

N73-25910

UAH Research Report No. 141

May 1973

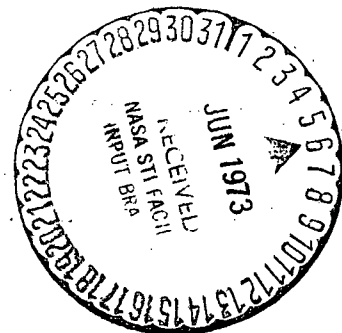
CASE FILE COPY

IMAGE PROCESSING TECHNIQUES AND APPLICATIONS TO THE EARTH RESOURCES TECHNOLOGY SATELLITE PROGRAM

by

R. J. Polge
B. K. Bhagavan
L. Callas

Final Technical Report



This research work was supported by
National Aeronautics and Space Administration
Under Contract NAS8-28545

The University of Alabama in Huntsville
Huntsville, Alabama

IMAGE PROCESSING TECHNIQUES AND
APPLICATIONS TO THE
EARTH RESOURCES TECHNOLOGY SATELLITE PROGRAM

by

R. J. Polge
B. K. Bhagavan
L. Callas

Final Technical Report

This research work was supported by
National Aeronautics and Space Administration
Under Contract NAS8-28545

The University of Alabama in Huntsville
Huntsville, Alabama

ABSTRACT

The purposes of this report are: (1) to study the Earth Resources Technology Satellite system with emphasis on sensors and data processing requirements, and (2) to evaluate image data compression using the Fast Fourier and Hadamard transforms.

The ERTS-A system and the fundamentals of remote sensing are discussed. Three user applications (forestry, crops, and rangelands) are selected and their spectral signatures are described. It is shown that additional sensors are needed for rangeland management. An on-board information processing system is recommended to reduce the amount of data transmitted.

Image data compression can be performed using transform techniques either by truncation or by coarse quantization. These two compression schemes were implemented on the digital computer using both the Fast Fourier and Hadamard transforms. Experiments were performed on three digitized images provided by MSFC to study distortion versus compression. Significant data compression can be obtained with these methods before distortion becomes noticeable.

PREFACE

The work performed under Contract NAS8-28545 consists of two tasks. The first task, Earth Resources Technology Satellite (ERTS) applications, is documented in Part I of this report, and the second task, Transform Techniques for Data Compression, is documented in Part II.

The main objective of the first task is to study data processing techniques for three selected user applications of ERTS-A and to recommend remote sensors and information processing for each application.

The main objective of the second task is to study data compression versus picture distortion, using Fourier and Hadamard transformations.

Part III of this report contains suggestions for future efforts.

Robert J. Polge

Huntsville, Alabama

May 1973

TABLE OF CONTENTS

	Page
Abstract	ii
Preface	iii
List of Illustrations	vii

PART I

EARTH RESOURCES TECHNOLOGY SATELLITE APPLICATION

CHAPTER 1	INTRODUCTION	1
CHAPTER 2	EARTH RESOURCES TECHNOLOGY SATELLITE	3
	2.1 Introduction	3
	2.2 Block Diagram for the ERTS-A System	3
	2.3 On-board System	3
	2.4 Ground System	6
	2.5 Conclusions	6
CHAPTER 3	REMOTE SENSING WITH ERTS-A	9
	3.1 Introduction	9
	3.2 Remote Sensing	9
CHAPTER 4	SELECTION OF THREE APPLICATIONS	13
	4.1 Applications of the ERTS-A	13
	4.2 Selection of User Applications	14
	4.3 Forestry (Increase of Yield).....	15
	4.4 Crops (Reduction of Losses).....	18
	4.5 Rangelands (Effective Management)	20
CHAPTER 5	DATA PROCESSING	23
	5.1 Introduction	23
	5.2 Potential Information Processing Techniques	23

5.3	Cataloging and Display	26
5.4	Recommended Changes on ERTS-A	27
CHAPTER 6	SUMMARY AND CONCLUSIONS.....	31
6.1	Summary	31
6.2	Conclusions	31
LIST OF REFERENCES		33

PART II

TRANSFORM TECHNIQUES FOR DATA COMPRESSION

CHAPTER 1	INTRODUCTION	35
CHAPTER 2	IMAGE TRANSFORMS	38
2.1	Introduction	38
2.2	Two-Dimensional Transforms	38
2.3	Discrete and Fast Fourier Transforms	39
2.3.1	Definition and Implementation	39
2.3.2	Fast Fourier Transform of Image Data	41
2.4	The Hadamard Transform	42
2.5	Computer Programs	44
CHAPTER 3	TRANSFORMATIONAL COMPRESSION OF IMAGE DATA	45
3.1	Introduction	45
3.2	The General Model	47
3.3	Quantization	47
3.4	Compression and Coding	51
3.4.1	Truncation	52
3.4.2	Coarse Quantization	54
3.5	Immunity to Noise	58
3.6	Summary	58

CHAPTER 4	COMPUTER IMPLEMENTATION OF THE EXPERIMENTS	59
4.1	Introduction	59
4.2	Hadamard Transform Compression	59
4.2.1	Transformation	59
4.2.2	Compression by Truncation	60
4.2.3	Compression by Coarse Quantization	60
4.2.4	Inverse Transformation	61
4.3	Fourier Transform Compression	61
4.3.1	Transformation	61
4.3.2	Compression by Truncation	62
4.3.3	Compression by Coarse Quantization	62
4.3.4	Inverse Transformation	63
4.4	Subroutines XREAD and XWRITE	63
CHAPTER 5	EXPERIMENTAL RESULTS	64
5.1	Introduction	64
5.2	Presentation of Experimental Results	65
5.3	Discussion of the Results	65
CHAPTER 6	SUMMARY AND CONCLUSIONS	76
6.1	Summary	76
6.2	Conclusions	76
LIST OF REFERENCES		78
APPENDIX A	The Fast Fourier Transform	79
APPENDIX B	The Hadamard Transform	83
APPENDIX C	Listings of Computer Programs	94

PART III

SUGGESTIONS FOR FUTURE EFFORTS	112
--------------------------------------	-----

LIST OF ILLUSTRATIONS

PART I

Figure		Page
2.1	ERTS-A System	4
2.2	ERTS-A On-Board System	5
2.3	ERTS-A Ground System	7
3.1	Spectral Reflectance from Corn Leaves	11
4.1	Spectral Reflectance from Pine Foliage	17
4.2	Spectral Reflectance from Cotton Leaves	19
5.1	Proposed ERTS On-Board Information Processing	28
5.2	ERTS-A Data Flow for Selected Applications	30

PART II

3.1	Model of an Image Transmission System	48
3.2	Linear Quantization Scheme	50
3.3	Nonlinear Quantization Scheme	50
4.1	The Simplified Experiment	59
5.1	The Test Pictures	66
5.2	Truncation with Hadamard Transform, CR = 2.4	67
5.3	Truncation with Hadamard Transform, CR = 6.5	68
5.4	Truncation with FFT, CR = 2.4	69
5.5	Truncation with FFT, CR = 6.5	70
5.6	Coarse Quantization with Hadamard Transform, CR = 2.4	71
5.7	Coarse Quantization with Hadamard Transform, CR = 3.5	72
5.8	Coarse Quantization with FFT, CR = 2.4	73
5.9	Coarse Quantization with FFT, CR = 3.5	74
B.1	Radamacher Functions	84
B.2	Walsh Functions	87
B.3	Hadamard Matrix of Order 8(Sequency Ordered)	89
B.4	Hadamard Matrix of Order 8(Generated Using Matrix Kronecker Products)	89

PART 1

EARTH RESOURCES TECHNOLOGY SATELLITE APPLICATIONS

CHAPTER 1

INTRODUCTION

The Earth Resources Technology Satellite (ERTS) is a satellite that is placed in a high orbit above the earth. It contains remote sensors that are used to detect objects and their change of state on the earth's surface or subsurface. The purpose of the ERTS program [1] is to demonstrate that remote sensing from space is a feasible and practical approach to efficient management of earth's resources. The remote sensors have been selected to cover a wide range of applications. In addition to the development of sensors and the satellites, there are significant technical problems in the areas of data transmission and data handling and analysis, where the prodigious quantities of data to be generated challenge man's ability to manage and digest them. For example, with its 496 nautical mile high inclination, sun-synchronous orbit and three high resolution, return-beam vidicon television cameras, it is estimated that approximately 250,000 pictures (85,000 sets of three) will be taken annually by the ERTS over the U. S., Canada, and selected international and foreign areas. The number of spectral channels will probably increase and the resolution decrease in future ERTS's. Thus, the data output will grow. Clearly, techniques which can reduce the data transmission rates are required in order to use efficiently the predicted increase in sensor capability.

This part documents the work done under Task I of Contract NAS8-28545. The goals of Part I of this report are to: (1) conduct a literature search and

define data collection and processing of the Earth Resources Technology Satellite (ERTS), (2) select three user applications for a detailed study, (3) study data processing techniques for the selected applications, and (4) recommend sensors, on-board processing and ground processing for each application.

Chapter 2 describes the ERTS system by tracing the signal flow through the on-board and ground systems. It is noted that no on-board information processing takes place. Chapter 3 discusses the fundamentals of remote sensing and points out that the remote sensors of ERTS-A cover the spectral bands of reflectance and radiance for many objects. Chapter 4 selects three user applications: (1) forestry (increase of yield), (2) crops (reduction of losses) and (3) rangelands (effective management). Spectral signatures are described for each application. In order to sense rangelands effectively, additional sensors are recommended. Chapter 5 discusses signal processing by ERTS-A. An on-board and ground information processing system is suggested. This system uses the Hadamard transform and delta modulation to reduce the amount of data transmitted without an appreciable loss of information.

CHAPTER 2

EARTH RESOURCES TECHNOLOGY SATELLITE

2.1 Introduction

The Earth Resources Technology Satellite (ERTS) is used to gather information remotely on the earth's surface or subsurface [1]. This information can be analyzed and interpreted to extract specific information for different scientific disciplines such as agriculture, forestry, geology, hydrology, oceanography or other disciplines concerned with man's environment. The first satellite, launched in 1972, is called ERTS-A and is described below.

2.2 Block Diagram for the ERTS-A System

A block diagram of the ERTS-A System is shown in Figure 2.1. The ERTS-A system can be divided into two systems: the on-board system and the ground system.

2.3 On-board System

Figure 2.2 is a simplified block diagram of the ERTS-A on-board system.

It is seen that ground radiance is the input to the two remote sensors: Return Beam Vidicon (RBV) Cameras and Multispectral Scanner (MSS). The RBV sensor consists of three cameras which sense spectral bands $.475 - .575 \mu$, $.580 - .680 \mu$, and $.698 - .830 \mu$, respectively. The MSS sensor is an optical scanner which detects ground radiance in the spectral bands $.5 - .6 \mu$, $.6 - .7 \mu$, $.7 - .8 \mu$, and $.8 - 1.1 \mu$. The outputs of the sensors, 3.2 mhz analog video for the RBV and 15 mbps digital data for the MSS, are sent directly to the wideband telemetry system for transmission to the ground or to the wideband video tape recorders for delayed transmission.

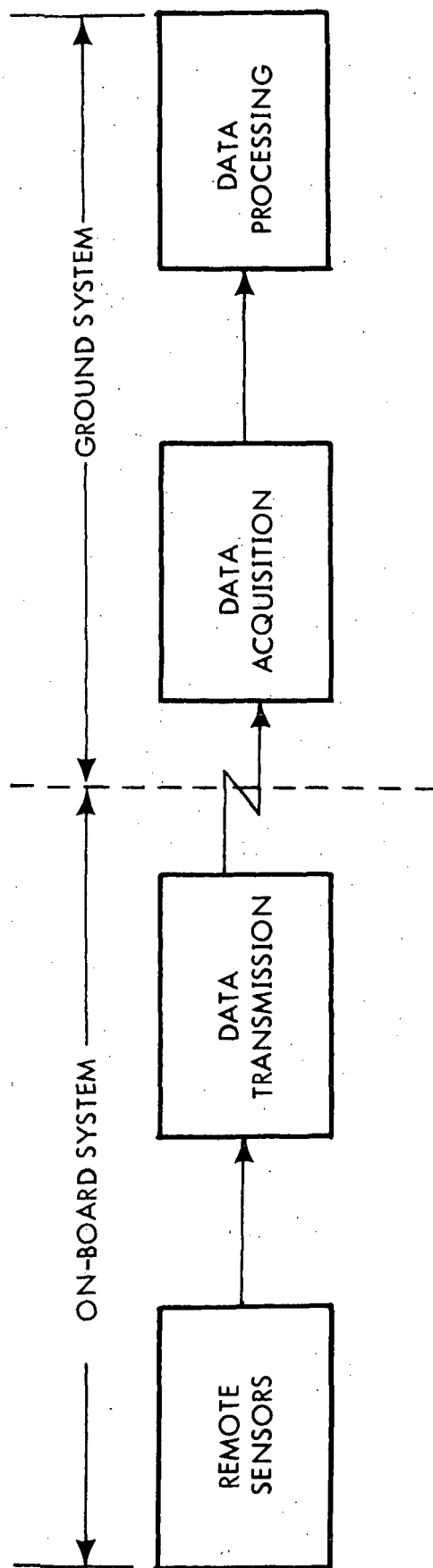


Figure 2.1 ERTS-A SYSTEM

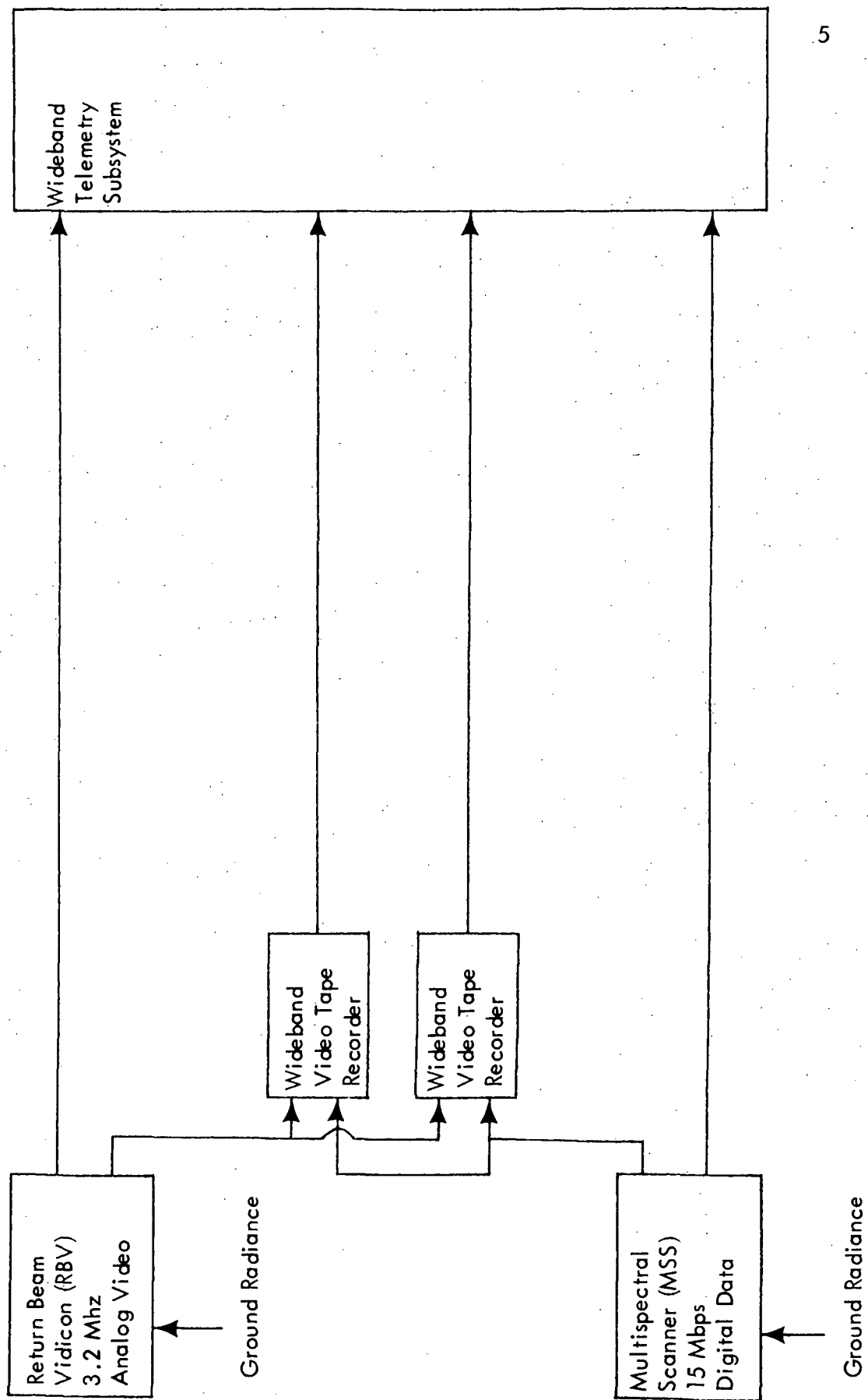


Figure 2.2 ERTS-A On-Board System

to the ground. A Data Collection System (DCS) is also part of the on-board system. It obtains data from remote, automatic data collection platforms and relays the data to ground stations.

2.4 Ground System

A block diagram of the ERTS-A ground system is depicted by Figure 2.3.

Wideband payload video data are received by space tracking and data network stations in Fairbanks, Alaska; Goldstone, California; and Greenbelt, Maryland.

The video data, in the form of magnetic tapes, are sent to the Operations Control Center (OCC) which controls the ERTS-A mission operations. The RBV and MSS video tapes are then sent to the NASA Data Processing Facility (NDPF). Three types of processing are done at the NDPF: (1) bulk processing to convert the video tapes to corrected 55 mm images on 70 mm film, (2) precision processing which uses the 70 mm film images to produce error-corrected film images on a 9 1/2 inch format, and (3) special processing which outputs data from the bulk or precision processing on computer compatible digital tapes for distribution. The NDPF output data products are disseminated to the user by work orders which are generated to match investigator requests for received data. A storage and retrieval system aids the investigator by making available microfilm browse files, listings of images returned, and listings of information available from the data collection system.

2.5 Conclusions

ERTS-A is well designed to sense remotely the earth spectral reflectance and radiance and to transmit images of ground scenes back to the earth for processing.

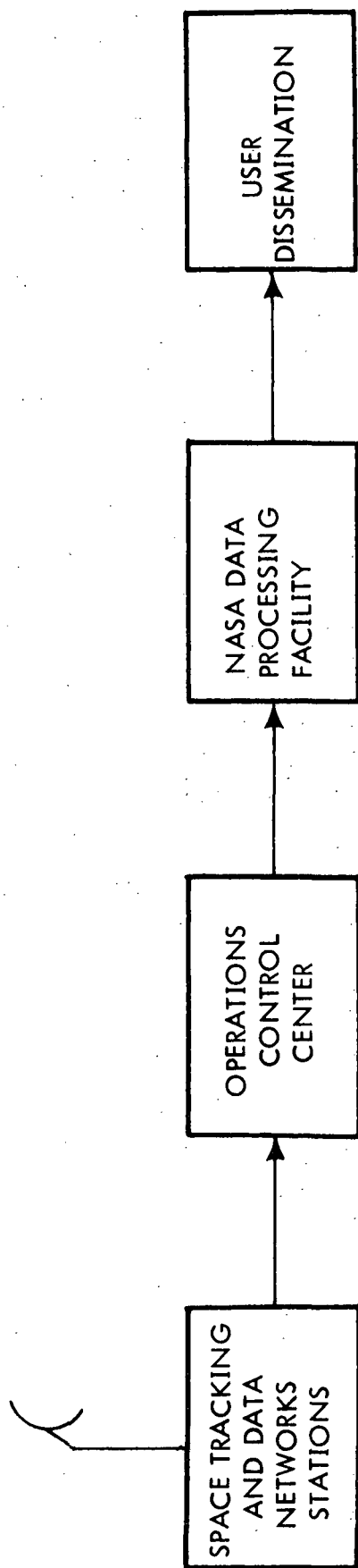


Figure 2.3 ERTS-A Ground System

However, the number of bits necessary to transmit a single image is enormous.

For example, PCM requires KN^2 bits/image where $N \times N$ is an array of picture elements encoded by a sequence of N^2 K -bit code words that specifies the gray levels of the image [2]. If $K = 8$ and $N = 256$, then $KN^2 = 524,288$ bits/image. In the ERTS-A only the signal is processed, not the information.

Clearly, it will be advantageous to compress the information before transmission. In this manner the number of bits transmitted will be reduced without an appreciable loss of information. One of the goals of this report is to investigate on-board and ground processing for a more efficient transmission.

CHAPTER 3

REMOTE SENSING WITH ERTS-A

3.1 Introduction

Remote sensing can be defined as the detection of some property of an object with a sensor that is located at a distance from the object. In the case of ERTS-A, the sensors are the return beam vidicon cameras and the multispectral scanner. The object of interest is the earth and its resources. The detection of a property of a particular object is based on the electromagnetic energy that it radiates or reflects. The fundamentals of remote sensing are discussed next.

3.2 Remote Sensing

The sun is the primary source of electromagnetic energy and emits photons (smallest units) of electromagnetic energy. This energy is of an oscillating nature and can be described in terms of wavelength (distance between peaks), velocity (speed of the peaks) and frequency (number of peaks passing a point in a given time) [3]. When the photons hit an object, one or more of the following interactions [4] between the photons and object can take place: the photons can be transmitted through, reflected by, absorbed by, emitted from, or scattered by the object.

The degree to which the above interactions between photons and object take place depends on the wavelength and is specific for each different object. Thus, it seems feasible for an object to be identified by remote sensing of one or more of the interactions given above. In practice only two of the interactions (reflectance

and emittance) are used in remote sensing. The electromagnetic spectrum that is most useful for remote sensing extends from wavelengths of .4 to 15 micrometers (microns μ). It has been shown [5] that "Solar reflectant power is decreased with increasing wavelengths until the radiation emitted by the object is dominant. The crossover point where the emitted radiation becomes dominant over the reflected radiation is at approximately 3μ ."

Figure 3.1 is an example of spectral reflectance of solar radiation from corn leaves [6]. Percent reflectance (measured by a spectrophotometer) is plotted as a function of wavelength in microns. The following observations are made. Below about $.4 \mu$ the reflectance is low due to atmospheric attenuation. Between $.4$ and 3μ , the reflectance is good. However, the band from 1.3 to 3μ is eliminated because of water absorption. Therefore, the band $.4$ to 1.3μ can be used effectively for sensing the reflectance of corn leaves [7]. The sensors of ERTS-A have been designed to cover most of this band. Comparisons of the spectral signatures for numerous resources reveal that all of the reflectances are similar but yet each reflectance is unique. It is this uniqueness of each spectral signature that implies the possibility of identifying a remote object by measuring its reflectance.

Although an object can be detected with a single wavelength band [8], identification of the object requires several wavelength bands. The term multi-spectral sensing is used to mean sensing an object within several different wavelength bands. Two objects may have the same reflectance in one band but will

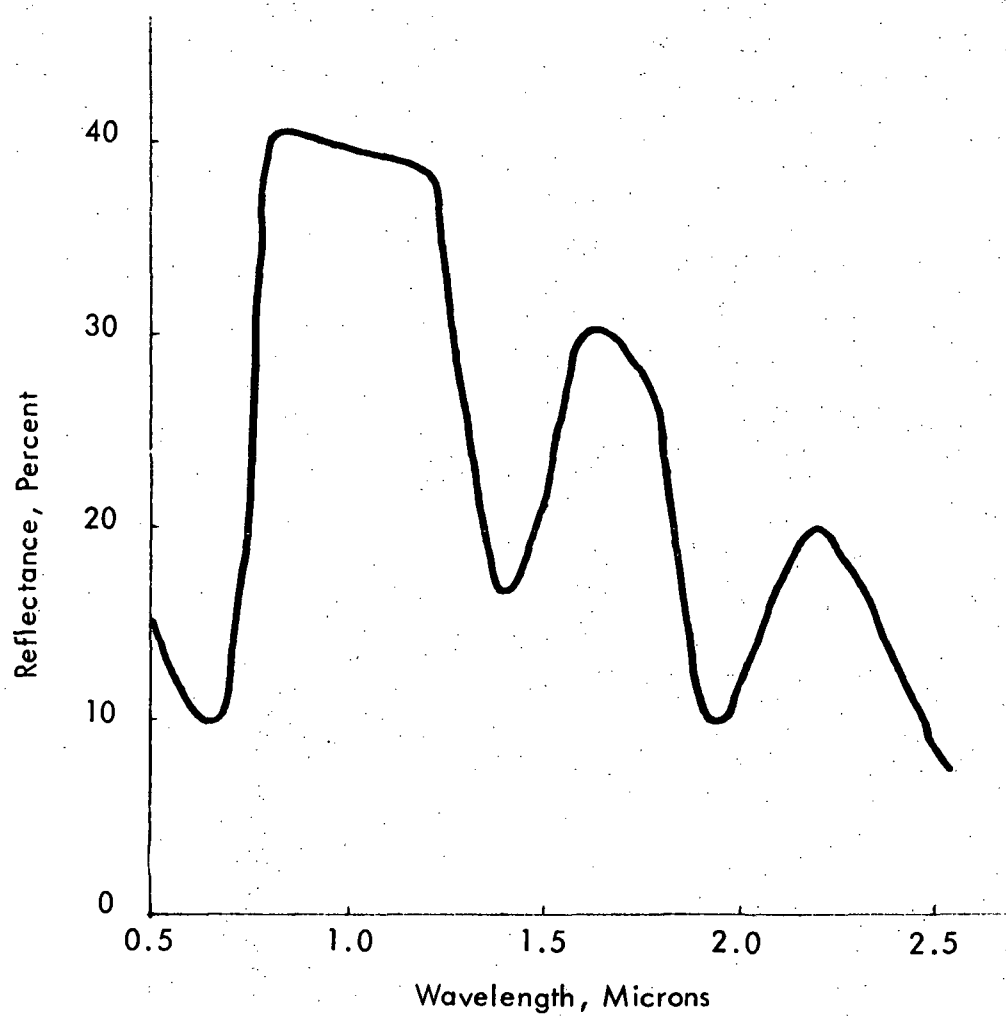


Figure 3.1 Spectral Reflectance from Corn Leaves

have different reflectances in another band. Thus, comparison of the reflectances for several bands enables one object to be distinguished from the other [9].

Measured values of reflectance within the wavelength bands listed above represent the output of the remote sensors of ERTS-A. Thus, the remote sensors of ERTS-A have been optimally selected to cover the region of the electromagnetic spectrum (.4 to 1.3 μ) where the interaction of reflectance occurs between the photons and the objects being detected.

CHAPTER 4

SELECTION OF THREE APPLICATIONS

4.1 Applications of the ERTS-A

It was pointed out in Chapter 2 that ERTS-A can be used to gather information remotely on the earth's surface or subsurface for applications such as agriculture, forestry, geology, hydrology, and oceanography. Other applications include terrain mapping, water pollution, air pollution, climatology, meteorology, zoology, sociology, economics and many others. Chapter 3 provided the fundamentals of remote sensing. It was shown that objects on the earth's surface radiate and reflect electromagnetic energy. Each object reflects and radiates energy at unique spectral bandwidths. And thus, the spectral signature of each object can be used to detect remotely its presence or change of state.

The ability of ERTS-A to detect the presence or change of state of objects depends on several factors. Some of these factors are (1) the distance from the sensor to the object, (2) resolving capability of the sensors, (3) condition of the atmosphere, (4) concealment of the object to be detected, and (5) reflectivity and/or emissivity of the object.

Therefore, the selection of applications should take into account both the user needs and the capability of the ERTS-A system. For example, it is difficult to gather information under the water surface because of the reflectivity of this surface.

4.2 Selection of User Applications

In each of the applications listed above there is a need for investigating some property. For example, in agriculture early detection of the areas of deterioration of crops due to disease and pestilence can be used to implement controls that will check further spread of the damaging agents to other areas. ERTS-A can be used to locate the problem areas and provide advance warning to the user so that he can apply the proper measures of control.

By 1985 the use of timber is expected to increase to about twice as much as it is now [10]. Therefore, there is a need to increase the yield of forestry in order to satisfy future requirements.

Each year millions of dollars worth of crops are lost due to insects and disease. Reduction in the loss of these crops not only represents a monetary savings, but also makes available food and other products which would otherwise be lost. Thus, the need exists to reduce the losses of crops due to insects and disease.

Grass is the main food for cattle and sheep. Since these animals are the source of meat, wool, leather and milk, it becomes desirable to create conditions that are conducive to their growth and reproduction. Effective management of pastures or rangelands is an important factor in increasing the number of cattle and sheep which the rangelands are capable of supporting.

Therefore, a need is established for investigating the user applications forestry (increase of yield), crops (reduction of losses), and rangelands (effective management). In order to utilize ERTS-A for remote sensing of the above user applications, it is

necessary to know their spectral reflectance (signatures). These spectral signatures are obtained by measuring spectral reflectance of the desired trees, crops or grasses with a spectrophotometer or other similar instrument. Once the spectral signatures are obtained, then the remote sensor is defined in terms of its required spectral band.

Each of the above user applications is discussed in the following sections.

The spectral signature for each application is described, and comparisons are made with the ERTS-A remote sensors to determine the compatibility need with the ERTS-A potential.

4.3 Forestry (Increase of Yield)

There are several ways that the yield of forestry can be increased. One way is to determine the location of suitable new areas. This can be accomplished by studying the environment that exists where forestry thrives abundantly. Some of the characteristics of the environment that can be analyzed are the soil variables, correlation between plant types and forest productivity, and the amount of rainfall or other water supply throughout the year.

Another way to increase the yield of forestry is through sequential observations of the change of vigor. In the case of an increase in vigor, advantage can be taken of noting those factors that are conducive to accelerated growth and attempting to recreate similar conditions in other areas. If a decrease in vigor is detected, then the possible causes can be investigated more closely through the use of large scale photography and ground observations. This method is known as multistage sensing. Generally, multistage sensing consists of sensing objects in several decreasing stages

of remoteness. For example, the first stage of sensing can be from the ERTS-A platform where large areas are contained in each image. Analysis of the resulting images shows the sections where a decline in tree vigor has occurred. More important, the analysis eliminates many areas where further sensing is not necessary. The next stage of sensing can be from a high altitude airplane. This time only those areas that were indicated in the first stage are sensed. The large scale imagery of the airplane further defines the exact location of the declining vigor trees. Following this, other stages such as low altitude flights and ground observations can be used to locate exactly the points of distress. Once the coordinates have been pinpointed, it may be possible to take certain measures to correct the problem of declining vigor.

A third method to increase the yield of forestry is by making timber yield estimates and predictions. The use of ERTS-A enables stratification or sectioning of the timber stands. The multistage sensing technique described above can be combined with variable probability sampling theory to reduce the sampling error by as much as 60 percent [11] .

Other methods to increase yield include locating the best areas in the forest for harvesting the trees and determining the area of timber for inventory purposes. Both of these methods can use multistage sensing techniques for optimum results.

Figure 4.1 shows the spectral signature for pine foliage [12]. Variations in reflectance as a function of wavelengths are evident in the region $.4 - .7 \mu$. Since this band is within the range of ERTS-A sensors, then ERTS-A can be used to detect changes in reflectance of pine trees due to increase or decrease of vigor.

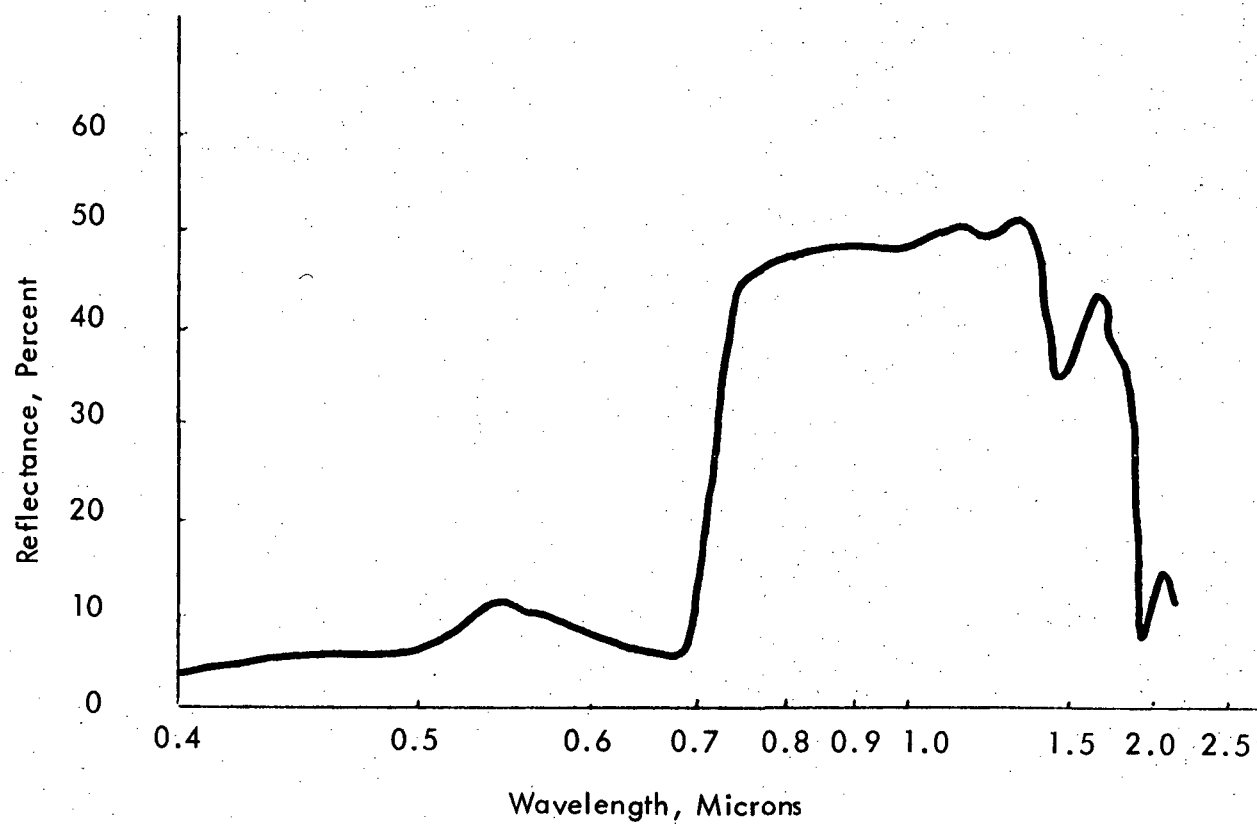


Figure 4.1 Spectral Reflectance from Pine Foliage

4.4 Crops (Reduction of Losses)

An important application of ERTS-A is to predict losses due to disease and insect infestation of crops so that corrective measures can be taken. The technique described below uses multispectral and multitemporal sensing to compare the actual predicted yield to a typical predicted yield. If the actual predicted yield is much less than the typical predicted yield, corrective steps can be taken to increase the yield.

In order to reduce losses of crops, it is necessary to distinguish healthy plants from unhealthy ones. This distinction is based on remote multispectral sensing. As previously mentioned, ERTS-A utilizes three bands for the return beam vidicon and four bands for the multispectral scanner. If several images of the same scene are projected simultaneously on a screen and combined optically through a filter system, a color composite will be obtained [13]. As a specific example, cotton, one of the major crops of the Southern U. S., is often attacked by an insect known as the pink boll worm. The leaves of the cotton plant are damaged, and this affects the quality and quantity of the crop. Through the use of its multispectral scanner and return beam vidicon sensors, ERTS-A is able to view the cotton fields remotely. Figure 4.2 shows the spectral signature of cotton leaves [14]. This graph of reflectance versus wavelength has regions of interest within the band of the ERTS-A sensors.

Evaluation of the condition of the cotton crop involves multitemporal sensing. By making observations of the same area at different times, a crop calendar can be obtained showing the spectral signature of the cotton field throughout the growing season. The crop calendar can then be used to compare an image with a previous

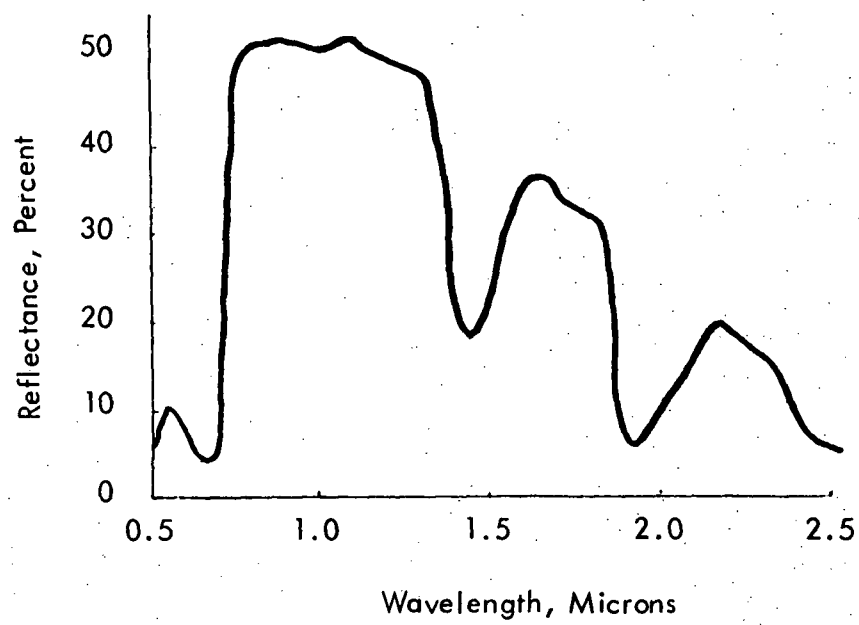


Figure 4.2. Spectral Reflectance from Cotton Leaves

image made under similar circumstances. Inferences can be made about the condition of the cotton by noting changes in the color. For example, the presence of certain diseases cause a change in the chlorophyll content of the leaves. This effect shows up as a darker tone or color on the color composite. Since the effects of certain diseases are most readily detected in the near infrared spectrum (.7 to .9 μ), the sensors should include this band. The multispectral scanner of ERTS-A includes this bandwidth, but the return beam vidicon does not include it all (up to .83 μ).

4.5 Rangelands (Effective Management)

The third application of ERTS-A to be discussed next is rangeland management. A considerable portion of the earth's surface consists of grasses and plants that are suitable for grazing cows, sheep and other herbivorous animals. Since the growing world population places an increasing demand on milk and meat from these animals, it is important that the rangelands be managed effectively. The term effective management considers the following items: (1) prevention of overgrazing, (2) prevention of undergrazing, (3) improvements of pastures, and (4) extension of present acreage.

The remote location of ERTS-A sensors enables the viewing of large areas (approximately 100 x 100 miles) of the earth's surface. The multitemporal imaging techniques described above facilitate the formulation of decisions about effective management of rangelands. For example, an indication of overgrazing is a decrease in the predicted yield of grasses or plants and shrubs associated with good rangelands. This effect is observed through a change in tone of the multispectral image. Some

of the results of overgrazing are a reduction of the moisture retention capabilities of the ground and increased erosion due to water run-off during rainy seasons. It is therefore important to be able to detect overgrazing and then apply effective measures such as proper fencing and efficient herding.

Similarly, it is advantageous to prevent undergrazing since a decrease in nutritive value of the grasses results. This is in part due to an increase in undesirable plants such as weeds. Spectral signatures for grasses are similar to those given previously for cotton.

Remote sensing can be used effectively to locate soil, water, temperature, and other factors conducive to rangeland growth. For example, it is known that certain types of plants or shrubs can be found in regions of flourishing rangelands. By transplanting healthy grasses to these regions, it is possible to extend the rangelands. ERTS-A can be utilized as a warning signal to detect signs of reduced yield. More specifically, the near infrared spectral band ($.7$ to $.9 \mu$) has been found to be most effective in indicating a decline in nutritive value of plants. Also, it has been observed [15] that moist range sites were detected on the $.32 - .38 \mu$ band, ponded spring water was detected on the $1.5 - 1.8 \mu$ band, and areas of moisture, open or concealed by vegetation, were most readily detected on the $8 - 14 \mu$ band.

Summarizing the three user applications: forestry (increase of yield), crops (reduction of losses) and rangelands (effective management), the following generalizations can be made. The ERTS sensors are adequate for detecting changes in vigor

of trees and plants. In order to extend the rangelands, it is desirable to sense the location of soils of high moisture content and bodies of water concealed by trees and undergrowth. For this purpose, sensors of $.32 - .38 \mu$, $1.5 - 1.8 \mu$, and $8 - 14 \mu$ are indicated.

The smallest area of the earth's surface that can be detected by ERTS-A is limited by the ground resolution distance of the sensors.

The ground resolution distance is inversely proportional to the height of a sensor above an object. Since it is desired to maintain a viewing angle of less than ± 10 degrees from the normal with a swath width of 100 nautical miles, the height of ERTS-A is fixed at about 496 nautical miles.

The ground resolution distances for the RBV and MSS are 126 meters and 224 meters, respectively, for high contrast objects [16]. However, this doesn't always mean that an object must be 126 meters to be resolved. Other factors such as the shape, background and decreased complexity of environment permit smaller objects to be discerned. Therefore, since the area of the minimum resolved ground element is large, it is recommended that the ERTS sensors be provided with variable focal length lenses. It is also recommended that ERTS be complemented with the multi-stage sensing techniques described above.

CHAPTER 5

DATA PROCESSING

5.1 Introduction

Chapter 2 described ERTS-A showing the different steps for sensing, transmitting, and processing an image. The ground scene is remotely sensed, transmitted to the ground and then processed. The bulk, precision and special processing convert the video tapes to film images and make radiometric and geometric corrections. It was pointed out that only signal processing takes place in the satellite. That is, there is no on-board information processing. The following sections describe some of the possibilities for on-board information processing. Cataloging and displays for the user are discussed. Recommendations are made for changes on ERTS-A to provide an on-board processing system.

5.2 Potential Information Processing Techniques

Various types of on-board information processing can be performed. Some of these types require more space than ERTS-A allows. Some possibilities have been suggested by W. E. Stoelzner [17]. For example, with trained personnel, on-board information processing and analysis could be performed. Then, only the important data would be transmitted to the ground thereby reducing the required bandwidth. The use of film and its tremendous storage capability could also reduce transmission requirements. Optical data processing would also be a definite possibility. Various other schemes could be enumerated, but will not be considered at this time.

In order to limit the scope of the investigations, the following assumption is made concerning the proposed satellite. Minimum configuration changes will be made to ERTS-A. Thus, the satellite will be unmanned, and on-board equipment is not retrievable during the mission. These restrictions impose a considerable limitation of the amount and types of on-board information processing that can be accomplished.

Since on-board equipment is not retrievable during the mission, the use of film as a storage device is not recommended. Therefore, although on-board optical data processing is feasible for some applications such as spectral analysis, complex spatial filtering, matched filtering and correlation [18], it would be difficult and uneconomical to implement on an unmanned satellite.

Another alternative for data processing (rather than optical) is the use of electronic data processing. Specifically, on-board digital information processing can be accomplished efficiently and economically up to a certain extent. The electronic data processing to be recommended falls under the category of data compression. Data compression is a method to decrease the time or bandwidth required to transmit a given amount of information. A general survey of the various schemes of data compression has been tabulated by C. M. Kortman [19]. He describes four general methods: parameter extraction, adaptive sampling, redundancy reduction and coding. Any usable data compression scheme must be information preserving. Therefore, the first method (parameter extraction) will not be considered since a loss of information is inherent with its use. The second method

(adaptive sampling) is not desirable since its complexity has so far prohibited any practical mechanization. The last two methods (redundancy reduction and coding) are definite possibilities for data compression on an unmanned satellite. These two methods are discussed next.

Redundancy reduction is a comparison technique to inhibit transmission of redundant samples. Thus, a saving of time or bandwidth may be obtained.

Redundancy reduction algorithms can be broadly categorized as predictors or interpolators. Predictors are used to estimate future data points within a specified tolerance knowing the past data points. The simplest predictor is the zero-order or straight line predictor which operates as follows. The first sample of a sequence of data is transmitted. Then as long as each succeeding sample has an amplitude within a specified tolerance band, it is not transmitted. When a sample occurs that is outside the tolerance band, it is transmitted. The amplitude of the transmitted sample is used to determine a new reference for the tolerance band. It has been found that the zero-order predictor is not only economical to implement, but it also yields a high compression efficiency. Interpolators use both past and future data points to achieve data compression. The zero-order interpolator is similar to the zero-order predictor except that the last sample of a sequence is transmitted (after an averaging process) instead of the first sample. A second type of interpolator is called the first-order interpolator. It is a first-order polynomial approximation to the curve of the transmitted data. Generally, higher compression ratios (up to about 6:1) can be obtained by the interpolators, but storage requirements are greater than those required by the predictors.

Encoding is a method of transforming data into a sequential format. An example of statistical encoding is Delta Modulation [20] which produces pulses that are dependent on the incoming signal. These pulses are integrated linearly or non-linearly and fed back for comparison with the input signal. The output consists of a series of positive and negative pulses. Only the positive pulses are transmitted. Non-linear integration of the pulses is probably more efficient than linear integration. Adaptive feedback can also be used to improve the performance.

Other types of data compression schemes applicable to image coding are the Fast Fourier Transform (FFT) and the Hadamard Transform [21]. Of these two the Hadamard Transform is the most efficient since it requires only $N \log N$ additions whereas the FFT requires $N \log N$ additions and $N \log N$ multiplications. The Hadamard operator is particularly suited to image coding since it is two-dimensional, self-inverse, symmetric, and orthogonal. The image coding method consists of transforming the data to the frequency domain, compressing the data (zonal or threshold sampling) and transmitting the data. At the receiving end an inverse transform is taken to obtain the approximate image. The advantage of transmitting in the frequency domain is that the effects of channel noise or errors are minimized by the averaging effect of the inverse transform.

5.3 Cataloging and Display

The information, display, cataloging and dissemination techniques for ERTS-A have been described in Section 4 of [1]. Various services are available to the user. The normal method for obtaining images or other ERTS-A data

is by request. The user submits an order for images or a request for data on file at the NASA Data Processing Facility. Catalogs of images, image descriptors, and data command system transmissions are distributed to the user. Other services available include microfilm images, a browse facility and a computerized data base which can be queried and searched to identify images of interest to the user.

The services listed above provide a variety of options for the user and appear to be adequate for the present state of the program. More information on the above can be found in [22, 23].

5.4 Recommended Changes on ERTS-A

In Chapter 4 three user applications were selected on the basis of an existing need. For each application detection of the objects of interest is based on individualistic spectral signatures. These spectral signatures differ from each other by the variations of response to radiance and reflectance with respect to wavelength. Thus, selection of the sensors depends on the spectral signatures. It was concluded that the present ERTS-A sensors are adequate for the selected user applications with the addition of sensors of wavelengths of $.32 - .38 \mu$, $1.5 - 1.8 \mu$, and $8 - 14 \mu$.

The processing techniques described in Section 5.2 are applicable to any of the wavelengths of the recommended ERTS-A sensors. Therefore, processing for each application is the same once the sensors have been fixed.

Figure 5.1 is a simplified block diagram of proposed ERTS-A on-board image processing. This figure is the same as Figure 2.2 except that additional blocks (wider lines) have been added to perform the on-board image processing. Figure 5.1 shows

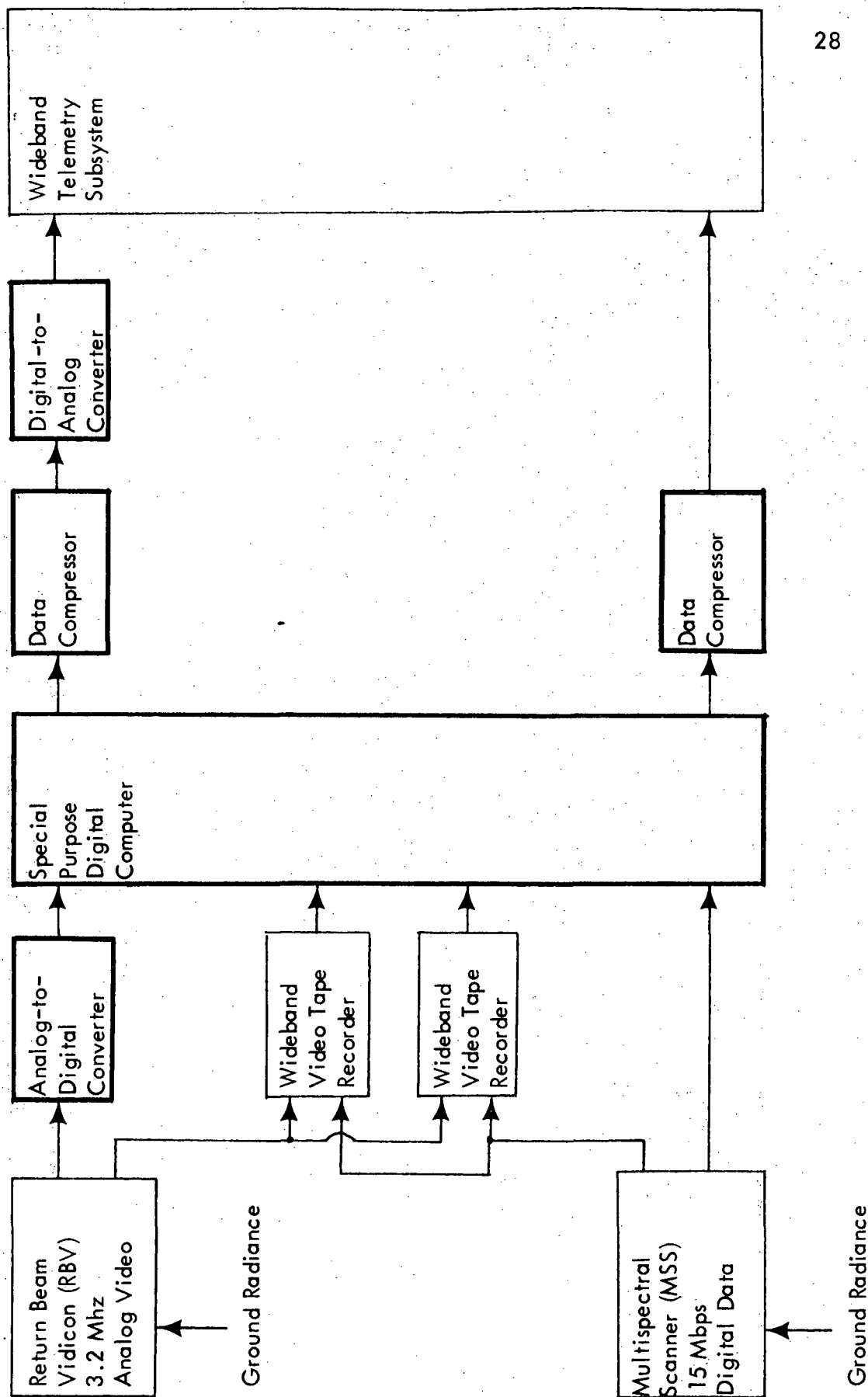


Figure 5.1 Proposed ERTS On-Board Information Processing

that one of the new blocks is a special purpose digital computer. The main functions of this digital computer are to take the Hadamard transform of the incoming image data and to compress the data by zonal or threshold sampling. Since the output of the RBV is analog video, it is necessary to include an analog-to-digital converter before the digital computer and a digital-to-analog converter at the input of the wideband telemetry system. The purpose of the two data compressors is to further compress the image data by redundancy reduction (e.g., zero-order predictor) or encoding (e.g., delta modulation).

After the image is received on the ground and before the magnetic tape is converted to film, another Hadamard transform must be taken to restore the image data to its original domain. This process is performed using a digital computer similar to the proposed on-board computer. One of the advantages of using the transform technique is that transmission noise is reduced due to the averaging characteristics of the transform. Detailed information on image data compression using the Hadamard transform is given in Part II of this report.

Figure 5.2 gives a recapitulation of the ERTS-A data flow for the selected user applications. The blocks in this figure contain either the pertinent information or the location in Part I of this report where the information can be found.

USER APPLICATION	SPECTRAL SIGNATURE	REMOTE SENSORS	ON-BOARD PROCESSING	DATA TRANSMISSION	GROUND PROCESSING	CATALOGING AND DISPLAY
Forestry (Increase Yield)	Figure 4.1	.6 - 1.0 μ	Figure 5.1	FM and PCM	Section 5.4 and Reference [1]	Section 5.3
Crops (Reduce Losses)	Figure 4.2	.6 - 1.0 μ	Figure 5.1	FM and PCM	Section 5.4 and Reference [1]	Section 5.3
Rangelands (Effective Management)	Similar to Figure 4.2	.32 - .38 μ .6 - 1.0 μ 1.5 - 1.8 μ 8 - 14 μ	Figure 5.1	FM and PCM	Section 5.4 and Reference [1]	Section 5.3

Figure 5.2 ERTS-A Data Flow for Selected Applications

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 Summary

The Earth Resources Technology Satellite (ERTS-A) system was described. Block diagrams were presented showing the flow of information from the remote sensors to the film or tape output. It was pointed out that on the satellite only signal processing takes place on board, but no information processing. Remote sensing fundamentals were given showing that each object being sensed has a particular spectral signature. Three user applications were selected: forestry (increase of yield), crops (reduction of losses), and rangelands (effective management). The spectral signature for each application was given. Then, the spectral bands of the remote sensors were determined for each application. On-board and ground information processing was investigated. It was shown that the Hadamard transform and delta modulation can be used to reduce the amount of data transmitted with no appreciable loss of information.

An investigation was made of the cataloging and display of ERTS-A output data. It was found that the user services provide a variety of options for the user and appear to be adequate for the present state of the program. References were given for further information on this subject.

6.2 Conclusions

ERTS-A is a well designed remote sensing system for detecting many objects on the earth's surface and observing their change of state over a period of time.

From the many reports in the literature ERTS-A has demonstrated that remote sensing from space is a feasible and practical approach to efficient management of most of the earth's resources.

For many applications the current remote sensors and processing are adequate. However, for other applications, e.g., rangeland management, changes must be made to the existing system to effectively sense all the spectral bands of interest. It was shown that for effective management of rangelands additional sensors of $.32 - .38 \mu$, $1.5 - 1.8 \mu$, and $8 - 14 \mu$ are needed. In addition, resolution can be improved by using variable focal length controlled from the ground.

An on-board information processing system using the Hadamard transform and delta modulation is feasible and recommended. Thus, the amount of data necessary to transmit an image can be reduced by a factor of 5 without an appreciable loss of information.

LIST OF REFERENCES

1. NASA "ERTS Data Users Handbook," Doc. No. 715D4249, Goddard Space Flight Center, Greenbelt, Maryland, September 1971.
2. Wintz, P. A., "Transform Picture Coding," Proc. of IEEE, Vol. No. 60, No. 7, July 1972.
3. Colwell, R. N., "Uses and Limitations of Multispectral Remote Sensing," Proc. Fourth Symp. Remote Sensing Environment, Univ. of Michigan, Ann Arbor, April 1966.
4. Gates et al., "Spectral Properties of Plants," Applied Optics, Vol. 4, No. 1, January 1965.
5. Suits, G. H., "The Nature of Infrared Radiation and Ways to Photograph It," Photogr. Engr. 26: 763-772.
6. Hoffer, R. M. and Johannsen, C. J., "Ecological Potentials in Spectral Signature Analysis," Remote Sensing in Ecology, University of Georgia Press, Athens, 1969, p.2.
7. Bauer, M. E. et al., "Detection of Southern Corn Leaf Blight by Remote Sensing Techniques," Proc. Seventh Symp. Remote Sensing of Environment, University of Michigan, Ann Arbor, May 1971.
8. Laboratory for Agricultural Remote Sensing, "Remote Multispectral Sensing in Agriculture," Volume No. 2, (annual report) Research Bulletin No. 832, Agriculture Experiment Station, Purdue University, July 1967.
9. Lauer, D. T., "Multispectral Sensing of Forest Vegetation," Photogrammetric Engr. 35: (4) 346-354.
10. School of Engineering, "Demeter, an Earth Resources Satellite System," Final Report, Stanford University, June 1968.
11. Langley, P. G., "New Multistage Sampling Techniques Using Space and Aircraft Imagery for Forest Inventory," Proc. Sixth Symp. Remote Sensing Environment, University of Michigan, Ann Arbor, October 1969.

12. Gates, D. M., "Characteristics of Soil and Vegetated Surfaces to Reflected and Emitted Radiation," Proc. Third Symp. Remote Sensing of Environment, University of Michigan, Ann Arbor, October 1964.
13. Lent, J. D., "Additive-Color Image Enhancement-Techniques and Equipment," Monitoring Earth Resources from Aircraft and Spacecraft, NASA SP-275, U. S. Government Printing Office, Washington, D. C., 1971, p. 11.
14. Gausman, H. W. et al., "Relation of Light Reflectance to Cotton Leaf Maturity," Proc. Sixth Symp. Remote Sensing of Environment, University of Michigan, Ann Arbor, October 1969.
15. Carneggie, D. M., "Applying Remote Sensing Technology for Improving Range Resource Inventories," Proc. Fifth Symp. Remote Sensing Environment, University of Michigan, Ann Arbor, April 1968.
16. Colvocoresses, A. P., "Image Resolutions for ERTS, Skylab, and Gemini/Apollo," Photogr. Engr. 38 (1) 33-35.
17. Stoelzner, W. E., "On-board Data Processing for Three Experiments Selected from the NASA Blue Book," Report No. TOR-0059 (6759-03) - 15, The Aerospace Corp., El Segundo, California, July 1970.
18. Husain-Abidi, A. S., "On-Board Spacecraft Optical Data Processing," NASA and Optical Society of America Conference on Holography and Optical Filtering, Huntsville, Alabama, June 1971.
19. Kortman, C. M., "Redundancy Reduction - A Practical Method of Data Compression," Proc. of IEEE, Vol. 55, No. 3, March 1967.
20. F. de Jager, "Delta Modulation, a Method of P. C. M. Transmission Using the 1-Unit Code," Phillips Research Report No. 7, 1952, pp. 442-466.
21. Pratt, W. K. et al., "Hadamard Transform Image Coding," Proc. of IEEE, Vol. 57, No. 1, January 1969.
22. Gurk et al., "Data Handling for Earth Resources Satellite Data," Proc. Sixth Symp. Remote Sensing of Environment, University of Michigan, Ann Arbor, October 1969.
23. Zeitler, E. O., and Bratton, R. D., "Uses of Earth Resources Data in the NASA/MSR Earth Resources Research Data Facility," Proc. Seventh Symp. Remote Sensing of Environment, University of Michigan, Ann Arbor, May 1971.

PART II

TRANSFORM TECHNIQUES FOR DATA COMPRESSION

CHAPTER 1

INTRODUCTION

This part documents the work done under Task II of contract NAS8-28545.

The main objective of this task is to study the effects of transformational data compression on the quality of a picture using discrete Fourier and Hadamard transforms.

The specific objectives are:

- (1) to develop computer programs for the fast computation of two-dimensional discrete Fourier and Hadamard transforms,
- (2) to define and discuss different transformational compression schemes and the corresponding compression ratios,
- (3) to experiment on a few digitized pictures, using both the Fourier and the Hadamard transforms, and different compression schemes to study distortion versus compression ratio.

The problem of interest in this part of the report is one of digital transmission of image data over a noisy communication channel. In such a situation, it is necessary to design a coding method which will minimize the bandwidth required to transmit a picture. The scheme should not degrade the quality of the image beyond certain fidelity limits, and furthermore, the coding method must not be overly sensitive to channel errors. Extensive investigation has been performed towards this end [1-3]. Most of these methods operate directly on the image, and they either do not exhibit satisfactory performance, or are too difficult to implement. A relatively new approach to the problem of bandwidth reduction is the use of transformational compression of images. This procedure achieves

a reasonably large bandwidth reduction and offers a certain immunity to channel errors. It is particularly attractive for use with the Fast Fourier and the Hadamard transforms, since fast computational algorithms are available for these transforms. The main intent of this part of the report is to study the effects of transformational data compression on the quality of a picture, using the Fast Fourier transform and the Hadamard transform.

Throughout this report, it is assumed that the original picture is already sampled and quantized, and the data are available in digital form. Sampling a picture corresponds to scanning in two dimensions where the output at each point is the corresponding gray level. After digitization, the image data can be arranged in the form of a matrix. It is further assumed that this matrix is square and that the number of rows (and columns) is an integral power of two.

The two transforms that are considered as candidates for transformational compression are the Fast Fourier transform and the Hadamard transform. Definitions and fast computation of these transforms as applied to two-dimensional data are briefly discussed in Chapter 2.

Two methods of data compression, truncation and coarse quantization, are examined in this report. General models of these schemes along with some preliminary justification for using transformational compression are discussed in Chapter 3. Suitable coding schemes are proposed for the two methods of compression. A measure of bandwidth reduction called the 'compression ratio' is defined based on these coding schemes. Also included in Chapter 3 is a brief

comment about the insensitivity to channel noises when the Fast Fourier or the Hadamard transform is used.

Implementations of the two compression schemes on a digital computer are explained in Chapter 4. This includes transformation, compression and inverse transformation.

The two compression schemes using both the transforms were applied to three typical pictures supplied by MSFC, and the results are presented in Chapter 5. Since no general measure of image distortion exists, the compressed pictures and the original pictures are examined visually to study the degradation in the quality of the picture. Experiments are conducted using various compression ratios and the results are compared.

A brief summary of the investigation and some concluding remarks are presented in Chapter 6.

Appendices A and B contain the development and implementation of the Fast Fourier and Hadamard transforms, respectively. FORTRAN listings of all the computer programs used in this study are included in Appendix C.

CHAPTER 2

IMAGE TRANSFORMS

2.1 Introduction

In this investigation, bandwidth reduction is achieved by transformational compression of images, where a transform of the image rather than the image itself is transmitted. Two discrete transforms, Discrete Fourier transform and the Hadamard transform, are considered. In this chapter, the definition and fast computation of these two transforms as applied to image data are discussed. It is assumed that the image is available in digitized form and is contained in a two-dimensional array f where the size of each dimension is $N = 2^m$, m being a positive integer.

2.2 Two-Dimensional Transforms

The general forward transform of a two-dimensional array f of size N in each dimension is defined as

$$F(i, j) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} a(i, j, m, n) f(m, n) \quad , \quad i, j = 0, 1, \dots, N-1 \quad (2.1)$$

where $a(i, j, m, n)$ is the forward transformation kernel. The inverse transformation is defined as

$$f(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b(i, j, m, n) F(i, j) \quad , \quad m, n = 0, 1, \dots, N-1 \quad (2.2)$$

where $b(i, j, m, n)$ is the reverse transformation kernel. The kernels $a(i, j, m, n)$ and $b(i, j, m, n)$ are said to be separable and symmetric if

$$a(i, j, m, n) = a_1(i, m) a_1(j, n) \quad (2.3)$$

$$\text{and } b(i, j, m, n) = b_1(i, m) b_1(j, n) . \quad (2.4)$$

A separable symmetric property is desirable for ease of implementation. Both the discrete Fourier transform and the Hadamard transform kernels exhibit this property.

2.3 Discrete and Fast Fourier Transforms

2.3.1 Definition and Implementation

The Discrete Fourier transform of a complex two-dimensional array f is defined by the kernel

$$a(i, j, m, n) = \frac{1}{N^2} w^{-(i m + j n)} \quad (2.5)$$

where w is a complex number given by

$$w = \exp(\sqrt{-1} \, 2\pi/N) . \quad (2.6)$$

It is easily seen that this kernel is a symmetric separable kernel and the forward transformation can be written as

$$F(i, j) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} w^{im} w^{jn} f(m, n) \quad (2.7)$$

where i and j are the spatial frequencies corresponding to the component $F(i, j)$.

Equation (2.7) can be implemented in two steps as follows:

$$f'(i, n) = \frac{1}{N} \sum_{m=0}^{N-1} w^{im} f(m, n) , \quad i, n = 0, \dots, N-1 \quad (2.8)$$

$$F(i, j) = \frac{1}{N} \sum_{n=0}^{N-1} w^{jn} f'(i, n) , \quad i, j = 0, 1, \dots, N-1 \quad (2.9)$$

It is well known [7] that $\frac{1}{N} w^{im}$ represents the forward transformation kernel of a one-dimensional discrete fourier transform. Therefore, equation (2.8) represents a set of N one-dimensional transforms, corresponding to each value of n , where the transform is taken along each column of the array f . Similarly, equation (2.9) represents a series of one-dimensional discrete transforms taken along each row of the matrix f' . Thus, the two-dimensional transform can be replaced by the one-dimensional transforms of each column followed by the one-dimensional transforms of each row.

If A is the $N \times N$ matrix which defines the discrete fourier transform, i.e., its (i, m) th element is $\frac{1}{N} w^{im}$, then equation (2.7) can be written as

$$F = A f A \quad (2.10)$$

This can be implemented as

$$F = \left(A (A f)^T \right)^T \quad (2.11)$$

since A is symmetric. The implementation is achieved in four steps: (a) transform all the columns, (b) transpose, (c) transform all the columns, and (d) transpose.

The Fast Fourier transform is an efficient algorithm for computing the Discrete Fourier transform. The four-step procedure outlined above is used to compute the two-dimensional transform starting from the one-dimensional Fast Fourier transform. The principle and implementation of the one-dimensional Fast Fourier transform are briefly discussed in Appendix A.

2.3.2 Fast Fourier Transform of Image Data

The Fourier transform is a complex transform, and hence requires the data to be stored in one complex two-dimensional array or two real two-dimensional arrays. However, the image data itself are real and so the imaginary part of each sample is zero.

An important property of the Discrete Fourier transform is the symmetry in the transform domain. If the imaginary part of the array f is zero, it can be easily seen from equation (2.7) that, since $w^N = 1$,

$$F(i, j) = F^*(N - i, N - j) \quad (2.11)$$

where superscript $*$ denotes complex conjugate. Therefore, for image data, all relevant information is contained in $\frac{N(N+1)}{2}$ complex samples given by

$$F(i, j), \quad i = 0, 1, \dots, N-1 \text{ and } j = 0, 1, \dots, i.$$

Another property inherent in the definition of the Discrete Fourier transform is the periodicity in the transform domain. It is easily shown that

$$F(i, j) = F(N + i, N + j) \quad (2.12)$$

and consequently

$$F(N-i, N-j) = F(-i, -j) \quad (2.13)$$

For a direct transformation kernel given by equation (2.5), it can be shown that the reverse transformation kernel is

$$b(i, j, m, n) = w^{im + jn} \quad (2.14)$$

Using this, it can be easily shown that

$$f = BFB \quad (2.15)$$

where the matrices B and A are related by

$$B = N A^* \quad (2.16)$$

Thus, the inverse transformation is an exact replica of the direct transformation except that w is replaced by its conjugate and the components are scaled by a factor of N .

2.4 The Hadamard Transform

The Hadamard transform is based on the Hadamard matrix which is a square array of plus and minus ones whose rows and columns are orthogonal to one another. If H_N is a symmetric Hadamard matrix of order N then

$$H_N H_N = NI \quad (2.17)$$

where I is the identity matrix. The matrix H_N may be made orthogonal by multiplying each element of the matrix by $1/\sqrt{N}$. It is known that if a Hadamard matrix of order N exists ($N > 2$), then $N \equiv 0 \pmod{4}$. The existence of a Hadamard matrix for all such values of N has not been shown, but a simple construction procedure is available when N is an integral power of 2. This construction procedure is

$$H_N = \underbrace{H_2 \times H_2 \times \dots \times H_2}_{m \text{ products}} \quad (2.18)$$

where $N = 2^m$, $\times$ denotes matrix Kronecker product and H_2 is a 2^{nd} order Hadamard matrix given by

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.19)$$

A frequency interpretation can be given to the Hadamard matrix [8]. The word "sequence" is used to designate the number of sign changes along each row of the Hadamard matrix. This frequency interpretation of the rows of a Hadamard matrix leads one to consider the rows to be equivalent to rectangular waves taking on values of plus and minus one. Such waves are called Walsh functions. Thus, the Hadamard transform can be considered as a decomposition of a function into a set of Walsh functions.

The Hadamard transform of a real two-dimensional array of size $N = 2^m$ in each dimension is defined as

$$F = \frac{1}{N} H_N f H_N \quad (2.20)$$

This definition is analogous to the Discrete Fourier transform and $\frac{1}{\sqrt{N}} H_N$ defines the one-dimensional Hadamard transformation matrix. A fast computational algorithm for computation of the one-dimensional Hadamard transform is discussed in Appendix B. This procedure can be used for computation of two-dimensional transform using the same four step process as in the case of the Fast Fourier transform.

It should be noted that the array obtained using equation (2.20) does not have the transform values in an increasing order of sequences. If it is necessary to arrange the components in an order of increasing sequences, an unscrambling procedure has to be implemented during the one-dimensional transformations.

Since H_N is symmetric and orthogonal, the inverse transformation is defined by

$$f = \frac{1}{N} H_N F H_N \quad (2.21)$$

Thus, the direct and inverse transformations are defined by the same relations. However, if the direct transformation were followed by an unscrambling operation, it is necessary to scramble this array before inverse transformation.

2.5 Computer Programs

Subroutines FFTON2 and SCHMD2 have been developed for the computation of the Fast Fourier transform and the Hadamard transform, respectively. These are based on the four-step procedure discussed above. A brief description of each program along with a FORTRAN listing is included in Appendix C.

CHAPTER 3

TRANSFORMATIONAL COMPRESSION OF IMAGE DATA

3.1 Introduction

Data compression is a technique to reduce the bandwidth needed to transmit a given amount of information in a given time, or to reduce the time needed to transmit a given amount of information in a given bandwidth. If a picture is represented by a set of discrete samples, in most cases it is found that the total information is almost uniformly distributed over all the samples. Furthermore, depending on the scanning rate, there exists a high degree of correlation between the samples. Therefore, to retain most of the information content, it becomes necessary to transmit a relatively large percentage of the samples. This, however, results in transmission of redundant data because of the correlation between the samples.

In order to make more efficient use of the available bandwidth, it is necessary to perform some preprocessing on the image. The main goals of the preprocessing operation are twofold:

- (1) to pack the total amount of information in a relatively small number of samples, so that most of the information content can be retrieved by transmission of a few samples,
- (2) to reduce the degree of correlation between the samples so that redundancy in transmission is minimized.

The simplest preprocessing technique would be a reversible linear transformation with the desirable properties. While the amount of decorrelation obtained

depends only on the transformation, the compaction of information using a given transform depends on the particular picture in question.

If a source consists of N samples, and only $M < N$ samples are transmitted, the transformation which minimizes the mean squared error is the Karhunen-Loeve transform [4]. Further, the transform samples are uncorrelated with each other, thereby minimizing redundancy. One big disadvantage of the Karhunen-Loeve transform is the fact that the transformation itself depends on the particular picture, resulting in a large computational effort and no general algorithm. Consequently, this method is not very practical and will not be investigated further.

In an effort to strike a compromise between performance and practicality, the Fast Fourier transform and the Hadamard transform will be considered as candidates for the transformation. While both have similar noise immunity properties, the Fourier transformation introduces more decorrelation, but requires more computational effort than the Hadamard transform. In terms of information packing, the performance depends on the picture and each transform is more efficient than the other in certain cases.

A general image transmission model is proposed in this chapter. The various components of this model, i.e., compression, quantization and coding, are discussed. Comments are made regarding the noise immunity property of the proposed model.

3.2 The General Model

The proposed model for digital image transmission is illustrated in Figure 3.1. The digitized picture in the form of a two-dimensional array f is transformed using a reversible linear transformation T . The transformed samples are then quantized for transmission over the digital transmission system. These quantized samples F are then subject to compression which is represented by a non-reversible linear transformation Q . The compressed transform samples F_1 are then coded for transmission. The received set of samples G_1 , which is the same as F_1 but contaminated by channel errors, is then decoded and an inverse transformation yields the output picture data.

The two linear transformations considered are the Fast Fourier transform and the Hadamard transform, which were discussed in detail in Chapter 2. Other components of the model will be studied in subsequent sections.

3.3 Quantization

When the digitized image is transformed using a digital computer, each transformed sample can assume one of a finite set of values, where the size of the set depends on the word length of the computer. However, the number of different values that can be transmitted over a digital channel is usually much less than the size of this set. For example, a typical computer could have a 36 bit word, whereas a typical transmission channel can accommodate 6 or 7 bits. Thus the transform samples can take one of 2^{36} different values, while only 2^6 or 2^7 different numbers can be transmitted on the channel. Consequently, it becomes necessary to approximate each transform sample by an available transmission level. Hence,

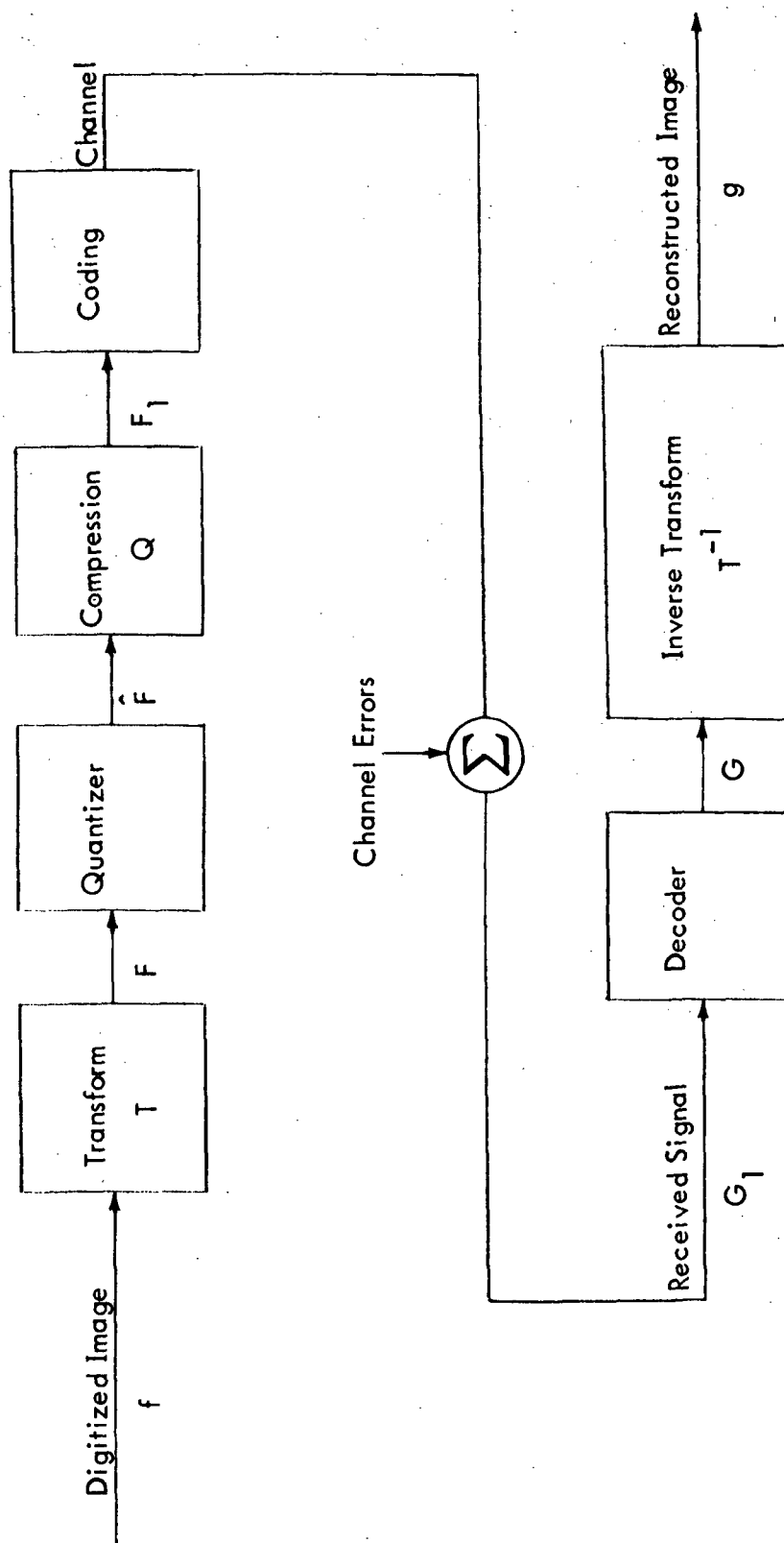


Figure 3.1 Model of an Image Transmission System

if a 6 bit channel is used, then each transform sample must be approximated by one of $2^6 = 64$ levels. The quantization levels should be selected such that the error due to quantization is minimized.

In order to determine the effect of quantization, it is necessary to define an error criterion. In general, the error criterion chosen depends on the particular application of the final reconstructed image. Thus, it is not possible to define a quantization scheme which is optimum for all applications. If $F(i, j)$ is the value of a sample before quantization, and $\hat{F}(i, j)$ is the corresponding quantized sample, Figures 3.2 and 3.3 depict linear and nonlinear quantization schemes, respectively. While the linear scheme is very simple to implement, it is necessary to use a suitable nonlinear scheme in order to satisfy fidelity criteria.

If the final application of the image is subjective viewing, the incremental brightness changes in the reconstructed image are much more noticeable if the brightness level is low than if it is high. Thus, the density of quantization levels should be greater for low level transform samples. From psychological tests, it is known that the human viewer is very sensitive to the location of the high frequency brightness transitions, but relatively insensitive to their actual magnitudes. Hence, the density of quantization levels at a given transform sample amplitude should be greater at higher frequencies (or sequences) than at lower frequencies (or sequences). In summary, the density of quantization levels should be:

- (a) greater at lower sample values for a given frequency (or sequence),
- (b) greater at higher frequencies (or sequences) for a given sample value.

If photometric measurements are to be made on an image, the cumulative mean squared error is a common fidelity criterion. In such a situation, the levels

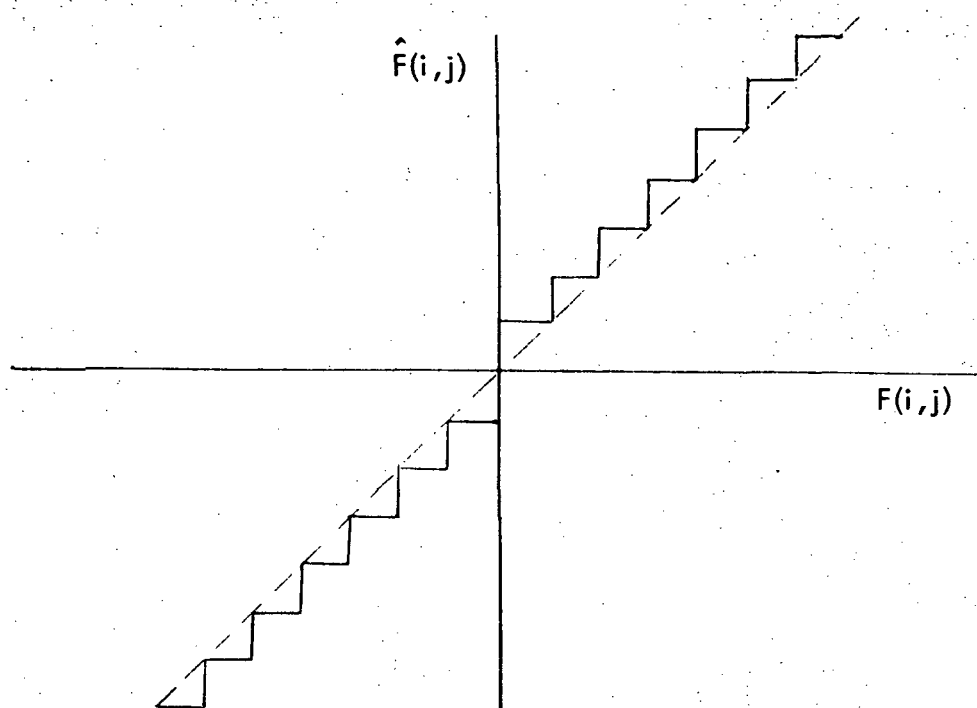


Figure 3.2 Linear Quantization Scheme

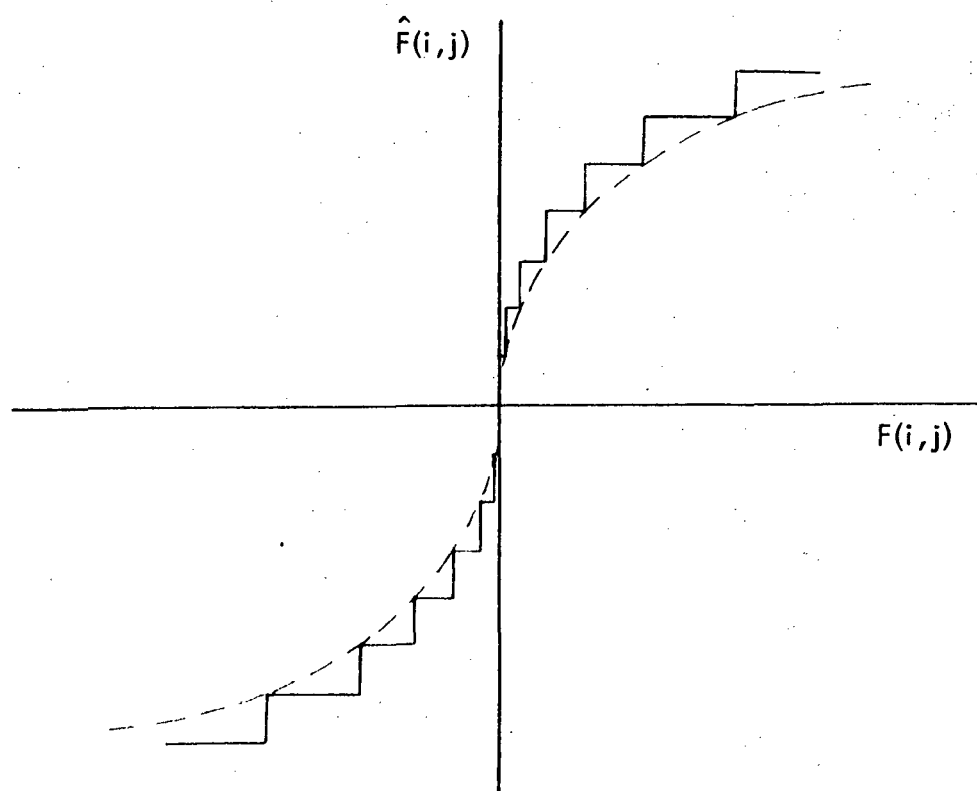


Figure 3.3 Nonlinear Quantization Scheme

of quantization in the transform domain must be selected to minimize the cumulative mean squared error in the spatial domain.

The effect due to quantization exists whether the image samples themselves are transmitted or the transform samples are transmitted. Thus the error due to quantization does not have a high influence on the error due to transformational compression. Since the main purpose of this investigation was to study the effects of compression, no further study of quantization was made, nor was a quantization scheme implemented in the experiments. A detailed review of different quantization schemes and their effects are to be found in [5].

3.4 Compression and Coding

Even after transformation and quantization, the set $\hat{F}$ contains the same number of samples as the original image f . Therefore, no bandwidth reduction is achieved if this set $\hat{F}$ is transmitted in its entirety. In order to obtain data compression, it is necessary to transmit only a few of the samples while still retaining a large portion of information. The set $\hat{F}$ should therefore be replaced by a set F_1 which is smaller in size than $\hat{F}$. Two types of compression, truncation and coarse quantization, are considered. Since each compression technique requires a different type of coding, a typical coding scheme is briefly discussed along with the compression procedure.

In order to determine the amount of bandwidth reduction achieved, a term called compression ratio (CR) is defined as follows,

$$CR = \frac{\text{Original number of samples}}{\text{Number of samples transmitted}}$$

3.4.1 Truncation

In most images of interest, the energy in the transform domain tends to be clustered at certain spatial frequencies or sequences. Specifically, most of the information is contained in the lower frequency (or sequence) components. Thus, the obvious method of conserving bandwidth is simply to not transmit the high frequency (or sequence) information. These components are replaced by zeros at the receiver before inverse transformation.

3.4.1.1 Truncation Using Hadamard Transformation

Let $\hat{F}$ be the quantized set of transform samples where the elements are sequence ordered. If the number of samples in each dimension is N and only $M < N$ samples are retained in each dimension, then the array after compression F_1 is a two-dimensional array of size M in each dimension where

$$F_1(i,j) = \hat{F}(i,j) \quad \text{for } \begin{matrix} i = 1, 2, \dots, M \\ j = 1, 2, \dots, M \end{matrix} \quad (3.1)$$

After decoding at the receiver, zeros are substituted in place of each of the transform samples that were not transmitted. Therefore, assuming no channel errors, the set G is given by

$$\begin{aligned} G(i,j) &= \hat{F}(i,j) \quad i, j \leq M \\ &= 0 \quad \text{Otherwise} \end{aligned} \quad (3.2)$$

where G is a two-dimensional array of size N in each dimension.

3.4.1.2 Truncation Using Fast Fourier Transformation

In this case, the transform consists of two arrays, magnitude and phase, each of which is a two-dimensional array of size N in either dimension. But it was shown in Chapter 2 that Fast Fourier transformation implies periodicity and that only $\frac{N(N+1)}{2}$ samples of each array need to be transmitted. Specifically, if $\hat{F}$ is the complex transformed array, then all the information is contained in

$$\{ \hat{F}(i,j), \quad i = 1, \dots, N; j = 1, \dots, i \}$$

If it is desired to retain only M sample in each dimension, then, using the periodicity property of the Fourier transform the compressed array G of size M in each dimension is given by

$$G(i,j) = \hat{F}(i,j) \quad (3.3)$$

$$\text{for (i) } i, j = 1, 2, \dots, M/2$$

$$\text{(ii) } i = 1, 2, \dots, M/2; j = N, N-1, \dots, N - \frac{M}{2} + 1$$

$$\text{(iii) } i, j = N, N-1, \dots, N - \frac{M}{2} + 1$$

$$\text{and (iv) } i = N, N-1, \dots, N - \frac{M}{2} + 1; j = 1, 2, \dots, M/2$$

Hence, the array G is a complex array which is two-dimensional of size M in each dimension. However, only $\frac{M(M+1)}{2}$ samples of the array need to be transmitted. After decoding at the receiver, zeros are filled in for the rejected samples as in the case of the Hadamard transform.

3.4.1.3 Coding for Truncation

Since the samples are already quantized to the number of levels available for transmission, no special coding scheme is necessary.

3.4.1.4 Compression Ratio

In the case of the Hadamard transform, it was seen that a total of M^2 samples have to be transmitted. Since the total number of samples to start with was N^2 , the compression ratio is given by

$$CR = \frac{N^2}{M^2} \quad (3.4)$$

In the case of the Fourier transform, it is required to transmit a complex array of size $\frac{M(M+1)}{2}$. Therefore, the compression ratio is

$$CR = N^2 / M(M+1) \quad (3.5)$$

3.4.2 Coarse Quantization

In the case of truncation bandwidth reduction was achieved by suppressing the high frequency (or sequency) components. It is a well-known fact that the high frequency transitions are important even though they are relatively few in number and contain a low proportion of the image energy. Further, in the case of truncation, large magnitude samples of high frequency are indiscriminately discarded. An obvious alternative is to retain only those samples whose magnitudes are above a given threshold level and the others are set to zero. With this type of compression, it becomes necessary to provide information as to the location of the significant samples.

3.4.2.1 The Compression Scheme

The compression technique is merely to eliminate certain samples in the transform domain. The other elements which are assumed to be insignificant are set equal to zero. Specifically, if $T > 0$ is a preselected threshold, then the compressed array F_1 is a two-dimensional array of size N in each dimension where

$$\begin{aligned} F_1(i,j) &= \hat{F}(i,j) && \text{if } |\hat{F}(i,j)| > T \\ &= 0 && \text{if } |\hat{F}(i,j)| \leq T \end{aligned} \quad (3.6)$$

The above equation, which holds good for both the Hadamard transform and the Fourier transform, suggests that the compressed array F_1 contains a large number of zeros if the image energy is concentrated in a few transform samples. However, F_1 still contains N^2 samples and no compression is achieved if all the samples are transmitted. In order to achieve compression a special coding scheme is devised which makes use of the fact that F_1 contains a large number of zeros. Using this special coding scheme, the output of the decoder at the receiver is exactly the same as F_1 if no channel errors exist. In this case, therefore, the best transform is the one which maximizes the number of transform samples which are zero or near zero.

3.4.2.2 Run Length Coding

In order to achieve a bandwidth reduction, it is necessary to code the positions of the significant samples as well as their values. Position coding, of course, adds

to the transmission bandwidth. Instead, a method called run-length coding is used to achieve bandwidth reduction. In this method, rather than transmitting a position code for each deleted sample, a code giving the number of adjacent samples which have been deleted (called run length) is transmitted. Experimental results have shown that enough deleted samples occur adjacent to each other to make this technique feasible.

As an example let the transform samples be quantized into 64 levels, -31 to +32, and let the threshold be set at a level of +4. Then, all samples that fall in the levels -3 to +4 are set equal to zero, which leaves a total of eight levels which are not used for coding the samples. These can be used for coding run lengths. Therefore, each sample and each run length requires the transmission of a 6-bit word.

The coding is done by scanning along each line of the transformed data, and, if a sample falls above the threshold, its amplitude is coded; if a sample falls below the threshold, then the coder waits to determine how many consecutive samples fall below the threshold. This number is then coded in lieu of coding the zero amplitudes.

One problem with this technique, however, is the error propagation effect occurring whenever an error occurs in one of the run-length words [6]. An error in a run-length word causes all of the subsequent words to be shifted in the transform domain. In order to compensate for error propagation effects, synchronization data must be included in the transmitted signal, so that the shifting of the data caused by an error does not extend over the entire data set.

3.4.2.3 Compression Ratio

The number of transmitted samples depends on the run lengths. Let K be the maximum run length allowed and let L_i , $i = 1, 2, \dots, K$ be the number of run

lengths of length i . Therefore, the total number of deleted samples is given by

$$N_Z = \sum_{i=1}^K L_i i \quad (3.7)$$

and the total number of run lengths is

$$L = \sum_{i=1}^N L_i \quad (3.8)$$

The total number of significant samples to be transmitted is $N^2 - N_Z$, and the total number of words to be transmitted including run lengths is $N^2 - N_Z + L$. Therefore, the compression ratio is given by

$$CR = \frac{N^2}{N^2 - N_Z + L} \quad (3.9)$$

It is easily seen that the compression ratio depends on the threshold level chosen.

For a given threshold, the compression ratio obtained depends on the particular picture.

3.4.2.4 Selection of Threshold

Since the threshold depends on the particular image for a desired compression ratio, it was made adaptive by defining it as a multiple of the absolute mean of the image transform. Variation of the constant multiplier yields different compression ratios.

3.5 Immunity to Noise

The effects of noise are reduced in two ways. First, since only a few samples are transmitted, the noise energy entering the system is much smaller than if all the samples were transmitted. Secondly, the effect of noise at the output is lower than that if the image were transmitted directly instead of its transform, since the noise contaminates the transform samples and any large disturbance of one frequency (or sequence) value is spread out over the entire output picture because of the averaging nature of the inverse transformations. This suggests that the transmission of transforms might be of great value even under a zero compression requirement.

3.6 Summary

A general model for the digital transmission of images was proposed. This model, wherein the transform of the image was transmitted instead of the picture itself, featured transformational data compression as a tool to bandwidth reduction. A brief outline of quantization of transform data was followed by a detailed description of two types of compression, truncation and coarse quantization. Each of the two compression techniques was discussed for application using the Fast Fourier transform and the Hadamard transform. Typical coding schemes and the corresponding compression ratios were proposed for both methods.

CHAPTER 4

COMPUTER IMPLEMENTATIONS OF THE EXPERIMENTS

4.1 Introduction

In order to evaluate their performance, the two compression techniques were experimented on three digitized pictures using both the Hadamard transform and the Fast Fourier transform. Since only the compression techniques are being evaluated, the experiments do not include quantization, coding and channel noises. The simplified experiment is illustrated in Figure 4.1.

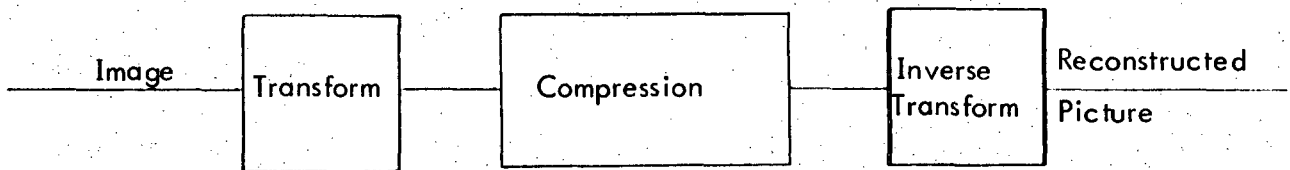


Figure 4.1 The Simplified Experiment

Three sets of image data were provided by the MSFC. Each picture was scanned to 256 samples in both dimensions and the data were stored on a magnetic tape. FORTRAN listings along with brief descriptions of all the subroutines used are included in Appendix C.

4.2 Hadamard Transform Compression

4.2.1 Transformation

The two-dimensional image data are read into an array X of size 256×256 .

Subroutine SCHMD2(N,M,KOPT) is used to obtain the Hadamard transform of

the array X . The result is also stored in array X which is transferred to the subroutine through a labelled common. The value of N is 256, that of M is 8 and $KOPT = 3$ is used so that the transform is sequency ordered.

4.2.2 Compression by Truncation

As explained in the previous chapter, all components corresponding to sequencies higher than a particular number are deleted. These deleted samples are replaced by zeros at the receiving end. Subroutine $CMPHM1(NTRUNC, N, M)$ is used for truncation. The transformed array X of size $N \times N$ is transmitted through a labelled common. In the subroutine, all components corresponding to sequencies greater than $NTRUNC$ are set equal to zero and the resulting array is stored in X . Thus,

$$X(I, J) = X(I, J) \quad \text{for } I = 1, 2, \dots, NTRUNC \text{ and } J = 1, 2, \dots, NTRUNC$$

and $X(I, J) = 0$ otherwise.

An inverse Hadamard transformation of the array X is taken before it is returned.

4.2.3 Compression by Coarse Quantization

In this case all components whose magnitudes are less than a threshold are set equal to zero. Before compression, it is necessary to fix a threshold, $TRSOLD$, and the number of levels available for run length coding, $KZERO$. $TRSOLD$ is computed as $XCOEFF$ times the absolute mean of the image transform. Subroutine $CMPHM2(JN, N, M, IRUN, TRSOLD, KZERO)$ is used for coarse quantization. The uncompressed transform array X of size $N \times N$ is transmitted through a labelled common. In the subroutine, all the elements of X whose magnitudes are below the threshold are set equal to zero and the resulting array is stored in X . Thus,

$$\begin{aligned}
 X(I,J) &= X(I,J) \text{ if } X(I,J) > \text{TRSOLD} \\
 &= 0 \quad \text{if } X(I,J) \leq \text{TRSOLD}
 \end{aligned}$$

An inverse Hadamard transformation of the array X is taken before it is returned.

In order to compute the compression ratio, it is necessary to calculate the number of run lengths in the compressed array. Subroutine CMPHM2 provides this information called IRUN as an output along with JN which is the total number of zeros in the array.

4.2.4 Inverse Transformation

As was explained in the previous subsections, the inverse transformation of the compressed and zero filled array is taken in the compression subroutine. The inverse Hadamard transform is obtained by calling subroutine SCHMD2 (N,M,KOPT) with $N = 256$, $M = 8$ and $KOPT = 1$. The array X to be transformed is transferred to the subroutine through a labelled common.

4.3 Fourier Transform Compression

4.3.1 Transformation

The two-dimensional image data are read into the array XR of size 256×256 . The array XI of the same size as XR is filled with zeros and corresponds to the array containing imaginary parts. The direct transformation is achieved by calling subroutine FFTON2(N,M,KRZIN, KRZOUT, KUNSCR) where $N = 256$, $M = 8$, $KRZIN = 1$, $KRZOUT = 2$ and $KUNSCR = 1$. The arrays XR and XI are transferred to FFTON2 through labelled commons and the transformed array is contained in XR and XI , where XR contains the magnitude samples and XI contains the phase samples.

4.3.2 Compression by Truncation

In this case all samples corresponding to frequencies higher than a particular value are set equal to zero. Subroutine CMPFT1(NTRUNC,N,M) is used for truncation and receives the transformed arrays XR and XI through labelled commons. The transformed data are compressed and the resulting arrays are again stored in XR and XI. Thus,

$$\left. \begin{array}{l} \text{XR(I,J)} = \text{XR(I,J)} \\ \text{and } \text{XI(I,J)} = \text{XI(I,J)} \end{array} \right\} \text{ for } \begin{array}{l} i, j = 1, 2, \dots, \text{NTRUNC} \\ i, j = 1, 2, \dots, \text{NTRUNC}; j = \text{N}, \text{N}-1, \dots, \text{N} + 1 - \text{NTRUNC} \\ i, j = \text{N}, \text{N}-1, \dots, \text{N} + 1 - \text{NTRUNC} \\ i = \text{N}, \text{N}-1, \dots, \text{N} + 1 - \text{NTRUNC}; j = 1, 2, \dots, \text{NTRUNC} \end{array}$$

An inverse Fast Fourier Transformation of XR and XI is taken before returning.

4.3.3 Compression by Coarse Quantization

Here all components which have magnitudes smaller than a threshold are set equal to zero. The threshold TRSOLD is computed as XCOEFF times the absolute mean of the image transform. Subroutine CMPFT2(JN,N,M,IRUN,TRSOLD,KZERO) is similar to subroutine CMPHM2 and receives XR and XI through labelled commons. The compressed data are also stored in XR and XI. Thus,

$$\left. \begin{array}{l} \text{XR(I,J)} = \text{XR(I,J)} \\ \text{XI(I,J)} = \text{XI(I,J)} \end{array} \right\} \text{ if } \text{XR(I,J)} > \text{TRSOLD}$$

$$\left. \begin{array}{l} \text{XR(I,J)} = 0 \\ \text{XI(I,J)} = 0 \end{array} \right\} \text{ if } \text{XR(I,J)} \leq \text{TRSOLD} .$$

An inverse Fast Fourier Transformation is taken before returning. As in the case of CMPHM2, the subroutine CMPFT2 also returns as outputs the total number of zeros

JN and the number of run lengths IRUN which are required for computation of the compression ratio.

4.3.4 Inverse Transformation

It is necessary to obtain the inverse transformation of the compressed arrays XR and XI in order to reconstruct the picture. This is achieved by calling subroutine FFTON2(N,M,KRZIN,KRZOUT,KUNSCR) with $N = 256$, $M = 8$, $KRZIN = 2$, $KRZOUT = 1$ and $KUNSCR = 1$. This subroutine receives XR and XI through labelled commons and the reconstructed picture is stored in the two-dimensional array XR.

4.4 Subroutines XREAD and XWRITE

The original image data were stored on a tape and it was decided to store the reconstructed image data also on a magnetic tape. Subroutines XREAD and XWRITE are used for reading one complete picture and writing one complete picture, respectively. In the case of the subroutine XWRITE, an end of file mark is written to mark the end of the picture data.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Introduction

Three images corresponding to three different frequency bands of the same subject were provided by MSFC. Each of the images had been scanned at 50 microns with a scanner giving 64 gray levels. Even though the original images supplied were of size 1024×1024 , only a portion of each image of size 256×256 was selected for experimental purposes. Experiments were conducted using the computer programs described in Chapter 4 on each of the pictures for the following cases.

- (1) Truncation with Hadamard transform,
- (2) Truncation with FFT,
- (3) Coarse quantization with Hadamard transform, and
- (4) Coarse quantization with FFT.

In each case the experiments were conducted for various compression ratios. After compression and reconstruction, the data were quantized to 64 levels and a 64 level photowriter was used to record the images on film at a 50 micron rate. Enlarged photographic prints of these films are used as the results of the experiments.

5.2 Presentation of Experimental Results

To facilitate easy comparison, the original pictures themselves were printed and are shown in Figure 5.1. Two values of NTRUNC, 162 and 100, were used for truncation with the Hadamard transform, resulting in compression ratios of approximately 2.4 and 6.5, respectively. The results of these are shown in Figure 5.2 and 5.3 and the corresponding results for truncation with FFT are shown in Figures 5.4 and 5.5.

Two values of XCOEFF, 0.8 and 1.2, were used for coarse quantization. These resulted in compression ratios of approximately 2.4 and 3.5 for each of the pictures and both the transforms. Results of coarse quantization with the Hadamard transform are given in Figures 5.6 and 5.7, and those with the Fourier Transform are given in Figures 5.8 and 5.9.

5.3 Discussion of the Results

It was mentioned in Chapter 3 that a drawback of truncation is the fact that large magnitude samples of high frequency (or sequency) are indiscriminately discarded. This is overcome using coarse quantization and from the results it is seen that coarse quantization is superior to truncation for a given compression ratio. However, coarse quantization requires a coding scheme which is far more complicated than the one used with truncation.

From the results it is seen that there is no appreciable degradation in the quality of the image for a compression ratio of 2.4 with either method and either transform. However, there is a defocussing effect in the case of truncation due to the elimination of high frequency transitions. Distortion becomes noticeable for compression ratios higher than 3.0 especially in the case of truncation.



Figure 5.1 The Test Pictures

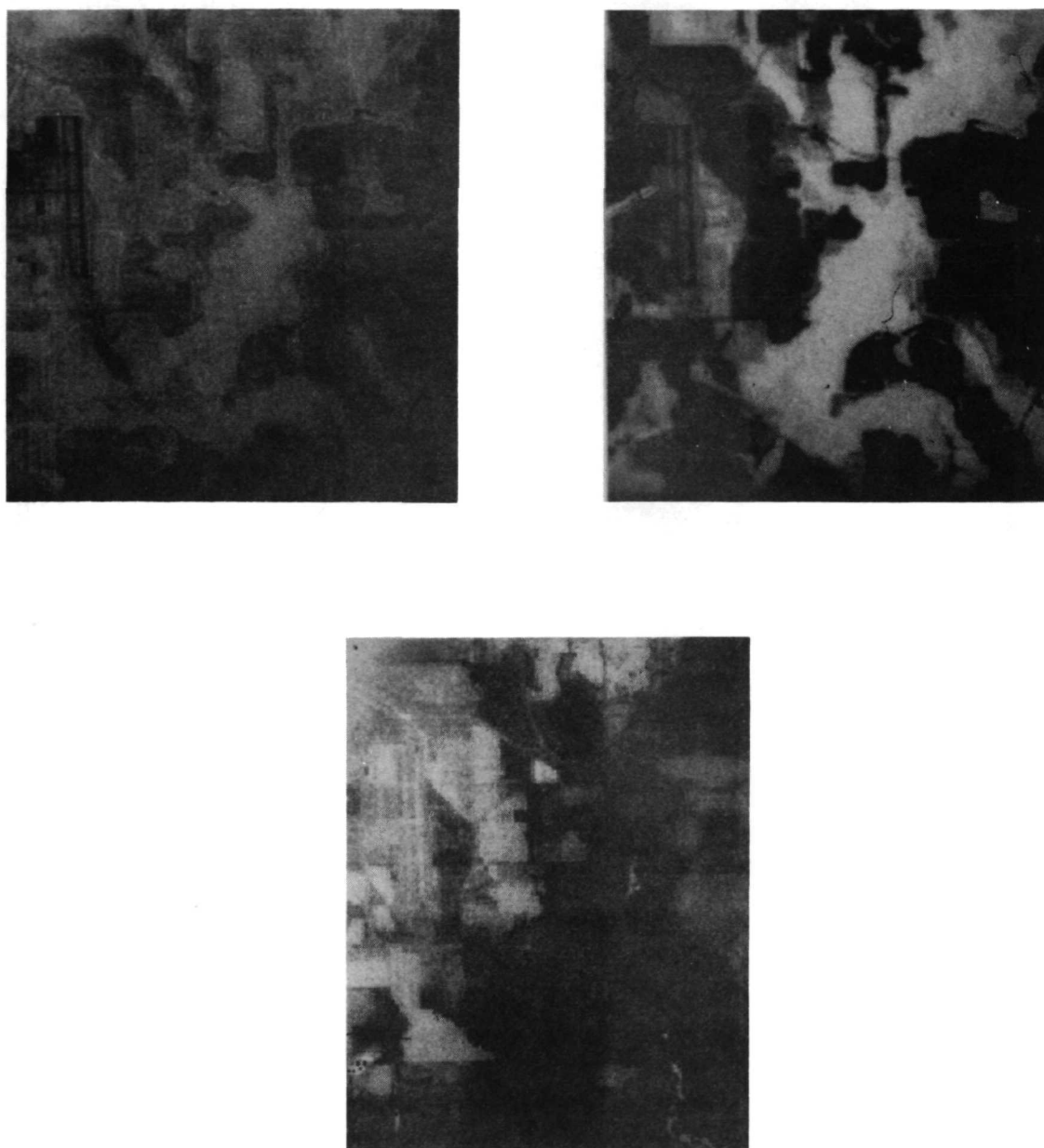


Figure 5.2 Truncation with Hadamard Transform, $CR = 2.4$

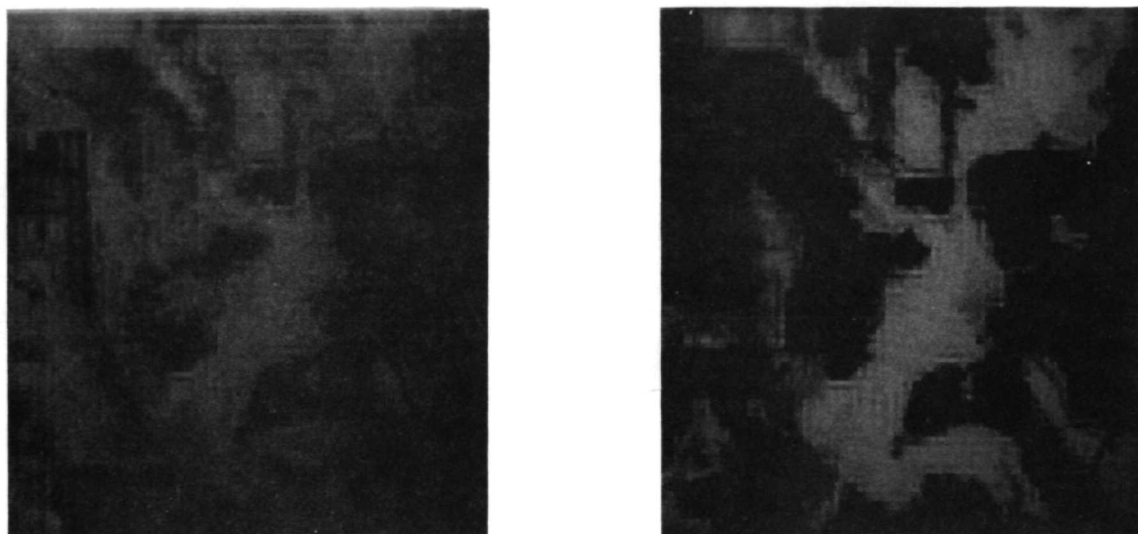


Figure 5.3 Truncation with Hadamard Transform, $CR = 6.5$

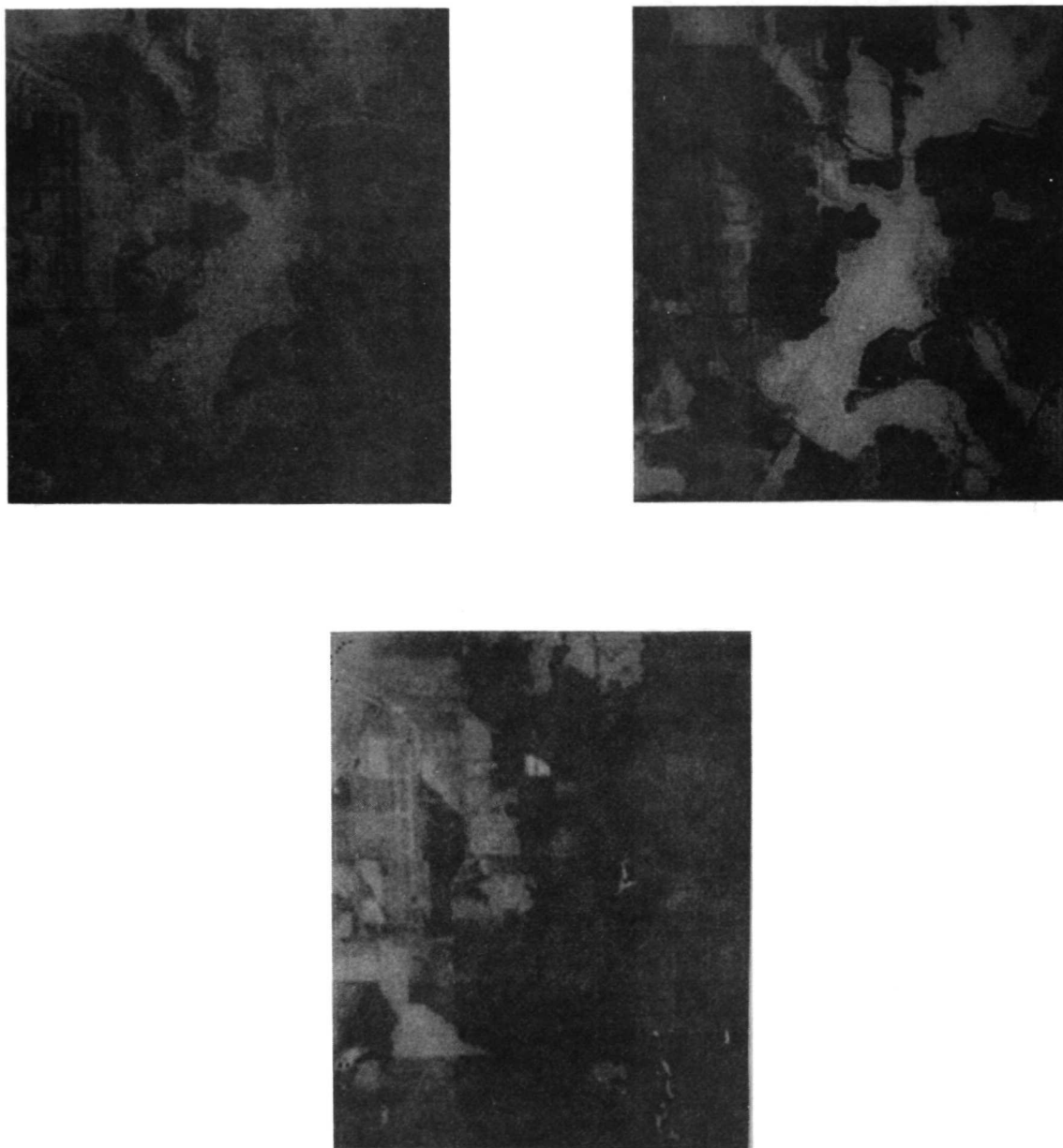


Figure 5.4 Truncation with FFT, CR = 2.4



Figure 5.5 Truncation with FFT, $CR = 6.5$

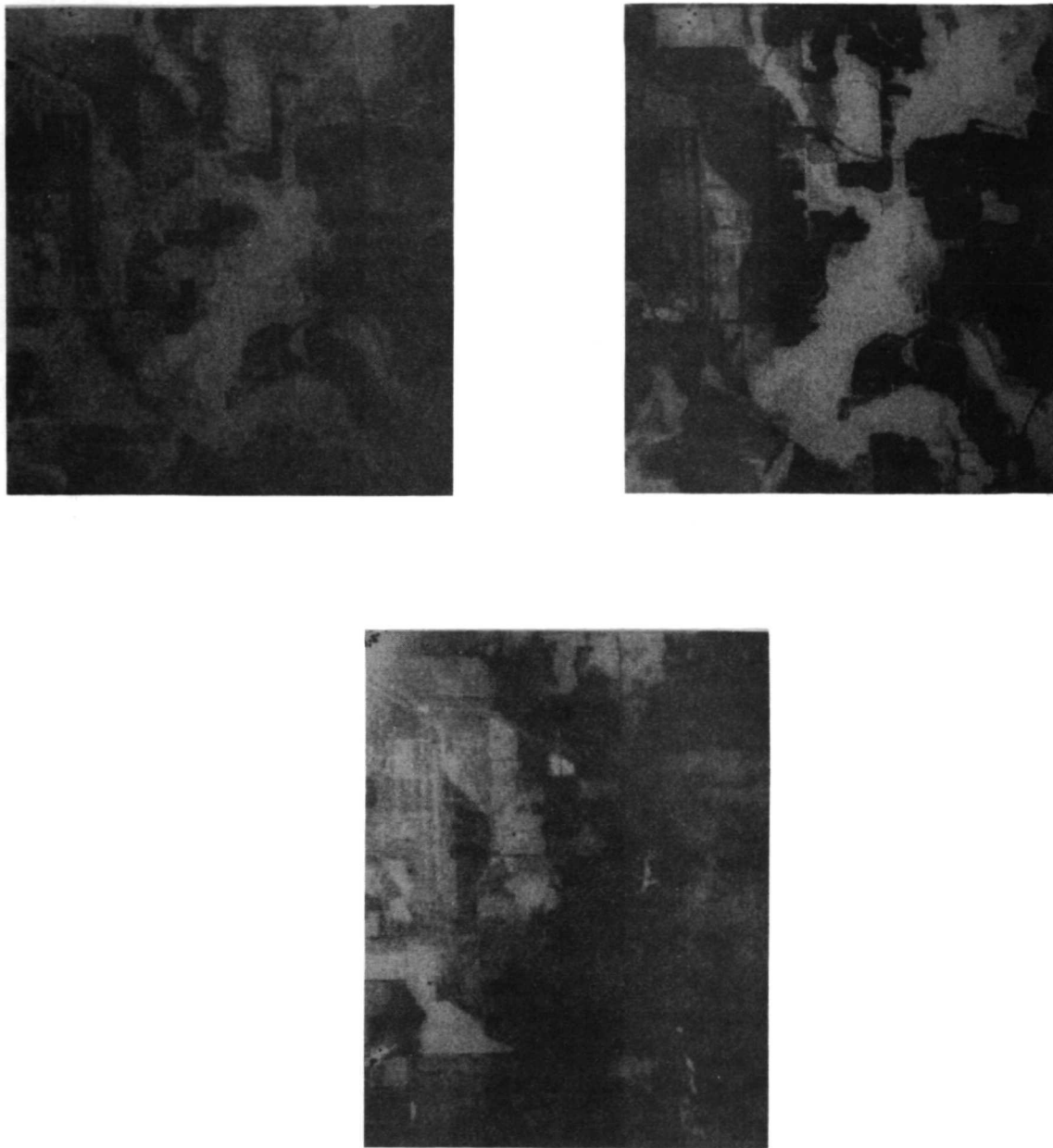


Figure 5.6 Coarse Quantization with Hadamard Transform, $CR = 2.4$

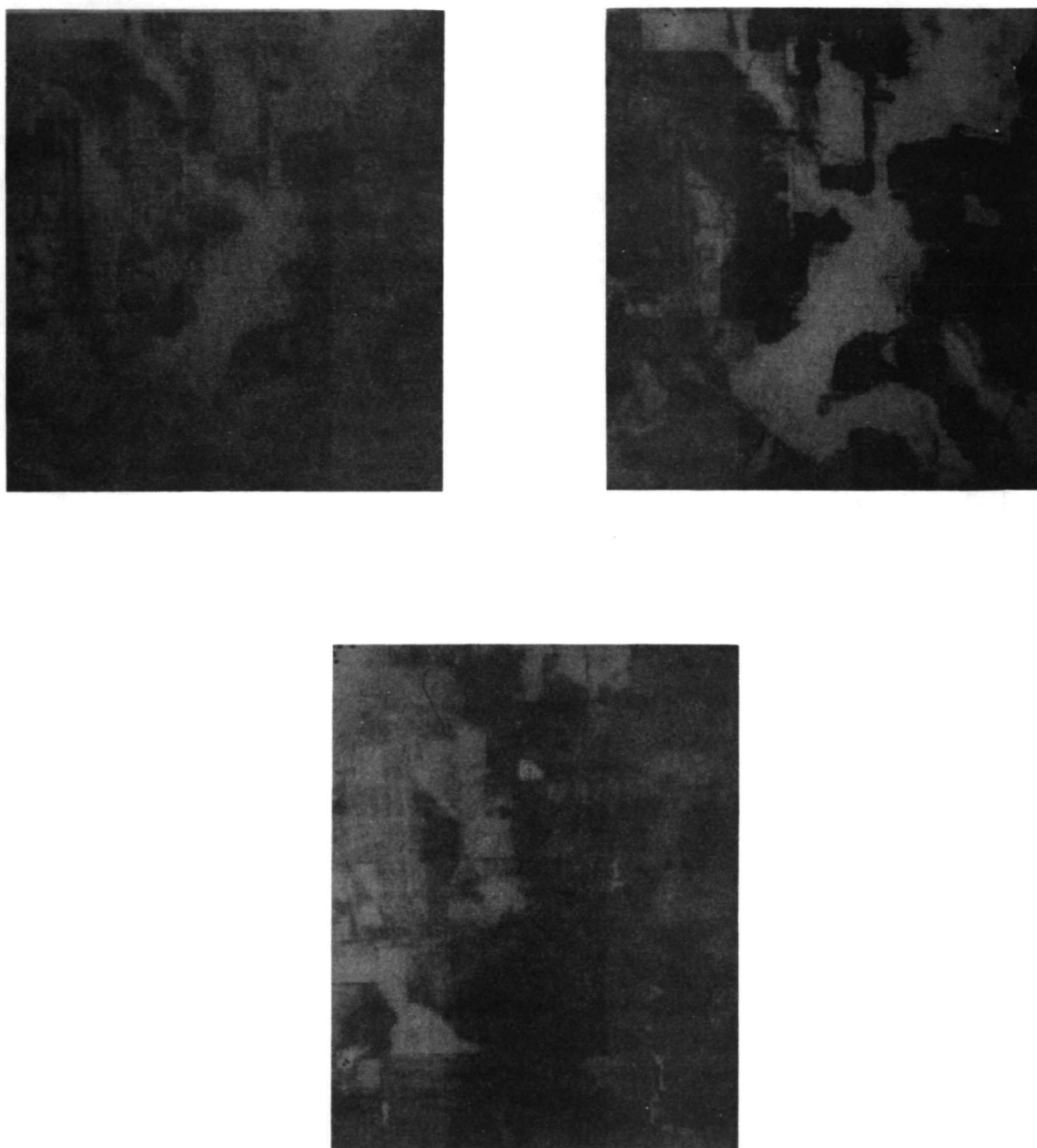


Figure 5.7 Coarse Quantization with Hadamard Transform, CR = 3.5

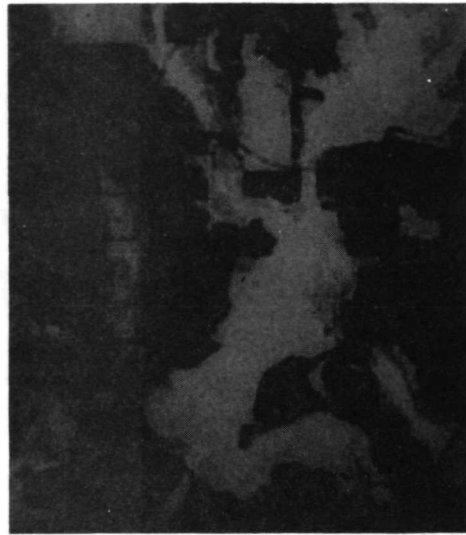


Figure 5.8 Coarse Quantization with FFT, $CR = 2.4$

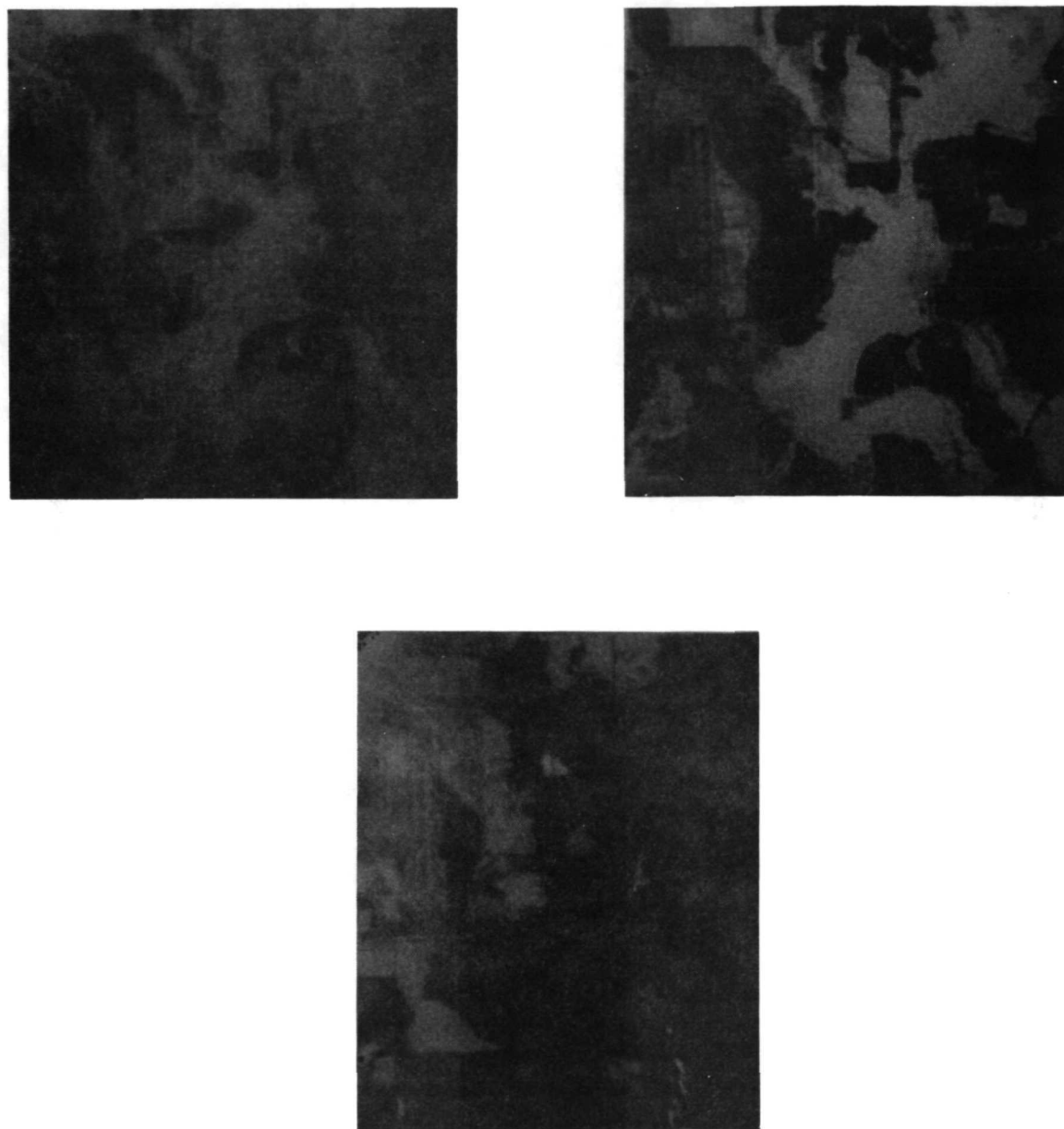


Figure 5.9 Coarse Quantization with FFT, CR = 3.5

For the pictures under test, the Fourier transform results in less degradation than the Hadamard transform for a given method of compression and a given compression ratio. However, the suitability of the transform depends on the particular picture in question. A major factor in favor of the Hadamard transform is the fact that the direct or inverse Fast Fourier transform requires much more computational effort than the Hadamard transform. The Hadamard transform of image data of the size $N \times N$ requires $2N^2 \log_2 N$ algebraic additions, whereas the Fast Fourier transform requires $2N^2 \log_2 N$ complex additions and $N^2 \log_2 N$ complex multiplications.

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 Summary

The effects of transformational image data compression on the quality of a picture was the subject of discussion in this part of the report. The two image transforms considered, the Fast Fourier transform and the Hadamard transform, were briefly described along with efficient schemes for their computations. A general digital image transmission model was outlined which includes transformational data compression. Two methods of compression, truncation and coarse quantization, were proposed. Suitable coding schemes were suggested for each method of compression and a measure of data compression called the compression ratio was defined. The compression schemes were implemented on a digital computer using both transforms. Experiments were performed using three digitized pictures and different compression ratios to study distortion versus compression.

6.2 Conclusions

Based on the results of the experiments on the three pictures, it is seen that considerable data compression can be achieved by using the Fast Fourier and the Hadamard transforms. The Fourier transform is slightly superior to the Hadamard transform in terms of distortion, but the ease with which the Hadamard transform can be computed makes it especially attractive. In addition to providing high data compression, the transmission of the transform also results in a certain amount of noise immunity.

The main problem in using the transform technique is one of data handling. The amount of data involved is so huge that all the data cannot be stored in the core of the computer and hence it must be stored externally. This poses serious problems during the computation of the transform [9] which requires a transposing of the array. Thus, it is necessary to develop efficient schemes for transposing large two-dimensional arrays which are stored externally. One such scheme has been developed by UAH and a FORTRAN listing of this program is available on request.

One way to overcome the problem of handling large arrays is to divide the original pictures into smaller subpictures and treat each subpicture as a separate picture [10]. This method will result in additional distortions at the boundaries of the subpictures and extreme care must be exercised while dividing the picture into subpictures.

LIST OF REFERENCES

1. W. K. Pratt, "A Bibliography on Television Bandwidth Reduction Studies," IEEE Trans. Information Theory, Vol. IT - 13, No. 1, January 1967.
2. A. Rosenfeld, "Bandwidth Reduction Bibliography," IEEE Trans. Information Theory, " Vol. IT-14, No. 4, July 1968.
3. Proc. IEEE, Special Issue on Redundancy Reduction, Vol. 55, No. 3, March 1967.
4. H. P. Kramer and M. V. Mathews, "A Linear Coding for Transmitting a Set of Correlated Signals," IRE Trans. Information Theory, Vol. IT-2, September 1956.
5. H. C. Andrews, Computer Techniques in Image Processing, Academic Press, New York, 1970.
6. S. A. Sheldahl, "Channel Identification Coding for Data Compression," EASCON Convention Record, 1968.
7. J. W. Cooley, P. A. Lewis, and P. D. Welch, "Historical Notes on the Fast Fourier Transform," Proc. IEEE, Vol. 55, October 1967.
8. H. F. Harmuth, "A Generalized Concept of Frequency and Some Applications," IEEE Trans. Information Theory, Vol. IT-14, No. 3, May 1968.
9. B. R. Hunt, "Computational Considerations in Digital Image Enhancement," Proceedings of the Two-Dimensional Digital Signal Processing Conference, Columbia, Missouri, October 1971.
10. E. J. Claire, S. M. Farber, and R. R. Green, "Practical Techniques for Transform Data Compression/Image Coding," IEEE Trans. Electromagnetic Compatibility, Vol. EMC-13, No. 3, August 1971.
11. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Comput. Vol. 19, No. 90, 1965.
12. W. M. Gentleman and G. Sande, "Fast Fourier Transforms for Fun and Profit," 1966 Fall Joint Computer Conference, AFIPS Proc., Vol. 29, Washington, D. C.:Spartan, 1966.
13. R. J. Polge, et al., Theory and Implementation of Fast Fourier and Hadamard Transforms, UAH Research Report No. , The University of Alabama in Huntsville, Huntsville, Alabama, April 1973.

APPENDIX A

THE FAST FOURIER TRANSFORM

The direct discrete Fourier transform (DFT) of a set X containing N complex samples is defined by the relation

$$x'(k) = \frac{1}{N} \sum_{j=0}^{N-1} w^{-jk} x(j) , \quad k = 0, 1, \dots, N-1 \quad (\text{A.1})$$

where the complex coefficient w is equal to $\exp(\sqrt{-1} 2\pi/N)$. Similarly, the inverse DFT is defined as

$$x(j) = \sum_{k=0}^{N-1} w^{jk} x'(k) , \quad j = 0, 1, \dots, N-1 \quad (\text{A.2})$$

The transformation defined by (A.1) requires N^2 complex multiplications and a direct implementation would be computationally inefficient.

The Fast Fourier Transform (FFT) is an efficient computational algorithm, originated by Cooley and Tukey [11] to compute the DFT. Many algorithms have been developed since. Gentleman and Sande [12] proposed an FFT algorithm which minimizes computer storage by using the same array for the input data and the transformed data. However, the transformed samples are not stored at the proper locations and the transformation must be followed by a permutation called unscrambling.

The algorithm explained here is based on the work of Gentleman and Sande and is useful only for arrays of size N where N is an integral power of two. Using this

algorithm, one can compute the DFT of an array of size $N = 2^M$ with only $\frac{1}{2}MN$ complex multiplications. The algorithm is based on the factorization of N into M factors, each of which is two. The method is illustrated for the case when $N = 8$ and $M = 3$. Using the binary expansion, the addresses j and k can be written as

$$j = 4j_2 + 2j_1 + j_0 \quad (\text{A.3})$$

$$k = 4k_2 + 2k_1 + k_0 \quad (\text{A.4})$$

where j_i and k_i are equal to 0 or 1. Substituting these into equation (A.2) and rearranging it yields

$$\begin{aligned} x(j_2^4 + j_1^2 + j_0) = & \sum_{k_0=0}^1 w^{4j_2 k_0} \sum_{k_1=0}^1 w^{2j_1 (2k_1 + k_0)} \sum_{k_2=0}^1 w^{j_0 (4k_2 + 2k_1 + k_0)} x'(k_2^4 \\ & + k_1^2 + k_0) , \end{aligned} \quad (\text{A.5})$$

where the factors $w^{8j_1 k_2}$, $w^{8j_2 k_1}$ and $w^{16j_2 k_2}$ were deleted because they always remain equal to one. Equation (A.5) is implemented in four steps:

$$x^{(1)}(j_0^4 + k_1^2 + k_0) = w^{j_0 (2k_1 + k_0)} \sum_{k_2=0}^1 w^{4j_0 k_2} x'(k_2^4 + k_1^2 + k_0) \quad (\text{A.6})$$

$$x^{(2)}(j_0^4 + j_1^2 + k_0) = w^{2j_1 k_0} \sum_{k_1=0}^1 w^{4j_1 k_1} x^{(1)}(j_0^4 + k_1^2 + k_0), \quad (\text{A.7})$$

$$x^{(3)}(j_0^4 + j_1^2 + j_2) = \sum_{k_0=0}^1 w^{4j_2 k_0} x^{(2)}(j_0^4 + j_1^2 + k_0), \quad (\text{A.8})$$

$$x(j_2^4 + j_1^2 + j_0) = x^{(3)}(j_0^4 + j_1^2 + j_2) \quad (\text{A.9})$$

where equations (A.6) to (A.8) are three partial transformations and (A.9) shows that a permutation is necessary because the samples are stored at a wrong address.

The transformation is said to be in place because the arrays X' , $X^{(1)}$, $X^{(2)}$, $X^{(3)}$ and X can share the same storage.

This procedure can be extended to a general case when $N = 2^M$ where the transformation is done in M steps and is followed by a permutation. The k^{th} step of the transformation can be shown to be equivalent to

$$x^{(k)}(s + gN_k) = x^{(k-1)}(s + gN_k) + x^{(k-1)}(s + gN_k + S_k)$$

$$x^{(k)}(s + gN_k + S_k) = w^{G_k s} \left[x^{(k-1)}(s + gN_k) - x^{(k-1)}(s + gN_k + S_k) \right] \quad (\text{A.10})$$

where $k = 1, 2, \dots, M$

$$s = 0, 1, \dots, (S_k - 1)$$

$$g = 0, 1, \dots, (G_k - 1)$$

$$N_k = 2^{M+1-k}, S_k = N_k/2 \text{ and } G_k = 2^{k-1}.$$

Equation (A.9) can be generalized to

$$x\left(\sum_{i=0}^{M-1} j_i 2^i\right) = x^{(M)}\left(\sum_{i=0}^{M-1} j_i 2^{M-1-i}\right) \quad (\text{A.11})$$

Equation (A.10) can be implemented so that no additional storage is required.

However, it is necessary to unscramble the array after the M^{th} step. The unscrambling given by equation (A.11) corresponds to a bit reversal. For a detailed study of the transformation algorithm and a variety of algorithms for implementing bit reversal, see reference [13].

APPENDIX B

THE HADAMARD TRANSFORM

B.1 Definition

In 1922 H. Rademacher exhibited a new set of orthogonal functions taking on only the values ± 1 , in the interval $[0, 1]$, which was of use in his study of the general theory of orthogonal functions. However, the Rademacher functions are not complete and there exist other non-trivial functions orthogonal to all of them. The Rademacher functions $\varphi_i(t)$ are defined on the interval $[0, 1]$ as

$$\varphi_0(t) = \begin{cases} 1 & 0 \leq t \leq \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \end{cases} \quad (\text{B.1})$$

$$\varphi_0(t+k) = \varphi_0(t) \text{ with } k \text{ integer} \quad (\text{B.2})$$

$$\varphi_n(t) = \varphi_0(2^n t) \quad 0 \leq t < 1 \quad (\text{B.3})$$

The first three Rademacher functions are shown in Figure B.1. It can be seen that $\varphi_n(t)$ has 2^{n+1} zero crossings in the interval $(0, 1)$.

In 1923, based on independent research, J. L. Walsh published a set of orthogonal functions which is complete in the interval $[0, 1)$. They take the values ± 1 and are similar in oscillation to the trigonometric functions. The set of Walsh functions includes the Rademacher functions. The Walsh functions $\psi_i(t)$

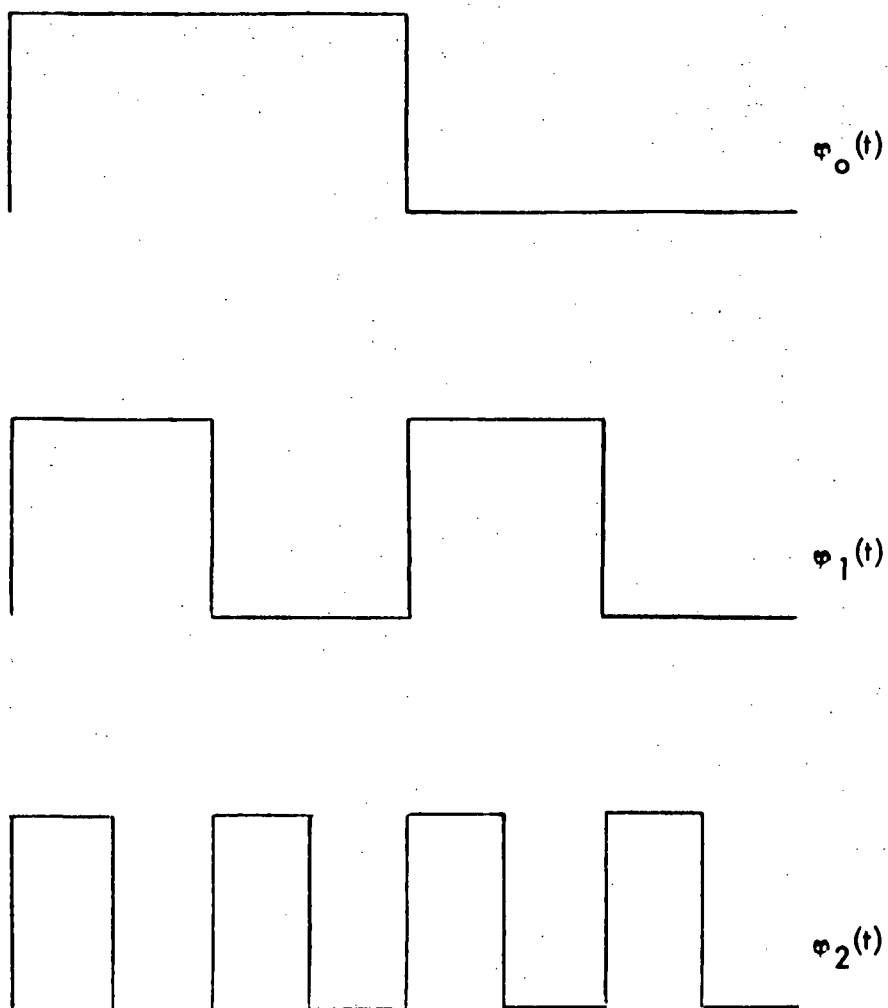


Figure B.1 Rademacher Functions

can be defined in terms of Rademacher functions as

$$\begin{aligned}
 \psi_0(t) &= 1 \\
 \psi_1(t) &= r_0(t) \\
 \psi_2(t) &= r_1(t) \\
 \psi_3(t) &= r_1(t) r_0(t) \\
 &\vdots \\
 \psi_n(t) &= r_{n1}(t) r_{n2}(t) \dots r_{nr}(t)
 \end{aligned} \tag{B.4}$$

where $n = 2^{n1} + 2^{n2} + 2^{n3} + \dots + 2^{nr}$

and $n1 > n2 > n3 \dots$

Thus, the Walsh functions are the products of all possible combinations of Rademacher functions, with the Rademacher functions taken according to the binary representation of n , where n is the order of the Walsh function. From this representation it can be seen that m Rademacher functions can be used to generate a set of $n = 2^m$ Walsh functions. The eight Walsh functions generated from the three Rademacher functions in Figure B.1 are shown in Figure B.2.

A frequency interpretation can be given to the Walsh functions. The word "sequency" was proposed by Harmuth to denote the number of zero crossings in the open interval $(0,1)$. The Walsh functions in Figure B.2 are also labeled according to this notation.

If $wal(k,t)$ denotes the Walsh function of sequency k , then a function $x(t)$ which is square integrable in $[0,T)$ can be expanded as

$$x(t) = \sum_{k=0}^{\infty} a_k \text{wal} \left(k, \frac{t}{T} \right) \text{ for } 0 \leq t < T$$

(B.5)

$$\text{where } a_k = \int_0^T x(t) \text{wal} \left(k, \frac{t}{T} \right) dt .$$

In the case of a discrete function with a sampling interval of τ , the Walsh transform coefficients $\{a_k\}$ become,

$$a_k = \tau \sum_{i=0}^{N-1} x(i\tau) \text{wal} \left(k, \frac{i\tau}{T} \right) \quad (\text{B.6})$$

where $T = N\tau$.

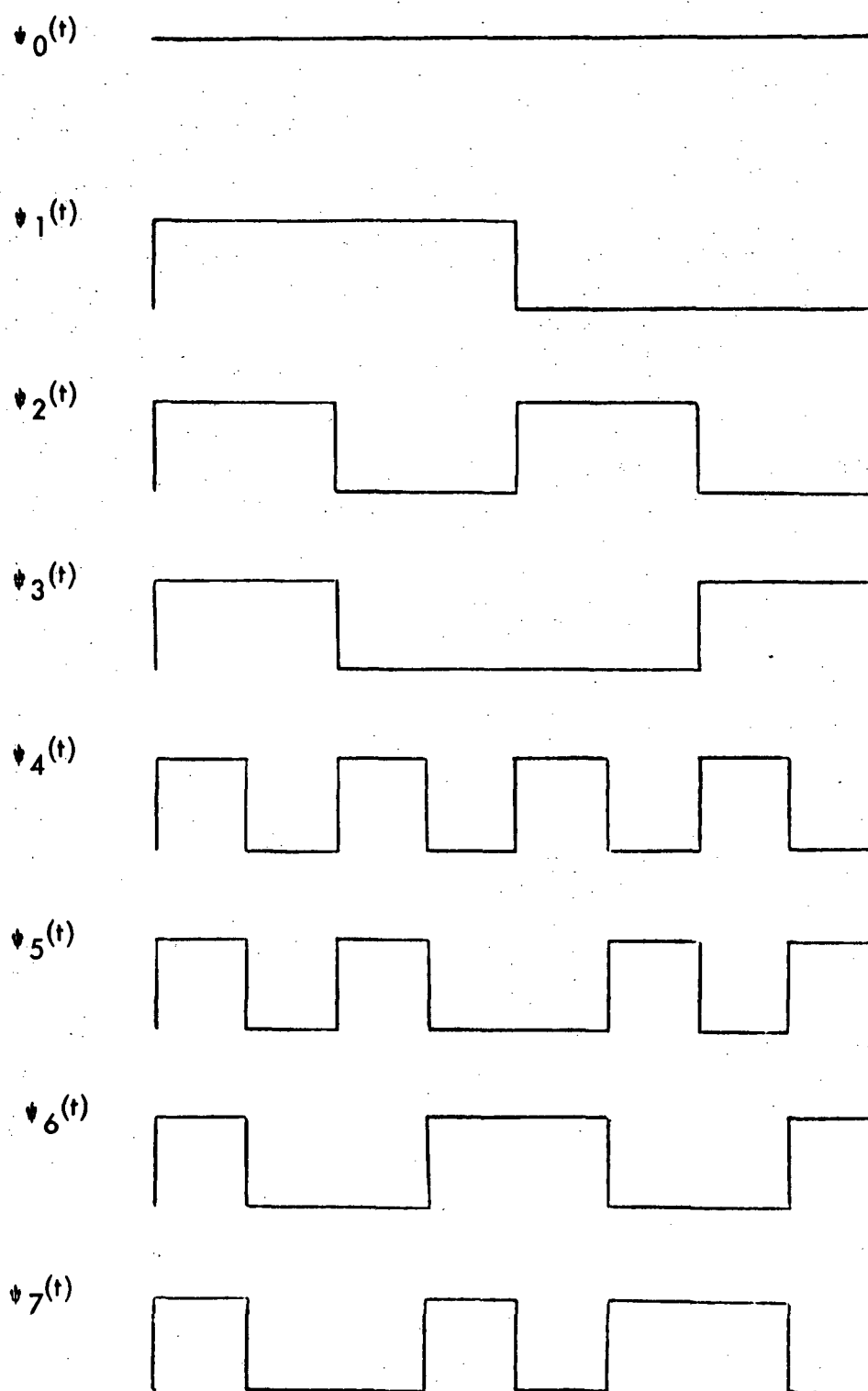
If only the first N terms of the expansion are taken, then the transformation becomes

$$\underline{a} = W \underline{x}$$

where $\underline{a}$ is the set of coefficients, and $\underline{x}$ is the set of sampled function values of the first N Walsh functions.

The Hadamard matrix is an orthogonal matrix made up of plus and minus ones. It is therefore easy to see that the matrix W as defined above is a Hadamard matrix of order N if a Hadamard matrix of order N exists. The matrix W corresponding to $N = 8$ is shown in Figure B.3. The Hadamard transform is the same as the discrete Walsh transform as generated using a Hadamard matrix. Hadamard matrices do not exist for all orders. A necessary (but not sufficient) condition for existence is that N is either equal to 2 or to 0 (mod 4).

Sequency



0

1

3

2

7

6

4

5

Figure B.2 Walsh Functions

B.2 Fast Computation of the Hadamard Transform

In the case when the order $N = 2^m$, where m is an integer, a very elegant procedure is available for the construction of the Hadamard matrix. This is based on the fact that the Hadamard matrix H_N of order N can be generated in a recursive fashion using the relation

$$H_{2K} = \begin{bmatrix} H_K & H_K \\ H_K & -H_K \end{bmatrix} \quad (\text{B.8})$$

$$\text{starting from } H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (\text{B.9})$$

This can be equivalently written as

$$H_N = \underbrace{H_2 \otimes H_2 \otimes \dots \otimes H_2}_{m \text{ products}} \quad (\text{B.10})$$

where $N = 2^m$ and $\otimes$ denotes matrix Kronecker product. This construction procedure can be used to develop a computational algorithm which can transform an array of size N using mN algebraic additions as opposed to N^2 additions for direct matrix multiplication. It should also be noted that a matrix generated in this manner is orthogonal and symmetric and hence is its own inverse except for a scaling factor. Specifically,

$$H_N H_N = NI \quad (\text{B.11})$$

where I is a unit matrix of order N and hence the forward and inverse transformations are the same except for a constant. The Hadamard matrix of order 8

Matrix	Sequency
$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	0
$\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$	1
$\begin{bmatrix} 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix}$	2
$\begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$	3
$\begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$	4
$\begin{bmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$	5
$\begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix}$	6
$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$	7

Figure B.3 Hadamard Matrix of Order 8 (Sequency Ordered)

Matrix	Sequency
$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	0
$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$	7
$\begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$	3
$\begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$	4
$\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$	1
$\begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix}$	6
$\begin{bmatrix} 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix}$	2
$\begin{bmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$	5

Figure B.4 Hadamard Matrix of Order 8 (Generated Using Matrix Kronecker Products)

generated using the above procedure is shown in Figure B.4, along with the sequences corresponding to each row.

The computational procedure to implement the transformation is based on the Kronecker products construction. The transformation is done in m steps for an array of size $N = 2^m$. If the transformation of $\underline{x}$ is defined by

$$\underline{X} = \frac{1}{\sqrt{N}} H_N \underline{x} \quad (\text{B.12})$$

$$\text{and if } \underline{y} = H_N \underline{x}, \quad (\text{B.13})$$

the computation of $\underline{y}$ proceeds as follows:

- (1) $\underline{x}$ is partitioned into two vectors $\underline{x}_1$ and $\underline{x}_2$ of size $N/2$ each.

The first step of the transformation is defined by

$$\underline{x}^1 = H_2 \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} = \begin{bmatrix} \underline{x}_1 + \underline{x}_2 \\ \underline{x}_1 - \underline{x}_2 \end{bmatrix} \quad (\text{B.14})$$

- (2) The array $\underline{x}^1$ is partitioned to two vectors of size $N/2$ each and step (1) is performed on each one of these partitions. Hence if

$$\underline{x}^1 = \begin{bmatrix} 1 \\ \underline{x}_1 \\ \hline 1 \\ \underline{x}_2 \\ \hline 1 \\ \underline{x}_3 \\ \hline 1 \\ \underline{x}_4 \end{bmatrix} \quad (\text{B.15})$$

then the second step of the transformation is defined by

$$\underline{x}^2 = \begin{bmatrix} H_2 \begin{bmatrix} \underline{x}_1^1 \\ \underline{x}_2^1 \\ -2 \end{bmatrix} \\ \hline H_2 \begin{bmatrix} \underline{x}_3^1 \\ \underline{x}_4^1 \\ -4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \underline{x}_1^1 + \underline{x}_2^1 \\ \underline{x}_1^1 - \underline{x}_2^1 \\ \hline \underline{x}_3^1 + \underline{x}_4^1 \\ \underline{x}_3^1 - \underline{x}_4^1 \end{bmatrix} \quad (\text{B.16})$$

(3) This procedure is repeated m times. In the k th step, ($k < m$), the array $\underline{x}^{k-1}$ is partitioned into 2^k partitions as

$$\underline{x}^{k-1} = \begin{bmatrix} \underline{x}_1^{k-1} \\ \hline \underline{x}_2^{k-1} \\ \vdots \\ \hline \underline{x}_{2^k}^{k-1} \end{bmatrix} \quad (\text{B.17})$$

The k th step of the transformation is defined by

$$\underline{x}^k = \begin{bmatrix} H_2 \begin{bmatrix} \underline{x}_1^{k-1} \\ \underline{x}_2^{k-1} \end{bmatrix} \\ \hline H_2 \begin{bmatrix} \underline{x}_3^{k-1} \\ \underline{x}_4^{k-1} \end{bmatrix} \\ \hline \vdots \\ \hline H_2 \begin{bmatrix} \underline{x}_{2^{k-1}}^{k-1} \\ \underline{x}_{2^k}^{k-1} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \underline{x}_1^{k-1} + \underline{x}_2^{k-1} \\ \underline{x}_1^{k-1} - \underline{x}_2^{k-1} \\ \hline \underline{x}_3^{k-1} + \underline{x}_4^{k-1} \\ \underline{x}_3^{k-1} - \underline{x}_4^{k-1} \\ \hline \vdots \\ \hline \underline{x}_{2^{k-1}}^{k-1} + \underline{x}_{2^k}^{k-1} \\ \underline{x}_{2^{k-1}}^{k-1} - \underline{x}_{2^k}^{k-1} \end{bmatrix} \quad (\text{B.18})$$

It can be seen that at each step, a total of N algebraic additions are needed, so that the transformation is achieved in mN algebraic additions.

B.3 Unscrambling

It can be seen from Figure B.4 that an unscrambling has to be done in order to arrange the Hadamard transform coefficients in an ascending order of sequences.

It is known that if $N = 2$ then the sequence numbers of the obtained transform are $(0, 1)$. If $N = 4$, the sequence numbers as they appear after the transformation are $(0, 3, 1, 2)$. This can be generalized to any order. Let the set $(m_1, m_2, \dots, m_N)$ be the sequence numbers for an N element transform and $(k_1, k_2, \dots, k_{2N})$ be the sequence numbers for a $2N$ element transform. Then

$$\left. \begin{array}{l} k_{2i-1} = m_i \\ \text{and } k_{2i} = 2N - 1 - m_i \end{array} \right\} i = 1, 2, \dots, N. \quad (\text{B.19})$$

These relations can be used to generate the sequence numbers and then arrange the transform values in the required order by using a sorting process. However, this requires additional storage. A recursive method to unscramble an array of size $2N$ is as follows:

(1) Rearrange the array by collecting alternate elements. That is, the sequence numbers are arranged as

$$(k_1, k_3, \dots, k_{2N-1}; k_2, k_4, \dots, k_{2N})$$

This can be done with no extra storage by careful manipulation.

(2) Partition into two arrays of equal size and interchange every alternate element of the second array with its succeeding element yielding a sequence number arrangement of

$$(k_1, k_3, \dots, k_{2N-1}; k_4, k_2, \dots, k_{2N}, k_{2N-2})$$

By using the relation between the set $\{k_i\}$ and $\{m_i\}$ given above, it can be shown that the sequence number arrangement is equivalent to

$$(m_1, m_2, \dots, m_N; N + m_1, N + m_2, \dots, N + m_N)$$

(3) The procedure is now repeated on the two halves of the array treating each as a separate N -dimensional array. The process is continued until the size of each partitioned array is two.

B.4 Inverse Transformation

The inverse transform of an array $\underline{X}$ is given by

$$x = \frac{1}{\sqrt{N}} H_N \underline{X} \quad (B.20)$$

which is defined by the same relation as the direct transform. This is due to the fact that the matrix H_N is a symmetric orthogonal operator. However, if the array $\underline{X}$ were unscrambled, it is necessary to scramble it before implementing equation (B.20). The scrambling operation is the precise opposite of the unscrambling operation and the same algorithm as that of unscrambling, but a reverse order can be used.

APPENDIX C

LISTINGS OF COMPUTER PROGRAMS

This appendix contains listings of all the programs described in Chapter 4.

These include the main programs for truncation and coarse quantization with FFT and Hadamard transform, and subroutines CMPHM1, CMPHM2, CMPFT1, CMPFT2, FFTON2, SCHMD2, XREAD and XWRITE.

All the programs are coded in FORTRAN IV and adequate comments are included where necessary.

Subroutine XWRITE

```

10 SUBROUTINE XWRITE(N)
20 .....
30 SUBROUTINE FOR WRITING ON THE OUTPUT TAPE
40 .....
50 OUTPUT TAPE IS ON UNIT # 9
60 .....
70 COMMON/BLKA/X(256,256)
80 DIMENSION XA(256)
90 DO 1 IREC=1,N
100 DO 2 J=1,N
110 XA(J)=X(IREC,J)
120 WRITE(9)(XA(J),J=1,N)
130 CONTINUE
140 END FILE 9
150 RETURN
160 END

```

Subroutine XREAD

```

10 SUBROUTINE XREAD(N)
20 .....
30 SUBROUTINE FOR READING OFF THE INPUT TAPE
40 .....
50 INPUT TAPE IS ON UNIT # 3
60 .....
70 COMMON/BLKA/X(256,256)
80 DIMENSION XA(256)
90 DO 1 IREC=1,N
100 READ(3)(XA(J),J=1,N)
110 DO 2 J=1,N
120 X(IREC,J)=XA(J)
130 CONTINUE
140 RETURN
150 END

```

Subroutine SCHMD2

```

10 SUBROUTINE SCHMD2(N,M,KOPT)
20 .....
30 TWO DIMENSIONAL HADAMARD TRANSFORM
40 .....
50 .....
60 C KOPT=1 == SCRAMBLE AND INVERT
70 C KOPT=2 == TRANSFORM OR INVERT
80 C KOPT=3 == TRANSFORM AND UNSCRAMBLE
90 .....
100 .....
110 C X IS THE ARRAY, M IS THE NUMBER OF BITS
120 C AND N IS THE NUMBER OF ELEMENTS IN THE ARRAY
130 .....
140 COMMON/BLKA/X(256,256)
150 IF(KOPT.NE.1) GO TO 32
160 DO 30 LL=1,2
170 MBIT=2
180 KSIZE=4
190 DO 1 KBIT=2,M
200 KHALF=KSIZE/2
210 KHALF1=KHALF+1
220 DO 2 K=KHALF1,KSIZE,2
230 DO 2 K1=K,N,KSIZE
240 DO 2 I=1,N
250 DUM=X(I,K1)
260 X(I,K1)=X(I,K1+1)
270 X(I,K1+1)=DUM
280 JUMP=KSIZE/4
290 KSPAN=KSIZE
300 DO 11 MBIT=2,MBIT
310 KST1=JUMP+1
320 DO 12 KST=KST1,KSIZE,KSPAN

```

Subroutine SCHMD2 (continued)

```

33• KSINMX=KST+JUMP-1
34• DO 12 KSIN=KST,KSINMX
35• DO 12 KSING=KSIN,N,KSIZE
36• DO 12 I=1,N
37• DUM=X(I,KSING)
38• X(I,KSING)=X(I,KSING+JUMP)
39• X(I,KSING+JUMP)=DUM
40• JUMP=JUMP/2
41• KSPAN=KSPAN/2
42• CONTINUE
43• KSIZE=KSIZE+KSIZE
44• MBIT=MBIT+1
45• CONTINUE
46• DO 31 I=1,N
47• DO 31 J=1,I
48• DUM=X(I,J)
49• X(I,J)=X(J,I)
50• X(J,I)=DUM
51• CONTINUE
52• DO 23 LL=1,2
53• NSPAN=N
54• NDIST=NSPAN/2
55• DO 20 I=1,M
56• DO 21 L=1,NDIST
57• JM=L+N-NSPAN
58• DO 21 J=L,JM,NSPAN
59• DO 21 K=1,N
60• DUM=X(K,J)
61• X(K,J)=DUM+X(K,J+NDIST)
62• X(K,J+NDIST)=DUM-X(K,J+NDIST)
63• CONTINUE
64• NSPAN=NDIST
65• NDIST=NSPAN/2

```

Subroutine SCHMD2 (continued)

```

66• 20 CONTINUE
67•   DO 22 I=1,N
68•   DO 22 J=1,I
69•   DUM=X(I,J)
70•   X(I,J)=X(J,I)
71•   X(J,I)=DUM
72• 22 CONTINUE
73•   DO 42 I=1,N
74•   DO 42 J=1,I
75•   X(I,J)=X(I,J)/FLOAT(N)
76• 42 CONTINUE
77•   IF(KOPT.NE.3) RETURN
78•   DO 40 LL=1,2
79•   KSIZE=N
80•   MBIT=M
81•   DO 3 KBIT=2,M
82•   KHALF=KSIZE/2
83•   KHALF1=KHALF+1
84•   JUMP=1
85•   KSPAN=4
86•   DO 13 NBIT=2,MBIT
87•   KST1=JUMP+1
88•   DO 14 KST=KST1,KSIZE,KSPAN
89•   KSINMX=KST+JUMP-1
90•   DO 14 KSIN=KST,KSINMX
91•   DO 14 KSING=KSIN,N,KSIZE
92•   DO 14 I=1,N
93•   DUM=X(I,KSING)
94•   X(I,KSING)=X(I,KSING+JUMP)
95• 14 X(I,KSING+JUMP)=DUM
96•   JUMP=JUMP+JUMP
97•   KSPAN=KSPAN+KSPAN
98• 13 CONTINUE
99•   DO 4 K=KHALF1,KSIZE,2
100•  DO 4 K1=K,N,KSIZE

```

Subroutine SCHMD2 (continued)

```

101• DO 4 I=1,N
102• DUM=X(I,K1)
103• X(I,K1)=X(I,K1+1)
104• X(I,K1+1)=DUM
105• KSIZE=KHALF
106• MBIT=MBIT-1
107• CONTINUE
108• DO 41 I=1,N
109• DO 41 J=1,I
110• DUM=X(I,J)
111• X(I,J)=X(J,I)
112• X(J,I)=DUM
113• CONTINUE
114• RETURN
115• END

```

4

3

41
40

Subroutine FFTON2

```

10 SUBROUTINE FFTON2(N,L,KRZIN,KRZOUT,KUNSCR)
20 DOUBLE PRECISION TPI,DPCON,DELANG,SDLANG,COLANG,SINANG,
30 ICOSANG,CHOLD
40 COMMON/BLKA/XR(256,256)
50 COMMON/BLKB/XI(256,256)
60 DIMENSION IX(128)
70 TPI=6.28318530717958648
80 M=IABS(L)
90 EPSFI=1.5E-7
100 EPSMO=2.E-7
110 IF(M.EQ.0)RETURN
120 N2=N**2
130 CON2=1./FLOAT(N2)
140 IF(KRZIN.EQ.2) CALL MPRI
150 DO 31 LL=1,2
160   NHALF=N/2
170   NHALFP=NHALF+1
180   DPCON=FLOAT(N)
190   DELANG=TPI/DPCON
200   IF(L.LT.0) DELANG=-DELANG
210   NSPAN=N
220   NDIST=NSPAN/2
230   CDLANG=DCOS(DELANG)
240   SDLANG=DSIN(DELANG)
250   DO 2 I=1,M
260     COSANG=1.
270     SINANG=0.
280     DO 3 J=1,NDIST
290       YR=COSANG
300       YI=SINANG
310     DO 5 K=J,N,NSPAN
320       KANDS=K+NDIST
330     DO 5 IR=1,N
340       HOLDR=XR(IR,K)-XR(IR,KANDS)
350       HOLDI=XI(IR,K)-XI(IR,KANDS)
360       XR(IR,K)=XR(IR,K)+XR(IR,KANDS)

```

Subroutine FFTON2 (continued)

```

37• XI(IR,K)=XI(IR,K)+XI(IR,KANDS)
38• XR(IR,KANDS)=TR*HOLDR-YI*HOLDI
39• XI(IR,KANDS)=YI*HOLDR+TR*HOLDI
40• 5 CONTINUE
41• CHOLD=COSANG
42• COSANG=COSANG*CDLANG-SINANG*SDLANG
43• SINANG=SINANG*CDLANG+CHOLD*SDLANG
44• 3 CONTINUE
45• NSPAN=NDIST
46• NDIST=NSPAN/2
47• CHOLD=CDLANG
48• CDLANG=CDLANG*CDLANG-SDLANG*SDLANG
49• SDLANG=CHOLD*SDLANG+CHOLD*SDLANG
50• 2 CONTINUE
51• DO 32 I=1,N
52• DO 32 J=1,I
53• DUMR=XR(I,J)
54• DUMI=XI(I,J)
55• XR(I,J)=XR(J,I)
56• XI(I,J)=XI(J,I)
57• XR(J,I)=DUMR
58• XI(J,I)=DUMI
59• 31 CONTINUE
60• IF(L.GT.0)GO TO 23
61• GO TO (18,19),KRZOUT
62• 18 DO 1 J=1,N
63• DO 1 I=1,N
64• XR(I,J)=XR(I,J)*CON2
65• 1 XI(I,J)=XI(I,J)*CON2
66• GO TO 21
67• 19 DO 20 J=1,N
68• DO 20 I=1,N
69• HOLDR=XR(I,J)*CON2
70• HOLDI=XI(I,J)*CON2
71• XR(I,J)=SQRT(HOLDR**2+HOLDI**2)
72• IF(XR(I,J).GT.EPSMO) GO TO 26
73• XI(I,J)=0.0

```

Subroutine FFTON2 (continued)

```

740 GO TO 20
750 26 IF (ABS(HOLDR),LT.EPSF1)HOLDR=SIGN(EPSF1,HOLDR)
760 X1(I,J)=ATAN2(HOLDI,HOLDR)
770 20 CONTINUE
780 21 CONTINUE
790 GO TO 4
800 23 IF (KRZOUT.EQ.2)CALL RIMP
810 4 IF (M.EQ.1)RETURN
820 IF (KUNSCR.EQ.0)RETURN
830 M2=M/2
840 M3=M-2*M2
850 K2M2=2*M2
860 K2MX=K2M2
870 IF (M3.EQ.1) K2MX=K2MX+K2MX
880 IX(1)=0
890 JLIH=1
900 DO 13 I=1,M2
910 DO 14 J=1,JLIH
920 IX(J)=IX(J)+IX(J)
930 IX(JLIH+J)=IX(J)+K2MX
940 JLIH=JLIH+JLIH
950 KUMX=K2M2-1
960 DO 41 LL=1,2
970 IF (M3.EQ.1) GO TO 10
980 DO 11 KU=1,KUMX
990 KVMN=KU+1
1000 DO 11 KV=KVMN,K2M2
1010 KPI=IX(KU) + KV
1020 KP2=IX(KV) + KU
1030 DO 15 I=1,N
1040 XSR=XR(I,KPI)
1050 XSI=XI(I,KPI)
1060 XR(I,KPI)=XR(I,KP2)
1070 XI(I,KPI)=XI(I,KP2)
1080 XR(I,KP2)=XSR
1090 XI(I,KP2)=XSI
1100 15 CONTINUE

```

Subroutine FFTON2 (continued)

```

1110 11 CONTINUE
1120 GO TO 33
1130 10 DO 12 KU=1,KUMX
1140 KVMN=KU+1
1150 DO 12 KV=KVMN,K2M2
1160 KP1=IX(KU) * KV
1170 KP2=IX(KV) * KU
1180 DO 16 I=1,N
1190 XSR=XR(I,KP1)
1200 XSI=XI(I,KP1)
1210 XR(I,KP1)=XR(I,KP2)
1220 XI(I,KP1)=XI(I,KP2)
1230 XR(I,KP2)=XSR
1240 XI(I,KP2)=XSI
1250 16 CONTINUE
1260 KP1=KP1+K2M2
1270 KP2=KP2+K2M2
1280 DO 17 I=1,N
1290 XSR=XR(I,KP1)
1300 XSI=XI(I,KP1)
1310 XR(I,KP1)=XR(I,KP2)
1320 XI(I,KP1)=XI(I,KP2)
1330 XR(I,KP2)=XSR
1340 XI(I,KP2)=XSI
1350 17 CONTINUE
1360 12 CONTINUE
1370 33 CONTINUE
1380 DO 42 I=1,N
1390 DO 42 J=1,I
1400 DUMR=XR(I,J)
1410 DUMI=XI(I,J)
1420 XR(I,J)=XR(J,I)
1430 XI(I,J)=XI(J,I)
1440 XR(J,I)=DUMR
1450 XI(J,I)=DUMI
1460 42 CONTINUE
1470 41 RETURN

```

Subroutine FFTON2 (continued)

```

148• SUBROUTINE RIMP
149• DO 22 J=1,N
150• DO 22 I=1,N
151•   HOLDR=XR(I,J)
152•   XR(I,J)=SQRT(HOLDR**2+XI(I,J)**2)
153•   IF(XR(I,J).GT.EPSH0) GO TO 27
154•   XI(I,J)=0.0
155•   GO TO 22
156• 27 IF(ABS(HOLDR).LT.EPSF1)HOLDR=SIGN(EPSF1,HOLDR)
157•   XI(I,J)=ATAN2(XI(I,J),HOLDR)
158• 22 CONTINUE
159• RETURN
160• SUBROUTINE MPRI
161• DO 25 J=1,N
162• DO 25 I=1,N
163•   HOLDR=XR(I,J)
164•   XR(I,J)=HOLDR*COS(XI(I,J))
165•   XI(I,J)=HOLDR*SIN(XI(I,J))
166• 25 RETURN
167• END

```

Main Program for Truncation with Hadamard Transform

```

10 C .....
20 C MAIN PROGRAM FOR TRUNCATION WITH HADAMARD TRANSFORM
30 C .....
40 C INPUT TAPE IS ON UNIT # 3
50 C OUTPUT TAPE IS ON UNIT # 9
60 C .....
70 C COMMON/BLKA/X(256,256)
80 C READ 100,N,M,NTRUNC
90 C CALL XREAD(N)
100 C CALL SCHMD2(N,M,3)
110 C CALL CMPHM1(NTRUNC,N,M)
120 C PRINT 102,NTRUNC
130 C CALL XWRITE(N)
140 C FORMAT(3110)
150 C FORMAT(110)
160 C STOP
170 C END

```

Subroutine CMPHM1

```

10 SUBROUTINE CMPHM1(NTRUNC,N,M)
20 .....
30 SUBROUTINE FOR TRUNCATION WITH HADAMARD TRANSFORM
40 .....
50 COMMON/BLKA/X(256,256)
60 NA=NTRUNC+1
70 DO 1 I=1,N
80 DO 1 J=NA,N
90 X(I,J)=0.0
100 X(J,I)=0.0
110 CALL SCHMD2(N,M,1)
120 RETURN
130 END

```

Main Program for Coarse Quantization with Hadamard Transform

```

10 C .....
20 C MAIN PROGRAM FOR COARSE QUANTIZATION WITH HADAMARD TRANSFORM
30 C .....
40 C INPUT TAPE IS ON UNIT # 3
50 C OUTPUT TAPE IS ON UNIT # 9
60 C .....
70 C COMMON/BLKA/X(256,256)
80 C READ 100,N,M,KZERO,XCOEFF
90 C CALL XREAD(N)
100 C CALL SCHMD2(N,M,3)
110 C S=0.0
120 C DO 4 I=1,N
130 C   DO 4 J=1,N
140 C     S=S+ABS(X(I,J))
150 C   XMEAN=S/FLOAT(N**2)
160 C   TRSOLD=XCOEFF*XMEAN
170 C   CALL CHPHM2(JN,N,M,IRUN,TRSOLD,KZERO)
180 C   PRINT 102,XCOEFF,JN,IRUN
190 C   CALL XWRITE(N)
200 C   FORMAT(3110,F10.5)
210 C   102 FORMAT(F10.5,2110)
220 C   STOP
230 C   END

```

Subroutine CMPHM2

```

10 SUBROUTINE CMPHM2(JN,N,M,IRUN,TRSOLD,KZERO)
20 .....
30 SUBROUTINE FOR COARSE QUANTIZATION WITH HADAMARD TRANSFORM
40 .....
50 COMMON/BLKA/X(256,256)
60 DIMENSION IZERO(8)
70 DO 6 I=1,8
80 IZERO(I)=0
90 JN=0
100 DO 2 I=1,N
110 K=0
120 ICOL=1
130 IF(ABS(X(I,ICOL)).GT.TRSOLD) GO TO 7
140 K=K+1
150 X(I,ICOL)=0.0
160 JN=JN+1
170 IF(K.EQ.KZERO) GO TO 3
180 ICOL=ICOL+1
190 IF(ICOL.GT.N) GO TO 9
200 IF(ABS(X(I,ICOL)).LE.TRSOLD) GO TO 8
210 IZERO(K)=IZERO(K)+1
220 K=0
230 ICOL=ICOL+1
240 IF(ICOL.GT.N) GO TO 2
250 GO TO 4
260 IZERO(K)=IZERO(K)+1
270 2 CONTINUE
280 IRUN=0
290 DO 10 I=1,KZERO
300 IRUN=IRUN+IZERO(I)
310 CALL SCHMD2(N,M,I)
320 RETURN
330 END

```

Main Program for Truncation with the FFT

```

10 C .....
20 C MAIN PROGRAM FOR TRUNCATION WITH THE FFT
30 C .....
40 C INPUT TAPE IS ON UNIT # 3
50 C OUTPUT TAPE IS ON UNIT # 9
60 C .....
70 COMMON/BLKA/XR(256,256)
80 COMMON/BLKB/XI(256,256)
90 READ 100,N,M,NTRUNC
100 CALL XREAD(N)
110 DO 1 I=1,N
120 DO 1 J=1,N
130 1 XI(I,J)=0.0
140 CALL FFTON2(N,-M,1,2,1)
150 CALL CHFFT1(NTRUNC,N,M)
160 PRINT 102,NTRUNC
170 CALL XWRITE(N)
180 100 FORMAT(3I10)
190 102 FORMAT(1I0)
200 STOP
210 END

```

Subroutine CMPFT1

```

1• SUBROUTINE CMPFT1(NTRUNC,N,M)
2• .....
3• C SUBROUTINE FOR TRUNCATION WITH THE FFT
4• C .....
5• COMMON/BLKA/XR(256,256)
6• COMMON/BLKB/XI(256,256)
7• NA=NTRUNC+1
8• NB=N-NTRUNC
9• DO 1 I=NA,NB
10• DO 1 J=1,N
11• XR(I,J)=0.0
12• XI(I,J)=0.0
13• XR(J,I)=0.0
14• XI(J,I)=0.0
15• CALL FFTON2(N,M,2,1,1)
16• RETURN
17• END

```

Main Program for Coarse Quantization with FFT

```

1000 .....
2000 MAIN PROGRAM FOR COARSE QUANTIZATION WITH THE FFT
3000 .....
4000 INPUT TAPE IS ON UNIT # 3
5000 OUTPUT TAPE IS ON UNIT # 9
6000 .....
7000 COMMON/BLKA/XR(256,256)
8000 COMMON/BLKB/XI(256,256)
9000 READ 100,N,M,KZERO,XCOEFF
10000 CALL XREAD(N)
11000 DO 1 I=1,N
12000 DO 1 J=1,N
13000 XI(I,J)=0.0
14000 CALL FFTON2(N,-M,1,2,1)
15000 S=0.0
16000 DO 4 I=1,N
17000 DO 4 J=1,N
18000 S=S+XR(I,J)
19000 XMEAN=S/FLOAT(N**2)
20000 TRSOLD=XCOEFF*XMEAN
21000 CALL CHFFT2(JN,N,M,IRUN,TRSOLD,KZERO)
22000 PRINT 102,XCOEFF,JN,IRUN
23000 CALL XWRITE(N)
24000 FORMAT(3I10,F10.5)
25000 FORMAT(F10.5,2I10)
26000 STOP
27000 END
100
102

```

Subroutine CMPFT2

```

10 SUBROUTINE CMPFT2(JN,N,M,IRUN,TR SOLD,KZERO)
20 .....
30 SUBROUTINE FOR COARSE QUANTIZATION WITH THE FFT
40 .....
50 COMMON/BLKA/XR(256,256)
60 COMMON/BLKB/XI(256,256)
70 DIMENSION IZERO(8)
80 DO 6 I=1,8
90 IZERO(I)=0
100 JN=0
110 DO 2 I=1,N
120 K=0
130 ICOL=1
140 IF(XR(I,ICOL).GT.TR SOLD) GO TO 7
150 K=K+1
160 XR(I,ICOL)=0.0
170 XI(I,ICOL)=0.0
180 JN=JN+1
190 IF(K.EQ.KZERO) GO TO 3
200 ICOL=ICOL+1
210 IF(ICOL.GT.N) GO TO 9
220 IF(XR(I,ICOL).LE.TR SOLD) GO TO 8
230 IZERO(K)=IZERO(K)+1
240 K=0
250 ICOL=ICOL+1
260 IF(ICOL.GT.N) GO TO 2
270 GO TO 4
280 IZERO(K)=IZERO(K)+1
290 2 CONTINUE
300 IRUN=0
310 DO 10 I=1,KZERO
320 IRUN=IRUN+IZERO(I)
330 CALL FFTON2(N,M,2,1,1)
340 RETURN
350 END

```

PART III

SUGGESTIONS FOR FUTURE EFFORTS

In this study, transform data compression techniques were investigated from the point of view of compression versus distortion. Since the selection of the method of compression and the compression ratio depends upon the type of picture and the information sought, further work is necessary to specify the most efficient data compression technique for each application.

One problem area in digital processing of pictures is the amount of storage required. For example, if a picture of size 50 x 50 mm is scanned at intervals of 50 microns, the size of the corresponding array is one million. This means that a peripheral storage device must be used which results in a considerable increase in computation time because of the many transfers required between the core and the peripheral device. The following three approaches are suggested to alleviate this problem.

- (1) A computer program which minimizes the number of transfers during the transformation can be developed.
- (2) Since most scanners define only 64 levels, it is likely that a full computer word is not required to store one image sample. It will be useful to study the minimum number of bits required during and after transformation for a specified accuracy. In this manner, the computer core will be used more efficiently.

- (3) It is likely that a hybrid processing including both optical and digital processing will be more efficient. This could be implemented with very few changes to the existing ERTS system.

Since the image data are contaminated by noise, data filtering and processing should be very useful. It is proposed to study both non-recursive and recursive filtering and enhancement techniques. The use of Fast Fourier transform and Hadamard transform are suggested for non-recursive filtering.

A very important area in image processing is information retrieval and pattern recognition. It is recommended to study the application of transform techniques to pattern recognition as well as maximum likelihood and other standard techniques.

Much work remains to be done on the selection and specification of sensors and of sensor mixes. For example, it was shown that the existing sensors were not adequate for rangeland management. We suggest that a comprehensive study be made of optimum selection of sensors for all planned applications.