frac{\bar{\sigma}_{i,j}^2}{\sigma_{i,j}^2} \right) \quad (4.25)$$

where σ is the modelled variance, and $\bar{\sigma}$ is the actual variance. Mauersberger, and Akansu and Haddad [55] used the mean mismatch ratio to optimize the model parameters to a set of images. The purpose of selecting the maximum of the ratios of the two variances is to avoid any preference towards too large or too small a variance. The computed mean mismatch ratios for variances shown in Tables 4, 5 and 6 are given in Table 7.

TABLE 7
MEAN MISMATCH RATIOS

P' Channel	1.453
C' Channel	1.227
Thermal Channel	1.899

Mauersberger [54] calculated a mean mismatch ratio of 1.260 for the set of 14 images that he used for the optimization of the generalized correlation model, and as high as 8.670 when a separable model was assumed for the same set of images. The results shown on Table 7 are close to those calculated in [54], and indicate that the generalized covariance model provides a good match to the multisensor imagery without any attempt to adjust or optimize the model parameters.

A quantitative evaluation of the usefulness of the model can be performed by comparing the bit allocation matrices obtained from the actual and the modelled variances. The bit allocation matrix defines the number of bits that will be transmitted for each of the 64 transform coefficients in each 8x8 block. Jayant and Noll [4] have derived the optimal bit allocation strategy for coding coefficients for any arbitrary block size ($N \times N$) using MSE as the optimization criterion. This optimal bit allocation depends only on the distribution of the coefficient variances, and is given by

$$B_{i,j} = B + \frac{1}{2} \log_2 \frac{\sigma^2(i,j)}{\left[\prod_{i,j=0}^{N-1} \sigma^2(i,j) \right]^{\frac{1}{N^2}}} \quad (4.26)$$

where B is the desired overall bit rate (in bits per sample), and $B_{i,j}$ is the number of bits allotted to code the $(i,j)^{th}$ coefficient. Difficulties in implementing this technique are due to the fact that fractional bit allocations normally result, and also negative $B_{i,j}$ are possible when low average bit rates are used. These shortcomings are due to the

fact that the optimization used to derive equation 4.26 was not constrained to only non-negative integer solutions. This property makes actual implementation of equation 4.26 suboptimal, and Rost [24] and Clarke [10] have shown that achievable distortions and rates increase when the computed bit rates are rounded to the nearest integer values and negative bit allocations are set to zero. These problems can be avoided by employing vector coding of the block of coefficients [62], but this technique will not be used in this study because of implementation considerations.

The results obtained from 4.26 by rounding values to the nearest non-negative integer are shown in Tables 8, 9 and 10. These results correspond to the DCT coefficient values given in Tables 4, 5, and 6, and assume an average bit rate B of one bit per pixel. The fact that the bit allocations calculated from the actual variances and from the modelled variances do not deviate by more than one bit for any DCT coefficient, and match exactly in most cases, indicate that the model is adequate for our multisensor image source. Application of this model to the development of the adaptive multisensor image compression scheme is described in the next chapter.

TABLE 8
BIT ALLOCATION MATRIX - P' CHANNEL

Computed from Actual Coefficient Variances

$$\begin{bmatrix} 5 & 3 & 2 & 1 & 1 & 1 & 0 & 0 \\ 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Computed from Modelled Coefficient Variances

$$\begin{bmatrix} 5 & 3 & 2 & 2 & 1 & 1 & 1 & 0 \\ 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

TABLE 9
BIT ALLOCATION MATRIX - C' CHANNEL

Computed from Actual Coefficient Variances

$$\begin{bmatrix} 4 & 3 & 2 & 2 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Computed from Modelled Coefficient Variances

$$\begin{bmatrix} 4 & 3 & 2 & 2 & 1 & 1 & 1 & 0 \\ 3 & 3 & 2 & 1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

TABLE 10
BIT ALLOCATION MATRIX - THERMAL CHANNEL

Computed from Actual Coefficient Variances

$$\begin{bmatrix} 5 & 3 & 3 & 2 & 1 & 1 & 1 & 1 \\ 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Computed from Modelled Coefficient Variances

$$\begin{bmatrix} 5 & 3 & 2 & 2 & 1 & 1 & 1 & 0 \\ 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 2 & 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

CHAPTER V

DEVELOPMENT OF MULTISENSOR IMAGE COMPRESSION SCHEME

Introduction

In this chapter we make use of the previously developed generalized covariance image model to select the transform type, and to design the bit allocation and quantization strategies that will be incorporated in the multisensor image compression scheme. The tradeoffs involved in adapting the transform type, bit allocation, and quantization strategies to the changes in data statistics are considered. A method of adapting the scheme to compensate for the nonstationarities of the image data is developed, and the complete adaptive transform compression algorithm is described. The implementation considerations are covered in the next chapter, and the evaluation of the algorithm is documented in Chapter 7.

Selection of Transform Type

Selection of a transform to be used in image data compression applications involves a number of tradeoffs between MSE performance, implementation considerations, and overhead transmission requirements. For example, for certain types of sources that can be modelled by Markov processes with correlation coefficients approaching 1.0, the KL transform has been shown to be optimal in the MSE sense. However, no general fast algorithm exists for implementing the KL

transform, and its computational complexity makes this transform impractical for real-time applications. In addition, the KL transform's basis vectors are data dependent and must be computed and transmitted as overhead information for proper reconstruction at the receiver. Conversely, transforms that have minimal implementation and overhead requirements such as the Walsh-Hadamard transform can have poor MSE performance for the same class of image data sources.

This section presents a procedure for evaluating the MSE performance of 2-D transforms. It is assumed that the transforms under consideration are limited to those that can be implemented via fast algorithms, and whose basis vectors are data independent so that overhead requirements are reduced. We therefore illustrate the procedure by evaluating the performance of the DCT, the Modified Hermite Transform (MHT), and the Walsh-Hadamard Transform (WHT) whose W matrices are listed in Table 11. W matrices for other transforms can be derived by means of Equations 4.14 through 4.19 [55].

It has been shown [10] that the DCT is superior to other transforms that have data independent basis vectors when the correlation coefficients are near unity and minimum MSE is used as the fidelity criterion. Therefore, the aim of this section is to determine if there are any advantages in changing the transform type when the correlation coefficients of the multichannel imagery are low. Analysis of a large number of 512x512 sections of multisensor images collected at various altitudes and background types indicates that the horizontal correlation coefficients range from 0.99

to 0.75, and the vertical correlation coefficients range from 0.98 to 0.58. A large percentage of the Thermal channel imagery had horizontal and vertical correlation coefficients near 0.96, while the P' and C' imagery averaged 0.93 and 0.90 for the horizontal and vertical correlation coefficients respectively. These values, and also the worst case values of 0.75 and 0.58, will be used to determine the effect of transform selection on compression performance.

The procedure used involves computing the 8x8 data covariance matrix by means of the generalized covariance model of Equation 4.24 using the previously stated correlation coefficient values. Equation 4.20 is then used to calculate an 8x8 coefficient covariance matrix for each of the three transforms of Table 11. Then we make use of a technique developed by Jayant and Noll [4] for determining the performance of a given transform as compared with PCM coding. This technique uses minimum MSE criterion, assumes optimum bit allocation, and is based on calculating the degree to which the total variance is concentrated in a small number of coefficients. The calculation of the gain of transform coding over PCM coding, G_{TC} , for a 2-D non-separable image model is given by

$$G_{TC} = \frac{\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \sigma^2(i, j)}{\left[\prod_{i,j=1}^N \sigma^2(i, j) \right]^{\frac{1}{N^2}}} \quad (5.1)$$

In effect, Equation 5.1 states that the gain of a specific transform over PCM coding

TABLE 11
W MATRICES FOR THREE TYPES OF TRANSFORMS

$$W_{DCT} = \begin{bmatrix} 1 & 1.750 & 1.500 & 1.250 & 1.000 & 0.750 & 0.500 & 0.250 \\ 1 & 1.367 & 0.599 & -0.125 & -0.653 & -0.890 & -0.816 & -0.480 \\ 1 & 0.987 & -0.353 & -1.133 & -1.000 & -0.280 & 0.353 & 0.426 \\ 1 & 0.419 & -1.252 & -1.051 & 0.270 & 0.796 & 0.162 & -0.345 \\ 1 & -0.250 & -1.500 & 0.250 & 1.000 & -0.250 & -0.500 & 0.250 \\ 1 & -0.919 & -0.869 & 1.258 & -0.270 & -0.589 & 0.544 & -0.154 \\ 1 & -1.487 & 0.353 & 0.633 & -1.000 & 0.780 & -0.353 & 0.073 \\ 1 & -1.867 & 1.522 & -1.081 & 0.653 & -0.316 & 0.108 & -0.019 \end{bmatrix}$$

$$W_{MHT} = \begin{bmatrix} 1 & 1.855 & 1.479 & 1.002 & 0.563 & 0.252 & 0.082 & 0.015 \\ 1 & 1.510 & 0.415 & -0.586 & -0.996 & -0.820 & -0.413 & -0.019 \\ 1 & 1.019 & -0.606 & -1.167 & -0.392 & 0.565 & 0.744 & 0.328 \\ 1 & 0.365 & -1.288 & -0.579 & 0.825 & 0.637 & -0.413 & -0.546 \\ 1 & -0.365 & -1.288 & 0.579 & 0.825 & -0.637 & -0.413 & 0.546 \\ 1 & -1.019 & -0.606 & 1.167 & -0.392 & -0.565 & 0.744 & -0.328 \\ 1 & -1.510 & 0.415 & 0.586 & -0.996 & 0.820 & -0.413 & 0.109 \\ 1 & -1.855 & 1.479 & -1.002 & 0.563 & -0.252 & 0.082 & -0.015 \end{bmatrix}$$

$$W_{WHT} = \begin{bmatrix} 1 & 1.750 & 1.500 & 1.250 & 1.000 & 0.750 & 0.500 & 0.250 \\ 1 & -1.750 & 1.500 & -1.250 & 1.000 & -0.750 & 0.500 & -0.250 \\ 1 & 0.250 & -1.500 & -0.250 & 1.000 & 0.250 & -0.500 & -0.250 \\ 1 & -0.250 & -1.500 & 0.250 & 1.000 & -0.250 & -0.500 & 0.250 \\ 1 & 1.250 & 0.500 & -0.250 & -1.000 & -0.750 & -0.500 & -0.250 \\ 1 & -1.250 & 0.500 & 0.250 & -1.000 & 0.750 & -0.500 & 0.250 \\ 1 & 0.750 & -0.500 & -0.750 & -1.000 & -0.250 & 0.500 & 0.250 \\ 1 & -0.750 & -0.500 & 0.750 & -1.000 & 0.250 & 0.500 & -0.250 \end{bmatrix}$$

equals the ratio of the arithmetic mean of the coefficient variances to their geometric mean.

Analysis of the results presented in Table 12 indicates that, as expected, DCT is considerably superior to the other transforms at higher values of correlation coefficients. The advantage of DCT over the other transforms and over PCM decreases with decreasing correlation coefficients; however, DCT is still the best performer for the range of coefficients representative of the multisensor imagery. The conclusion that can be drawn from these results is that, for the image data sources considered in this study, there is little to be gained from switching transform types to adapt to changing data statistics.

TABLE 12
TRANSFORM GAIN FOR THREE TYPES OF TRANSFORMS

ρ_h	ρ_v	Transform	G_{TC}
0.96	0.96	DCT	17.8315
		WHT	13.9843
		MHT	11.0523
0.93	0.90	DCT	9.970
		WHT	7.832
		MHT	7.038
0.75	0.58	DCT	2.442
		WHT	2.090
		MHT	2.297

This conclusion agrees with Clarke's conjecture [10] that the important gains in an adaptive transform coding scheme are to be found in the bit allocation and quantization schemes rather than in the optimization or adaptation of the transform type. Therefore, the adaptive transform coding scheme used in this study will incorporate a single transform type, the DCT. In general, however, the procedure described in this section is useful for evaluating performance of candidate transforms for a given image source. It also is essential when selecting a transform for sources that have negative correlation coefficients where the DCT can be inferior to a number of other transform types [10, 61].

Bit Allocation Strategy

The bit allocation strategy involves using the individual DCT coefficients' variances to calculate the number of bits to be assigned to each of the coefficients. As is standard practice [10, 63], a full 8 bits will be allocated to the DC coefficient in order to prevent blocking artifacts caused by coding errors that result in brightness level differences between adjacent blocks. Techniques such as ADPCM, where the DC coefficient of each block is coded as the residual of a prediction based on DC coefficient values from preceding blocks, can be used to reduce the number of bits required to code these coefficients [10]. Due to its small effect on the overall bit rate, ADPCM of the DC coefficients will not be used in this study, but should be considered in cases where channel capacity is very limited.

The available bits for coding the remaining 63 AC coefficients will be

allocated by means of the optimal bit allocation scheme presented previously in Equation 4.26. The variances used in this equation will be derived from the generalized covariance model in order to maintain low overhead requirements. Mauersberger [64] has investigated the effect of variance mismatch on quantizer performance, and he has concluded that the matching of a quantizer relative to the variance is not very critical. Since the results of the previous chapter indicate that the model accurately predicts the distribution of coefficient variances, the errors that will result from using modelled rather than computed variances should be small. In order to test this assertion, a number of images were coded using actual variances and modelled variances computed over 512×512 pixel areas (assuming stationarity), and the reconstructed images were analyzed. These images were visually indistinguishable, and the PSNR's of images coded by both methods were computed using Equation 2.4 and showed very small variations. For example, the P' Channel image shown in Figure 12 had a PSNR of 34.42 dB when coded using actual variances, and a PSNR of 34.20 dB when modelled variances were used.

Model parameter updating, and adaptivity of the bit allocation scheme to deal with changing image statistics will be considered later in this chapter.

Quantization of Transform Coefficients

In order to implement an efficient transform coefficient quantization scheme, it is important that the variance of the individual coefficients and their probability density function (pdf) be estimated. The variance is used to normalize the individual

coefficients, and the pdf is used to select the decision intervals and reconstruction value for minimum MSE quantization. If it is assumed that all of the AC coefficients have the same pdf, they can be normalized by dividing by their individual standard deviation, and then identical unit variance quantizers for each integer bit allocation can be used to code and reconstruct the data. If this assumption is false, then individual quantizers must be implemented for each integer bit allocation and each pdf type, and information matching the specific quantizer to each coefficient must be supplied to the receiver for proper decoding. As described later in this section, we will operate under the assumption that all of the AC coefficients have the same pdf while the DC coefficients may have a different pdf; therefore, normalization of the AC coefficients will be required for efficient quantization.

As in the case of transform selection and bit allocation, the normalization procedure also involves tradeoffs between implementation complexity and overhead transmission requirements. The transformed data can be used to calculate the variances of each of the 63 AC coefficients, and those variances can be used to normalize and code the data; however, this requires added computations and also the matrix of normalization factors has to be transmitted as overhead for proper reconstruction at the receiver.

Chen and Smith [63] developed a technique for estimating the normalization factors that reduce the overhead requirements. Their technique involves calculating a normalization constant c which is set equal to the maximum standard deviation of

those DCT coefficients which were assigned 1 bit for coding. The normalization factors for the remaining AC coefficients are then dependent on the number of bits assigned. For a given normalization constant c and allocated number of bits N_b , the normalization factor k is given by

$$k = c 2^{N_b - 1} \quad (5.2)$$

Therefore, assuming that the bit allocation matrix is available at the receiver, only the factor c is required to reconstruct the normalization matrix.

The largest standard deviation is used in this procedure in order to prevent coefficient clipping. This technique was applied to the multisensor imagery using the generalized covariance model for bit allocation and variance estimation. The resulting imagery was of very poor quality with excessive blocking artifacts particularly in low-detail areas. This problem was due to the fact that this technique places greater emphasis on the high frequency DCT coefficients that have the greatest amount of modelling error. As Tables 4 through 6 show, the modelled variances do not decrease with increasing frequency as rapidly as the actual variances. As a result, the k factors for the low frequency coefficients that have 3 or more bits allocated are much larger than required since they include an exponential increase in the error present in the c estimate. The resulting coarse quantization of the low frequency coefficients affects those blocks that have little detail due to large coding errors in the critical low frequency coefficients.

The normalization technique used in this study consists of employing the covariance model to estimate the coefficient standard deviations which are then used to normalize the corresponding AC coefficients. Overhead requirements are limited to the horizontal and vertical correlation coefficients. The same model is then used at the receiver to reconstruct the normalization and the bit allocation matrices. This technique avoids the error growth inherent in Equation 5.2.

After allocating the bits and normalizing the DCT coefficients, the next requirement is to optimally quantize these coefficients. In order to accomplish this, an estimate of the pdf of each coefficient is required. A considerable amount of research effort has been devoted to determining the pdf of DCT coefficients, but the results have been inconsistent. The conclusions based on theoretical considerations differ substantially from those based on empirical analyses, and the latter also vary depending on the type of image data sources used. For example, Pratt [66] and Netravali and Limb [3] have postulated that the DC coefficients should have Rayleigh pdf's since they are composed of the sums of positive values, and that the AC coefficients should have Gaussian pdf's due to the Central Limit theorem. Conversely, Clarke [10] theorizes that the DC coefficients are best modelled by Uniform pdf's, and the low order AC coefficients have Gaussian pdf's while the high order AC coefficients are Laplacian. Azadegan [8], and Reininger and Gibson [65] have documented tests that indicate that the DC coefficients are best approximated by a Gaussian pdf while the AC coefficients have Laplacian statistics.

As a result of these conflicting conclusions, it is important to estimate the effect of pdf mismatch on quantizer performance. Mauersberger [64] conducted an in-depth analysis of quantizers operating over a range of parameter mismatches. He concluded that the quantizers are relatively robust to variance mismatches, but they are quite sensitive to pdf errors. His results show that the performance of optimum quantizers designed for a Gaussian pdf perform poorly when applied to sources having Laplacian statistics. On the other hand, the performance of optimal Laplacian quantizers did not deteriorate significantly when used to code a Gaussian source. In cases where the pdf of the DCT coefficients cannot be accurately determined a priori, the selection of Laplacian quantizers would therefore be preferred.

To test this hypothesis, a number of 512 X 512 sections of multisensor images were transformed and the coefficients normalized and then quantized by means of non-uniform Laplacian and Gaussian optimum quantizers that were designed using Max's method [4, 10]. No adaptivity to changing statistics were incorporated in these tests, and an average bit rate of 1 bit per pixel was used in the bit allocation scheme. The results clearly show that the images processed with the Laplacian quantizers were of considerably higher visual quality and had very slightly higher PSNR than those coded with the Gaussian quantizers. For example, the images shown on Figures 20 and 21 show the P' channel data of Figure 12 after compression and reconstruction using Laplacian and Gaussian non-uniform quantizers at an average rate of 1 bit per pixel. Since for this test a single bit allocation matrix was used to code the



Figure 20. Reconstructed P' Imagery Using Laplacian Non-Uniform Quantizers



Figure 21. Reconstructed P' Imagery Using Gaussian Non-Uniform Quantizers

entire frame, the portions of the imagery that have sharply different statistics such as the edges of the circular targets are distorted. However, it can be observed that the image coded with the Gaussian quantizer has higher levels of edge distortion, particularly around the two circular objects on the right side of the image. The PSNR of the image in Figure 20 is 34.30 dB while that of Figure 21 is 34.27 dB. In the remainder of this study, non-uniform optimum Laplacian quantizers will be used. For critical image compression application, reasonable estimates of transform coefficients pdf's can be obtained by the Kolmogorov-Smirnoff (KS) test which is described in [22] and [65], and its implementation is presented in [27]. A real-time compression scheme that incorporates KS testing to provide quantizer adaptivity would result in a significant increase in system complexity and computational performance requirements. From the results presented here and in [64] and [65], the expected improvements in compression performance are not sufficient to justify the incorporation of the KS technique.

To summarize the conclusions of this chapter up to this point, we have identified the parameters that can be made adaptive to changes in the statistics of the image data source. These include the transform type, the bit allocation strategy, and the quantizer selection. The results presented in this chapter indicate that adaptation of the transform and quantizer types does not substantially affect the performance of a compression scheme for the image sources considered in this study. Therefore, the scheme developed in the remainder of this chapter will focus on adapting the bit

allocation strategy to the changes in source statistics.

Adaptivity to Nonstationary Statistics

In the previous chapter, the properties of the imagery in both the spatial and frequency domains were determined assuming a wide-sense stationary source. In this section we incorporate techniques that adapt to the fact that real world images have spatially varying mean and covariance function. A number of techniques have been proposed to deal with these nonstationarities. The formally correct approach to characterizing an image is to assume the existence of a very large set of typical test images and then to examine the sequence of intensity values at a particular location (fixed over all test images) as a member of the "ensemble" of outputs of the source which was assumed to generate the test images in the first place [10]. While theoretically correct, this approach is unrealizable since we normally only have a single realization of an image, not an "ensemble" from which to compute the statistics of each pixel location.

A more practical and widely used approach is to divide the imagery into small blocks each of which is assumed to be wide-sense stationary. That is, each block is treated as a sample function of a stationary process, but there is a different process for each block. Pearlman [52] devised a scheme where each individual block of imagery is characterized by a 1-D AR model of order ranging from 6 to 16. The computational expense of calculating AR model parameters for each block makes this approach unsuited for real-time applications. This is illustrated by Pearlman's

published results which show that coding of a single 512 x 512, 8 bit image requires from 16 to 33 minutes of CPU time on a Micro Vax II.

The model-based approach that we have incorporated in this study assumes that the majority of the 8x8 blocks of pixels in large sections of the multisensor imagery can be modelled by a single 2-D model. This model is then used to adapt the bit allocation and coefficient normalization strategies for that particular section of imagery. In contrast to Pearlman's approach, the fact that a single model is used for a large number of blocks greatly reduces the implementation complexity as well as the overhead transmission requirements resulting in a scheme that is suitable for implementation in real-time systems.

The first level of adaptivity of our compression scheme involves updating the model parameters to compensate for changes in the image data. The technique developed in this study consists of loading and buffering square sections of image data from corresponding portions of each channel. The sections, which can range from 64x64 to 512x512 pixels, are used to calculate the parameters of the covariance model which in turn are used to develop the bit allocation matrices and the normalization factors. The sections are then partitioned into small blocks (8 x 8) which are then transformed and coded. The mean of each small block is assumed to be spatially varying so that the DC coefficient of the covariance model is ignored, and the mean of each 8x8 block (which is calculated by proper scaling of the first DCT coefficient) is coded separately with 8 bits. The model parameters are updated every time a new

section of image data is loaded, and new coefficient variance estimates, bit allocation matrices and normalization factors are computed. The result of coding coding the images previously shown in Chapter 4 are presented in Figures 22 through 25. The average bit rate for coding was 1 bit per pixel, and the model parameters were computed on the overall image section (512x512). The original and the reconstructed images are presented for side-by-side comparison. It is very apparent that while the scheme is very good for preserving overall fidelity, it is very poor for coding areas that vary considerably from the average area of the scene. For example, the multisensor images show considerable coding errors around the round objects and in the high contrast area near the top center of the imagery that corresponds to a small bush. The mandril imagery exhibits the greatest coding error around the original low detail areas of the nose and cheeks. No noticeable improvement was achieved by coding the multisensor imagery using smaller blocks for the model estimates.

Based on these results, it is obvious that another level of adaptivity is required. The specific application of the multisensor imagery employed in this study is used as the basis for the next level of adaptivity. Since the application of this type of high resolution imagery is to search large areas in order to find very small targets, and the multisensor is specifically designed to assure an adequate level of contrast (in at least one channel) when imaging a target of interest, the compression scheme will be made adaptive to these characteristics. The result of these sensor-specific considerations is that we can assume that the areas of interest (that is, the targets) will correspond to

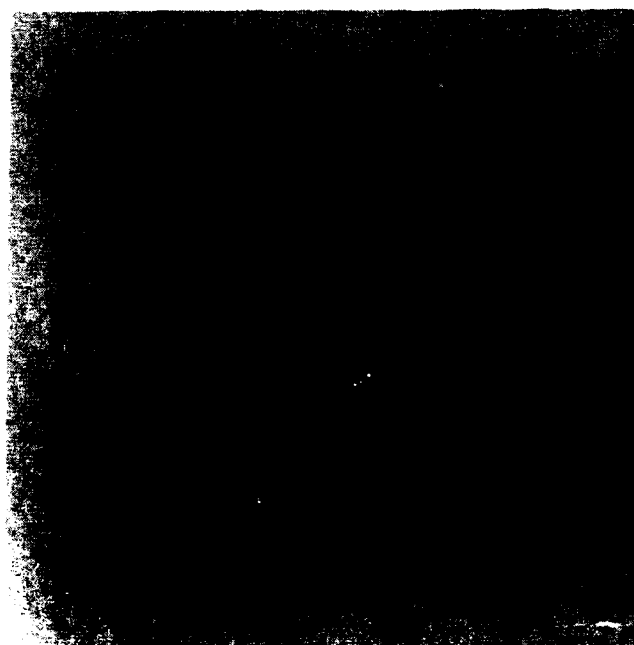


(a) Original

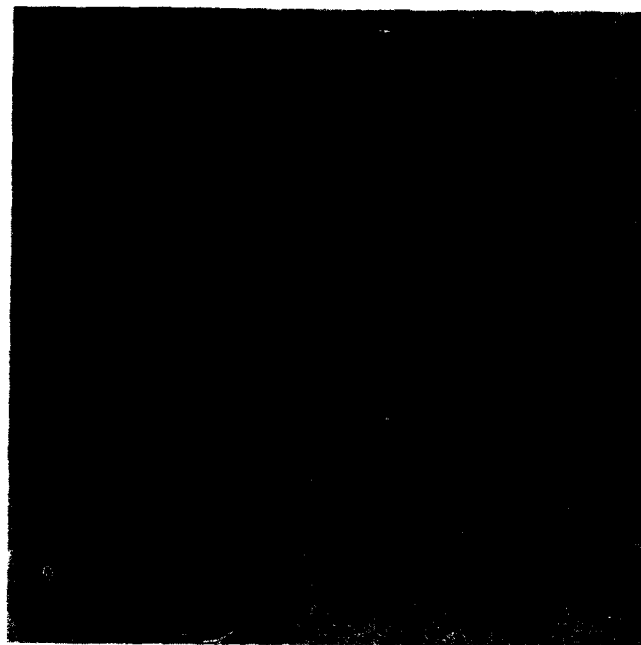


(b) Reconstructed

Figure 22. Partially Adaptive Compression and Reconstruction of P' Imagery



(a) Original

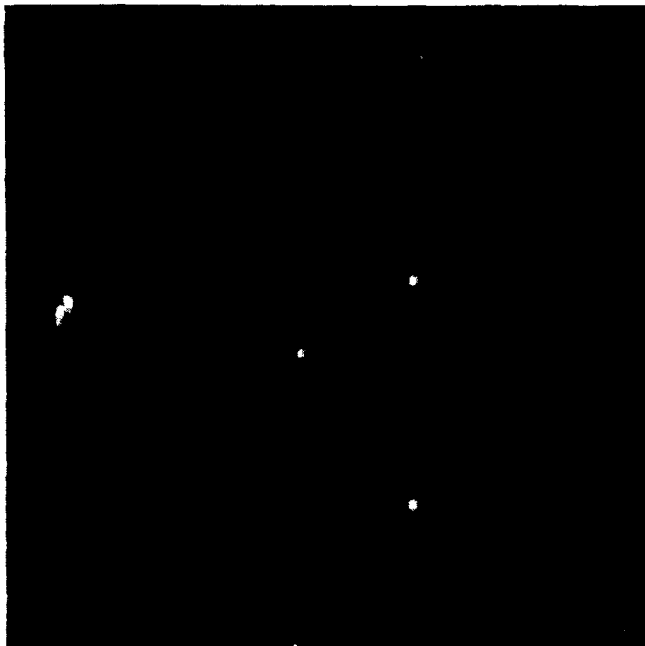


(b) Reconstructed

Figure 23. Partially Adaptive Compression and Reconstruction of C' Imagery

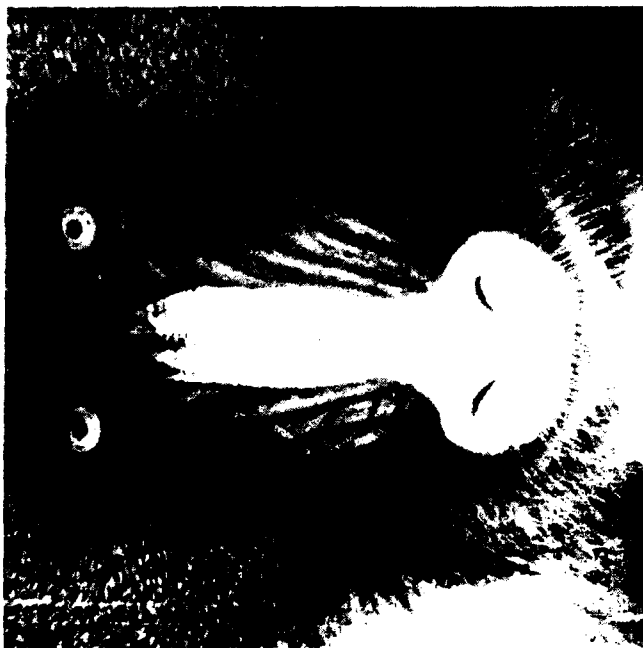


(a) Original



(b) Reconstructed

Figure 24. Partially Adaptive Compression and Reconstruction of Thermal Imagery



(a) Original



(b) Reconstructed

Figure 25. Partially Adaptive Compression and Reconstruction of Mandril Imagery

anomalies in the imagery, and these anomalies can be expected to comprise a very small portion (less than 1%) of the data collected. The anomalies are characterized by relatively sharp discontinuities (edges) which can be found in one or more of the image channels, and therefore can be characterized by a DCT coefficient distribution that differs considerably from the average of the scene. The variance of the AC DCT coefficients has been used effectively as a measure of discontinuities (edges) that occur within given block [23], [63] and will likewise be used in our scheme for defining the blocks of interest and coding these differently. The fact that the targets are very small place an upper bound on the number of blocks that must be coded differently.

The second level of adaptivity, therefore, involves calculating the variance of the AC coefficients of each block by means of Equation 5.3.

$$CVAR = \sum_{u=0}^7 \sum_{v=0}^7 [(X(u,v))^2 - (X(0,0))^2] \quad (5.3)$$

A specified number of blocks having the highest variance are flagged in each channel, and these are used to calculate the variances of the individual DCT coefficients. The values of the DCT coefficient variances are then used in the same manner as the modelled variances to compute bit allocation and normalization factors for coding. Since these high variance blocks have more detail than the average, they are allocated higher average bit rates. However, since they comprise a very small percentage of the total blocks, the fact that rates of up to 4 bits per pixel are allocated for these

blocks has a small effect on overall rates. The results presented in Chapter 7 show a marked improvement in the reconstruction of the imagery that contains small targets.

Adaptive Transform Coding Algorithm Description

The algorithm used for compression of the multisensor imagery is shown in block diagram form in Figure 26, and it consists of the following operations:

1. Read in corresponding sections of P, C, and Thermal Channels (up to 512x512 pixel sections).
2. Compute mean of P and mean of C over entire section.
3. Calculate mean polarization d (Equation 4.1), rotation angle ϕ (Equation 4.2), and transformation matrix (Equation 4.3).
4. Calculate P' and C' by rotating P and C channels using Equation 4.3.
5. Calculate data variance, and horizontal and vertical correlation coefficients of the P', C', and Thermal channels for each section.
6. Compute 8x8 covariance matrix for each channel using computed correlation coefficients in the generalized covariance model (Equation 4.26).
7. Compute bit allocation matrices using Equation 4.26 for the desired average pixel rate.
8. Compute normalization factors for each channel.
9. Divide each section of image data into 8x8 blocks and perform 2-D DCT on each block.
10. Determine variance of the AC coefficients in each 8x8 block and select

those blocks that have variances in the top 1% in each channel.

11. Calculate variance of each of the AC coefficients in the high variance blocks, and calculate bit allocation matrix for each channel.

12. Use appropriate normalization and bit allocation matrix to quantize and code each block of coefficients.

The image decompression scheme for reconstruction at the receiver is presented in block diagram format in Figure 27. In order to simplify the diagram, the decompression of the Thermal channel is not shown in Figure 27. However, decompression of the Thermal channel is identical to the decompression of the P' and C' channels except that it does not require the final channel rotation stage. The decompression scheme basically consists of reversing the compression steps outlined above. Overhead information required to calculate the covariance model, bit allocation matrices, normalization factors, and inverse rotation matrix are indicated by large arrows. It should be noted that since the rotation matrix $[A]$ (Equation 4.3) is a unitary matrix, the inverse rotation matrix is easily computed by transposing $[A]$. The implementation of this adaptive transform coding algorithm is described in the following chapter, and examples of the markedly improved reconstructed image quality are presented.

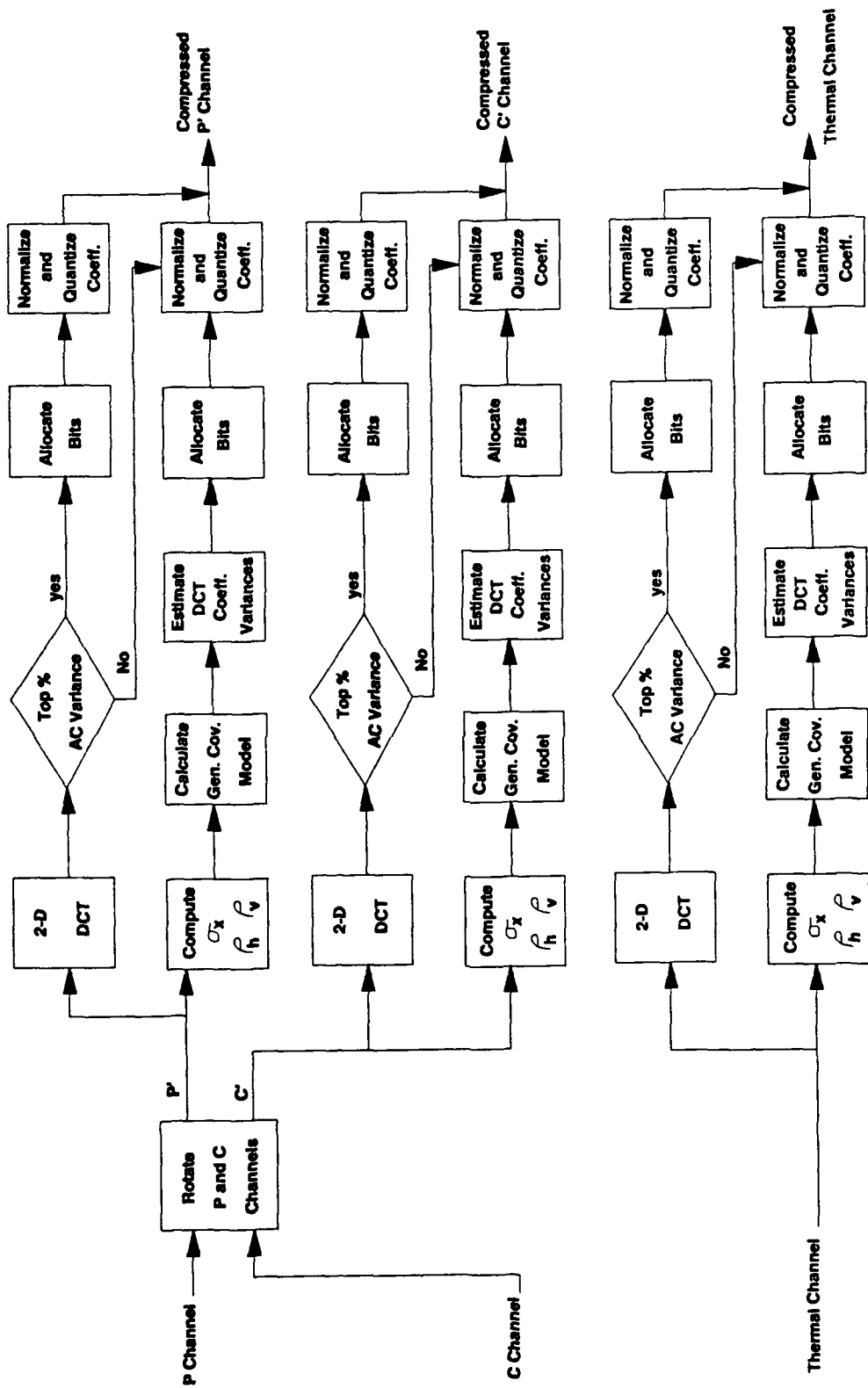


Figure 26. Compression Stage of Adaptive Transform Coding Algorithm

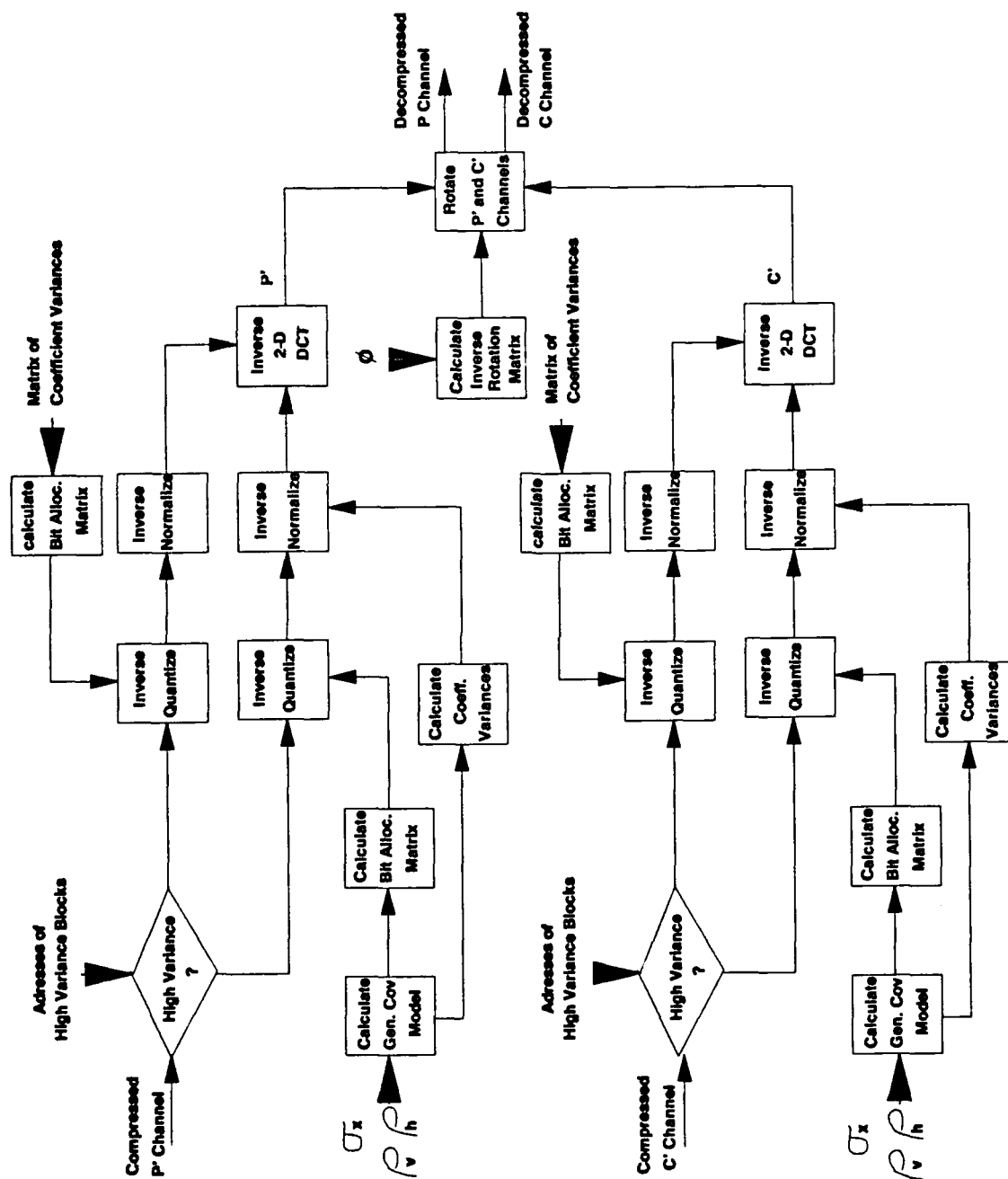


Figure 27. Reconstruction Stage of Adaptive Transform Coding Algorithm

CHAPTER VI

IMPLEMENTATION AND TESTING

Introduction

In this chapter, the compression scheme developed in Chapter 5 is adapted to a fine-grained (SIMD) parallel processor. The inherent parallelism involved in subsectioning each image channel into small blocks and then independently transforming each block is exploited by appropriate mapping into the local memory of the array of processors. An efficient and stable implementation of the DCT is used, and the adaptivity described in the previous chapter is incorporated in the algorithm. Efficient implementation of the model and the normalization and quantization strategies are also described. Timing analysis and possible bottlenecks are identified. The implementation is tested by processing a large number of multisensor images, and the average bit rates and SNR are calculated.

DCT Implementation

The DCT was first defined in 1974 by Ahmed, Natarajan and Rao [38] who also proposed an efficient implementation using FFT techniques. Since that time, a number of improved DCT algorithms have been developed. One of the most efficient techniques was developed by Hou [49] and it results in a fast, numerically stable algorithm that requires fewer multiplications and avoids overflow problems caused by

inversion of the cosine coefficients as required by other methods (e.g. Lee's algorithm [50]). In addition, Hou's technique consists of a recursive algorithm that is efficiently mapped into a SIMD architecture.

For these reasons Hou's 1-D algorithm was selected, and the required 2-D transform is obtained by sequentially applying the 1-D algorithm to the rows and columns of the image data. It should be noted that slightly more efficient techniques are available such as those that directly compute 2-D DCT's, but these are not easily extended to large block sizes, and their implementation is considerably more complicated [51]. An overview of the DCT and Hou's method is presented prior to describing the implementation method.

The DCT of a 1-D data sequence $x(n)$, $n=0,1,\dots,N-1$ was defined by Ahmed et al as

$$\begin{aligned} X(0) &= \frac{\sqrt{2}}{N} \sum_{n=0}^{N-1} x(n) \\ X(k) &= \frac{2}{N} \sum_{n=0}^{N-1} x(n) \cos\left(\frac{(2n+1) k \pi}{2N}\right) \end{aligned} \quad (6.1)$$

for $k = 1, 2, \dots, N-1$, and $X(k)$ is the k^{th} DCT coefficient.

The inverse DCT is defined as

$$x(n) = \sum_{k=0}^{N-1} X(k) \cos\left(\frac{(2n+1) k \pi}{2N}\right) \quad (6.2)$$

Equation 6.1 can be written in matrix form as

$$X = \frac{2}{N} T(N) x \quad (6.3)$$

where X and x are column vectors composed of the elements of the original and transformed data sequences arranged in natural order, and $T(N)$ is an $N \times N$ matrix whose elements are the factors of $x(n)$ defined in equation 6.1. Hou chose to neglect the $2/N$ factor in the development of the fast algorithm, and it must be inserted at the end of the computation to preserve the orthonormality of the DCT.

Hou's fast algorithm is based on the fact that, by properly rearranging the X and x vectors, the transformation matrix can be decomposed recursively until only simple 2 point DCT's are required. For any size N input vector, the permuted DCT matrix $\hat{T}(N)$ can be recursively computed from

$$\hat{T}(N) = \begin{bmatrix} \hat{T}\left(\frac{N}{2}\right) & \hat{T}\left(\frac{N}{2}\right) \\ K\hat{T}\left(\frac{N}{2}\right)Q & -K\hat{T}\left(\frac{N}{2}\right)Q \end{bmatrix} \quad (6.4)$$

where

$$Q = \text{Diag} [\cos \phi_m] \quad (6.5)$$

$$\text{where } \phi_m = \frac{(4m + 1) \pi}{2N} \quad \text{for } m = 0, 1, 2, \dots, \frac{N}{2}-1 \quad (6.6)$$

$$\text{and } K = R L R \quad (6.7)$$

where R is the permutation matrix for performing the bit reversal, and L is a lower diagonal matrix defined as

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & . & . & . & 0 \\ -1 & 2 & 0 & 0 & 0 & . & . & . & 0 \\ 1 & -2 & 2 & 0 & 0 & . & . & . & 0 \\ -1 & 2 & -2 & 2 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ -1 & 2 & -2 & 2 & -2 & . & . & . & 2 \end{bmatrix} \quad (6.8)$$

Equation 6.3 can therefore be rewritten as

$$\hat{X} = \hat{T}(N) \hat{x} \quad (6.9)$$

In this equation, $\hat{X}$ is obtained by bit reversing the transformed data sequence. To rearrange the original data sequence $x(n)$ into $\hat{x}$, we first divide the sequence into even-indexed and odd-indexed samples, then column order it by placing the former samples in the top half and the latter samples in the bottom half. Next, the samples in the bottom half (odd-indexed samples) are bit inverted (e.g the sequence $x(1)$, $x(3)$,

$x(5)$, is reordered as $x(5), x(3), x(1)$). After this reordering, the matrix implementation becomes

$$\begin{bmatrix} \hat{X}_{top} \\ - \hat{X}_{bottom} \end{bmatrix} = \begin{bmatrix} \hat{T}\left(\frac{N}{2}\right) & \hat{T}\left(\frac{N}{2}\right) \\ K\hat{T}\left(\frac{N}{2}\right)Q & -K\hat{T}\left(\frac{N}{2}\right)Q \end{bmatrix} \begin{bmatrix} \hat{x}_{top} \\ - \hat{x}_{bottom} \end{bmatrix} \quad (6.10)$$

The block diagram implementation of equation 6.10 is shown in Figure 28.

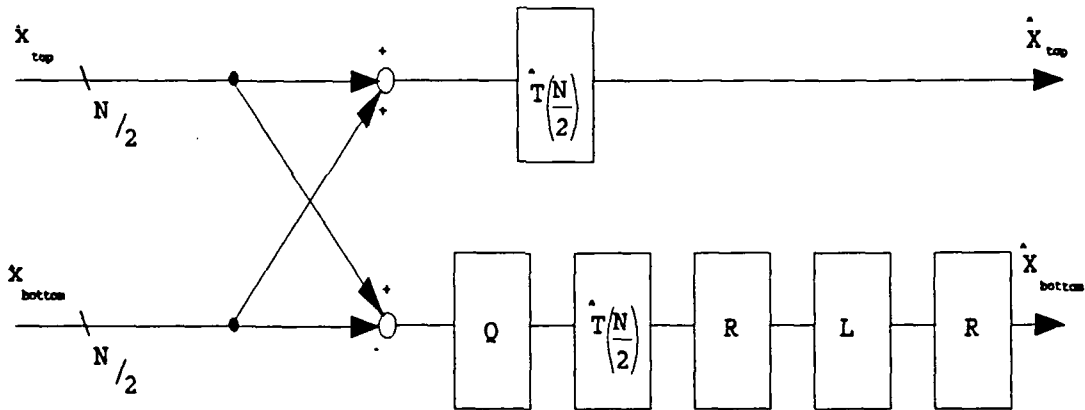


Figure 28. Block Diagram of Hou's Algorithm (Decimation-in-Frequency DCT)

From Figure 28, it is apparent that higher order DCT's are calculated from two identical lower order DCT's, so that any finite data sequence of length 2^N can be decomposed to a number of 2 point DCT's. In this study we employ 8 x 8 point 2-D DCT's that are implemented by sequential application of 8-point 1-D DCT's. The implementation of Hou's algorithm that was used in this study is as shown in Figure

29, where (a) is the top level (8-point DCT), and (b) and (c) are recursively lower levels.

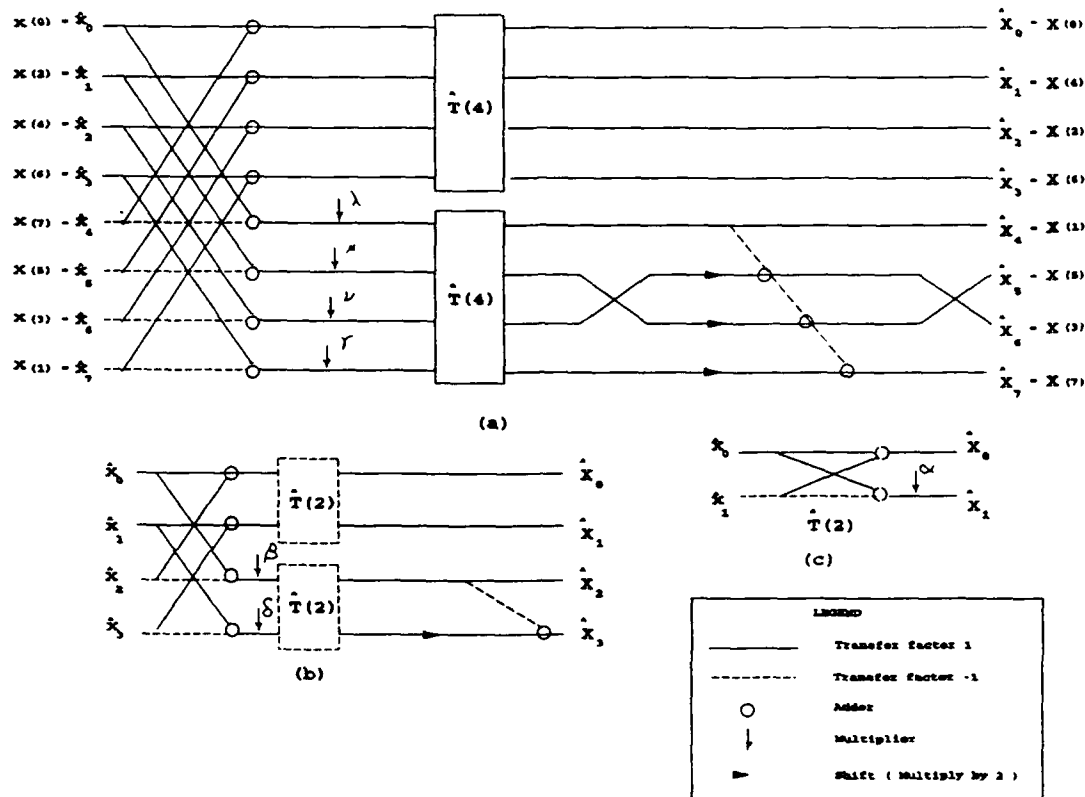


Figure 29. Recursive Computation of DCT for $N = 8$

In Figure 29, the constants α , β , δ , λ , μ , ν , and γ , are derived from equations 6.5 and 6.6 by applying the appropriate m .

For efficient implementation in a massively parallel processor that has limited interprocessor connectivity, it is required that the image data be mapped so that all of the pixels in an 8×8 block are assigned to the same processing element PE, and all

PE's can then execute the same instruction simultaneously. This is accomplished by remapping the incoming image data into a crinkled format where the three channels of image data are sub-sectioned into 8x8 blocks, and each block is stored in column major order in the local memory of one of the 4096 available PE's. This can be done very rapidly by means of parallel data transforms (PDT's) to remap the image data from one format to another [53].

The calculation of an 8x8 DCT on the DAP therefore consists of the following steps:

- (1) Reorder each row (or column) of data
- (2) Perform butterfly addition/subtraction
- (3) Multiply bottom half by $\cos(\phi_m)$
- (4) Call $\hat{T}(N/2)$ subroutines for top and bottom halves
- (5) Perform bit-reversal reordering of the bottom half
- (6) Perform multiplication by 2 (shift) and subtraction for bottom half
- (7) Perform bit-reversal reordering for the bottom half
- (8) Repeat steps (1) through (7) for each column of transformed data

The actual FORTRAN Plus code used to implement this DCT algorithm is listed in Appendix B.

Adaptivity Implementation

The adaptivity described in the previous chapter consists of periodically updating the model parameters and of selecting a percentage of the high activity

blocks. The goal of both of these data-dependent operations is to assign the appropriate number of bits and normalization factors to the significant transform coefficients of each 8x8 block. This section describes the tradeoffs that were involved in implementing the adaptivity portions of the compression algorithm on a massively parallel processor.

The model parameter update operation was implemented by dividing each frame of imagery into square sections and mapping each section of imagery in crinkled format (all the pixels in each 8x8 block are assigned to the same PE). The algorithm operations are then performed simultaneously over all the PE's and the process is looped as many times as required to complete the frame. For example, if the model parameters are recomputed for every 64x64 pixel section of imagery, then the algorithm must be looped 64 times in order to process each 512x512 section of imagery. In addition, only 64 of the 4096 processors available are utilized. This looping technique was found to be very slow, with processing times of approximately 1 minute required for the compression/reconstruction operation. Fortunately, there were no advantages to updating the model parameters at this fast rate. Excellent results were obtained at bit rates as low as .55 bits per pixel when the model parameters were updated only once for each 512x512 pixel section. Execution times were substantially improved by processing these large sections simultaneously, since all 4096 PE's were then utilized, and the times to compress/reconstruct each 512x512 section of imagery were reduced to under 1 second.

The second part of the adaptive scheme, the identification and separate coding of the high activity blocks was implemented using a scheme that did not reduce the parallelism exploited in the model parameter updating operation. The technique used involved creating a logical matrix that corresponds to the address of each of the 4096 8x8 blocks in one section of image data. The AC coefficient variances of all the blocks are computed simultaneously by application of Equation 5.3, and the algorithm searches through each of the 4096 variance values for the maximum. Once a maximum is found, the logical mask is set to false at the corresponding address location, and the process is repeated until the preset number of high variance blocks are selected. The logical mask is then used to block the model-based coding computations for those PE's that contain the high variance blocks. In effect, all of the low variance blocks are coded simultaneously using the modelled variances, and then all of the high variance blocks are coded simultaneously using the actual DCT coefficient variances. The code for this technique is contained in the FORTRAN-Plus subroutine "adapt" which is listed in Appendix B.

Another important part of the compression scheme is the coefficient quantization section. In this study, unity variance, optimum non uniform Laplacian and Gaussian quantizers were utilized. A number of tests in which actual imagery data were compressed and reconstructed indicated that the Laplacian quantizers resulted in higher PSNR. Each of these quantizers were implemented using two lookup tables. One lookup table (named "bound" in the program listing) is used to

define the decision levels for a given number of coding bits, and a second table ("cval") defines the assignment values. Each normalized AC coefficient is assigned a particular location in the decision table based on the number of bits allocated and the range of levels that bracket its value. The coefficient's assigned location then defines the assigned value that is found in the "cval" matrix. The program listing in Appendix B shows the decision and assignment values that correspond to the Laplacian quantizers having 2 to 32 quantization levels (1 to 5 bit quantizers). It should be noted that the bit allocation method (Equation 4.26) is limited to a maximum of 5 bits for any AC coefficient (in order to keep the lookup tables small). Any coefficient which requires more than 5 bits by application of Equation 4.26, is therefore only allocated 5 bits.

To illustrate the high quality of the reconstructed imagery using the adaptive compression algorithm, the multisensor images previously shown in Figures 12, 13, and 14 were compressed at an average rate of 0.8 bits per pixel with the number of high-variance blocks fixed at 64. The final channel rotation was not performed in order to compare results with those obtained in the previous chapter. The results presented in Figures 30, 31 and 32 for the P', C', and Thermal channels show that even extremely small details such as the dark centers of the round objects in the Thermal image are preserved. Comparison with the results obtained by coding with the partially adaptive algorithm (Figures 22, 23 and 24) illustrate the decided advantage of adapting the scheme to changes in block variances. Even at the lower

bit rate, the images shown here do not exhibit any of the blocking artifacts observed earlier. Additional examples of results obtained from application of the adaptive algorithm are presented in Chapter 7. Also included in that chapter is an evaluation of the rate distortion performance and of the effects on automatic target cueing algorithms.



Figure 30. P' Channel Image Coded at 0.8 Bits Per Pixel

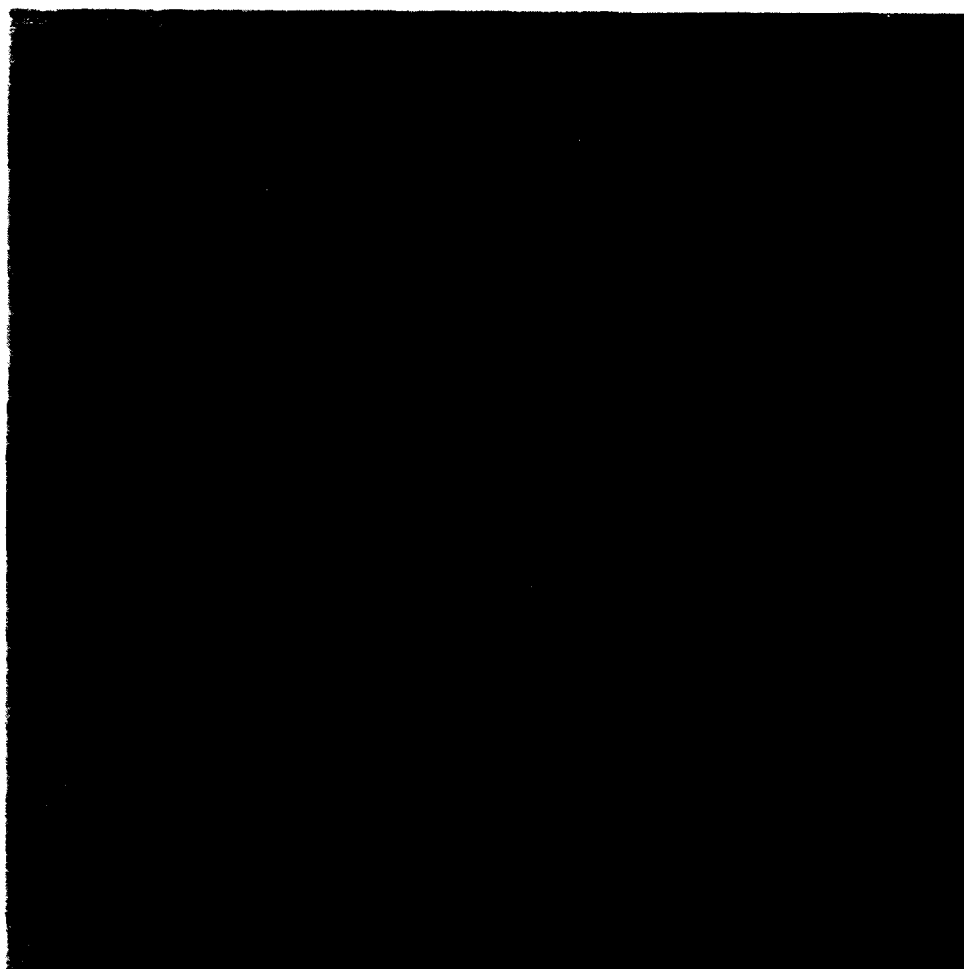


Figure 31. C' Image Coded at 0.8 Bits Per Pixel

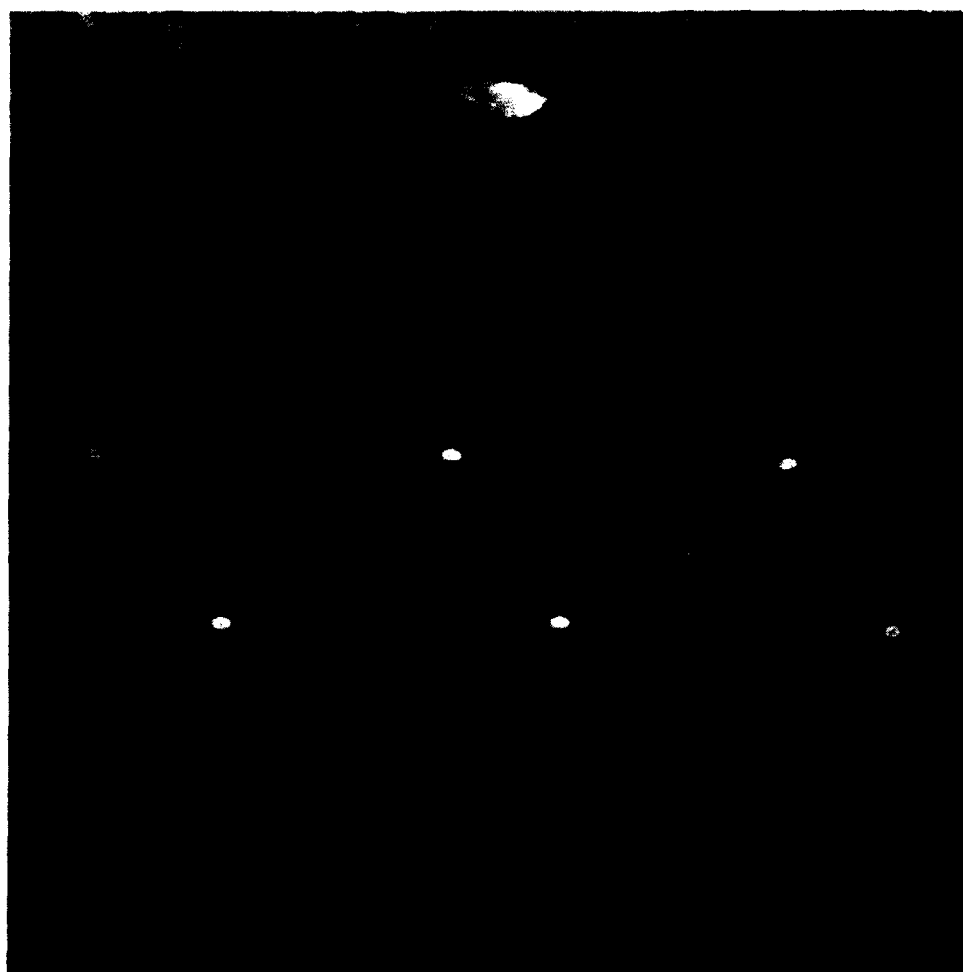


Figure 32. Thermal Image Coded at 0.8 Bits Per Pixel

CHAPTER VII

ANALYSIS OF RESULTS

Introduction

In this chapter, a number of compressed and reconstructed images are presented and analyzed to determine the rate versus distortion performance of the adaptive compression scheme. The utility of the imagery is evaluated by processing the original and the reconstructed multisensor imagery with automatic target cueing algorithms that rely on shape and edge contrast fidelity, as well as grey level value, for object detection. In addition, examples of original and reconstructed images that contain very small man-made objects in background clutter are presented in order to qualitatively evaluate the effects of compression/reconstruction, and also to estimate the usefulness of the imagery in applications that require human photo-interpretation.

Rate-Distortion Performance

The rate distortion function of an adaptive transform coding scheme is very difficult to define mathematically even for those cases where simple image models and distortion metrics are used [8]. Analytical expressions for rate distortion functions that do not account for adaptivity are not very useful in this study, because our scheme is designed to maintain high fidelity of only a small percentage of important blocks while allowing the background blocks to be coded less accurately. This

characteristic of the adaptive scheme would be neglected by rate-distortion functions that estimate average, non-adaptive performance. The approach that is implemented here is to select a number of representative images, compress and reconstruct them over a range of average bit rates, and then compute the PSNR for each bit rate considered. In addition, sections of the original and reconstructed images are visually inspected for artifacts that may affect the performance of automatic target cueing algorithms. The bit rates considered range from 0.2 to 1.5 bits per pixel.

Performance improvements at higher bit rates cannot be evaluated because the compression scheme, as presently implemented, allows a maximum of 5 bits for any single AC coefficient. It is unlikely, however, that an adaptive transform coding scheme would be considered for applications allowing such high bit rates. Simpler non-adaptive transform and predictive schemes can satisfactorily operate at rates greater than 1.5 bits per pixel.

The rate distortion performance of the scheme using a fixed number (64) of allowable high variance blocks is as shown in Figures 33, 34 and 35 for the P, C and Thermal channels respectively. Sections of original and reconstructed imagery at two different average bit rates are presented in Figures 36 through 44.

The bit rates used in these results were calculated by means of the actual number of bits in the two bit allocation matrices used for each channel. One matrix, called `bit_all` in the source code, corresponds to the bit assignment calculated from the model, and the second matrix, `bitall2`, is determined from the actual coefficient

variances of the 64 high AC variance blocks. The average bit rate for each 512x512 section of imagery is given by

$$bitrate = \frac{(4096 - 64) * \sum [bit_all] + (64) * \sum [bitall2]}{512 * 512} \quad (7.1)$$

The three rate distortion curves presented here show that the scheme's PSNR decreases significantly below 0.4 bits per pixel, and the improvement with increased bit rate tapers off after approximately 1.0 bits per pixels. Inspection of the reconstructed imagery also indicates that significant blocking artifacts become visible when operating below 0.60 bits per pixel. Thus, without modifications (such as entropy coding of the output data), this scheme should be considered for applications requiring average bit rates of approximately 0.8 bits per pixel.

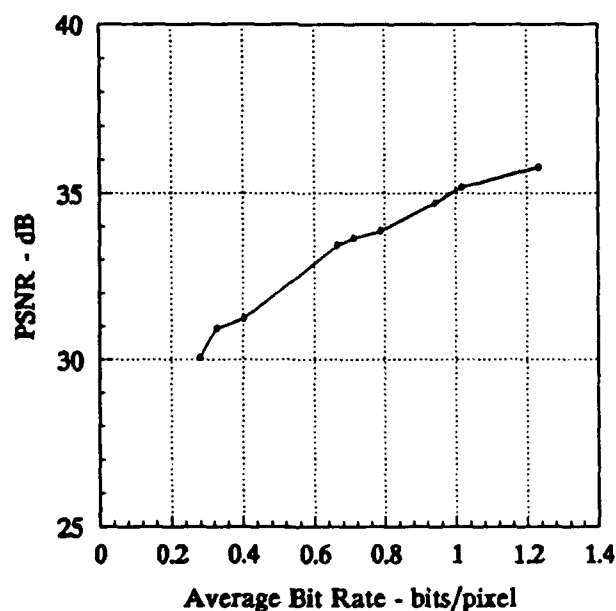


Figure 33. Rate Distortion Performance for P Channel Data

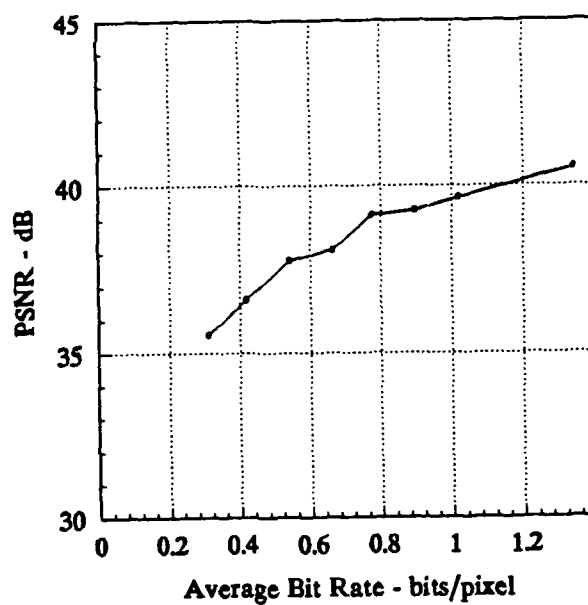


Figure 34. Rate Distortion Performance for C Channel Data

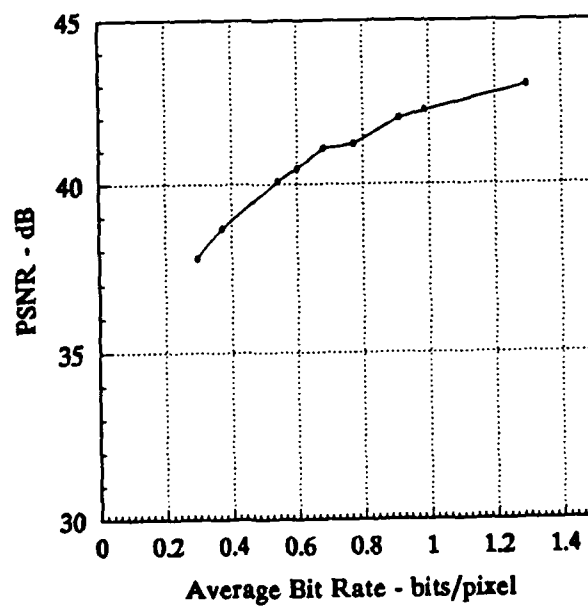


Figure 35. Rate Distortion Performance for Thermal Channel Data

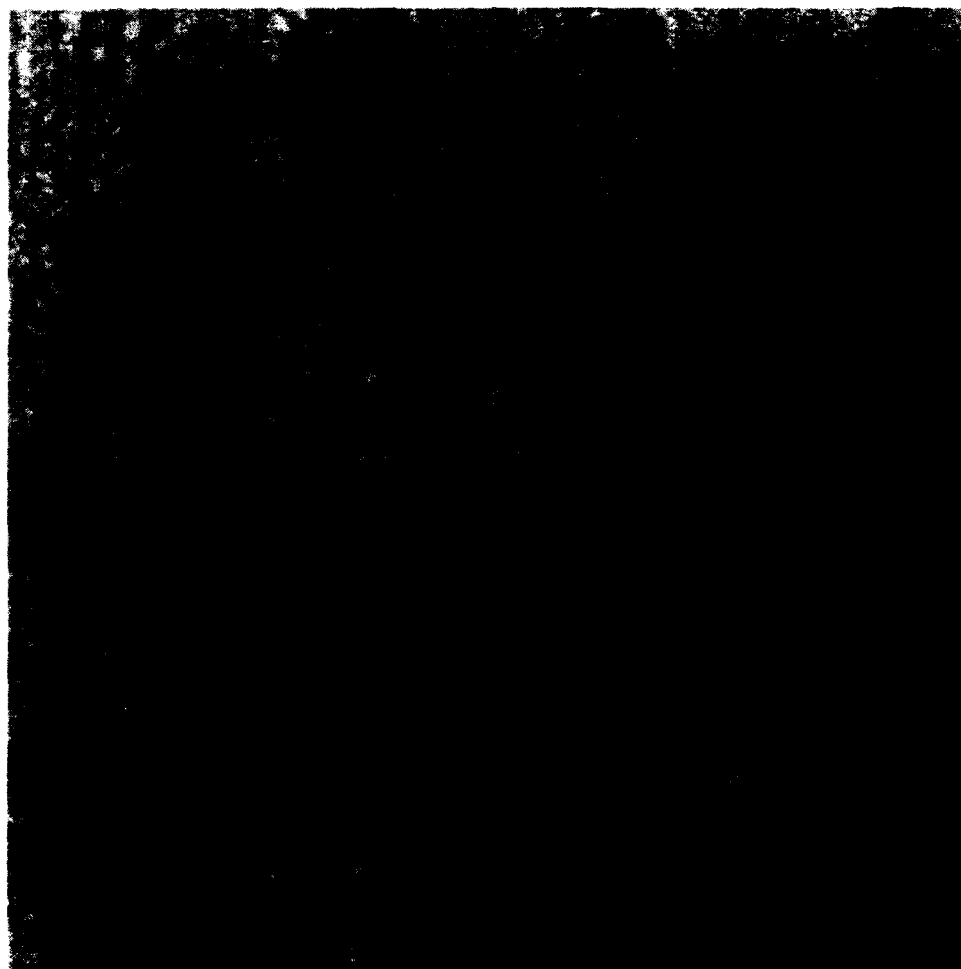


Figure 36. Original P Channel Imagery



Figure 37. Original C Channel Imagery

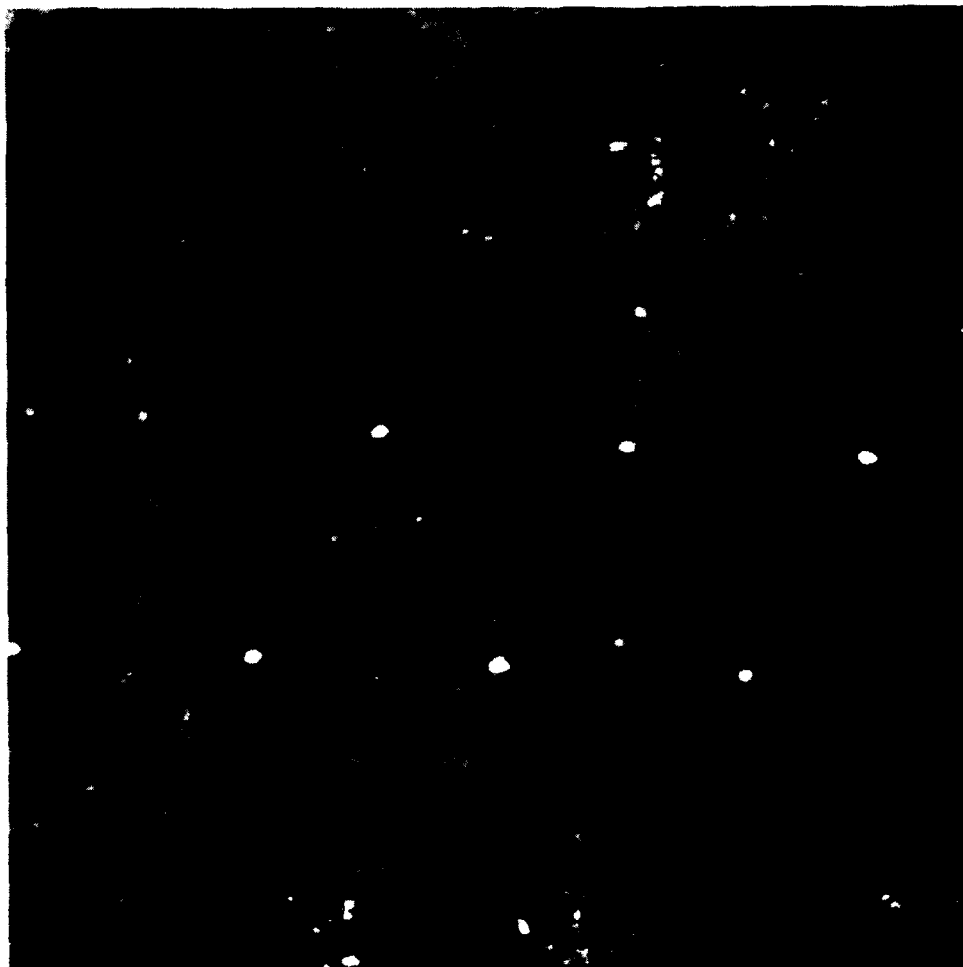


Figure 38. Original Thermal Channel Imagery

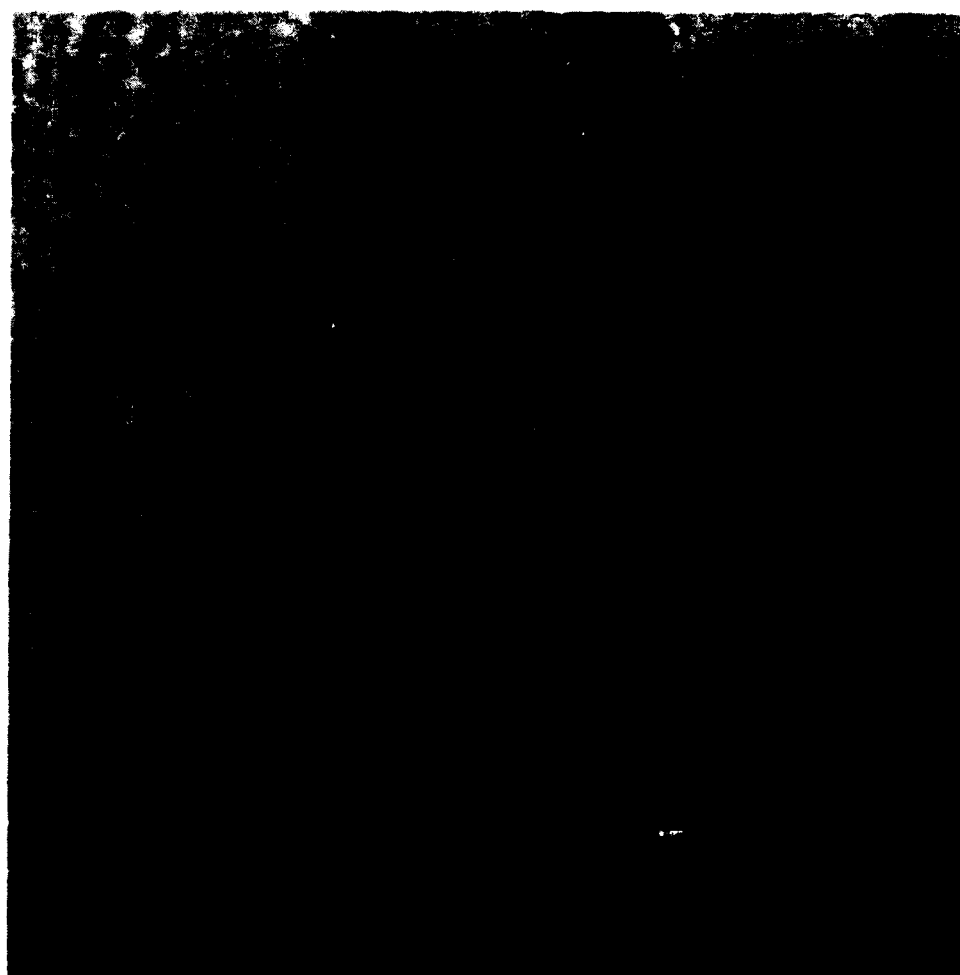


Figure 39. P Channel Imagery Compressed at 0.55 Bits Per Pixel



Figure 40. C Channel Imagery Compressed at 0.55 Bits Per Pixel

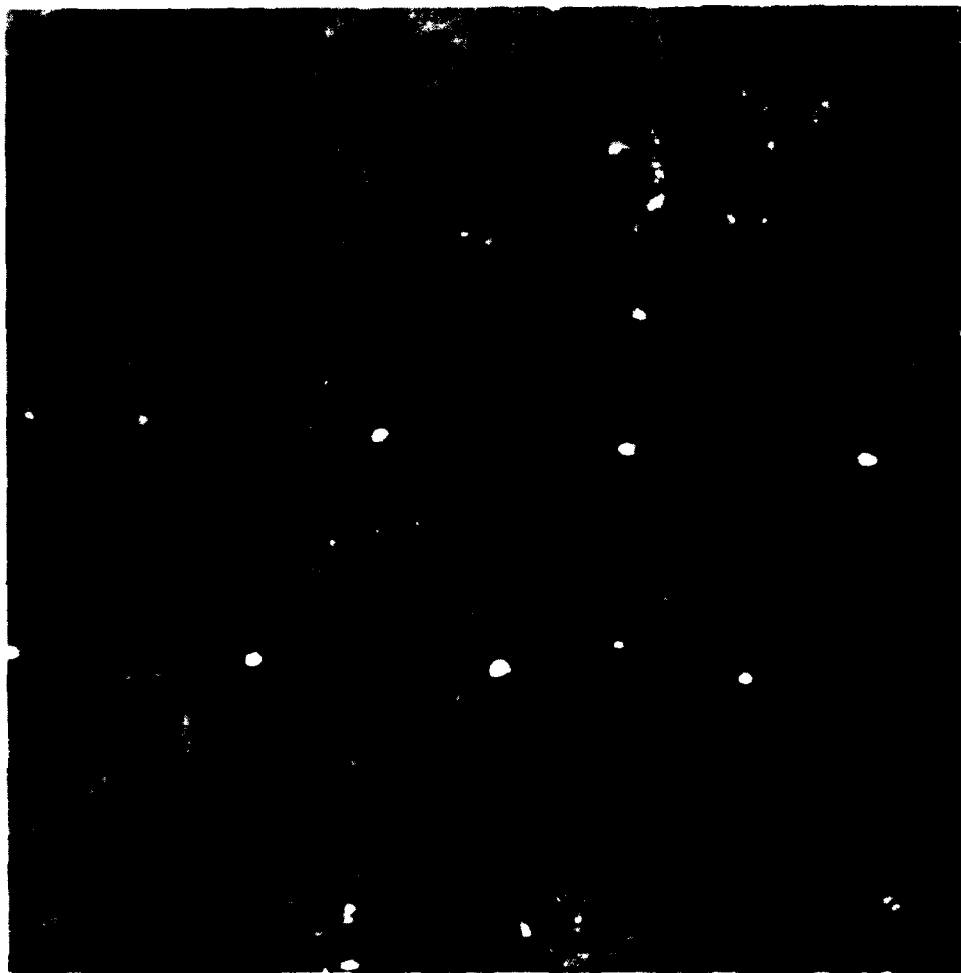


Figure 41. Thermal Channel Imagery Compressed at 0.55 Bits Per Pixel



Figure 42. P Channel Imagery Compressed at 0.80 Bits Per Pixel

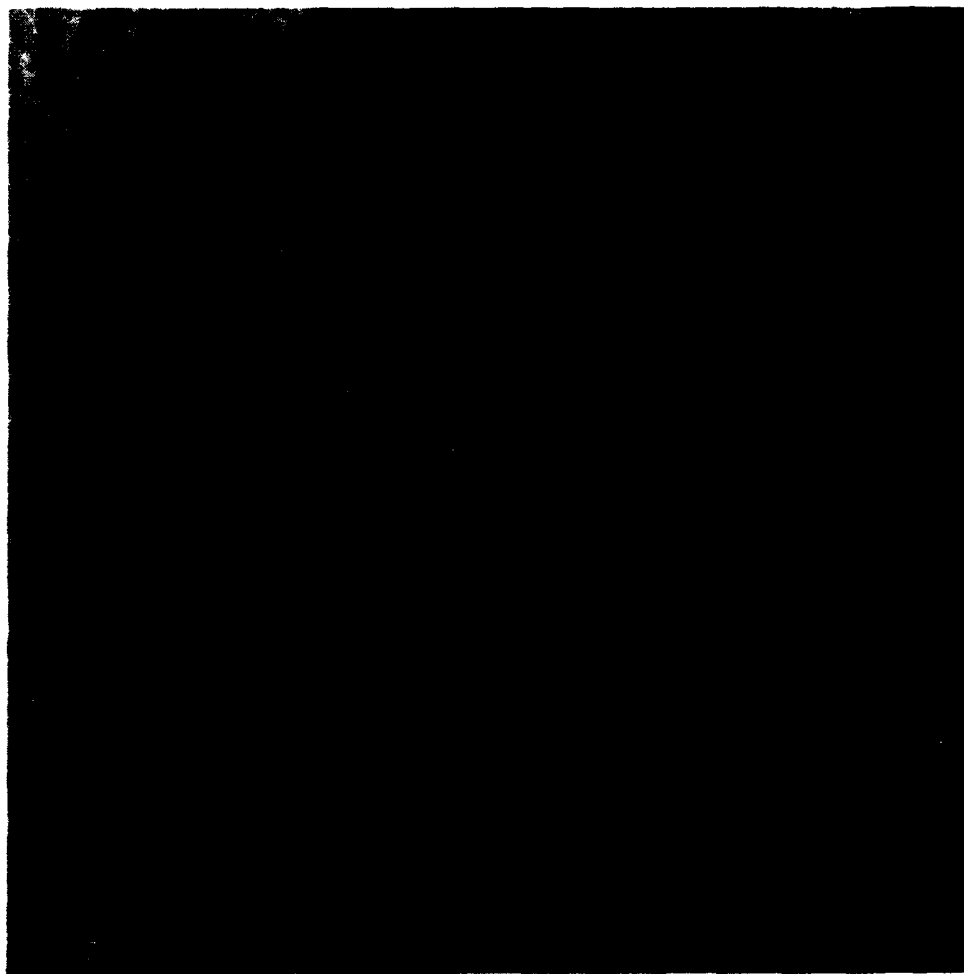


Figure 43. C Channel Imagery Compressed at 0.80 Bits Per Pixel

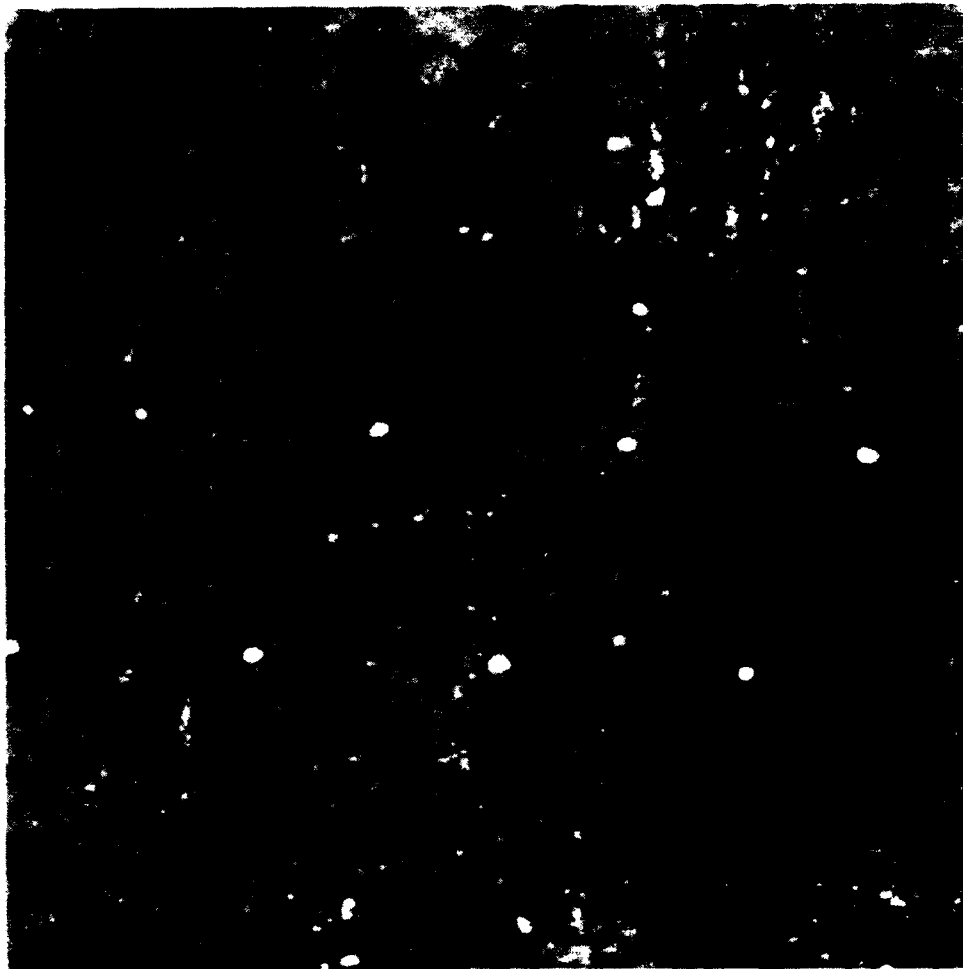


Figure 44. Thermal Channel Imagery Compressed at 0.80 Bits Per Pixel

Utility of Reconstructed Imagery

The image shown in Figure 45 consists of three rows of man-made targets placed in a tall grass background. An automatic target cueing algorithm [17] similar to that shown in Figure 3 was applied to the three image channels, and the results are displayed as red boxes overlaid on one of the image channels. All of the 14 boxes correspond to actual targets, with no false alarms or missed targets. The three channels were then compressed and the results processed with the same algorithm (no changed thresholds or filter settings). The bit rates were lowered equally for all channels until either false alarms or missed targets were observed. It was found that the target detection algorithm is quite robust, since imagery that had considerable blockiness in one or more channels was still perfectly cued. This is due to the fact that the adaptive coding preserves the sharp edges even at low overall rates. A false alarm was finally observed when the bit rate was lowered to 0.5 bits per pixel (Figure 46). It should be noted, however, that the C' channel bit rates could be reduced to below 0.3 without affecting the target cueing performance. This is due to the fact that the image rotation (Equation 4.3) transfers most of the variance of the C channel into the P' data thereby leaving the C' channel with very low level of detail. The number of high variance blocks could also be reduced to as low as 12 with no measurable increase in distortion of the C' data.

A second set of tests were performed to determine the ability to visually detect very small targets in the reconstructed imagery. In this case, the targets had

signatures ranging from one to five pixels. Figure 47 shows two such targets that consist of 1x4 pixels each. Figure 48 shows that the small targets are still very visible after compression at 0.8 bits per pixel.

In summary, it can be concluded that the multisensor image considered in this study can be rapidly compressed by the adaptive transform coding algorithm developed in Chapter 5, and the reconstructed imagery can be used for automatic and manual target detection applications over a range of very low bit rates.

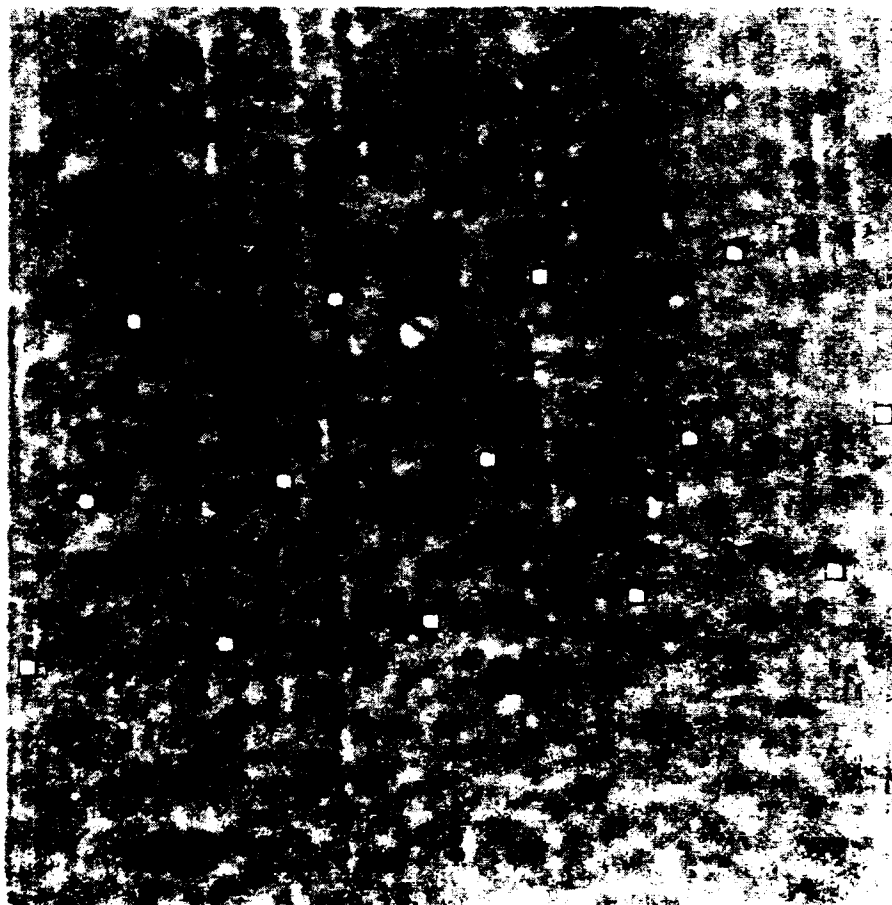


Figure 45. Target Cueing Performance - Original Imagery

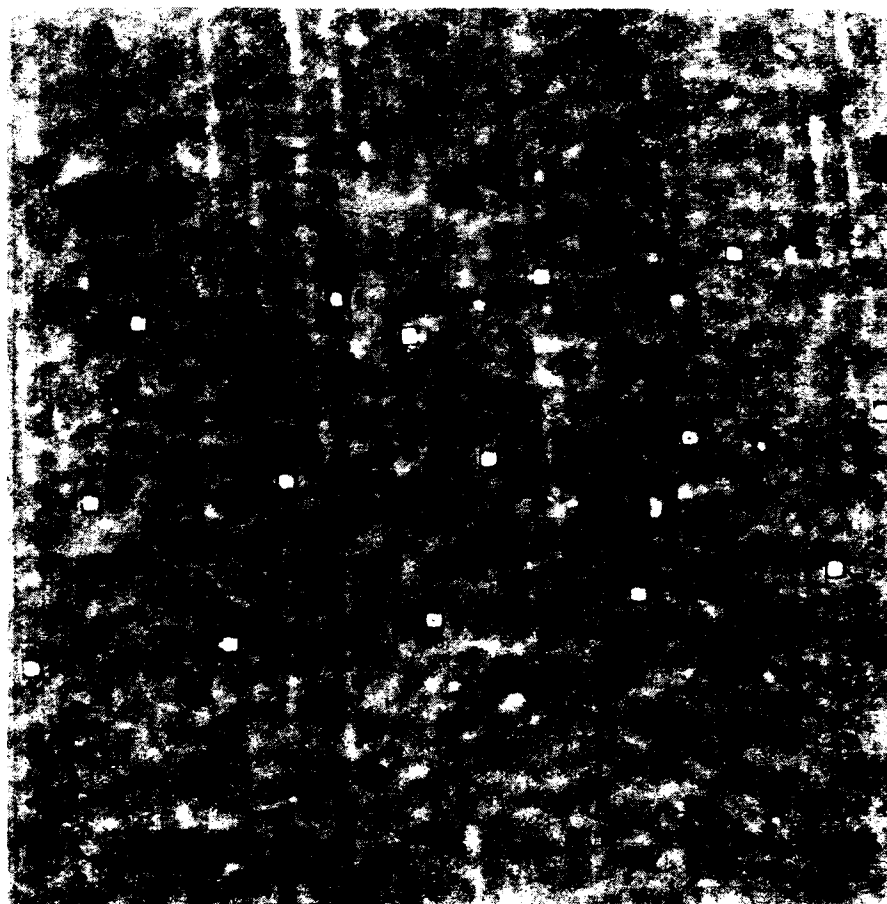


Figure 46. Target Cueing Performance - Image Coded at 0.50 Bits Per Pixel

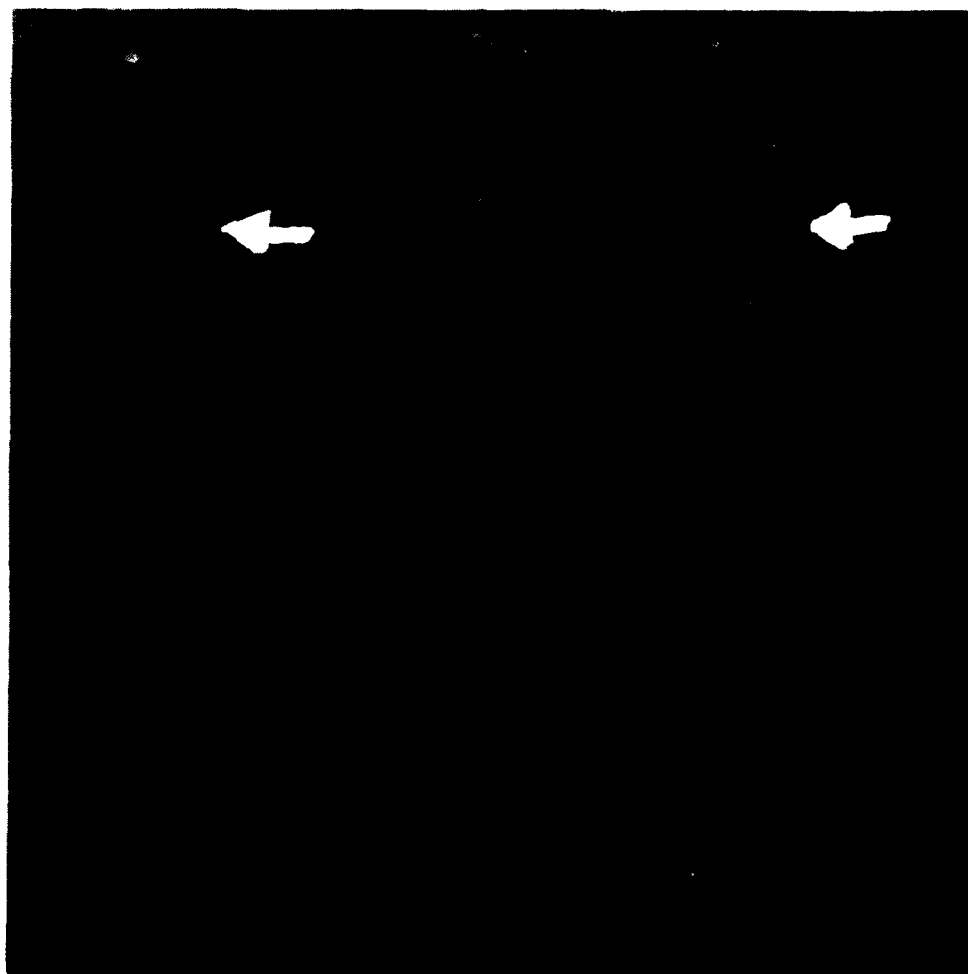


Figure 47. Small Targets in Background Clutter - Original Thermal Imagery

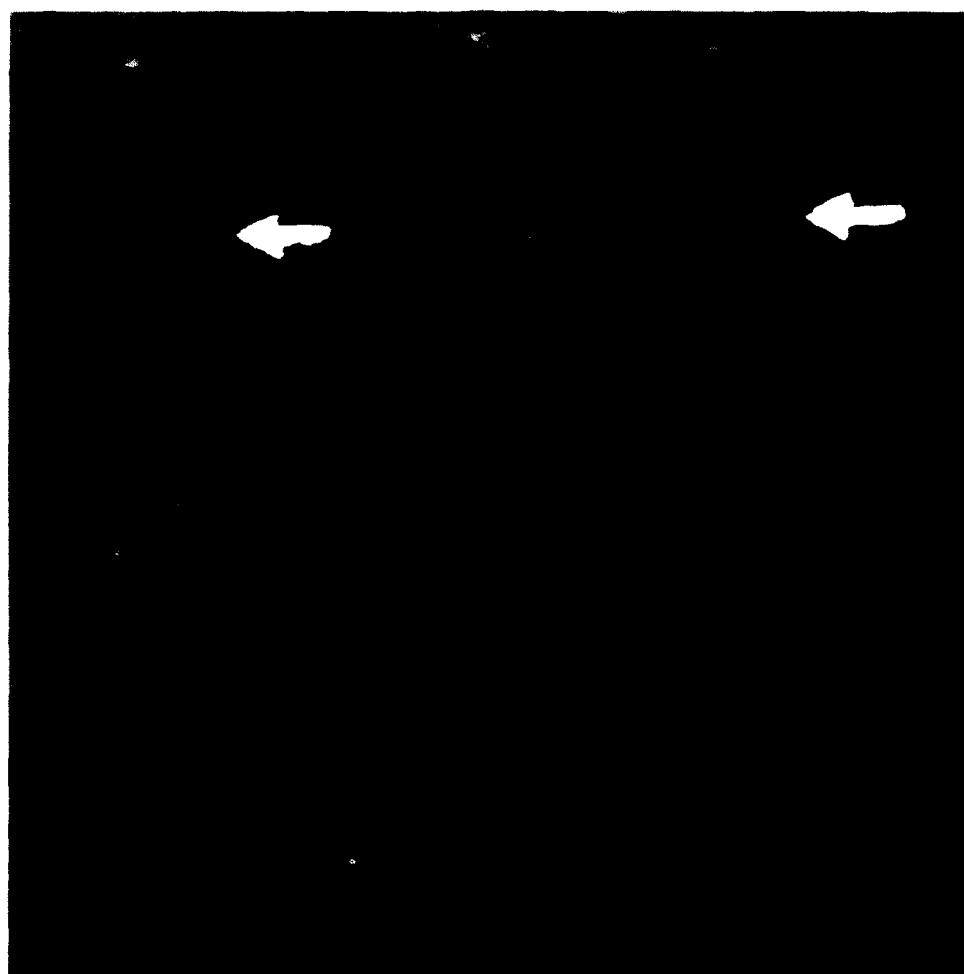


Figure 48. Small Targets in Background Clutter - Thermal Imagery at 0.80 Bits Per Pixel

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS

Results

This study has presented the development and testing of a DCT-based multisensor image compression scheme that adapts to changing source statistics and is suitable for real-time applications. After describing the imaging system used to collect the imagery for this study, a detailed analysis of three channels of image data was conducted. An efficient and robust method of removing a major portion of the correlation between the laser channels (P and C) by means of an approximation to a principal components rotation was developed. The spectral and spatial domain statistics of the two rotated laser channels and the Thermal-IR channel were computed. These statistics were then used to develop a generalized covariance model under wide-sense stationarity assumption. The model was used to determine the efficiency of various transforms, and to develop the bit allocation strategy and the coefficient normalization and quantization scheme. The nonstationary nature of the imagery was considered, and a method of updating the model parameters and classifying the blocks that have high information content was developed. The resulting scheme was implemented in a massively parallel processor using fast algorithms. The compression scheme was tested using a number of images from a

very large multisensor image database, and its performance characteristics were defined. The goal of developing an efficient scheme that compresses the imagery while maintaining adequate performance of target cueing algorithms was demonstrated.

A summary of the major contributions of this study is as follows:

1. Developed a methodology for the design and implementation of adaptive compression schemes for multisensor image data.
2. Adopted and verified a mathematical model for the multisensor source, and applied it to the development and implementation of an adaptive image compression algorithm.
3. Developed and tested a transform based image compression technique suitable for real-time applications.
4. Developed a technique for removing multiplicative noise from laser imagery.
5. Developed a novel technique for rapidly removing interchannel correlations.
6. Demonstrated that multisensor image data can be substantially compressed without affecting the performance of automatic target cueing algorithms.
7. Demonstrated that multisensor image data can be substantially compressed without degrading photo-interpretation capabilities.

Future Work

During the course of the research for this dissertation, a number of areas requiring further work were identified. Some topics were considered outside of the scope of this study, but they will need to be fully explored before a truly optimal multisensor image compression scheme can be designed. Some of the topics which were not considered in detail include:

1. The optimality of the generalized covariance model should be investigated further. In particular, the fixed model parameters derived by Mauersberger [54] should be optimized using actual multisensor image data, and the effect of varying the update rate of the adaptive parameters of the model should be fully investigated.
2. The scheme for classifying the high information (high AC variance) blocks should be made more adaptive to changes in source statistics. For example, rather than establishing the number of high variance blocks a priori, a scheme that determines when the block variances approach the model variances should be considered.
3. The effect of varying the transform block size was not considered. It is very likely that larger block sizes can provide higher compression ratios for the same allowable distortion. However, the tradeoffs between higher compression performance and resolution of very small targets need to be considered before going to larger blocks.
4. Recently developed transforms such as the wavelet transform should be

considered.

5. Since the proposed scheme generates variable length codes, buffering will be required if a constant rate transmission channel is used. The buffer fill state should be considered as a method of adjusting the adaptivity of the compression scheme for improved performance.

6. The effects of channel errors on the compression scheme should be investigated. It is expected, since this scheme relies on transform coding of small blocks which limits the extent of distortions caused by channel errors, that only a small portion of the overhead information needs to be transmitted with error correcting codes. The effects of errors in the received values of the model parameters and the rotation angle on image reconstruction should be investigated.

A number of techniques are readily available for improving the scheme presented in this work. These techniques are relatively uncomplicated to implement, and should be included in any high performance DCT-based adaptive compression. These improvements include:

1. Replacement of the indirect 2-D DCT implementation (applying 1-D algorithm twice) with a more computationally efficient, direct 2-D DCT algorithm [51].

2. Implement the DCT using integer arithmetic rather than the floating point implementation used in this study. However, the effect of reduced dynamic range and possible truncation and overflow problems must be evaluated.

3. Increase the execution efficiency by coding the algorithms in assembly language rather than using relatively inefficient high level languages such as FORTRAN - Plus.
4. Make the scheme adaptive to changes in sensor resolution (a function of altitude and groundspeed).
5. Arithmetic coding of the quantized transform coefficients and some of the overhead data such as the variance matrices and the model parameters should be incorporated in order to increase the compression ratios.

APPENDIX A

REMOVAL OF COHERENT LASER POWER VARIATIONS

Introduction

The purpose of this appendix is to describe the operations performed on the P and C laser channels in order to correct image distortions caused by instabilities of the laser source. It is assumed that future operational laser imaging systems will be capable of preventing or eliminating these distortions prior to the image compression stage. For this reason, noise removal processing was not included as part of the image compression system requirements. Coherent laser noise removal techniques are included here as separate pre-processing steps needed in order to ensure that the models and algorithms in this study are developed using imagery with the highest signal-to-noise ratio possible.

Removal of Coherent Laser Noise

Figures 49,50, and 51 show 512 by 512 pixel sections of imagery from the thermal, P, and C channels respectively of a scene composed of a camouflaged truck in a desert environment. The high frequency noise that is very visible in the active (P and C) channels, but absent from the passive thermal-IR channel has been attributed to instabilities in the output of the laser source induced by mechanical vibrations in the helicopter platform. A laser power monitor circuit was added in an attempt to pre-process the active imagery prior to digitization. This effort was unsuccessful, and



Figure 49. Unprocessed Thermal Infrared Channel Imagery

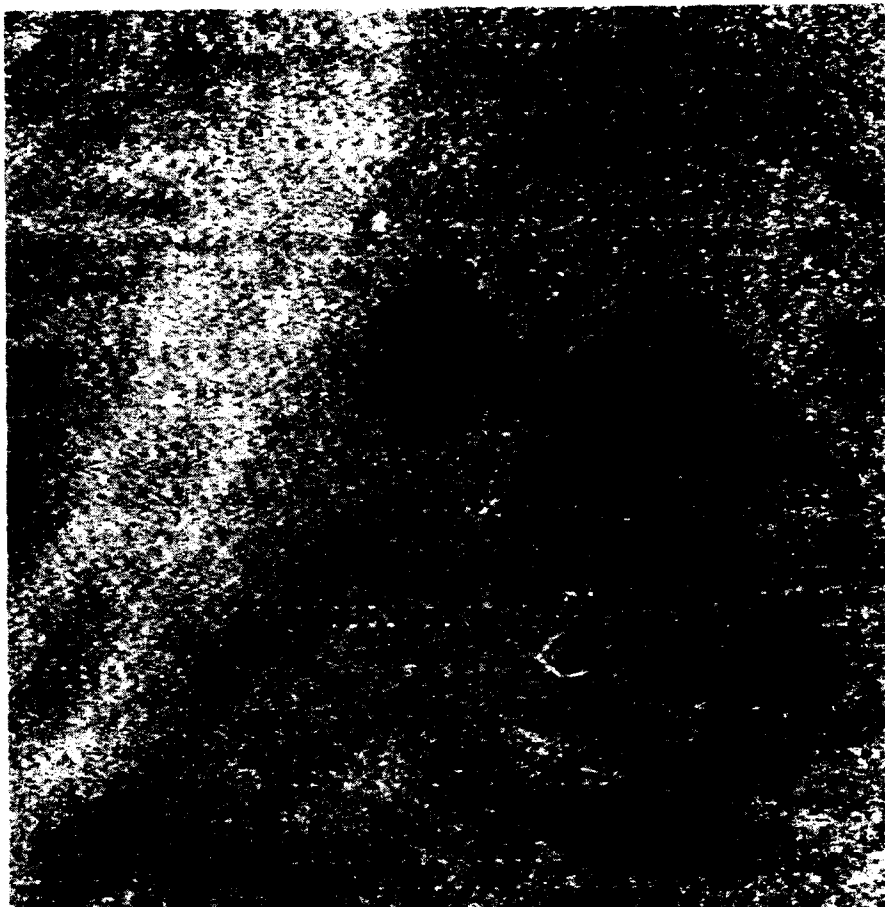


Figure 50. Unprocessed P Channel Imagery

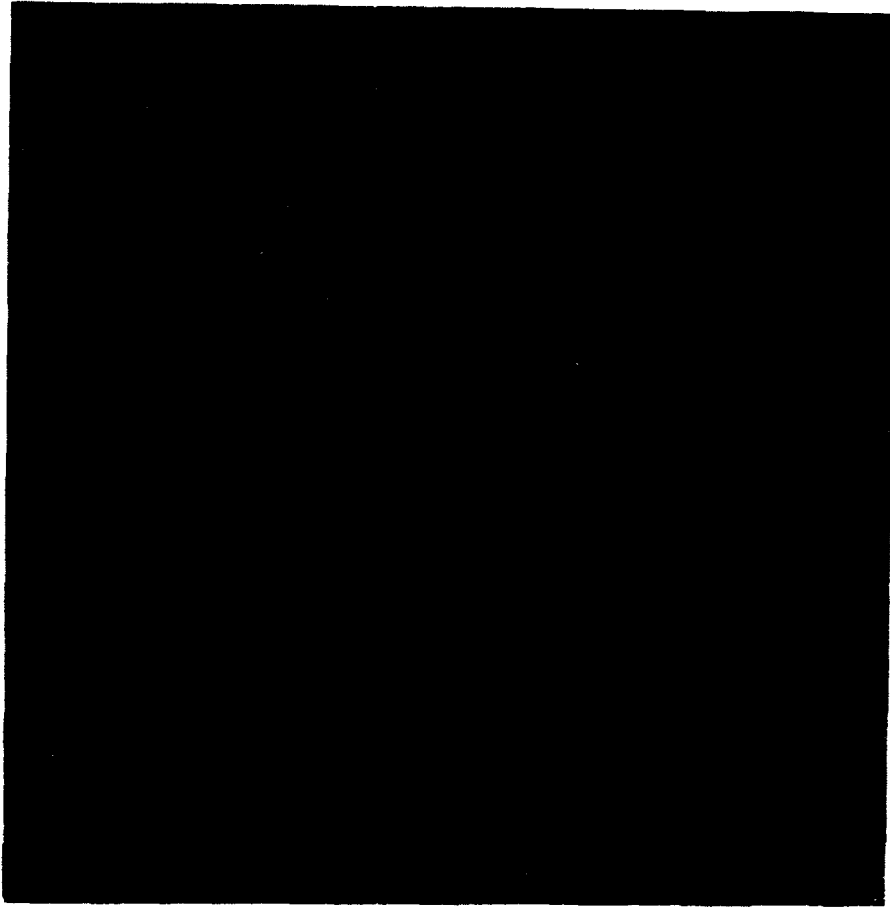


Figure 51. Unprocessed C Channel Imagery

post-processing of the digital data using the digitized laser power monitor data is presently required to clean the active imagery. Figure 52 shows the digitized laser power variations from the power monitor circuit.

Two dimensional power spectra plots of these images are shown in Figures 53, 54, 55, and 56, which correspond to the thermal, P, C, and laser power channels respectively. The horizontal and vertical frequency axes have been normalized by the sampling frequencies (1.05 MHz horizontal and 350 Hz vertical). Comparison of these plots show a high degree of similarity between the active channels and the laser power. In addition, from a data compression perspective, comparison of the power spectra of the active channels with the thermal channel indicate that the laser power noise significantly increases the bandwidth of the image data and therefore decreases the compressibility of the data. It should be pointed out that spatial domain analysis of this data also shows that the vertical correlation coefficients (ρ_v) of the P and C image data are considerably lowered by the laser noise.

In these spectra plots, the variations in laser power, being almost pure sinusoids of fixed horizontal spatial frequency, appear as impulses on the horizontal axis. Since the onsets of the power fluctuations are random, the phase of the observed image oscillations is also random, and therefore the impulses tend to form noise stripes parallel to the vertical axis [5]. Initially it would appear to be a simple

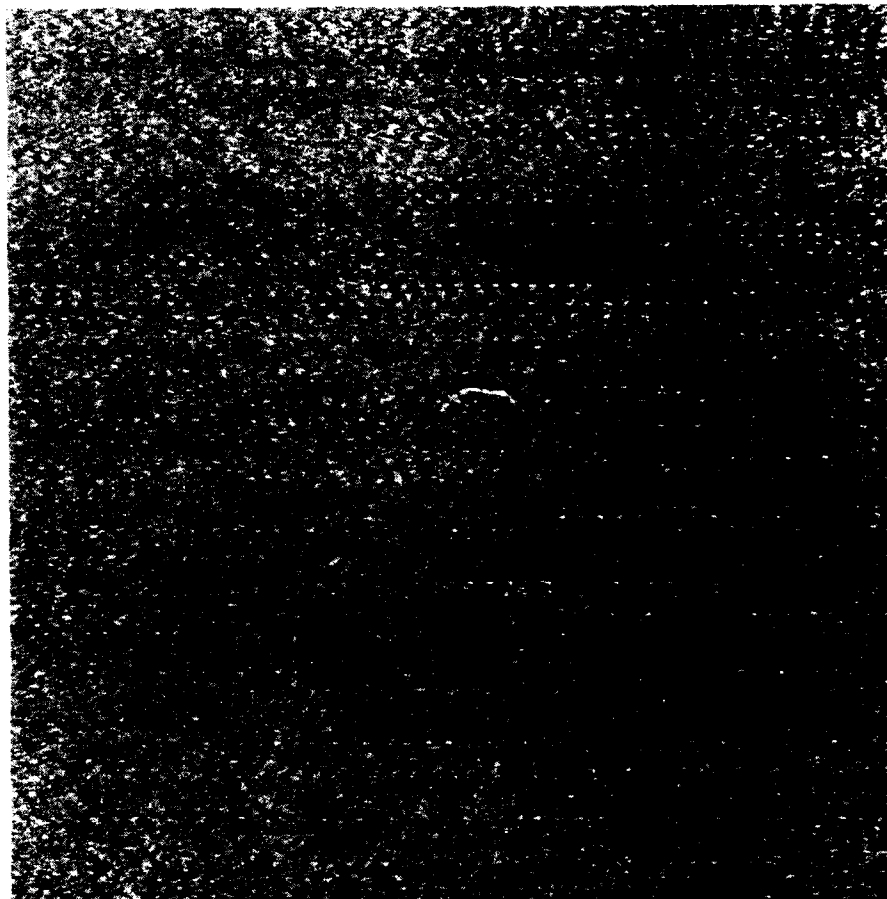


Figure 52. Laser Power Monitor Channel

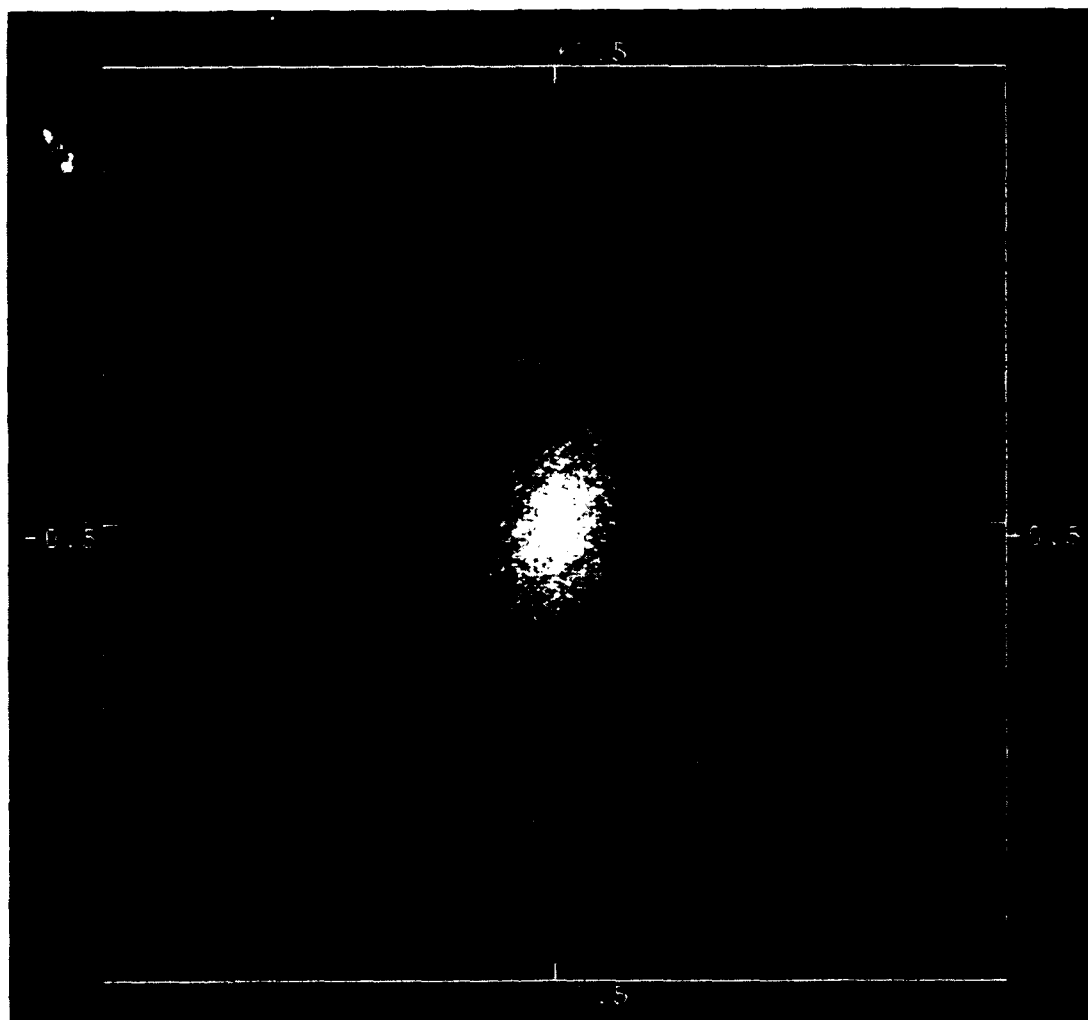


Figure 53. Two Dimensional Power Spectrum of Figure 49

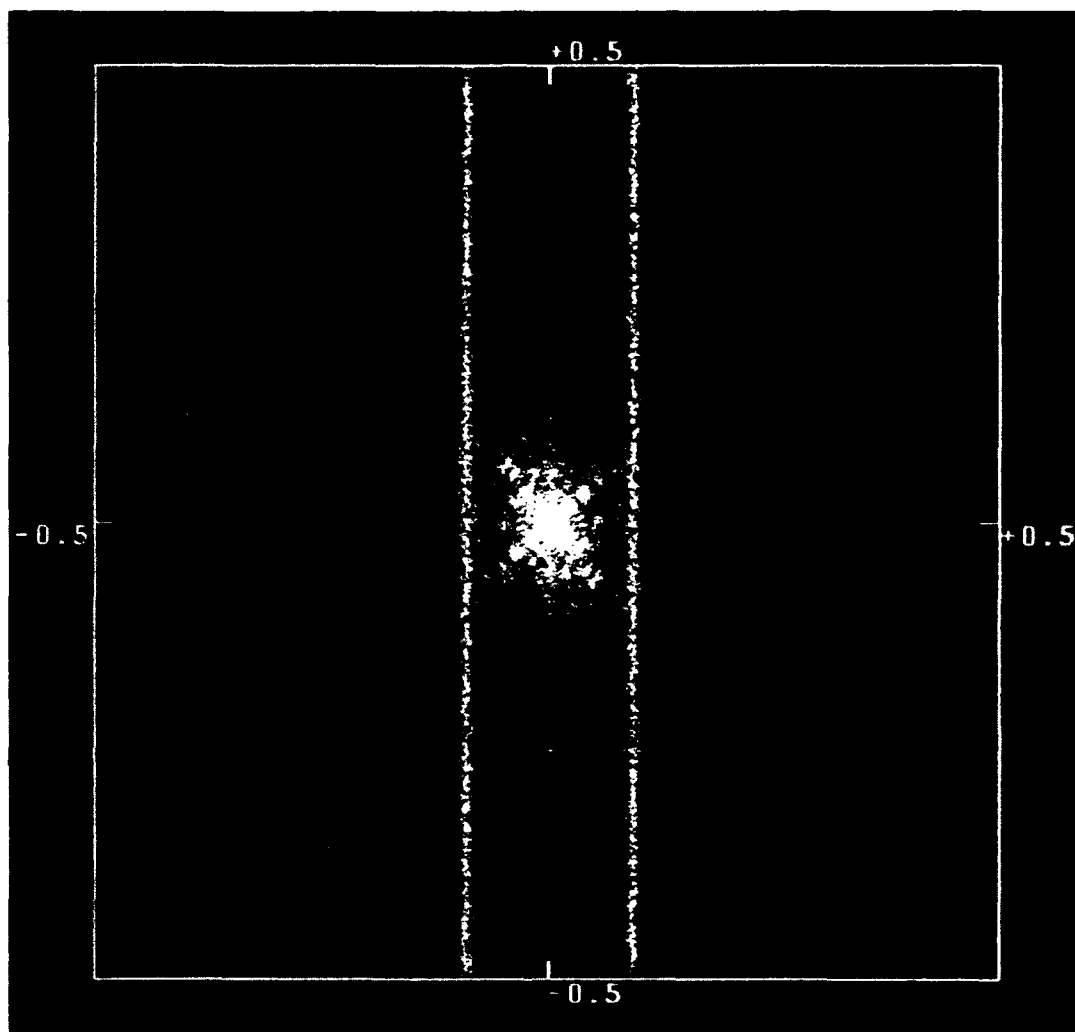


Figure 54. Two Dimensional Power Spectrum of Figure 50

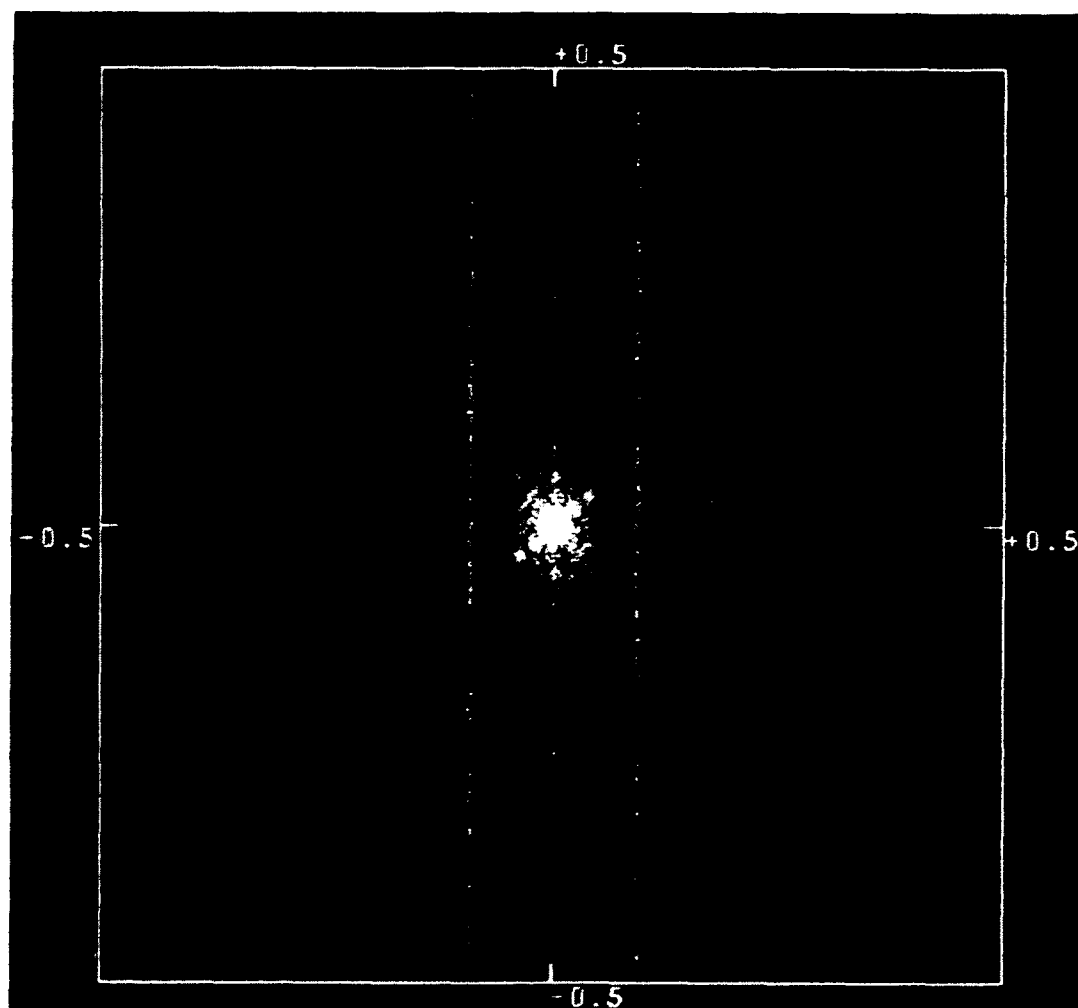


Figure 55. Two Dimensional Power Spectrum of Figure 51

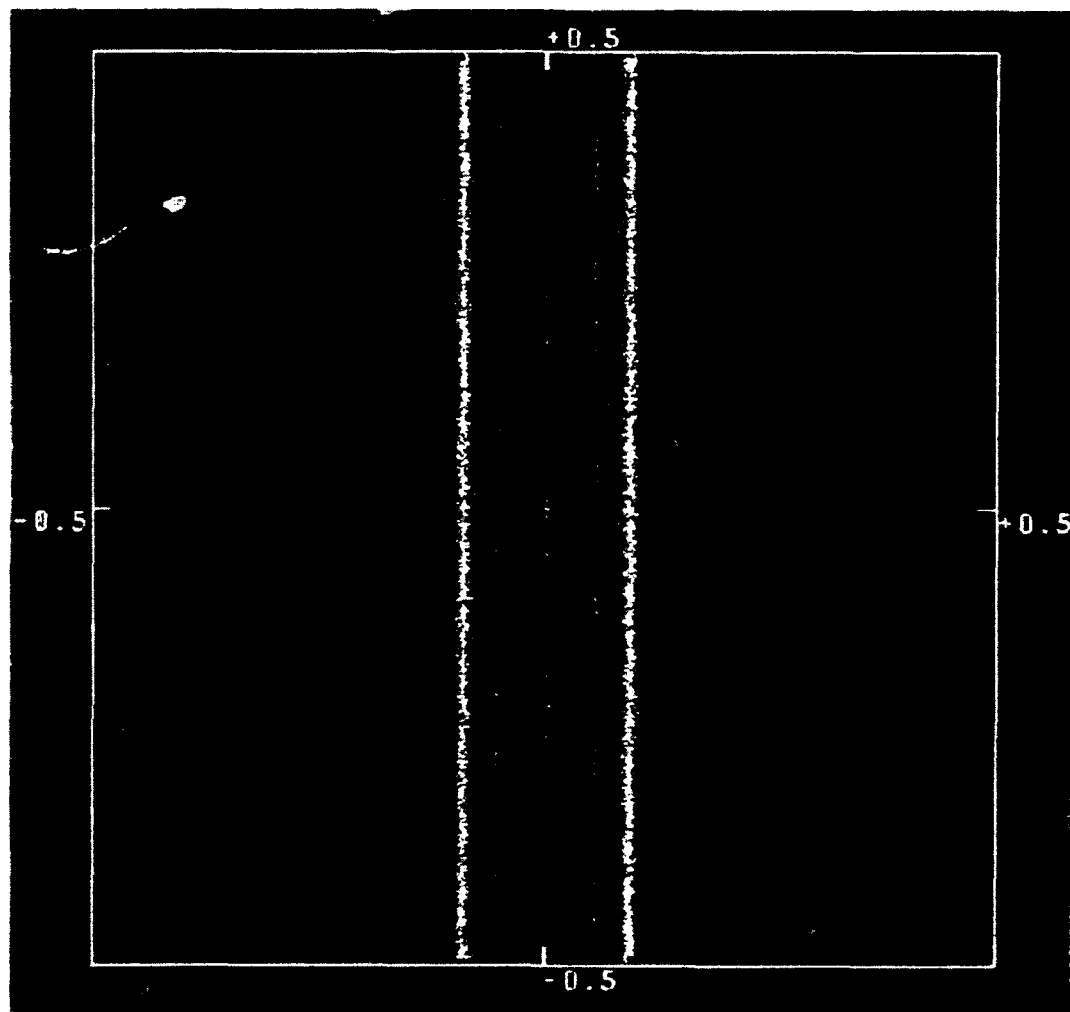


Figure 56. Two Dimensional Power Spectrum of Figure 52

problem to remove these noise spikes by using the techniques described by Moik [5]. These techniques, however, are not applicable in this case because here we are dealing with multiplicative rather than additive noise. In addition, the noise frequency components fall within a band that contains a significant amount of image information.

In a standard restoration problem where the image data has been corrupted by additive noise, one technique that has proven successful involves isolating the noise, creating a noise-only image, and subtracting it from the original degraded image. The gains and offsets of the two images are iteratively (or interactively) adjusted in order to reduce the noise component on the original image to zero. In the present case where we are dealing with multiplicative noise, the goal is to adjust the noise image so that the noise component in the original image is reduced to a factor of one. In order to define the operations required to achieve this goal, a close analysis of a large number of images was performed. Figures 57, 58, and 59 show single line plots of corresponding lines of P, C, and laser power image data. These plots consist of the intensity values of one 710 pixel wide line which transects the truck shown in the previous imagery. The truck image data corresponds to the sharply lower intensity values located between pixels number 200 and 300. The laser power variations are predominant near the beginning and the end of the scan line.

Expanded plots of a noisy section of image data together with the corresponding laser data are shown in Figures 60 and 61. From these plots it can be

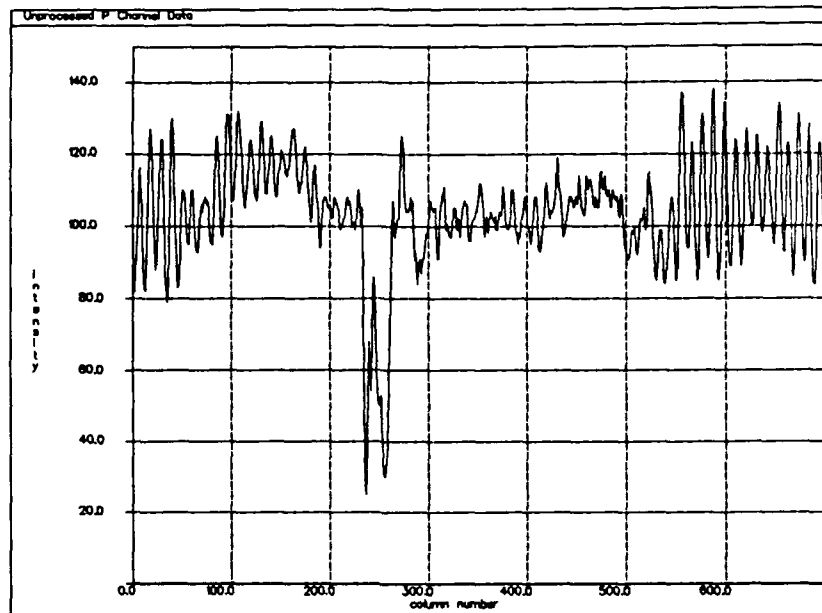


Figure 57. Plot of Single Line of Noisy P Channel Imagery

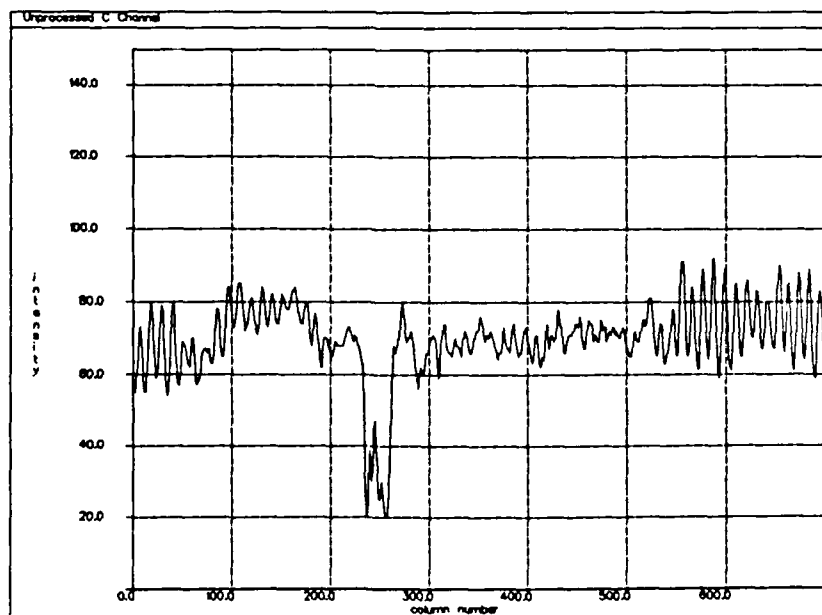


Figure 58. Plot of Single Line of Noisy C Channel Imagery

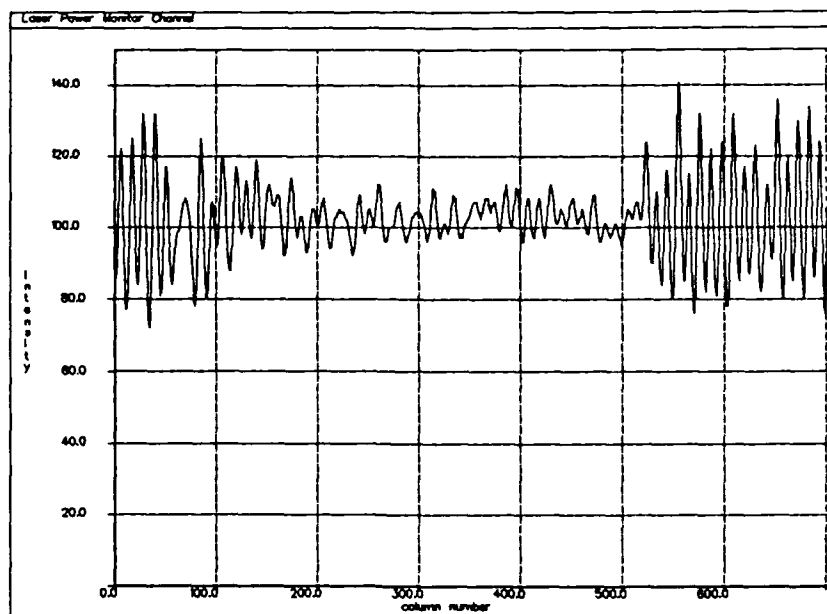


Figure 59. Plot of Single Line of Laser Power Channel

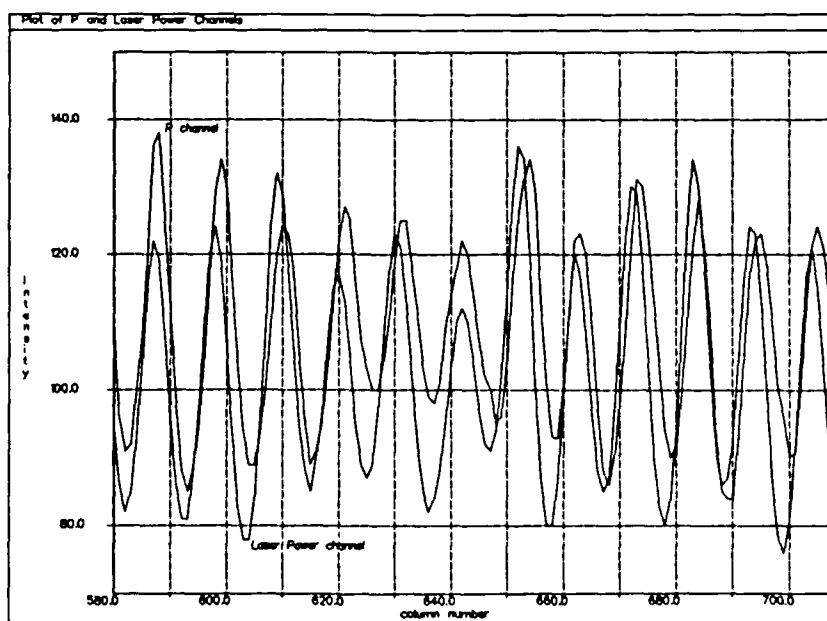


Figure 60. Plot of P and Laser Power Channels

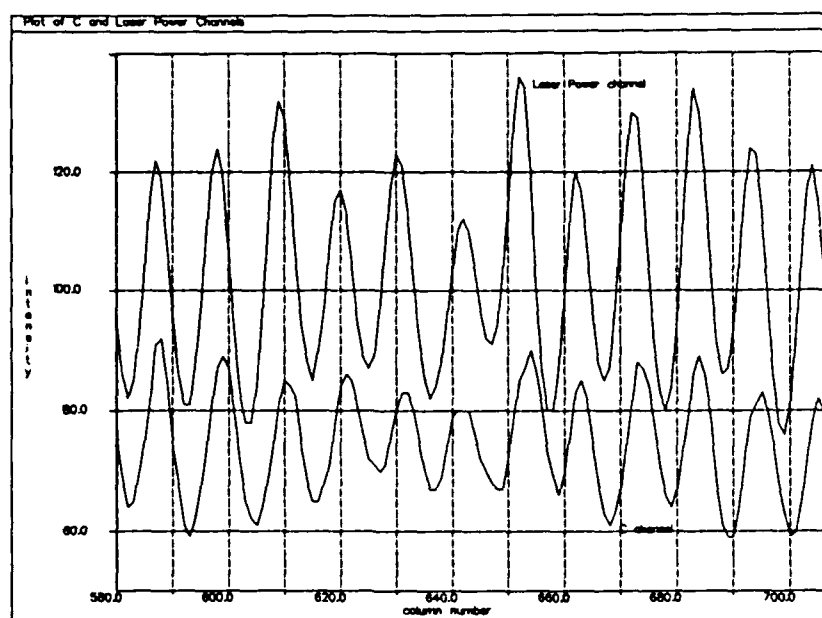


Figure 61. Plot of C and Laser Power Channels

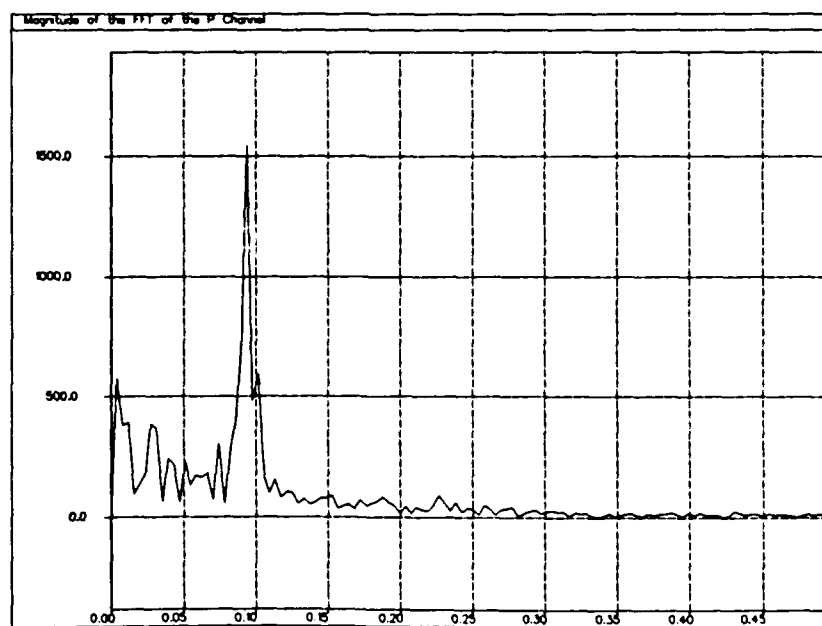


Figure 62. Magnitude of the FFT of Noisy Section of P Channel

determined that scale factors, offsets, and phase adjustments of the laser power data are required to remove the multiplicative noise from the image channels. Figure 62 shows the magnitude of the one dimensional FFT of the noisy section of P channel data from Figure 60. The laser power contribution appears as a sharp peak that was used to estimate the phase correction and scale factor to be applied to the laser power data. The image correction operations required to remove the coherent laser power variations are shown in the flowchart of Figure 63.

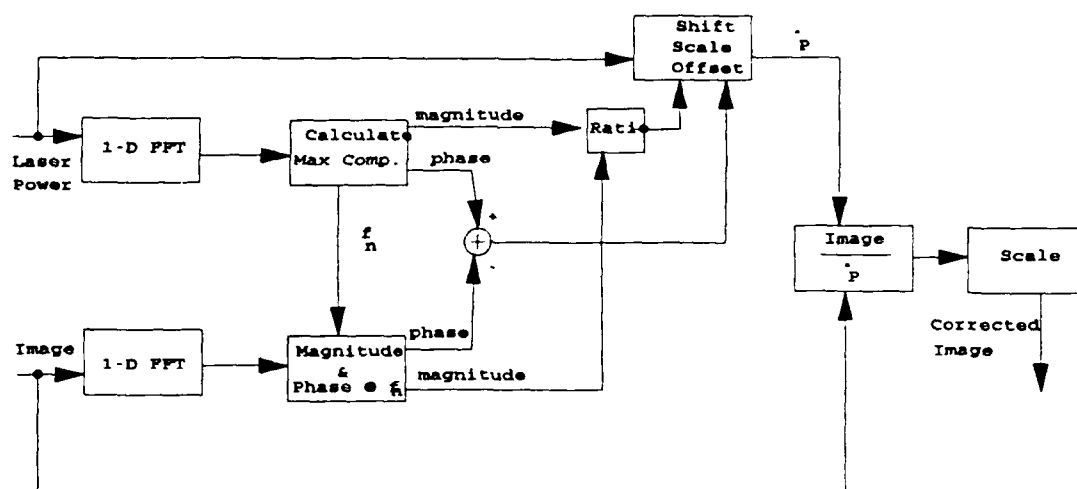


Figure 63. Flowchart of Image Correction Operations

In this processing scheme, the laser power data is transformed line by line using a one dimensional FFT that outputs the complex-valued results in polar format (magnitude and phase). The non-zero frequency component having the largest magnitude is found, and the magnitude, phase, and frequency (f_n) are determined. The frequency f_n is used to select the corresponding component in the image channel.

The corresponding magnitudes and phases at f_n are used to scale and shift the laser power data. It should be noted that the phase shifts required ranged from 0 to 45 degrees so that there are no phase unwrapping requirements in this application. The D.C. level of the laser power channel is also changed by an offset computed from global averages of the image and laser power data. The original image channel data are then divided pixel-by-pixel by the corrected laser power data. The results are then re-scaled by the global mean of the original image data.

Figure 64 shows the results of these operations on a single line of noisy data from the P channel. Figure 65 shows the results on the 512 by 512 pixel image data from Figure 50. It can be observed that the high frequency laser noise has been significantly attenuated. Processing of the active channel imagery by this technique resulted in an average 25 dB attenuation of the laser power noise.

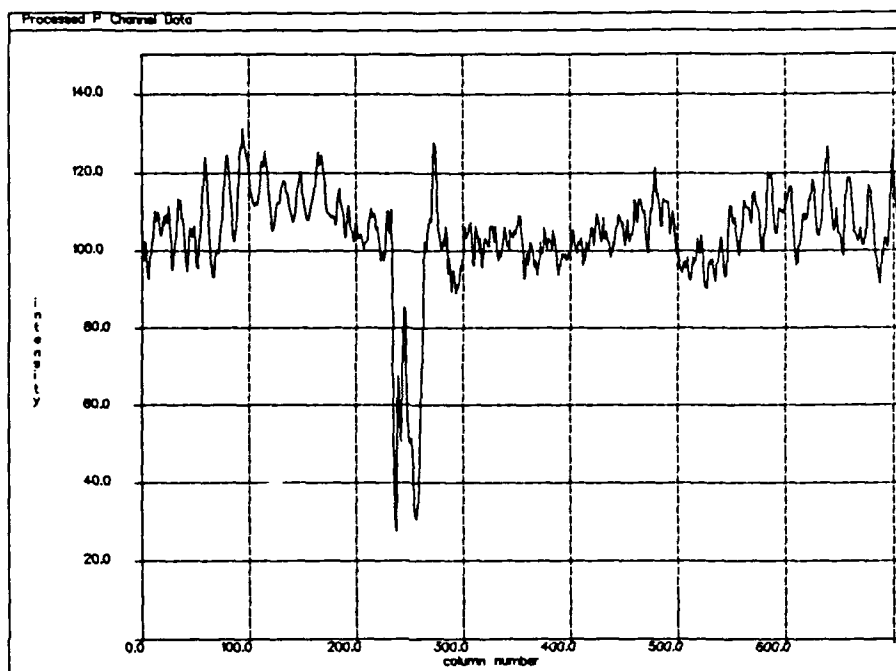


Figure 64. Data From Figure 57 After Processing to Remove Noise



Figure 65. Processed P Channel Imagery

APPENDIX B

IMAGE PROCESSING ALGORITHM SOURCE CODE

Introduction

This appendix includes the source code of the program used to implement the multichannel adaptive transform coding scheme developed in this study. The algorithm is coded in FORTRAN-Plus which is an extension of FORTRAN-XX that facilitates parallel operations. A complete description of the language is included in [58]. It should be noted that the objective of this study is to develop and test techniques that are broadly applicable to compression of imagery generated by multisensor systems; therefore, no attempt has been made to optimize the software implementation of a scheme used for the specific imagery used in this study.

FORTRAN Plus Source Code

```
entry subroutine start
c*****
c      This subroutine contains the main DAP program that performs all of the
c      model calculations, compression and decompression, and display operations
c*****
#include compress.h

integer*1 obuf(,sxx)
integer*4 obufi(8192)
real vcor(3), hcor(3), var(3), covar(8,8,3), cvar(8,8,3), btrate
integer bit_all(8,8,3), norm(8,8,3)
character filein(32)
```

```

character fileout(32)
character*1 keyboard(32) //'dev/tty'
character mess(32,8) 'Enter p filename      : ',
&                'Enter c filename      : ',
&                'Enter thermal filename : ',
&                'Enter rotated p filename : ',
&                'Enter rotated c filename : ',
&                'Enter decompressed p filename : ',
&                'Enter decompressed c filename : ',
&                'Enter decompressed t filename : '
equivalence (obuf, obufi)

#pdt MAP sh2 = (a:8...6^1 2...0^0|r:5...0^2|c:5...0^1)
#pdt MAP in = (a:5...0^2 |r:8...3^1|c:2...0^1 2...0^0)

c    Initialize display

    call start_graphics(0,disp,ierr)
    call set_lut (1,ierr)
    do 5 i=1, sx
        do 5 j=1, sy
            disp(.,j,i) = 0
5    continue
    call put_frame (ierr)

c    Open terminal for reading and writing

    call amt5_open (keyboard, 'r', ifd, ifail)
    call amt5_open (keyboard, 'w', ofd, ifail)

c    Read input files and display

    do 50 k=1, 3

c        Open input file

        call write_char (mess(1,k), 32)
        call read_char (filein)
        call amt5_open (filein, 'r', ifd2, ierror)

c        Read image and convert format

```

```

      do 20 i=1, syy
        call amt5_read (ifd2, obufl, 32768, numsent, ifail)
#pdt REMAP FROM=in, TO=sh2, DATA=obuf, LENGTH=1
        do 10 j=1, sxx
          im(.,i,j,k) = obuf(.,j)
10      continue
20      continue

c      Zoom image and display

      do 30 i=1, sxx
        do 30 j=1, syy
          disp(.,j+4,i+4) = im(.,j,i,k)
30      continue
        call zoom
        call put_frame (ierr)
50      continue

c      Rotate P and C images into P' and C' and display

      call rotpc (im(.,1,1,1), im(.,1,1,2),pol,theta)
      do 70 k=1, 2
        do 60 i=1, sxx
          do 60 j=1, syy
            disp(.,j+4,i+4) = im(.,j,i,k)
60      continue
c      Zoom image 2x to fill 1024x1024 display
        call zoom
        call put_frame (ierr)
70      continue

c      Calculate data variances for each 512x512 section of imagery

      call variance (im(.,1,1,1), var(1))
      call variance (im(.,1,1,2), var(2))
      call variance (im(.,1,1,3), var(3))

c      Calculate the vertical and horizontal correlation coefficients

```

```

call auto_corr (im(.,1,1,1), vcor(1), hcor(1))
call auto_corr (im(.,1,1,2), vcor(2), hcor(2))
call auto_corr (im(.,1,1,3), vcor(3), hcor(3))

```

- c Calculate the data covariances using the generalized covariance model

```

call covariance (var(1), vcor(1), hcor(1), covar(1,1,1))
call covariance (var(2), vcor(2), hcor(2), covar(1,1,2))
call covariance (var(3), vcor(3), hcor(3), covar(1,1,3))

```

- c Estimate the DCT coefficient variances using the model and the W matrix

```

call coef_var (covar(1,1,1), cvar(1,1,1))
call coef_var (covar(1,1,2), cvar(1,1,2))
call coef_var (covar(1,1,3), cvar(1,1,3))

```

- c Select the desired average bit rate and
c generate the bit allocation matrix and
c the normalization factors for each channel

```

btrate = .25
call bit_alloc (cvar(1,1,1), bit_all(1,1,1), norm(1,1,1), btrate)
btrate = .25
call bit_alloc (cvar(1,1,2), bit_all(1,1,2), norm(1,1,2), btrate)
btrate = .25
call bit_alloc (cvar(1,1,3), bit_all(1,1,3), norm(1,1,3), btrate)

```

- c Compress/decompress and display result

```

do 100 k=1, 3
  do 80 i=1, sxx
    do 80 j=1, syx
      disp(.,j+4,i+4) = im(.,j,i,k)
80    continue
    call zoom
    call put_frame (ierr)
    pause 400
    call dct (im(.,1,1,k),bit_all(1,1,k),norm(1,1,k),cvar(1,1,k))
    do 90 i=1, sxx
      do 90 j=1, syx
        disp(.,j+4,i+4) = im(.,j,i,k)
90    continue

```

```

        call zoom
        call put_frame (ierr)
        pause 500
100  continue
c    Perform inverse rotation to recover P and C channels
        call revrot(im(,,1,1,1),im(,,1,1,2),pol,theta)
        do 102 k=1,3
            do 101 i=1,sxx
                do 101 j=1,syy
                    disp(,,j+4,i+4) = im(,,j,i,k)
101  continue
                call zoom
                call put_frame (ierr)
                pause 501
102  continue
c    Write decompressed images to disk

        do 120 k=1, 3

c        Open output file

            call write_char (mess(1,k+5), 32)
            call read_char (fileout)
            call amt5_open (fileout, 'w', ofd2, ierror)

c        Convert format and write image

            do 110 i=1, syy
                do 105 j=1, sxx
                    obuf(,,j) = im(,,i,j,k)
105  continue
            #pdt REMAP FROM=sh2, TO=in, DATA=obuf, LENGTH=1
                call amt5_write (ofd2, obufi, 32768, numsent, ifail)
110  continue
120  continue

        call amt5_close (ifd2, ifail)
        return
end

```

```

subroutine dct (im1, bit_all, norm, cvar)
c*****
c    This subroutine implements an 8x8 2-D DCT using Hou's
c    recursive method. It also performs the quantization of
c    the coefficients by means of optimal non-uniform
c    Laplacian quantizers.
c*****

#include compress.h

integer*1 im1(,syy,sxx)
integer bit_all(8,8), norm(8,8), bitall2(8,8)
real cvar(8,8)
integer*2 itemp(,)
logical l1(,), mask2(,)
real dctblk(,)
real temp(,)

c
c    The following matrix defines the decision levels of an optimum non-uniform
c    Laplacian quantizer with number of bits allocated ranging from 1 to 5
c
real bound(33,4)/-9999.0,-1.127,0.0,1.127,9999.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,
&      -9999.0,-2.380,-1.253,-.533,0.0,.533,
&      1.253, 2.380,9999.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      -9999.0,-3.725,-2.597,-1.878,-1.345,-.92,
&      -.577,-.264,0.0,.264,.577,.92,1.345,1.878,
&      2.597,3.725,9999.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      -9999.0,-5.13,-4.0,-3.28,-2.75,-2.32,-1.97,
&      -1.67,-1.4,-1.17,-1.0,-.76,-.59,-.43,-.27,-.13,
&      0.0,.13,.27,.42,.59,.76,1.0,1.17,1.4,1.67,
&      1.97,2.32,2.75,3.28,4.0,5.13,9999.0/

c
c    The following matrix defines the assignment values of an optimum non-uniform
c    Laplacian quantizer with number of bits allocated ranging from 1 to 5
c

```

```

      real cval(32,4)/-1.834,-.42,.42,1.834,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      -3.087,-1.673,-.833,-.233,.233,.833,1.673,
&      3.087,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      0.0,0.0,-4.432,-3.017,-2.178,-1.578,-1.111,-.729,
&      -.405,-.124,.124,.405,.729,1.111,1.578,
&      2.178,3.017,4.432,0.0,0.0,0.0,0.0,
&      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
&      -5.83,-4.42,-3.58,-2.98,-2.51,-2.13,-1.81,-1.53,
&      -1.28,-1.06,-.86,-.67,-.5,-.35,-.2,-.06,.06,.2,
&      .35,.5,.67,.86,1.06,1.28,1.53,1.81,2.13,2.51,
&      2.98,3.58,4.42,5.83/
      integer*2 qdct(,syy,sxx)

      do 10 i=1, sxx
        do 10 j=1, syy
          dctbuf(,j,i) = im1(,j,i)
          dctbuf(dctbuf(,j,i).lt.0,j,i) = dctbuf(,j,i) + 256
10      continue
        call sheet_crink (dctbuf, 32, syy, sxx)
        do 20 i=1, sxx
          call dct8(dctbuf(,1,i))
20      continue
        do 22 i=1,sxx
          do 22 j=1,syy
            if(j .eq. 1) then
              dctbuf(,j,i) = dctbuf(,j,i) * .353553
            else
              dctbuf(,j,i) = dctbuf(,j,i) * .5
            endif
22      continue
        #pdt MAP colm = (a:2...0^1 2...0^2 4...0^0|r:8...3^2|c:8...3^1)
        #pdt MAP rowm = (a:0...2^2 2...0^1 4...0^0|r:8...3^2|c:8...3^1)
        #pdt REMAP FROM=colm, TO=rowm, DATA=dctbuf, LENGTH=1
        do 25 i=1, sxx
          call dct8(dctbuf(,1,i))
25      continue

      do 26 i=1,sxx

```

```

do 26 j=1,syy
  if(j .eq. 1) then
    dctbuf(.,j,i) = dctbuf(.,j,i) * .353553
  else
    dctbuf(.,j,i) = dctbuf(.,j,i) * .5
  endif
26  continue
#pdt REMAP FROM=colm, TO=rowm, DATA=dctbuf, LENGTH=1
  pause 3250
  dctblk = 0
  do 27 i=1, sxx
    do 27 j=1, syy
      if (i.eq.1 .and. j .eq. 1) goto 27
      dctblk = dctblk + (dctbuf(.,j,i)*dctbuf(.,j,i))
27  continue
  call adapt(dctblk, mask2)
  btrate = 4
  call bit_alloc(dctvar, bitall2, norm, btrate)
  do 300 i=1, sxx
    do 300 j=1, syy
      if (i.eq.1 .and. j.eq.1) then
        qdct(.,j,i) = dctbuf(.,j,i) / 8.0 + .5
        dctbuf(.,j,i) = qdct(.,j,i) * 8
        go to 300
      endif
      sd = sqrt(cvar(j,i))
      sd2 = sqrt(dctvar(j,i))
      temp(.not. mask2) = dctbuf(.,j,i) / sd
      temp(mask2) = dctbuf(.,j,i) / sd2
      nb = bit_all(j,i)
      nb2 = bitall2(j,i)
      if (nb .gt. 5) nb = 5
      if (nb2 .gt. 5) nb2 = 5
      if(nb .eq. 0) then
        qdct(.not. mask2,j,i) = 0
        dctbuf(.not. mask2,j,i) = 0.0
      else if(nb .eq. 1) then
        qdct(.not. mask2,j,i) = 0
        qdct((temp.gt.0).and.(.not. mask2),j,i) = 1
        dctbuf(.not. mask2,j,i) = -0.707*sd
        dctbuf((temp.gt.0).and.(.not. mask2),j,i) = 0.707*sd

```

```

else
  do 290 k=1, 2**nb
    l1 = (.not. mask2) .and. (temp.gt.bound(k,nb-1)) .and.
    &      (temp.lt.bound(k+1,nb-1))
    qdct(l1,j,i) = k-1
    dctbuf(l1,j,i) = cval(k,nb-1)*sd
290    continue
  endif
  if(nb2 .eq. 0) then
    qdct(mask2,j,i) = 0
    dctbuf(mask2,j,i) = 0.0
  else if(nb2 .eq. 1) then
    qdct(mask2,j,i) = 0
    qdct((temp.gt.0).and.(mask2),j,i) = 1
    dctbuf(mask2,j,i) = -0.707*sd2
    dctbuf((temp.gt.0).and.(mask2),j,i) = 0.707*sd2
  else
    do 295 k=1, 2**nb2
      l1 = (mask2) .and. (temp.gt.bound(k,nb2-1)) .and.
      &      (temp.lt.bound(k+1,nb2-1))
      qdct(l1,j,i) = k-1
      dctbuf(l1,j,i) = cval(k,nb2-1)*sd2
295    continue
    endif
300    continue
    pause 120

#pdt REMAP FROM=rowm, TO=colm, DATA=dctbuf, LENGTH=1
    do 28 i=1,sxx
      do 28 j=1,syy
        if(j .eq. 1) then
          dctbuf(.,j,i) = dctbuf(.,j,i) * 2.828427
        else
          dctbuf(.,j,i) = dctbuf(.,j,i) * 2.0
        endif
28      continue

      do 30 i=1, sxx
        call idct8(dctbuf(.,1,i))
30      continue
#pdt REMAP FROM=rowm, TO=colm, DATA=dctbuf, LENGTH=1

```

```

do 32 i=1,sxx
  do 32 j=1,syy
    if(j .eq. 1) then
      dctbuf(.,j,i) = dctbuf(.,j,i) * 2.828427
    else
      dctbuf(.,j,i) = dctbuf(.,j,i) * 2.0
    endif
32  continue

  do 35 i=1, sxx
    call idct8(dctbuf(.,1,i))
35  continue
  call crink_sheet (dctbuf, 32, syy, sxx)
  do 40 i=1, sxx
    do 40 j=1, syy
      itemp = dctbuf(.,j,i)
      itemp(itemp .gt. 255) = 255
      itemp(itemp .lt. 0) = 0
      itemp(itemp .gt. 127) = itemp - 256
      im1(.,j,i) = itemp
40  continue

  return
end

```

subroutine dct8 (raw)

real raw(.,8), t(.,8)

```

t(.,1) = raw(.,1) + raw(.,8)
t(.,2) = raw(.,3) + raw(.,6)
t(.,3) = raw(.,5) + raw(.,4)
t(.,4) = raw(.,7) + raw(.,2)
call dct4 (t)
t(.,5) = (raw(.,1) - raw(.,8)) * cos(0.19635)
t(.,6) = (raw(.,3) - raw(.,6)) * cos(0.98175)
t(.,7) = (raw(.,5) - raw(.,4)) * cos(1.76715)
t(.,8) = (raw(.,7) - raw(.,2)) * cos(2.55254)
call dct4(t(.,5))

```

```

t(,,7) = 2*t(,,7) - t(,,5)
t(,,6) = 2*t(,,6) - t(,,7)
t(,,8) = 2*t(,,8) - t(,,6)
do 10 i=1, 8
    raw(,,i) = t(,,i)
10  continue

return
end


subroutine dct4 (raw)

real raw(,,4), t(,,4)

t(,,1) = raw(,,1) + raw(,,3)
t(,,2) = raw(,,2) + raw(,,4)
call dct2 (t)
t(,,3) = (raw(,,1) - raw(,,3)) * cos(0.39270)
t(,,4) = (raw(,,2) - raw(,,4)) * cos(1.96350)
call dct2 (t(,,3))
t(,,4) = 2*t(,,4) - t(,,3)
do 10 i=1,4
    raw(,,i) = t(,,i)
10  continue

return
end


subroutine dct2 (raw)

real raw(,,2), t(,,2)

t(,,1) = raw(,,1) + raw(,,2)
t(,,2) = (raw(,,1) - raw(,,2)) * cos(0.78540)
raw(,,1) = t(,,1)
raw(,,2) = t(,,2)

```

```

return
end

```

```

subroutine idct8 (raw)

```

```

real raw(:,8), t(:,8)

```

```

t(:,1) = raw(:,1)
t(:,2) = raw(:,2)
t(:,3) = raw(:,3)
t(:,4) = raw(:,4)
call idct4 (t)
t(:,5) = raw(:,5)
t(:,6) = (raw(:,7) + raw(:,6)) / 2.0
t(:,7) = (raw(:,5) + raw(:,7)) / 2.0
t(:,8) = (raw(:,6) + raw(:,8)) / 2.0
call idct4(t(:,5))
t(:,5) = t(:,5) / cos(0.19635)
t(:,6) = t(:,6) / cos(0.98175)
t(:,7) = t(:,7) / cos(1.76715)
t(:,8) = t(:,8) / cos(2.55254)
raw(:,1) = (t(:,1) + t(:,5)) / 2.0
raw(:,3) = (t(:,2) + t(:,6)) / 2.0
raw(:,5) = (t(:,3) + t(:,7)) / 2.0
raw(:,7) = (t(:,4) + t(:,8)) / 2.0
raw(:,8) = (raw(:,1) - t(:,5))
raw(:,6) = (raw(:,3) - t(:,6))
raw(:,4) = (raw(:,5) - t(:,7))
raw(:,2) = (raw(:,7) - t(:,8))

```

```

return
end

```

```

subroutine idct4 (raw)

```

```

real raw(:,4), t(:,4)

```

```

t(:,1) = raw(:,1)
t(:,2) = raw(:,2)
call idct2 (t)
t(:,3) = raw(:,3)
t(:,4) = (raw(:,3) + raw(:,4)) / 2.0
call idct2 (t(:,3))
t(:,3) = t(:,3) / cos(0.39270)
t(:,4) = t(:,4) / cos(1.96350)
t(:,1) = (t(:,1) + t(:,3)) / 2.0
t(:,2) = (t(:,2) + t(:,4)) / 2.0
t(:,3) = t(:,1) - t(:,3)
t(:,4) = t(:,2) - t(:,4)
do 10 i=1,4
    raw(:,i) = t(:,i)
10 continue

```

```

return
end

```

```

subroutine idct2 (raw)

```

```

real raw(:,2), t(:,2)

```

```

t(:,1) = raw(:,1)
t(:,2) = raw(:,2) / cos(0.78540)
t(:,1) = (t(:,1) + t(:,2)) / 2.0
t(:,2) = t(:,1) - t(:,2)
raw(:,1) = t(:,1)
raw(:,2) = t(:,2)

```

```

return
end

```

```

subroutine read_char (char)

```

```

#include compress.h

```

```

character*1 char(32)

call amt5_read (ifd, cbuf, 128, numsent, ifail)
call convhs1 (cbuf, 128)

if (numsent .gt. 33) numsent = 33
do 10 i=1, numsent - 1
    char(i) = cbuf(i)
10  continue

do 20 i=numsent, 32
    char(i) = ' '
20  continue

return
end

subroutine write_char (char, numchar)

#include compress.h

character char(128)

do 10 i=1, numchar
    cbuf(i) = char(i)
10  continue
call convsh1 (cbuf, 128)
call amt5_write (ofd, cbuf, numchar, numsent, ierr)

return
end

subroutine zoom

#include compress.h

```

```

    ix = sx/4
    iy = sy/4
    nx = sx/2
    ny = sy/2

    do 10 i=1, nx
        do 10 j=1, ny
            disp(,j,i) = disp(,j+iy,i+ix)
10      continue

    do 20 i=1, nx
        do 20 j=1, ny
            disp(,j+ny,i) = disp(,j,i)
            disp(,j,i+nx) = disp(,j,i)
            disp(,j+ny,i+nx) = disp(,j,i)
20      continue

#pdt MAP sheet = (a:9...6^1 9...6^2 2...0^0|r:5...0^2|c:5...0^1)
#pdt MAP yshuff = (a:9...6^1 8...5^2 2...0^0|r:4...0^2 9^2|c:5...0^1)
#pdt REMAP FROM=sheet, TO=yshuff, DATA=disp, LENGTH=1
#pdt MAP xshuff = (a:8...5^1 9...6^2 2...0^0|r:5...0^2|c:4...0^1 9^1)
#pdt REMAP FROM=sheet, TO=xshuff, DATA=disp, LENGTH=1

    return
    end

```

subroutine rotpc (p, c,pol,theta)

#include compress.h

```

integer*1 p(,syy,sxx), c(,syy,sxx)
real ptemp(,), ctemp(,)
integer*4 itemp(,)

```

c Convert images to signed values

```

call unsig_two (p, 8, syy, sxx)
call unsig_two (c, 8, syy, sxx)

```

c Find mean of p and c

```

meanp = 0
meanc = 0
do 10 i=1, sxx
    do 10 j=1, syy
        meanp = meanp + sum(p(:,j,i))
        meanc = meanc + sum(c(:,j,i))
10  continue
meanp = meanp / (xx*yy) + 128
meanc = meanc / (xx*yy) + 128

```

c Calculate rotation angle

```

pol = float(meanp-meanc) / float(meanp+meanc)
theta = atan((1.0-pol)/(1.0+pol))

```

c Transform images

```

costh = cos(theta)
sinth = sin(theta)
do 20 i=1, sxx
    do 20 j=1, syy
        ptemp = length(p(:,j,i),2) + 128
        ctemp = length(c(:,j,i),2) + 128
        itemp = ptemp*costh + ctemp*sinth + .5
        itemp(itemp.gt.255) = 255
        p(:,j,i) = itemp - 128
        itemp = ctemp*costh - ptemp*sinth + .5
        itemp(itemp.gt.127) = 127
        itemp(itemp.lt. -128) = -128
        c(:,j,i) = itemp
20  continue

```

c Convert images back to unsigned values

```

call two_unsig (p, 8, syy, sxx)
call two_unsig (c, 8, syy, sxx)

```

```

return
end

```

```

subroutine revrot (p1, c1,pol,theta)

#include compress.h

integer*1 p1(.,syy,sxx), c1(.,syy,sxx)
real ptemp(,), ctemp(,)
integer*4 itemp(,)

c   Convert images to signed values

call unsig_two (p1, 8, syy, sxx)
call unsig_two (c1, 8, syy, sxx)

c   Reverse transform P' and C' images

costh = cos(theta)
sinth = sin(theta)
do 20 i=1, sxx
  do 20 j=1, syy
    ptemp = length(p1(.,j,i),2) + 128
    ctemp = length(c1(.,j,i),2)
    itemp = ptemp*costh - ctemp*sinth + .5
    itemp(itemp.gt.255) = 255
    itemp(itemp.lt.0) = 0
    p1(.,j,i) = itemp - 128
    itemp = ctemp*costh + ptemp*sinth + .5
    itemp(itemp.gt.255) = 255
    c1(.,j,i) = itemp - 128
20  continue

c   Convert images back to unsigned values

call two_unsig (p1, 8, syy, sxx)
call two_unsig (c1, 8, syy, sxx)

return
end

```

```
subroutine auto_corr (im1, vcor, hcor)
```

```
#include compress.h
```

```
integer*1 im1(,,syy,sxx), im2(,,syy,sxx)
integer*4 itemp(,), itemp1(,), rsum0(), csum0()
integer*4 rsum1(), csum1(), itemp2(,)
real hcor1(), vcor1()
```

c Convert image to signed values in crinkled format

```
call unsig_two (im1, 8, syy, sxx)
call sheet_crink (im1, 8, syy, sxx)
```

c Find mean

```
mean = 0
do 10 i=1, sxx
  do 10 j=1, syy
    mean = mean + sum(im1(,,j,i))
10 continue
mean = mean / (xx*yy)
```

c Calculate horizontal correlation

```
call shift_image_east_p (im1, 8, syy, sxx, im2, 1)
rsum0 = 0
rsum1 = 0
do 20 i=1, sxx
  itemp = im1(,,1,i) - mean
  itemp1 = im2(,,1,i) - mean
  itemp2 = itemp * itemp
  rsum0 = rsum0 + sumc(itemp2)
  itemp2 = itemp * itemp1
  rsum1 = rsum1 + sumc(itemp2)
20 continue
hcor1 = float(rsum1) / float(rsum0)
hcor = sum(hcor1) / 64.0
```

c Calculate vertical correlation

```

call shift_image_north_p (im1, 8, syy, sxx, im2, 1)
csum0 = 0
csum1 = 0
do 40 i=1, syy
    itemp = im1(.,i,1) - mean
    itemp1 = im2(.,i,1) - mean
    itemp2 = itemp * itemp
    csum0 = csum0 + sumr(itemp2)
    itemp2 = itemp * itemp1
    csum1 = csum1 + sumr(itemp2)
40  continue
vcor1 = float(csum1) / float(csum0)
vcor = sum(vcor1) / 64.0

c  Convert image back to unsigned values in sheet format

call two_unsig (im1, 8, syy, sxx)
call crink_sheet (im1, 8, syy, sxx)

return
end

```

```

subroutine variance (im1, var)

```

```

#include compress.h

```

```

integer*1 im1(.,syy,sxx)

call unsig_two (im1, 8, 8, 8)
itot = 0
do 10 i=1, sxx
    do 10 j=1, syy
        itot = itot + sum(im1(.,j,i))
10  continue
xmean = float(itot) / 262144.0

xtot = 0.0
do 20 i=1, sxx
    do 20 j=1, syy

```

```

        xtot = xtot + sum((im1(:,j,i)-xmean)**2)
20  continue
    var = xtot/262144.0
    call two_unsig (im1, 8, 8, 8)

    return
end

subroutine covariance (var, vcor, hcor, covar)

#include compress.h

    real covar(8,8)

    alpha = -log(hcor)
    beta = -log(vcor)
    do 10 i=1, 8
        do 10 j=1, 8
            temp1 = ((alpha*(i-1)**1.137)**1.4142135)
            temp2 = -(((temp1+((beta*(j-1)**1.09)**1.4142135))**.7071068)
            covar(j,i) = var * exp(temp2)
10  continue

    return
end

subroutine coef_var (covar, cvar)

    real covar(8,8), cvar(8,8)
    real dctmat(8,8)/1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
& 1.749,1.367,0.987,0.419,-0.250,-0.919,-1.487,-1.867,
& 1.499,0.598,-0.353,-1.252,-1.500,-0.869,0.353,1.522,
& 1.250,-0.125,-1.133,-1.051,0.250,1.258,0.633,-1.081,
& 1.000,-0.653,-1.000,0.270,1.000,-0.270,-1.000,0.653,
& 0.750,-0.890,-0.280,0.796,-0.250,-0.589,0.780,-0.316,
& 0.500,-0.815,0.353,0.162,-0.500,0.544,-0.353,0.108,
& 0.250,-0.480,0.426,-0.345,0.250,-0.154,0.073,-0.019/
    real d(,), dt(,), cv(,), temp(,)

```

external real matrix function F01_G_MM

```

d = 0.0
dt = 0.0
cv = 0.0
temp = 0.0
do 10 i=1, 8
    do 10 j=1, 8
        dt(i,j) = dctmat(j,i)
        d(j,i) = dctmat(j,i)
        cv(j,i) = covar(j,i)
10  continue

temp = F01_G_MM (d, cv, 8, 8, 8, ifail)
cv = F01_G_MM (temp, dt, 8, 8, 8, ifail)

do 20 i=1, 8
    do 20 j=1, 8
        cvar(j,i) = cv(j,i)
20  continue

return
end

```

subroutine bit_alloc (cvar, bit_all, norm, btrate)

```

real btrate
real cvar(8,8)
integer bit_all(8,8), norm(8,8)
real*8 vprod

vprod = 1.0
do 10 i=1, 8
    do 10 j=1, 8
        vprod = vprod * (cvar(j,i)**0.25)
10  continue
vprod = vprod**(1.0/16.0)

do 20 i=1, 8
    do 20 j=1,8

```

```

        temp = log(cvar(j,i)/vprod) / log(2.0)
        bit_all(j,i) = btrate + 0.5 * temp + 0.5
        if (bit_all(j,i) .lt. 0) bit_all(j,i) = 0
20  continue
    vmax = 0.0
    do 30 i=1, 8
        do 30 j=1, 8
            if (bit_all(j,i) .gt. 1) go to 30
            if(cvar(j,i) .gt. vmax) vmax = cvar(j,i)
30  continue
    maxsd = sqrt(vmax)

    do 40 i=1, 8
        do 40 j=1, 8
            if (bit_all(j,i) .eq. 0) then
                norm(j,i) = 0
            else
c      norm(j,i) = maxsd * 2**(bit_all(j,i)-1)
                norm(j,i) = maxsd * bit_all(j,i)
            endif
40  continue
    bit_all(1,1) = 8
    norm(1,1) = 8

    return
end

```

```

    subroutine adapt(dctblk, mask2)
#include compress.h
    real dctblk(.)
    logical mask2(.)
    logical mask1(.)
    real temp(.)
    mask2 = .true.
    do 13 i=1, 64
        mask1 = maxp(dctblk, mask2)
        mask2 = mask2 .and. (.not. mask1)
13  continue
    mask2 = .not. mask2
    do 35 i=1, sxx
        do 35 j=1, syy

```

```
        temp = dctbuf(.,j,i)
        temp(.not. mask2) = 0
        dctvar(j,i) = sum (temp*temp)/64.0
35  continue
    return
end
```

REFERENCES

- [1] Davisson, L. D. and Gray, R. M., *Data Compression*, Dowden, Hutchinson and Ross, Pennsylvania, 1976.
- [2] Jain, A. K., "Image Data Compression: A Review", Proceedings of the IEEE, vol. 69, No. 3, March 1981.
- [3] Netravali, A. N. and Limb, J. O., "Picture Coding: A Review", Proceedings of the IEEE, vol. 68, No. 3, March 1980.
- [4] Jayant, N. S. and Noll, P., *Digital Coding of Waveforms - Principles and Applications to Speech and Video*, Prentice Hall, New Jersey, 1984.
- [5] Moik, J. G., *Digital Processing of Remotely Sensed Images*, NASA SP-431, Washington, 1980.
- [6] Moorhead, R. J. and Rajala, S. A., "A Survey of Video Teleconferencing Algorithms", North Carolina State University Report CCSP-WP-84/11, March 1984.
- [7] Ng, S. B. and Schiff, L., "Two-Tier DPCM for Video Conferencing", IEEE Transactions on Communications, vol. 37, No. 4, April 1989.
- [8] Azadegan, F., "Rate-Distortion Performance of Image Compression Schemes", Ph.D. Dissertation, Rensselaer Polytechnic Institute, Troy, New York, December 1988.
- [9] Rabbani, M. and Jones, P. W., *Digital Image Compression Techniques*, vol. TT7, SPIE Optical Engineering Press, Washington, 1991.
- [10] Clarke, R. J., *Transform Coding of Images*, Academic Press, London, 1985.
- [11] Jain A. K., *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliff, New Jersey, 1989.
- [12] Gonzalez, R. C. and Wintz, P., *Digital Image Processing*, Addison Wesley, Massachusetts, 1988.

- [13] Hall, E. L., *Computer Image Processing and Recognition*, Academic Press, New York, New York, 1979.
- [14] Cress, D. H. and Smith, P. A., "Airborne Active / Passive Scanner For Surface Minefield Detection (U)", *Proceedings of the IRIS Specialty Group on Active Systems*, Naval Postgraduate School, Monterey, Calif., November 1985.
- [15] Cespedes, E. R. and Cress, D. H., " Analysis of Passive Imaging Concepts For Remote Minefield Detection Applications (U)", *Proceedings of the 1986 Army Science Conference*, US Military Academy, Westpoint, N. Y.
- [16] Cress, D. H., Cespedes, E. R., and Ginsberg, I. W., "Active/ Passive Airborne Scanner: Development And Processing For Standoff Surface Minefield Detection (U)", *Proceedings of the IRIS Specialty Group On Active Systems*, Orlando, Fla., October 1987.
- [17] Cespedes, E. R., Goodson, R. A. and Ginsberg, I. W., "Multisensor Image Processing Techniques For Real-Time Standoff Minefield Detection", *Proceedings of the First National Sensor Fusion Symposium*, Orlando, Fla., April 1988.
- [18] Hansen, G. M., Cress, D. H., Smith, P. A., More, K. A., and Stanich, C. G., "Development of a Multisensor Airborne Scanner For Remote Minefield Detection," *Proceedings of the First National Sensor Fusion Symposium*, Orlando, Fla., April 1988.
- [19] Goodson, R. A., Cress, D. H., and Cespedes, E. R. "Application of Expert System Concepts To Remote Detection of Surface Minefields (U)", Technical Report EL-89-3, US Army Engineer Waterways Experiment Station, Vicksburg, Miss., May 1989.
- [20] Candy, J.V., *Signal Processing: The Model-Based Approach*, McGraw-Hill, New York, New York, 1986.
- [21] Rogne, T. J., Smith, F. G., and Rice, J. E., "Passive Target Detection Using Polarized Components of Infrared Signatures", OptiMetrics Inc. Report, 1990.
- [22] Press, W. H. and Teukolsky, S. A., "Kolmogorov-Smirnov Test for Two-Dimensional Data", *Computers in Physics*, August 1988.

- [23] Rose, K., Heiman, A., and Dinstein, I., "DCT/DST Alternate Transform Image Coding", IEEE Transactions on Communications, vol. 38, No. 1, January 1990.
- [24] Rost, M. C., "Data Compression Using Adaptive Transform Coding", Ph.D. Dissertation, University of Nebraska, Lincoln, Nebraska, 1988.
- [25] Jain, A. K., "A Fast Karhunen-Loeve Transform for a Class of Random Processes", IEEE Transactions on Communications, September 1976.
- [26] Chen, W., Smith, C. H., and Fralick, S. C. "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Transactions on Communications, vol. COM-25, No. 9, September 1977
- [27] Press, W. H., et al, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, Massachusetts, 1988.
- [28] Hunt, B. R. "Nonstationary Image Models (And Their Application to Image Data Compression)", *Image Modeling*, Academic Press, New York, 1981.
- [29] Shannon, C. E., "Coding Theorems for a Discrete Source with a Fidelity Criterion", *IRE National Convention Record*, 1959
- [30] Berger, T., *Rate Distortion Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 1971
- [31] Gray, R. M., *Source Coding Theory*, Kluwer Academic Publishers, Massachusetts, 1990
- [32] Witten, I. H., Neal, R. M., and Cleary, J. G., "Arithmetic Coding for Data Compression", Communications of the ACM, vol. 30, No. 6, June 1987.
- [33] Moffatt, A., "Linear Time Adaptive Coding", IEEE Transactions on Information Theory, vol. 36, No. 2, March 1990.
- [34] Langdon, C. G., "An Introduction to Arithmetic Coding", IBM Journal Res. Dev. vol 28, No. 2, March 1984.
- [35] Sayood, K., and Schekal, S. M, "Use of ARMA Predictors in the Differential Encoding of Images", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 36, No. 11, November 1988.

- [36] Reeve, H. C., and Lim, J. S., "Reduction of Blocking Effects in Image Coding", *Optical Engineering*, vol. 23, Jan 1984.
- [37] Dinstein, I., Rose, K., and Heiman, A., "Variable Block-Size Transform Image Coder", *IEEE Transactions on Communications*, vol. 38, No. 11, November 1990.
- [38] Ahmed, N., Natarajan, T., And Rao, K. R., "Discrete Cosine Transform", *IEEE Transactions on Computers*, January 1974.
- [39] Hinman, B. L., Bernstein, J. G., and Staelin, D. H., "Short Space Fourier Transform Image Processing", *Proceedings of ICASSP 84*, San Diego, California, 1984.
- [40] Malvar, H. S. and Staelin, D. H., "The LOT: Transform Coding Without Blocking Effects", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, No.4, April, 1989.
- [41] Nasrabadi, N. M. and King, R. A., "Image Coding Using Vector Quantization: A Review", *IEEE Transactions on Communications*, COM-36, 1988.
- [42] Knotts, M. E. and O'Donnell, K. A., "Anomalous Light Scattering from a Perturbed Grating", *Optics Letters*, vol.15, No. 24, December 1990.
- [43] Mendez, E. R. and O'Donnell, K. A., "Observation of Depolarization and Backscattering Enhancement in Light Scattering from Gaussian Random Surfaces", *Optics Communications*, vol. 61, No. 2, January 1987.
- [44] O'Donnell, K. A. and Mendez, E. R., "Experimental Study of Scattering from Characterized Random Surfaces", *Journal of the Optical Society of America*, vol. 4, No. 7, July 1987.
- [45] Wolff, L. B., "Classification of Material Surfaces from the Polarization of Specular Highlights", *Proceedings of the Conference on Optics, Illumination, and Image Sensing for Machine Vision 2*, SPIE, Cambridge, Massachusetts, 1988.
- [46] Leader, C. J., "Analysis and Prediction of Laser Scattering from Rough-Surface Materials", *Journal of the Optical Society of America*, vol. 69,

No. 4, April 1979.

- [47] Pattanayak, D. N., and Wolf, E., "General Form and a New Interpretation of the Ewald-Oseen Extinction Theorem", Optics Communications, No. 6, 1972.
- [48] Ballard, J., Miles, B.H., Castellane, R.M., and Cress, D.H., "Reflectance and Polarization Study Conducted During Standoff Detection Technical Demonstration Fort Hunter Liggett, California". Technical Report EL-91-, U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS, March 1991.
- [49] Hou, H.S., "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, 1987
- [50] Lee, B.G., "A New Algorithm to Compute the Discrete Cosine Transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol 32, December 1984..
- [51] Wu, H.R., and Paoloni, F.J., "A Two-Dimensional Fast Cosine Transform Algorithm", IEEE Transactions on Signal Processing. Vol. 39, No. 2, 1991.
- [52] Pearlman, W.A., "Adaptive Cosine Transform Image Coding with Constant Block Distortion", IEEE Transactions on Communications, vol. 38, No. 5, May 1990.
- [53] "Parallel Data Transforms", Active Memory Technology Corporation Technical Publication man022.02, November 1988.
- [54] Mauersberger, W., "Generalised Correlation Model for Designing 2-Dimensional Image Coders", Electronics Letters, vol 15, No. 20, May 1990.
- [55] Akansu, A.N., and Haddad, R.A., "Factorization of the Coefficient Variance Matrix in Orthogonal Transforms", IEEE Transactions on Signal Processing, vol. 39, No. 3, March 1991.
- [56] Rioul, O., and Vetterli, M., "Wavelets and Signal Processing", IEEE Signal Processing Magazine, October 1991.

- [57] Mallat, S., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, No. 7, July 1989.
- [58] "FORTRAN-PLUS Enhanced", Active Memory Technology Corporation Technical Publication man102.01, October 1990.
- [59] Natarajan, T.R., and Ahmed, N., "Performance Evaluation of Transform Coding Using a Nonseparable Covariance Model", IEEE Transactions on Communications, vol. 26, 1978.
- [60] Powell, M. J., "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives", Computer Journal, vol. 7, 1964.
- [61] Haddad, R.A., and Akansu, A.N., "On Asymmetrical Performance of the Discrete Cosine Transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, January 1990
- [62] Segall, A., "Bit Allocation and Encoding for Vector Sources", IEEE Transactions on Information Theory, vol. 22, No. 2, March 1976.
- [63] Chen, W.H., and Smith, C.H., "Adaptive Coding of Monochrome and Color Images", IEEE Transactions on Communications, vol. 25, No. 11, November 1977.
- [64] Mauersberger, W., "Experimental Results on the Performance of Mismatched Quantizers", IEEE Transactions on Information Theory, vol. 25, No. 4, July 1979.
- [65] Reininger, R.C., and Gibson, J. D., "Distributions of the Two-Dimensional DCT Coefficients for Images", IEEE Transactions on Communications, vol. 31, No. 6, June 1983.
- [66] Pratt, W.K., *Digital Image Processing*, Wiley-Interscience, New York, New York, 1978.
- [67] Schaming, W.B., and Bessette, O.E., "Empirical Determination of Processing Parameters for a Real Time Two-Dimensional Discrete Cosine Transform Coding (2D-DCT) Video Bandwidth Compression System", *Advances in Image Transmission II*, SPIE vol. 249, 1980.