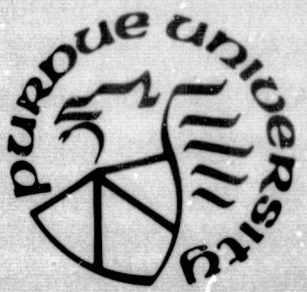
NSG-5010

DATA COMPRESSION FOR SATELLITE IMAGES

Po Hsiin Chen
and
Paul A. Wintz



School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

TR-EE 77-9
December, 1976

This work was supported by NASA Contract No. NSG 5010 and by ARPA Contract No. F30602-75-C-0150.

FINAL REPORT

DATA COMPRESSION FOR SATELLITE IMAGES

by

Po Hsiin Chen

and

Paul A. Wintz

TR-EE 77-9

December, 1976

School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

This work was supported by NASA Contract No. NSG 5010 and by
ARPA Contract No. F30602-75-C-0150.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

α^2	= variance
C	= class of codes, class number, or total picture complexity
C(i)	= cosine transform of U(ℓ)
C_i	= picture complexity of the figure i
Δ's	= indices resulting from DDC and PDQ
e	= error, or base of the natural logarithm
E_i^2	= squared error
E( )	= expectation value of
f( )	= function of
h	= entropy per pixel
L	= run length
m	= number of bits to represent grey levels
M_i	= mapping i
m_i	= number of runs
N_i	= number of index pairs
P	= prefix of a code word, or perimeter of a figure
P(I)	= probability density function of I
P_i	= probability of the data falls in the ith interval
q_i	= quantizing error
R	= data compression ratio
R_{ij}	= covariance of the ith and jth pixel
S	= suffix of a code word or sequence of digits

S_i = end points of an object in a scan line

S^2 = signal power

T = threshold value

U_i = transition level i

V_i = quantization level i

W = code word consisting of a sequence of bits

W_i = word count

$X(t)$ = stationary signal

X^T = transpose of a measurement vector for a pixel

X_{ij} = grey level of a pixel in a two dimensional array

Y_{ij} = grey level of a pixel in a reconstructed two-dimensional array

Z = binary representation of an integer

ABSTRACT

An efficient data compression system is presented for satellite pictures and two grey level pictures derived from satellite pictures. The compression techniques take advantage of the correlation between adjacent picture elements.

Several source coding methods are investigated. Double delta coding is presented and shown to be the most efficient. Both predictive differential quantizing technique and double delta coding can be significantly improved by applying a background skipping technique.

We have investigated several classes of codes to represent the derived data resulting from source coding techniques. An extension code is constructed. This code requires very little storage space and operates efficiently. Simulation results are presented for various coding schemes and source codes.

A method is presented to measure the complexity of pictures. Data compression ratios can be predicted from the measurement of picture complexities.

A compressed data format is designed to pack the encoded derived data into a more compact form for better data compression. This format also protects the data so that the number of objects in the picture can be decoded correctly.

Three efficient coding techniques for satellite pictures are investigated. All three can be used in a nearly real time, multi-rate data transmission system. One of them is a modified DPCM system where various codes and quantization schemes are used. A hybrid coding combining DPCM and a transformation technique is also presented. This technique minimized the rms error. The linear interpolation technique is also examined. The advantage of this technique is its low cost of implementation. Performance comparisons among these techniques are made.

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

The discrete multispectral scanner data are sampled and quantized into a fixed number of bits for each picture element. This is referred to as a satellite picture, which usually requires 6 to 8 binary digits per picture element (pixel). Some two-level and multi-level pictures were obtained from picture classification on satellite pictures.

Due to the redundancy in the pictures, various data compression techniques were developed to reduce the data bulk to be transmitted. In this chapter, we make a technical review of the literature, followed by the introduction to the objective of this thesis.

1.1 Technical Review of the Literature

People have attempted to acquire information from measurements made by a satellite at a distance above the earth. This major area of research provides a means to predict and control man's environment [1,2]. The data is collected by a sensor mounted on the aerospace platform. The reflected energy in certain regions of the electromagnetic spectrum are collected and digitized to form multispectral data [3].

To extract information from the multispectral data, and to interpret it as certain useful objects, one has to relate the properties of the object with the measurements on the spectral, spatial, or temporal domains.

Basically, there are two approaches toward the classification of digitized imagery; namely, statistical and linguistic approaches. More research has been done on the statistical approach than on the linguistic approach. The advantage of the statistical approach is its generality, although complete statistical information of the objects is impossible. Most of linguistic methods transform input pictures into line drawings or regions [4,5], and apply natural or artificial language processing. Haralick, Shanmugan and Dinstein [6] investigated the idea of using textural features for image classification where the spatial relationship between picture elements was used to train the classifier [7].

Wilkins [8] and Gupta [9] attempted to extend the blob algorithm to find the natural boundary. But two picture elements belonging to the same class may belong to different populations in certain channels. Thus, blob algorithm may give misclassification in this circumstance.

A subset of multispectral measurements is found to contain most of the information which would provide excellent classification results. Feature selection techniques can be used to find the best combination of channels from the original N channels [10,11].

LARS, Purdue University [12], developed a multispectral data analysis system. Based on the assumption of Gaussian distributed data, the clustering analysis [13,14,15], was used to decompose the non-Gaussian density function into several Gaussian density clusters. This will provide the statistics for training classes corresponding to each cluster and can be repeated to refine the training samples [16].

The table look-up algorithm for picture classification [17] is an improved technique. This approach has reduced the processing time significantly compared with the approach by LARS on the same computer.

Many research projects have been conducted on the data compression for binary and continuous-tone pictures. The process of scanning reduces a picture from a two-dimensional array of pixels to a one-dimensional sequence of pixels, which can be shown to be a first order Markov process [18,19]. Later Habibi [20] and others [21,22] extended the model to a two-dimensional random field.

Two-level and multi-level pictures are characterized by sharp edges due to limited number of grey levels. Thus, they often can be compressed with no degradation. Run length coding [23] and the scheme suggested by Huang [24] were developed to compress the binary pictures such as text and weather maps where the figures are mostly line drawings. Run length coding takes into account only horizontal correlations in the scan line. The two-dimensional run length coding known as predictive differential quantization (PDQ) [25] was developed to take into account both horizontal and vertical correlations.

Bit-plane coding [26,27] was used to encode the multi-level pictures. This algorithm transfers a n-bit per pel picture into n pictures of 1 bit per pel, followed by the run length coding (RLC), et.al.. Thus, the same coding process of RLC was repeated n times. The data compression achieved is not significantly better than other schemes. Indeed, it really depends on the complexity of objects.

Because of the complexity of the continuous-tone pictures, any error free coding technique can not provide significant data compression. Those coding schemes which result in certain amount of distortion had been investigated. Kretzmer [28] and Harrison [29] exploited the correlation in the video signal in order to reduce the average power required to transmit it, and found that a substantial reduction in the average power

could be obtained using just the previous element, little further reduction was possible using more of the adjacent elements on the same and preceding lines. Schreiber [30] measured the second and third order statistics of a scan line and found considerable correlations between adjacent elements.

Piecewise-linear approximation for interpolative coding was investigated by Youngblood [31]. By dividing the picture into small blocks, Cunningham [32] represented the grey level of each block by its average grey level. Graham [33] regarded all points with the gradient higher than a threshold as edges. By transmitting changes in contour directions, gradient [34] directions, and magnitude using Huffman codes [35], pictures can be reconstructed at the receiving end. Later, contour coding was used with the blob algorithm [8,9]. However, its shortcomings are the large memory space for storing the whole frame and relative complexity of the IP and T algorithms.

DPCM is based on the invention of Cutler [36]. The differences between the prediction and the signal are quantized into one of m grey levels and a code is transmitted to indicate which of the m levels occurred. The quantization scale could be either uniform or nonuniform. Theoretical studies of the nonuniform quantizer had led to Max's [37] criteria for minimizing the rms quantization distortion. The most desirable characteristics of DPCM is the ease of design and the speed of the operation. The limitation of this system is its sensitivity to the picture statistics and the propagation of the channel error.

The transform coding [38-41] was designed to create a domain in which the data is uncorrelated and the signal energy is compacted

into a small number of components. The typical transform coding includes: Fourier, Hadamard, Hotelling and linear transform. The advantage of the transform coding is its superior coding performance at lower bit rates and the degradation is less objectionable to a human viewer, in addition to its less sensitivity to the data statistics and channel errors. But this scheme usually requires a large memory space during the processing.

An extensive survey of published papers and reports dealing with data compression, picture properties, picture coding and transmission can be found in Wilkins and Wintz [42], Huang, et.al. [43], and Connor, et.al. [45].

After the redundancy in the digitized picture is removed by some coding scheme, efficient source codes are used to encode the derived data of the coding schemes such that they can be transmitted over the communication channel.

There are two types of codes - fixed length codes and variable length codes. The natural code and Gray code [46] are two typical fixed length codes. In the natural code, the numbers are represented by their binary representations. Gray code is an inverted binary code. The advantage of the fixed length code is that the code length of each code word remains the same which makes the design of the data format a lot easier. It also has a drawback that the redundancy still exists in the code words due to nonuniform distribution of the data.

The variable length codes are designed to map shorter code words to the data with higher probability of occurrence, and long code words to rarely occurring data. The Huffman code [35] is known to be the optimal variable length code, but it is not practical for coding purposes.

Huang, et.al. developed a code to encode the run lengths for run length coding of binary pictures. The prefix code [47] is another variable length code, but it is not suitable to encode the run lengths and the data where their values could be an integer.

In general, certain codes may be good only for certain type of data. To match the code words with the statistics of the data is the key in designing the efficient source codes. For more theoretical treatment of the coding aspects, Stiffler [47] and Gallager [48] are good references.

1.2 The Objective and Contributions

The objective of this research is to investigate an efficient data compression system for both continuous tone satellite pictures and those pictures with two or more grey levels. A block diagram of the data compression system is shown in Figure 1.1. The system is divided into two subsystems for compression and decompression purposes.

A digitized picture is processed by the redundancy remover to achieve the first level of data compression. The output of the redundancy remover is called the derived data. Some efficient source codes are used to encode the derived data and remove the redundancy in the data due to their nonuniform distributions. This processing is also referred as the second level of compression. In general, the output of the encoder for each scan line is a variable length bit string. Block packing will pack the bit strings into fixed length blocks such that data transmission can be achieved at a constant bit rate. The received messages can be decoded in reverse order as shown in the figure to obtain the reconstructed picture.

Chapter 2 briefly describes the picture classification (or pattern recognition) algorithms which convert the multispectral satellite pictures

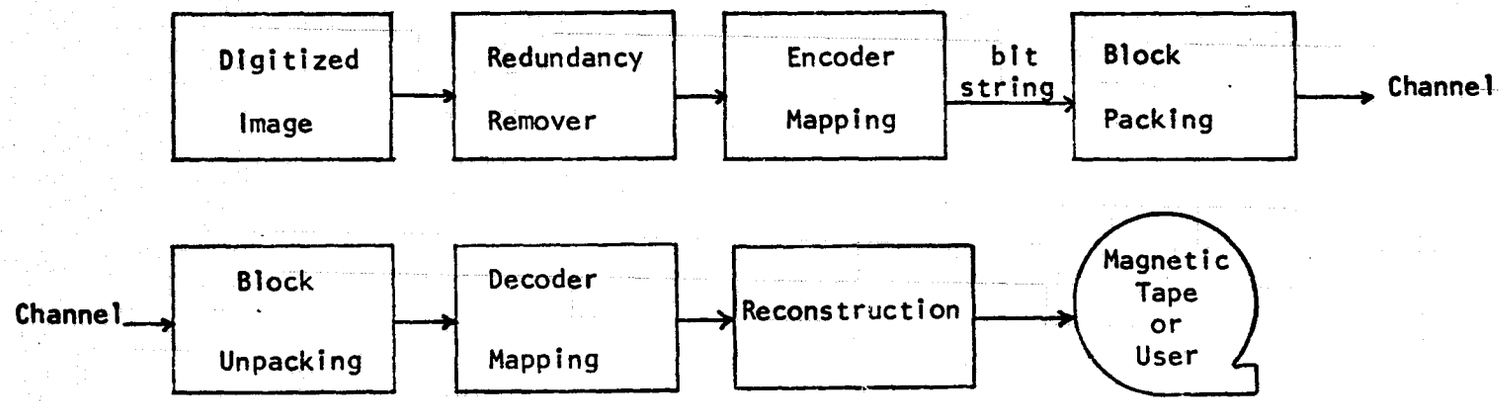


Figure 1.1 A Block Diagram of a Data Compression System

into two-level and multi-level pictures [49]. Run length coding (RLC) and predictive differential quantizing (PDQ) were originally developed for data compression on the binary pictures. We have modified the PDQ to ease the design of an efficient compressed data format. An improved data compression technique is introduced as the double delta coding (DDC). DDC is shown to be more efficient than PDQ. We find that the complexity of the figures in a picture plays an important role in the data compression ratio.

Efficient source codes are developed to encode the derived data and achieve the second level of data compression. In Chapter 3, we present the algorithm to construct the extension code which combines the prefix code and folding technique [50]. Three classes of the non-prefix codes are also investigated. The code words of the non-prefix code consist of blocks of fixed length digits. The first bit of each block is the identification bit, such that the start and end of the code word can be recognized. The extension code and non-prefix code are designed to match the source statistics.

For a multi-level or two-level picture, certain grey levels are more likely to occur than others, and we define this grey level as the majority class. Regarding the majority class as the background class, background skipping will enable us to process those figures represented by the minority classes. Thus, the processing time required will be reduced significantly in addition to the improvement of the data compression ratio. In Chapter 4, we also include the computer simulation results by applying the coding techniques and codes discussed in the previous chapters. An efficient compressed data format is designed to pack the coded messages such that they can be decoded correctly at the receiving end. A mathematical model for the derived data is set up, from which an upperbound for the entropy is defined.

Chapter 5 deals with the compression techniques which provide the data compression ratios of four and six for the satellite images. In this case, any error free coding scheme can not guarantee the desired data compression. Thus, the distortion resulting from the data compression is inevitable. An error criterion is defined to serve as the basis in evaluating the performance of the compression schemes. Three different schemes are developed and simulated. The overall performance has to take into account both the processing time and implementation complexity.

CHAPTER 2

DATA COMPRESSION ON SATELLITE IMAGES AFTER CLASSIFICATION

In this chapter we will investigate some efficient picture coding techniques on two-level and multi-level pictures resulting from image classification on multispectral satellite images. In section 2.2, we briefly describe the picture classification. Then, run length coding and modified predictive differential quantizing are investigated in section 2.3 and 2.4. Finally, an improved version of a coding technique called double delta coding is introduced in section 2.5. This chapter is closed by a section which investigates the complexity of the objects contained in the pictures, and its influence on the selection of the appropriate coding technique.

2.1 Benefits of Efficient Image Coding

A digitized image is a two-dimensional array of picture elements resulting from the quantization of sampled data. Each picture element is assigned one of the 2^m grey levels when a m-bit quantizer is used, where m is an integer. When a multispectral image is involved, the third dimension is the spectral bands. For example, the LANDSAT image consists of four spectral bands. That is to say, there are four pictures of the same scene.

For continuous tone pictures, usually a value of m between 6 and 8 is required to obtain a satisfactory picture representation. The size of

the multispectral satellite images is about 2350 by 3260. There are usually four or more pictures for the same scene. Thus, we need about 120 frames to cover the complete information for our research interest. It is rather difficult and time consuming to handle such a huge amount of data. Fortunately, there is a lot of redundancy in the pictorial data, since the entropy of the picture is less than the bits used to represent the picture. The redundancy reduction techniques developed can be divided into two categories. One of them is for continuous tone pictures which are characterized by smooth edges and fine variations between samples. The data compression schemes for continuous tone pictures will be discussed later in Chapter 5. The other is for binary and multi-level pictures which are characterized by sharp edges due to the limited number of grey levels. Therefore, this type of pictures should be compressed without any degradation.

The two-level and multi-level pictures with which we are concerned were obtained from image classification on multispectral satellite images. Typical examples of this type of data are water/non-water map (two-levels), land use map (five levels), and snow cover map (three levels). To transmit this type of data accurately within a short period of time will require a very efficient data compression method.

To evaluate the performance of the data compression techniques, many factors will have to be considered, for example, the computational efforts to do the encoding and decoding, the data compression ratios, and the effect of transmission errors. A sophisticated coding scheme may provide an attractive data compression ratio, but it may also require more computation time both for encoding and decoding. Our goal is to

investigate those coding techniques which will require minimum equipment complexity and operation efforts. Based on this principle, we will investigate some promising candidates in later sections.

2.2 Information Extraction by Image Classification

For each ground resolution element, the sensor on board the satellite generates a set of n measurements corresponding to n spectral channels. We can regard these n measurements as a point in n-dimensional Euclidean space (or measurement space). To reduce the size of the data bulk and computational efforts, a subset of n components of this measurement space is selected, and used in the picture classification. Feature selection techniques select four out of n channels as a best subset which could provide good classification results.

There are several classification algorithms developed for computer processing. Among them, the Gaussian maximum likelihood classifier [12] and the table look-up method [17] are well known. In the Gaussian classification scheme, the data is assumed to be multivariate Gaussian distributed. If the original data is non-Gaussian distributed, clustering [51] would provide a mechanism for dividing the training class into approximately Gaussian subclasses. Then, these subclasses can be used as the refined training samples to train the classifier [7].

Based on the maximum likelihood decision rule [52], [53], each picture element is classified into an appropriate class. A flow chart in Figure 2.1 shows the procedures of the Gaussian maximum likelihood classification.

The table look-up method is a powerful classification technique which substantially reduces the processing time compared with the

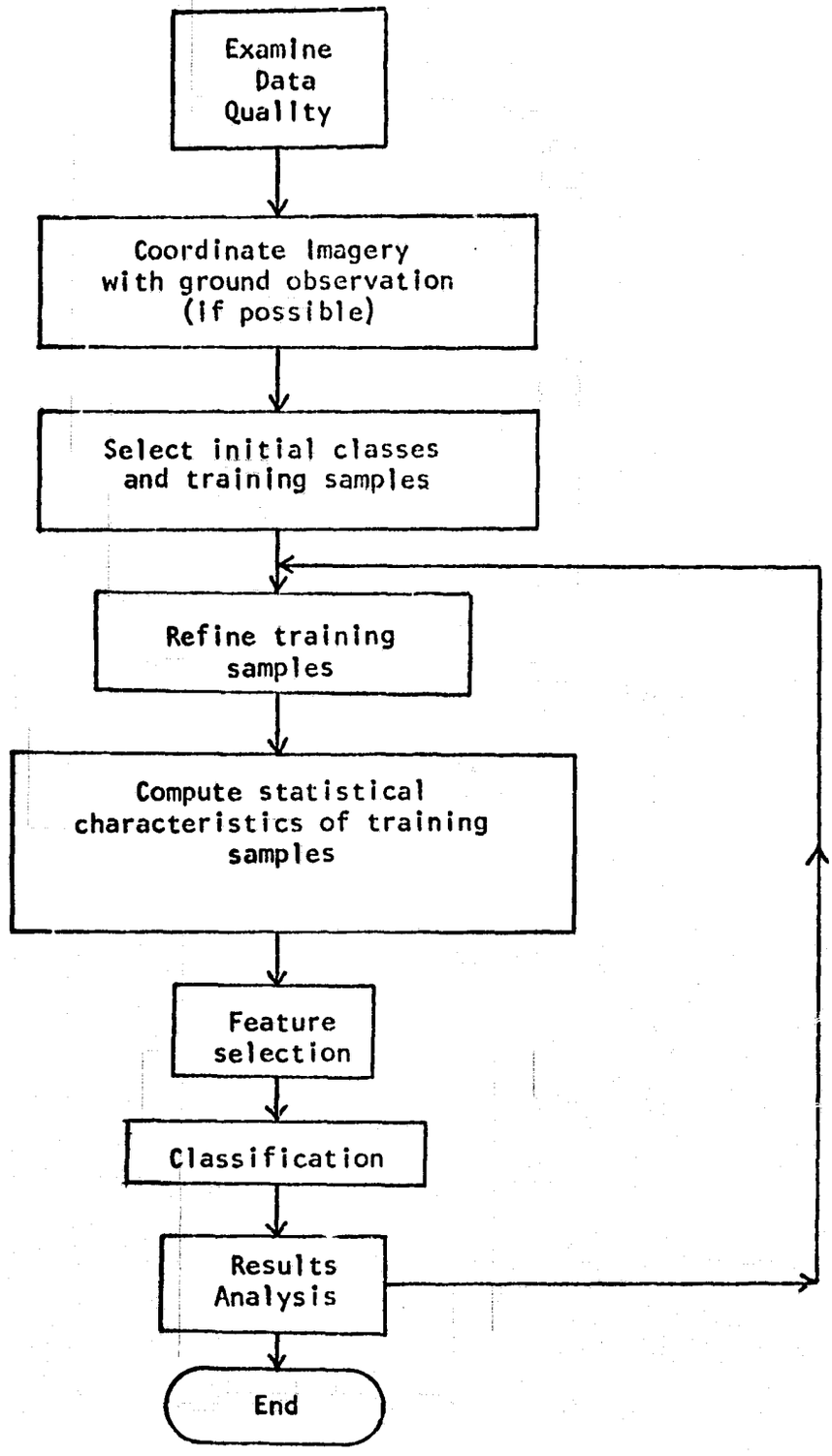


Figure 2.1 Flow Chart for the Progression of Class and Subclass Formation

Gaussian classifier on the same computer. This method makes it possible to process multispectral data on the minicomputer without additional special-purpose hardware. The classification procedure for each picture element involves a sequential, tree-structure search in which each measurement is tested to determine whether or not it is inside the region of measurement space assigned to the hypothesized class. The partitioning of measurement space is accomplished by the program which prepares the table. At each stage, the permissible range is a function of the class and the values in the preceding channels; for example, with a 4-channel satellite picture, the picture element is represented by the measurement vector $X^T = (X_1, X_2, X_3, X_4)$. Let X_{4L} and X_{4H} be the minimum and maximum values that X_4 can have and still be inside that region of a four-dimensional measurement space associated with class C. If $X_{4L} \leq X_4 \leq X_{4H}$, the hypothesis C holds, otherwise a new hypothesis is formed and tested. X_{4L} and X_{4H} are functions of C, X_1, X_2, X_3. If at any stage, an input data value is outside its permissible range, a new class hypothesis is formed. Testing continues in this way until the desired class is found such that all four data values are within their permissible ranges.

We have classified the LANDSAT pictures, taken in normal and flooding periods, into a reservoir map, river map, snow cover map, and land use map. These maps will be used in the picture coding simulation later. Classification results are presented in Figures 2.2 through 2.9. It is worthwhile to mention that classifying the picture into water and non-water using only two channels did provide comparable results with that using all four channels. This is verified by Figures 2.2(b) and 2.2(c). Furthermore, the processing time is reduced significantly.

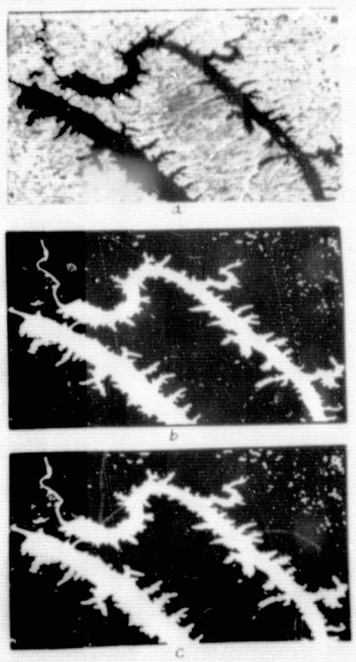


Figure 2.2 Classification Result of Non-Flooded Reservoir
(a) Original (channel 4)
(b) Classification Result Using Four Channels
(c) Classification Result Using Channels 2 and 4

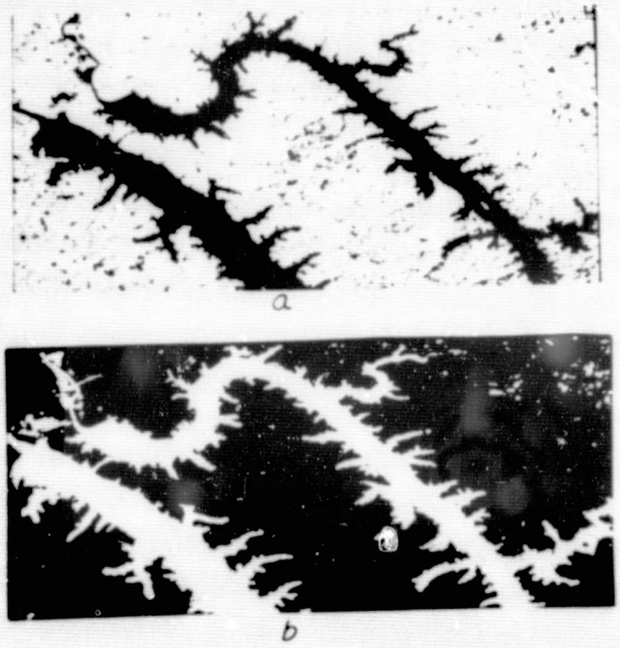


Figure 2.3 Thresholding and Classification Results
(a) Thresholding Display of Figure 2.2a Non-Flooded Reservoir
(b) Classification Result of Flooded Reservoir

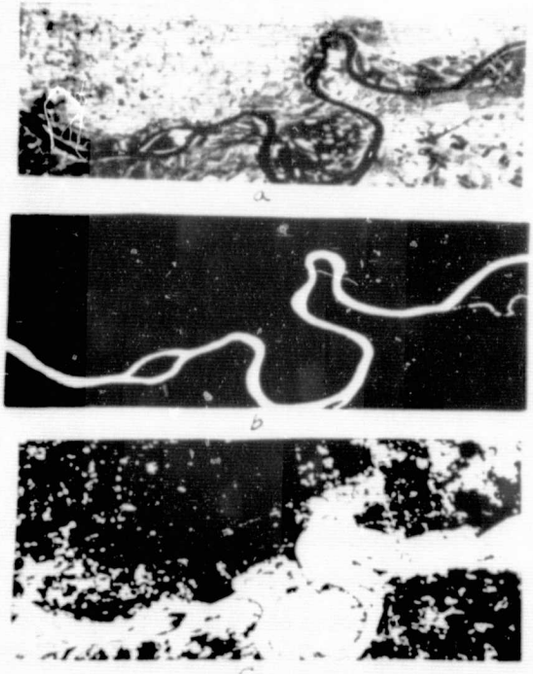


Figure 2.4 Classification Results of River Map
(a) Original Non-Flooded River Map (channel 4)
(b) Classification Result of (a)
(c) Classification Result of Flooded River Map

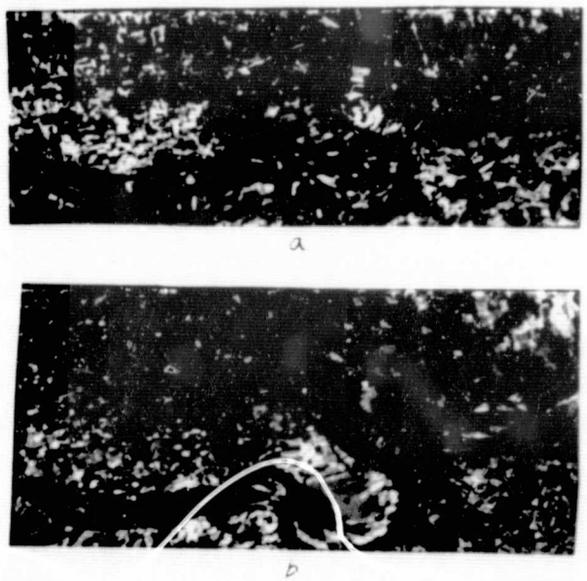


Figure 2.5 Color Infra-red River Map
(a) Original Non-Flooded River
(b) Original Flooded River

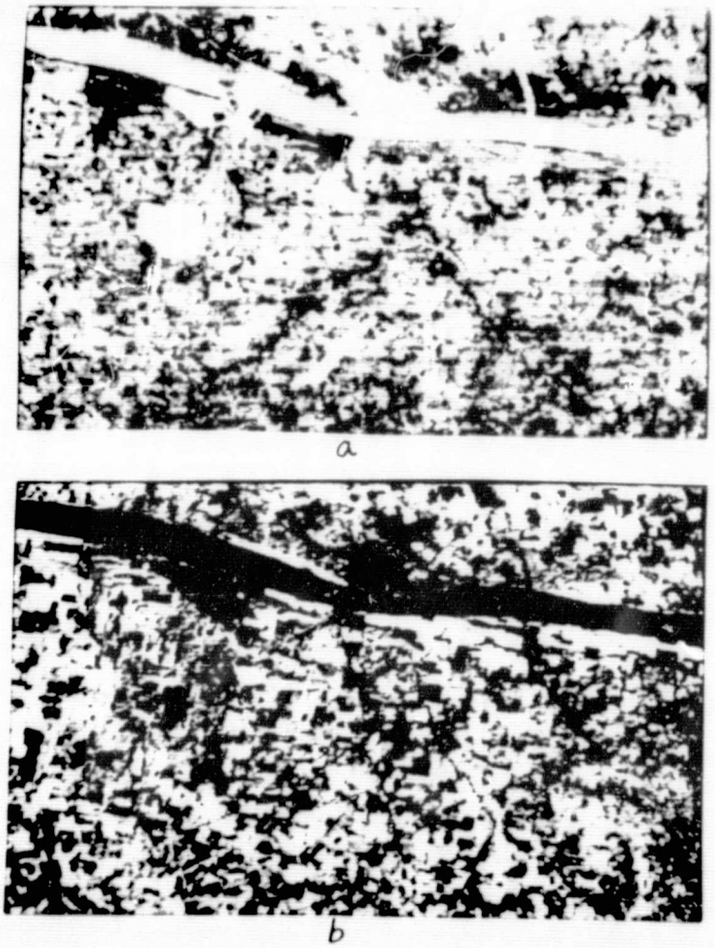


Figure 2.6 Original Land Use Map
(a) Channel 1
(b) Channel 4

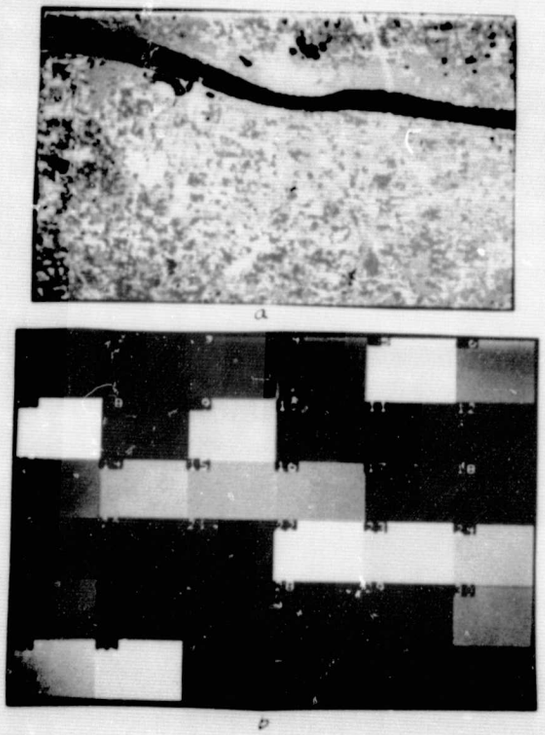


Figure 2.7 Land Use Map and Color Chart
(a) Classification Result of Figure 2.6
(b) Color Chart Where Each Color is Identified by a Number

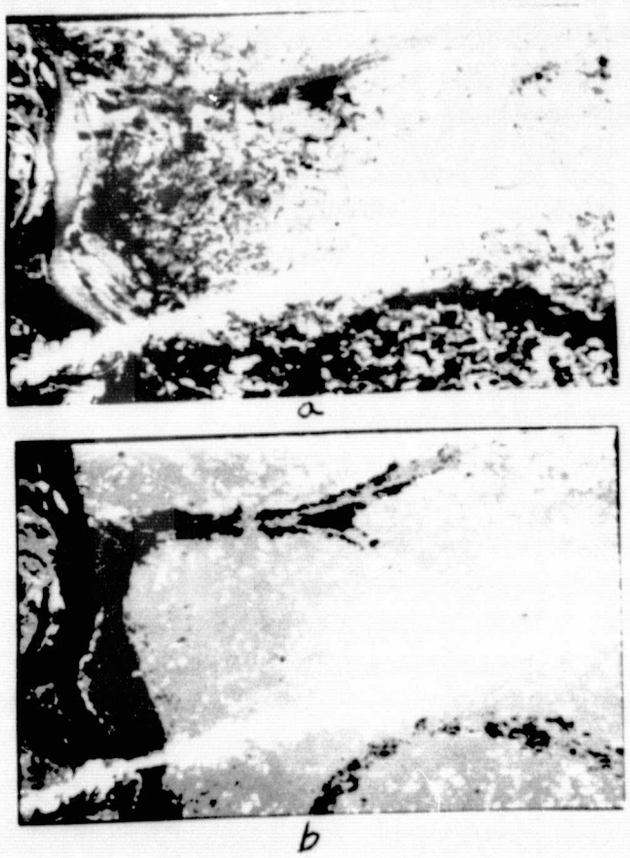


Figure 2.8 Snow Cover Map
(a) Original (Channel 2)
(b) Classification Result of (a)

Figure 2.9 Color Infra-red Snow Cover Map and Its Classification Result

(a) Color Infra-red Snow Cover Map
(b) Color Coded Classification Result of Figure 2.8(a)

Since water reflects little of the light in channel 4 of the LANDSAT picture, it appears as a significantly dark area. An easy way to classify the picture into water and non-water is to threshold the picture (channel 4) with an appropriate threshold value, instead of running the classification programs mentioned above. The CPU time used by the thresholding technique is minimized. Figure 2.3(a) shows a thresholded picture resulting in a reservoir map. In Figure 2.6(a), the interstate highway, runway of the airport, and buildings in the residential areas which are made of concrete, reflect most of the light in channel 1, and therefore appear as the bright areas.

The color picture shown in Figure 2.5 is a simulated infrared (IR) picture using channel 1, 2, and 4. The forest is shown as red in IR picture. This picture is classified into five classes: concrete/highway, water, soil, vegetation, and forest. Each class is identified with a color as shown in Figure 2.7(a). The colors in the color chart as shown in Figure 2.7(b) were used to represent the classification results. Figure 2.8(a) was taken in March of 1974, which provided the information about the snow cover on the ground. It is classified into three classes: water, non-water, snow. In Figure 2.8(b), these three classes were represented by dark, grey, and white respectively. A color coded classification result of Figure 2.8(a) is shown in Figure 2.9(b).

2.3 Run Length Coding (RLC)

Those pictures obtained by the picture classification of multispectral images consist of sequences of class numbers, where each class number corresponds to one of the grey levels. For example, in the binary water-surface extent map, grey levels 0 and 1 represent water and non-water respectively.

The neighboring picture elements of a binary or multi-level picture are highly correlated. In other words, the chances for the neighboring picture elements to be of the same grey level are very good. Instead of transmitting the class numbers point by point, the run lengths of each class and the class number are transmitted. Thus, this coding scheme is named run length coding. At the receiving end, simply repeat the class number as specified by the run lengths to reconstruct the picture.

There are two parameters resulting from run length coding: the class number and run length. Thus, the derived data will be pairs of integers (C,L), where C stands for the class number, L is the run length. After the application of run length coding, the picture can be represented by a set of sequences:

$$(C_1, L_1), (C_2, L_2), \ldots\ldots$$

The only constraint is that the successive class number C_i should not be the same.

Suppose that the length of a run depends only on the class number of this run, and the class number statistically depends on the class number of the previous run; then, the entropy of a typical sequence of n runs can be represented by

$$H(C_1,L_1,C_2,L_2,\ldots C_n,L_n) = \sum_{K=1}^{n} H(L_K|C_K) + \sum_{K=1}^{n} H(C_K|C_{K-1}) \qquad (2.1)$$

The entropy of a sequence of m_1 runs of class 1, m_2 runs of class 2, ... etc. is

$$\sum_{K=1}^{n} m_K\, H(L_K|C_K) + \sum_{K=1}^{n} m_K\, H(C_K|C_{K-1}) \qquad (2.2)$$

By the definition of the entropy:

$$H(L) = -\sum_{I=1}^{\infty} P(I)\, \log_2 P(I) \qquad (2.3)$$

where P(l) is the probability of the run length. The average number of the picture elements in the above sequence is

$$\sum_{K=1}^{n} m_K E(L|C_K)$$

where $E(L|C_K)$ is the average run length for the class K.

Therefore, the entropy per picture element is

$$h(L) = \frac{\sum_{K=1}^{n} m_K H(L_K|C_K) + \sum_{K=1}^{n} m_K H(C_K|C_{K-1})}{\sum_{K=1}^{n} m_K E(L|C_K)} \qquad (2.4)$$

This is the lower bound on the number of bits per picture element required by a code provided that the successive runs are independent.

2.4 Predictive Differential Quantizing (PDQ)

Predictive differential quantizing (PDQ) is an extension of the run length coding. In run length coding, the correlations among the picture elements along the scan lines are taken into account. This is generally referred to as one-dimensional R.L.C.. PDQ takes into account both horizontal and vertical correlations.

The main idea of PDQ is to transfer the differences between corresponding run lengths of successive scan lines. The symbols Δ' and Δ'' as shown in Figure 2.10 are transmitted, along with signals of new starts and merges. Δ' indicates the change of location, while Δ'' tells how the width changes. We modified PDQ to include new starts and merges as special cases of Δ', such that the new start signal will be coded by its starting column number, and the merge signal is encoded by a termination code. This also makes the format design easier. In our simulations, the

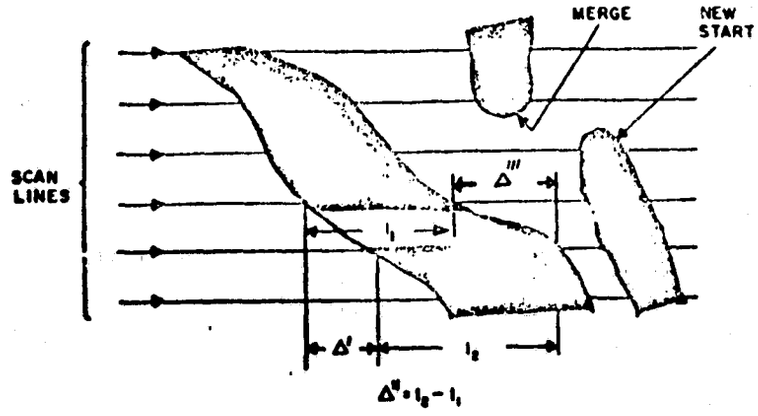


Figure 2.10 Predictive Coding

merge signal (termination code) is represented by the number of samples, n, of a scan line plus one (i.e. n+1). More details about the codes will be given in Chapters 3 and 4. We defined the process of converting the picture into sequences of (Δ', Δ'') pairs as the first level of compression. And the process of encoding the Δ's with appropriate code words is the second level of compression.

Huang [25] concluded in his paper that PDQ out performs both run length coding and block coding when it is used on weather maps, but it is inferior to run length coding when used on printed material. We have applied PDQ to our two-level and multi-level pictures. From the simulation results, it showed that the complexity of the figures is a major factor in the data compression ratio achieved.

2.5 Double Delta Coding (DDC)

As shown in Figure 2.10, instead of transmitting Δ' and Δ'', we propose to transmit Δ' and Δ''' such that both Δ' and Δ''' indicate

the changes of the edges. Again, the new start and merge signals are regarded as special cases of Δ'. In these cases, Δ''' is used to represent the width of starting or merging figure on that scan line.

Let us consider the following figure

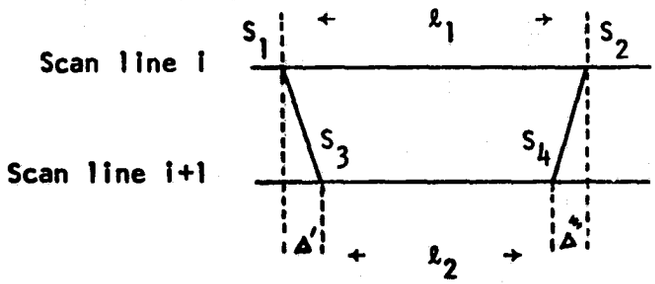


Figure 2.11 Double Delta Coding

and define the Δ's in the strict sense.

$$\begin{aligned} \Delta' &= S_3 - S_1 \\ \Delta'' &= \ell_2 - \ell_1 \\ \Delta''' &= S_4 - S_2 \end{aligned} \tag{2.5}$$

Assuming that straight lines could be used to approach end points S_1, S_3, and S_2, S_4 respectively. As the strict sense of the definitions of Δ's in equation (2.5) is concerned, we can assume that the probability distribution functions of Δ' and Δ''' are approximately the same from their definitions. Thus, we have $H(\Delta') \simeq H(\Delta''')$. However, we have used the general definition of Δ' in our computer simulation for convenience.

The entropy for new starts and merges were included in $H(\Delta')$. $H(\Delta')$ is, therefore, higher than $H(\Delta''')$.

The probability density functions of Δ's are shown in Figures 4.3 through 4.6. We see that there is always a highest spike dominating over the rest of them in the distribution of Δ'''. However, there is no such dominant spike in the distribution of Δ''. From the properties of the entropy function [48], we conclude that the entropy of Δ''' is less than that of Δ'' (i.e. $H(\Delta''') < H(\Delta'')$). This conclusion is verified by our computer simulation results as shown in Table 4.6.

It is more efficient to transmit Δ' and Δ''' than Δ' and Δ'', not only because the number of digits required to be transmitted is reduced, but also from the practical point of view. This is the advantage of the double delta coding over predictive differential quantizing. The computer simulation results will be shown in Chapter 4.

2.6 The Complexity of the Figures

Most of the source coding techniques are data dependent. In other words, certain coding schemes may be good for certain pictures, but it is less efficient for other pictures because of the design of the coding algorithm.

For those pictures shown in section 2.2 which will be used in the computer simulations, the complexity of the picture is a major factor in the performance of the coding scheme.

Let us define the complexity of the figure i in the picture as follows:

$$C_i = \left| \frac{4\pi A_i}{P_i^2} - 1 \right| \quad i = 1, 2, \ldots n \tag{2.6}$$

where A_i is the area of the figure i and P_i is the perimeter of the figure i.

Among all the figures, the circle is the simplest figure with $C_i = 0$. Thus, the complexity for the circle is the minimum for all figures. On the other hand, a line can be regarded as an extreme case of a complicated figure with infinite perimeter and infinitesimal area, thus its complexity is 1. The complexity of any figure should lie between 0 and 1.

Since the area and perimeter of a figure can be obtained as a by-product in the processing of run length coding or predictive coding (PDQ or DDC), there will not be much extra computer time required to determine the complexity of the figures.

The complexity of the whole picture, C, can be defined as the sum of the complexity of each individual figure in the picture. That is,

$$C = \sum_{i=1}^{n} C_i \tag{2.7}$$

From our computer simulation, we found that run length coding outperformed PDQ and DDC for the pictures with high complexity. For example, the flooded river map is a picture with relatively high complexity. On the other hand, PDQ and DDC provide better data compression ratios for the pictures with little complexity. The river and reservoir maps taken in the normal season are typical examples. For the multi-level pictures like the snow cover map and the land use map, run length coding would be more efficient because of the picture complexity.

In Figure 2.12, we show the relationship between the compression ratio and the picture complexity for various coding schemes which were used in our simulations. Thus, a threshold, T, for the picture complexity

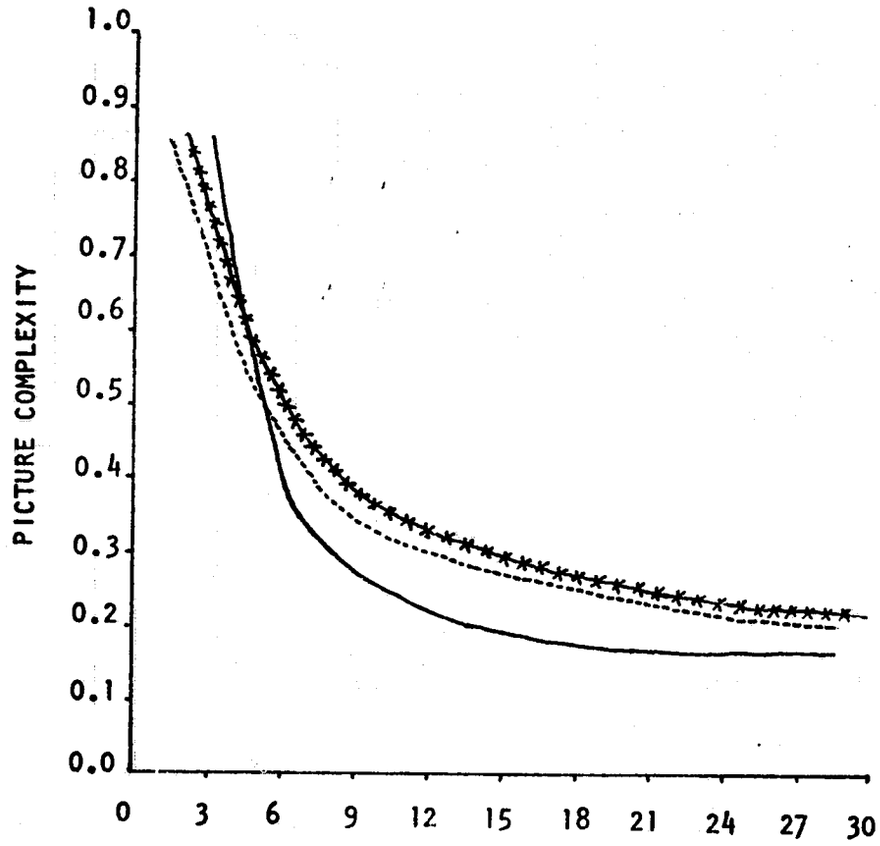


Figure 2.12. Data Compression Ratio vs Picture Complexity for Simulated Pictures

can be chosen properly such that whenever $C \geq T$, the RLC will be used as the redundancy remover, whereas PDQ or DDC is chosen if $C < T$.

CHAPTER 3

EFFICIENT SOURCE CODES

The first level of redundancy removing was investigated in Chapter 2 in which most of the coding schemes take into account the high correlations of the neighboring picture elements.

Through the redundancy remover, we convert the input data (digitized image) into sequences of run lengths or index pairs, which are defined as the derived data. In this chapter, we will investigate the properties of the derived data. Several classes of code words were designed to match the statistics of the derived data.

3.1 Source Codes to Match the Data Statistics

From the probability density functions of the runs and Δ's which are shown in Figure 4.3 through 4.6, we see that the statistics of the derived data of the computer simulated pictures has a characteristic that the data falls in the range -31 to 31 with the probability of 0.6. Furthermore, the probability density functions can be approximated by exponential and symmetric functions.

The optimal code for this type of data is to assign shorter code words to the data with higher probability of occurrence, and longer code words to the data with less probability of occurrence. The Huffman code [35] is the optimal code for nonuniformly distributed data, but the construction of this code is too complicated to be implemented in practical

operations. Thus, the code words which we will investigate for the derived data are sub-optimal codes.

3.2 Properties of the Prefix Code

A sequence, P, of binary symbols is called a prefix of ω if ω=ps. A class of codes is said to have the prefix property if no code word is a prefix of any other code word. Thus, if a class of codes has the prefix property, each code word can be uniquely decoded as soon as it is received in its entirety. However, the classes of codes which will be discussed later in section 3.4, still can be uniquely decoded even though they are not prefix codes.

To demonstrate the importance of the prefix property, let us consider a class of codes C = {11, 010, 100, 110, 101, 0011} which does not possess the prefix property. Suppose that a sequence of symbols S = 11010101010101 was received. S can be decoded into 11, 010, 101, 0101, or 110, 101, 010, 101. That is to say that S cannot be decoded uniquely. Therefore, a uniquely decodable code has to be a prefix code or use a special design of identification bit in each code block, such as the R_1 code in section 3.4.

Prefix codes can be constructed through the tree structure [54] which is the easiest way to form the code. The prefix code is utilized as the first step of the construction of extension codes in section 3.3.

3.3 Algorithm CT - to Construct the Extension Code

The first step of the construction of Extension Codes is to form a class of prefix codes by the tree structure as mentioned in section 3.2. The unique decodability of a class of codes can be verified by a method suggested and proved by Sardinas and Patterson [55].

For those pictures we have simulated, the normal operating range of the derived data is about 700 integers. To encode the derived data with prefix codes will require a large amount of code words which make the construction of prefix codes impractical for coding purposes due to the code length being increased too fast.

Our algorithm is to construct a class of prefix codes which consist of a few code words. Applying the extension blocks to some of the prefix codes, we can extend the prefix code with a limited number of code words to encode a large number of source messages (i.e. derived data). And the coded sequence can be decoded without ambiguity.

It is clear that the storage space required to store those prefix codes at the encoder and decoder will be very small. The implementation of the extension code can be done with very little complexity. It is easier to explain the algorithm CT with an example.

Suppose that the source messages consist of integers ranging from 1 to 560. We construct a code dictionary C = { 00, 111, 1100, 1101, 1000, 1001, 1010, 1011, 0100, 0101, 0110, 0111}. The code C is a class of prefix codes. Since any sequence of words from C beginning with any suffix of any word in C can make the decoder automatically come into correct synchronism, the code dictionary C is a completely self-synchronizing dictionary.

Again, we assume that the small integers possess higher probability of occurrence. Figure 3.1 shows the mapping of the code words to the source messages. Of course, the number of code words in C can be adjusted based on the number of source messages (or derived data). This code requires very little implementation complexity and very small storage space to store the code words. All of these features make the extension code a practical source code.

Code Words	Mapping To Source Messages
00	1
111	2
0101	3
1101	4
1000	5
0111	6
0110	7
1011	8
0100	9
1100	1 block of extension (3 bits)
1010	2 blocks of extension (6 bits)
1001	3 blocks of extension (9 bits)

0101 + □□□ will cover 10 to 17 (16 code words)

0110 + □□□ □□□ will cover 18 to 81 (64 code words)

0111 + □□□ □□□ □□□ will cover 82 to 593 (512 code words)

In other words there are 2^n possible ways to fill the block/blocks.

Figure 3.1 Extension Code for Integers 1 through 593

The code words shown in Figure 3.1 can be used to encode the run lengths. To encode the derived data from predictive coding, where the integers are ranged from -275 to 1060, another mapping will be used, as shown in Figure 3.2. For Δ', we assign the shortest code word to initiate a figure termination. The code words 0100, 1100 followed by one or two blocks of extensions are assigned to encode the positive numbers such that 0100 plus 5 bits of natural codes will represent the integers 4 through 35, and 1100 plus 10 bits of natural codes are used to encode the integers 36 through 1060. Similarly, code words 1010 and 1001 are extended to encode -14 through -19 and -20 through -275 respectively. Note that the number of bits in the extension blocks can be adjusted such that the extension code can cover the whole range of the derived data.

3.4 Non-Prefix Code

We have said that the uniquely decodable code has to possess the prefix property. In this section, we will investigate a class of codes which is not a prefix code. The structure of this code is different from that of the prefix code. In order to uniquely identify each code word, the code word is divided into several fixed length blocks of two bits. The first bit of each block is the identification bit followed by an information bit. All the identification bits within a code word have to remain the same so that it can be recognized and decoded correctly at the receiving end.

The following codes were designed to match the statistics of the derived data:

Code Words	Δ'	Δ''	Δ'''
00	Figure termination	0	0
111	N = 0	N = 1	N = 1
0101	1	-1	-1
1101	-1	2	2
1000	2	-2	-2
0111	-2	3	3
0110	3	-3	-3
1011	-3	4	4
0100	positive 1 block extension (5 bits)	positive 1 block extension (4 bits)	positive 1 block extension (5 bits)
1100	positive 2 blocks extension (10 bits)	positive 2 blocks extension (8 bits)	positive 2 blocks extension (10 bits)
1010	negative 1 block extension (4 bits)	negative 1 block extension (4 bits)	negative 1 block extension (4 bits)
1001	negative 2 blocks extension (8 bits)	negative 2 blocks extension (8 bits)	negative 2 blocks extension (8 bits)

For Δ' if $N > 0$ and $4 \leq N \leq 35$ code length = 9 bits

$36 \leq N \leq 1060$ code length = 14 bits

If $N < 0$ $4 \leq |N| \leq 19$ code length = 8 bits

$20 \leq |N| \leq 275$ code length = 12 bits

Figure 3.2 Encoding Δ's by Extension Code

(a) R_1 Code - for Run Lengths

This code was first developed by Huang, etc. [56] to encode the run lengths. The optimum Huffman code is impractical due to the fact that the run lengths may be any possible integer. Each code word consists of several fixed length blocks of two bits. Figure 3.3 shows an example of the R_1 code; it consists of three blocks. The first bit in each block is the identification bit which is used to identify the starting and end of a code word. The second bit is the information bit. All of the identification bits of a code word have to be either all zero or all one.

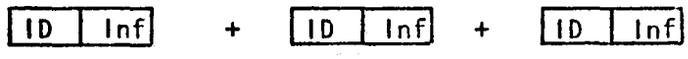


Figure 3.3 An Example of R_1 Code with Three Blocks

When a two-level picture is coded, the identification bit can also be used as the color code. For a multi-level picture, the expansion of one identification bit into two or more bits is not recommended, since there is a better way to handle this problem as disucssed in chapter 4.

Because there is only one information bit per block, the total number of possible code words consisting of n blocks is 2^n. The encoding algorithm is as follows:

(1) Determine the range of the run length L.

If $2^n-1 \leq L < 2^{n+1} - 1$

Then L is mapped into a code word with n blocks.

(2) Determine the n information bits.

The binary representation of Z coded in the n information bits is

$$Z = L - 2^n + 1.$$

The major attraction of R_1 code is the ease of implementation. A computer simulation with the R_1 code will be presented in chapter 4.

(b) C_1 code - for PDQ or DDC (i.e. for Δ', Δ'' and Δ''')

Through the predictive coding, PDQ or DDC, the input data (images) will be converted into sequences of index pairs: (Δ', Δ'') or (Δ', Δ'''). The range of these indices of the simulated pictures is from -260 to 1040. During the first level of PDQ or DDC processing, the figure termination is represented by n+1 for a picture with n samples per scan line.

The application of the same code to Δ' and Δ'' or Δ' and Δ''' may simplify the implementation complexity of the encoder and decoder, but it also has a drawback, since no single code can be optimal to two source messages with different statistics.

The encoding procedure for C_1 code is as follows:

Let Δ denote any index: $(\Delta', \Delta''$ or $\Delta''')$

(1) The mapping M_1 will map Δ's into N

M_1 : if $\Delta \leq 0$ $\quad N = 2 \times (-\Delta)$

$\Delta > 0$ $\quad N = 2 \times \Delta - 1$

(2) The mapping M_2 will map N into C_1 codes

If $2^n - 1 \leq N+1 < 2^{n+1} - 1$, then the code word for N consists of n blocks. The first bit of each block is the identification bit, followed by the information bit.

(3) The binary number Z coded in n information bits of the code word is

$$Z = N - 2^n + 2$$

Figure 3.4 shows some code words in C_1 code. The code length for figure termination is the longest.

C_1 Code - for Δ', Δ'', or Δ''''

Δ	$\xrightarrow{M_1}$	N	$\xrightarrow{M_2}$ Code
0		0	ID 0
1		1	ID 1
-1		2	ID 0 ID 0
2		3	ID 0 ID 1
-2		4	ID 1 ID 0
3		5	ID 1 ID 1
-3		6	ID 0 ID 0 ID 0
4		7	ID 0 ID 0 ID 1
-4		8	ID 0 ID 1 ID 0

Figure 3.4 C_1 Code

(c) 0_1 code - for Δ' only

The statistics of Δ' for most of the simulated pictures shows that the probability of occurrence of the figure termination is very high. Therefore, it will be more beneficial to construct a code mapping which assigns a shorter code word to represent the figure termination. Thus, we can apply the 0_1 code for Δ' and the C_1 code for Δ'' or Δ''' to achieve a better data compression ratio.

The encoding algorithm for 0_1 code is as follows:

(1) The mapping M_1 maps Δ' into N

M_1: The figure termination is mapped to N = 0

for $\Delta' > 0$ $\quad N = 2 \times \Delta'$

$\Delta' \leq 0$ $\quad N = 2 \times (-\Delta') + 1$

(2) The mapping M_2 is the same as M_2 in C_1 code encoding procedure

Figure 3.5 shows some code words of the 0_1 code where the figure termination is mapped into a code word with only two bits.

For a picture with size greater than the normal operating size, which is determined by the practical situation and the capacity of the computer in use, it is better to divide the picture into several sections such that no lengthy code word will be used. And the segmentation of the pictue will also reduce the transmission errors which occur in the lengthy code words when the segmentation is not used.

All of the codes developed in this chapter are simulated to determine their performance in chapter 4.

O_1 Code - for Δ' only

Δ'	$\xrightarrow{M_1}$ N	$\xrightarrow{M_2}$ Code
Terimation	0	ID 0
0	1	ID 1
1	2	ID 0 ID 0
-1	3	ID 0 ID 1
2	4	ID 1 ID 0
-2	5	ID 1 ID 1
3	6	ID 0 ID 0 ID 0
-3	7	ID 0 ID 0 ID 1

Figure 3.5 O_1 Code

CHAPTER 4

BACKGROUND SKIPPING AND ANALYSIS OF THE SIMULATED DATA

Background skipping enables us to deal with the data which has less probability of occurrence such that more saving on the processing time and higher data compression can be obtained. The background class can be determined from the statistics of the picture.

In this chapter, we will present some computer simulation results by applying those coding techniques and codes investigated in the previous chapters, to binary and multi-level pictures. The comparisons among these coding schemes and codes are also made.

An analysis of the derived data from the computer simulation is described in section 4.3. Finally, a compressed data format is developed to pack the encoded indices (i.e. run lengths or Δ's) in an efficient manner that the data compression can be achieved with minimum errors.

4.1 Background Skipping Technique (BST)

Definition of Background Class

The major class of a picture with two or more classes is defined to be the class with the highest probability of occurrence. If the picture is too complicated and the statistics vary significantly from region to region, we can divide the picture into several sections, and define a major class for each section. Those classes which are not selected as the major class are defined to be the minor classes.

We select the major class as the background class for a section of the picture or the entire picture depending on where the major class is defined.

Advantage of BST When Applied to PDQ and DDC

The characteristics of PDQ and DDC is to transmit the picture with fewer digits by updating the the changes of the figures between the successive scan lines. However, if we randomly selected those figures represented by the major class, we will have to spend more time in the processing of encoding and decoding in addition to the detection of index pairs in the scan lines. When the multi-class picture is concerned, the waste in time and data compression will be more severe.

With the background skipping, a class code for the background class is transmitted at the beginning of data transmission. Then, the index pairs of the figures represented by the minor classes are transmitted. At the receiver, the storage space for the reconstructed picture is first filled with the background class. When the decoding is completed, the transmitted figures are formed in the appropriate locations.

Table 4.1 shows the advantage of the application of the background skipping in the simulations of PDQ where the processing time and the data compression are considerably improved.

The statistics of the two-level pictures is presented in Table 4.2. Since the land is the major class, it is defined as the background class.

4.2 Computer Simulation of PDQ and DDC

There are two steps in the data compression routines of PDQ and DDC. The first level of compression converts the pictures into sequences of index pairs. A flow chart in Figure 4.1 shows the algorithm of the first

Table 4.1 Advantage of Using Majority Class as Background Class in PDQ

PICTURE	PDQ (C_1 code) Background Class ≠ Majority Class	PDQ (C_1 code) Background Class = Majority Class
No Flood Reservoir 4522 words	R = 4.48611 comp: 54.30300 sec decomp: 94.30300 sec 1008 words	R = 5.25203 comp: 14.02 sec decomp: 53.204 sec 861 words
Flooded Reservoir 3705 words	R = 3.79222 comp: 52.4 sec decomp: 76.415 sec 977 wprds	R = 4.48004 comp: 12.097 sec decomp: 44.255 sec 827 words
Flooded Reservoir 6960 words	R = 0.80937 comp: 92.841 sec decomp: 107.318 sec 8599 words	R = 1.00686 comp: 52.538 decomp: 79.928 sec 6913 words
No Flood River 3060 words	R = 7.48166 comp: 46.829 sec decomp: 74.828 sec 409 words	R = 12.9113 comp: 6.525 sec decomp: 34.526 237 words

Table 4.2 Statistics of LANDSAT-2 Images

Percentage / Picture	CLASS 1 (LAND)	CLASS 2 (WATER)
1. No Flood Reservoir	80.23%	19.77%
2. Flooded Reservoir	76.78%	23.22%
3. Flood River	86.27%	13.73%
4. No Flood River	94.03%	5.97%

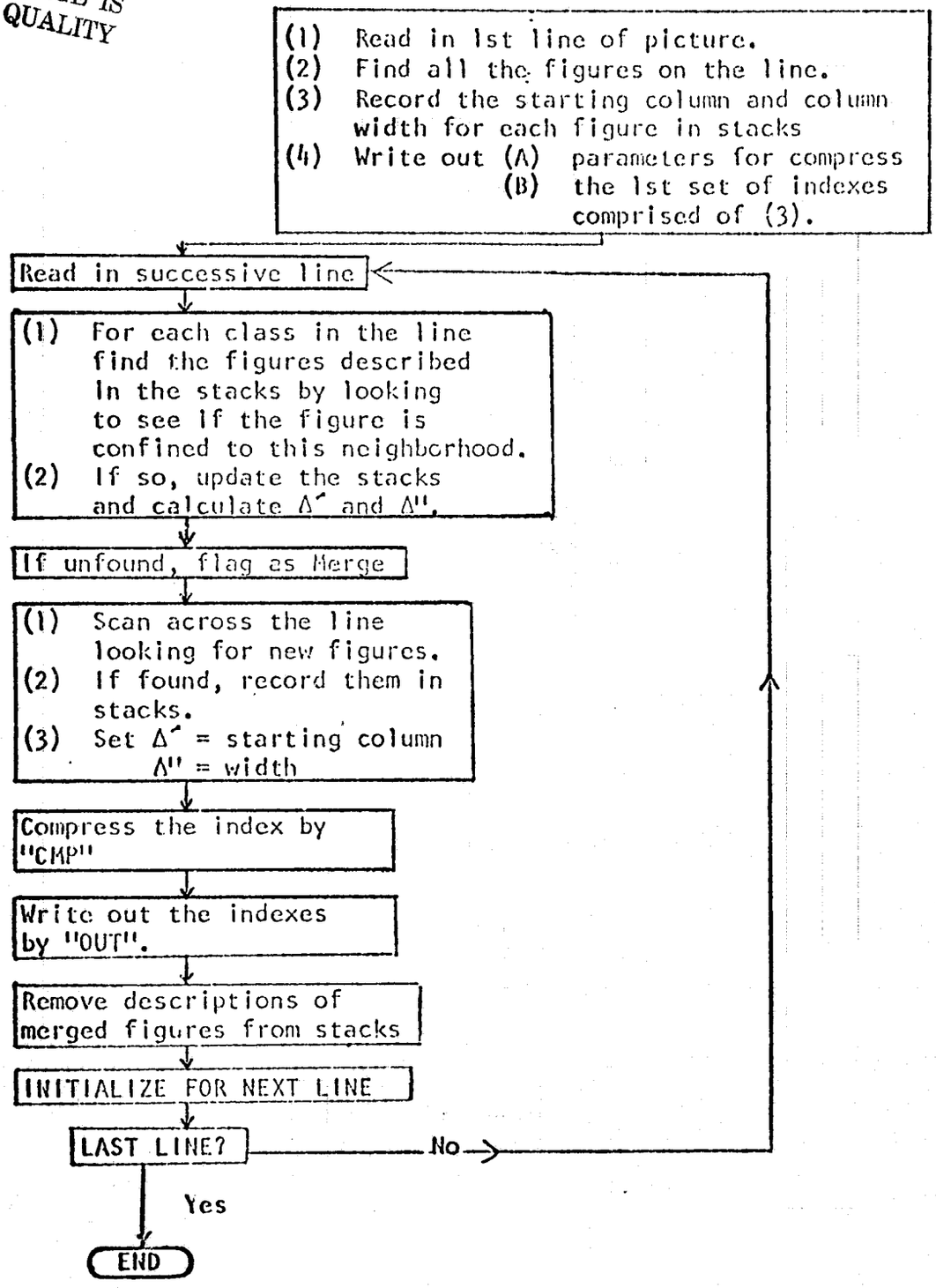


Figure 4.1 Flow Chart of the 1st Level Compression

level of compression for PDQ. The second level of compression converts the index pairs into coded messages according to the mapping of the selected code.

When the first level of compression is completed, the index pairs of each scan line are arranged in the format as shown in Figure 4.2.

N	(Δ_1', Δ_1''), (Δ_2', Δ_2''), (Δ_3', Δ_3''), . . .

N denotes the number of index pairs for this scan line

Figure 4.2 Arrangement of the Index Pairs

During the first level of compression, we have used the following algorithm to find the indices:

(1) Locate all the figures on the scan line and record the starting column and width for each figure.

(2) Group the index pairs of the same class together such that only one class number is used to identify each group.

(3) For each class of the next scan line, find the figures as described in step (1) and (2) by looking to see if the figures are confined to their neighborhood, calculate Δ's. Then, find the new figures of each class, and locate the index pairs.

(4) Repeat the step (3) for the rest of scan lines until the last one is reached.

Recall that code words of the non-prefix codes (i.e. C_1 code or O_1 code) consist of blocks of two bits. One of them is the identification bit which can be used as the class code for the two-level picture. Therefore, there is no need to pack a class code in front of the sequence of index pairs of the same class. When a multi-level picture is concerned,

it is not worthwhile to extend one identification bit into two or more bits. Instead, we can pack a class code in front of the group of index pairs of the same class. This will optimize the second level of compression by saving a lot of identification bit which would occur in each block of the code words.

As described in Chapter 2, we have classified the multispectral images into (1) two-level pictures which include the river map, and reservoir map both in the regular and flooded seasons; (2) multi-level pictures which include the land use map and snow cover map.

These pictures were used in the computer simulations by various coding techniques and codes which were discussed in Chapters 2 and 3. The performances of the simulated results were evaluated.

For convenience, we divided the results into three parts - two-level and multi-level pictures plus the data statistics:

(a) Simulation Results for Two-Level Pictures

Table 4.3 shows the data compression ratios obtained through the applications of RLC, PDQ and DDC using the R_1 code, C_1 code, and O_1 code, respectively. In general, the flooded-period pictures are more complicated than those of the regular season because there are more figures which contribute more new starts, figure terminations, and short runs. The advantage of PDQ and DDC over RLC no longer holds in these pictures from the fact that the algorithms of PDQ and DDC are designed to take advantage of the small changes of the figure between successive scan lines.

We made another simulation by implementing two options of the extension code with PDQ and DDC. One of these is to encode the index

Table 4.3 Simulation Results for Two-Level Pictures where R_1 and C_1 Codes were Used

Picture \ Coding Scheme	RLC (R_1)	PDQ (C_1)	DDC (C_1)
1. No Flood Reservoid 4522 words (1 word = 60 bits)	R = 5.4306 Comp: 9.994 sec Decomp: 50.015 sec 833 words	R = 5.25203 Comp: 14.02 sec Decomp: 53.204 sec 861 words	R = 5.74587 Comp: 14.655 sec Decomp: 52.174 sec 787 words
2. Flooded Reservoir 3705 words	R = 4.28979 Comp: 8.445 sec Decomp: 40.439 sec 864 words	R = 4.48004 Comp: 12.097 sec Decomp: 44.255 sec 827 words	R = 4.77448 Comp: 11.735 sec Decomp: 41.776 sec 776 words
3. Flooded River 6960 words	R = 2.15597 Comp: 22.206 sec Decomp: 78.008 sec 3228 words	R = 1.00686 Comp: 52.538 sec Decomp: 79.928 sec 6913 words	R = 1.07146 Comp: 50.993 sec Decomp: 78.103 sec 6881 words
4. No Flood River 3060 words	R = 9.5625 Comp: 5.632 sec Decomp: 33.087 sec 320 words	R = 12.9113 Comp: 6.526 sec Decomp: 34.526 sec 237 words	R = 14.6411 Comp: 6.467 sec Decomp: 33.119 sec 209 words

Comp = CPU Time Used for Compression
Decomp = CPU Time Used for Decompression
R = Data Compression Ratio

pairs with the same version of the extension code. The other is to select two different versions of the extension code to encode Δ' and Δ'' (or Δ''') respectively. Table 4.4 gives a comparison of the performances between these two options. It is obvious that the second option is definitely better.

We also simulated the two-level pictures with PDQ and DDC, where O_1 code was used to encode Δ', and C_1 code was selected for encoding of Δ'' or Δ'''. As shown in Table 4.5, we see that the data compression ratios achieved are better than that shown in Table 4.3 where the C_1 code was used to encode all the indices (Δ', Δ'', and Δ''').

Table 4.6 shows a list of entropies of run lengths and Δ's in addition to the optimal data compression ratios we can achieve. This table is prepared to be used as a reference to evaluate the codes we have discussed. The optimal code known to us is the Huffman code, but it is not practical in the coding operations. We found that the data statistics for the same type of picture have the same characteristics. Thus, we can predetermine the code for each type of picture. From the simulations, we are convinced that the codes we have developed can be used to operate in an efficient manner without the knowledge of the statistics of the individual picture. For example, RLC operates much better than other techniques for the flooded river maps, while DDC and PDQ are the best for the three other types of pictures.

(b) Simulation Results for Multi-Level Pictures

Among the multi-level pictures we simulated, the snow cover map is considerably more complicated than the land use map even though there are more classes in the land use map. We also noted that the data compression

Table 4.4 Simulation Results of Extension Codes

PICTURE	PDQ Same Codes Used for Δ', Δ''	PDQ Different Codes Used for Δ', Δ''	DDC Different Codes Used for Δ', Δ'''
1 No Flood Reservoir	794 words	716 words	706 words
4522 words	R = 5.69521	R = 6.31564	R = 6.40509
2 Flooded Reservoir	843 words	823 words	739 words
3705 words	R = 4.39501	R = 4.50197	R = 5.00878
3 Flooded River	6950 words	6643 words	5378 words
6960 words	R = 1.00141	R = 1.04776	R = 1.29423
4 No Flood River	236 words	235 words	219 words
3060 words	R = 12.9661	R = 13.0212	R = 13.9726

Table 4.5 Simulation Results where O_1 Code is Used for Δ' and C_1 Code is Used for Δ'' & Δ'''

PICTURE	PDQ	DDC
1 No Flood Reservoir 4522 words	670 words R = 6.74925	527 words R = 8.58064
2 Flooded Reservoir 3705 words	797 words R = 4.64852	714 words R = 5.18553
3 Flooded Reservoir 6960 words	6523 words R = 1.06706	5338 words R = 1.36379
4 No Flood River 3060 words	229 words R = 13.3624	201 words R = 15.2238

Table 4.6 Entropy Resulted from Various Coding Schemes

PICTURE	RLC	PDQ	DDC
1 No Flood Reservoir 259290 bits	E(L)=5.6437 bits Entropy = 38524 bits R = 6.7306	E(Δ′)=3.2121 bits E(Δ′′)=3.8829 bits Entropy = 29665 bits R = 8.7406	E(Δ′)=3.2121 bits E(Δ′′′)=2.7287 bits Entropy = 24839 bits R = 10.4388
2 Flooded Reservoir 210270 bits	E(L)=5.8484 bits Entropy = 38302 bits R = 5.4897	E(Δ′)=4.8771 bits E(Δ′′)=4.1455 bits Entropy = 35414 bits R = 5.9375	E(Δ′)=4.8771 bits E(Δ′′′)=3.1413 bits Entropy = 31473 bits R = 6.6809
3 Flooded River 405640 bits	E(L) = 4.2678 bits Entropy = 133745 bits R = 3.0329	E(Δ′)=6.3333 bits E(Δ′′)=4.3941 bits Entropy = 291548 bits R = 1.3913	E(Δ′)=6.3333 bits E(Δ′′′)=2.5929 bits Entropy = 242597 bits R = 1.6721
4 No Flood River 171600 bits	E(L)=7.3967 bits Entropy = 13255 bits R = 12.946	E(Δ′)=3.6020 bits E(Δ′′)=3.2269 bits Entropy = 5969 bits R = 28.7485	E(Δ′)=3.6020 bits E(Δ′′′)=2.9276 bits Entropy = 5707 bits R = 30.0683

E(L) = Entropy of run length E(Δ′) = Entropy of Δ′

R = Optimal data compression ratio

Entropy = Entropy of the entire picture

ratios achieved by PDQ and DDC are not as good as that achieved by RLC. Furthermore, it shows that the complexity of the figures of the picture is a major factor in the data compression ratios even though the number of classes (or grey levels) in a picture is somewhat important.

As shown in Table 4.7, the R_1 code and the C_1 code were selected for the simulation of multi-level pictures. We have come to a conclusion that RLC is more efficient than PDQ or DDC as far as the multi-level pictures are concerned. Especially, when there are lots of figures with high complexity in the picture, any error free coding technique such as RLC, PDQ or DDC will not serve as a good technique for data compression.

Some coding schemes with distortion like the one suggested by Ziv [57] may be the alternative which provides the constant bit rate at the output of the encoder. This scheme considers blocks of data symbols of a predetermined length, and assigns a string of binary digits, that has a predetermined length, to each block. The length of the blocks depends on the output bit rate desired. This technique is not error free because a single code word was assigned to a number of different messages.

(c) Data Statistics

Figure 4.3 through Figure 4.6 are the measurements of the various data statistics for binary pictures. The curves represent the envelopes of the relative probability of occurrence. Numbers shown on the abscissa represent the extent of the data. The vertical axis represents the relative probability. Thus, the cases with zero probability of occurrence do not imply that they did not occur in the picture. Instead, they are just less likely to occur with a probability close to zero.

Comparing the probability distribution of Δ''' against that of Δ'', we find that there is a dominant spike located at the center of the curve

Table 4.7 Simulation Results for Multi-Level Pictures

Picture	RLC (R_1)	PDQ (C_1)	DDC (C_1)
Snow Cover Map 3134 words (3 Levels)	R = 1.85443 Comp: 18.273 sec Decomp: 36.066 sec 1690 words	R = 1.04142 Comp: 28.214 sec Decomp: 41.42 sec 3009 words	R = 1.27205 Comp: 27.473 sec Decomp: 39.572 sec 2464 words
Land Use Map 1548 words (5 Levels)	R = 3.23849 Comp: 15.518 sec Decomp: 17.653 sec 478 words	R = 1.4452 Comp: 31.567 sec Decomp: 23.823 sec 1071 words	R = 1.76197 Comp: 30.321 sec Decomp: 20.483 sec 879 words

Comp = Compression Time Decomp = Decompression Time

R = Data Compression Ratio

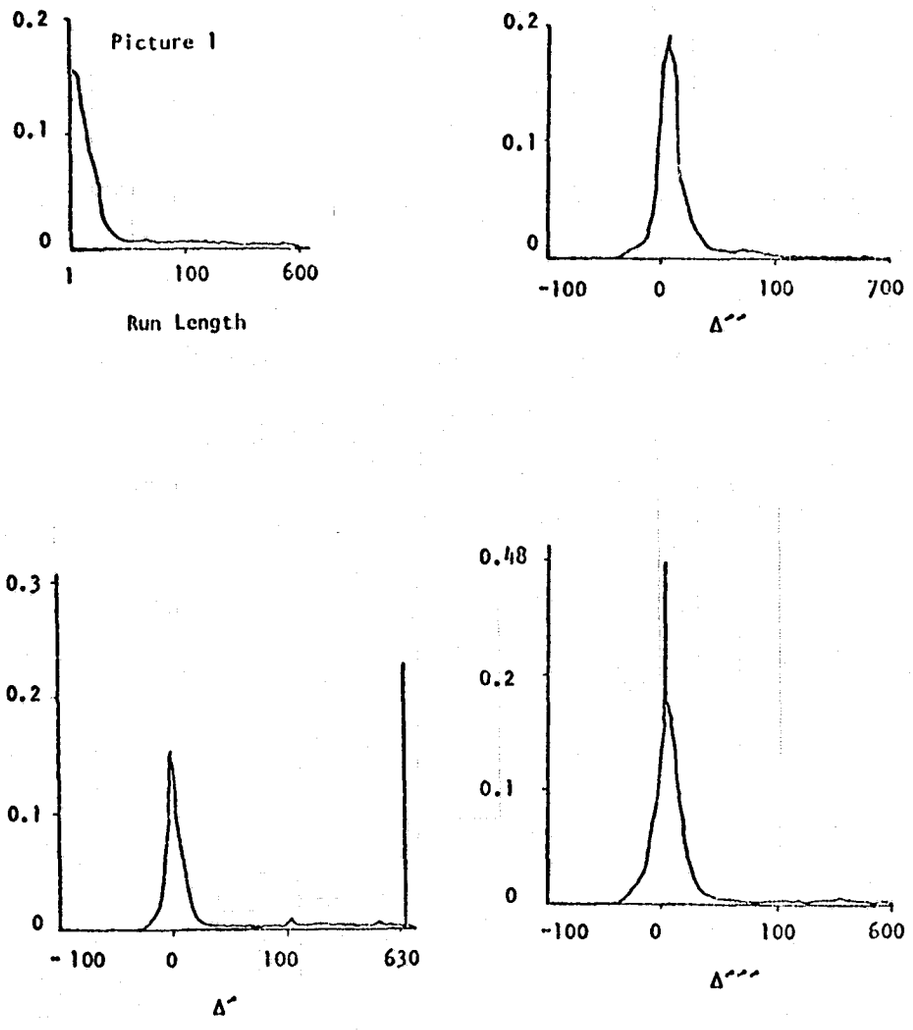


Figure 4.3 Envelope of the Probability Distribution of Indices of Picture 1

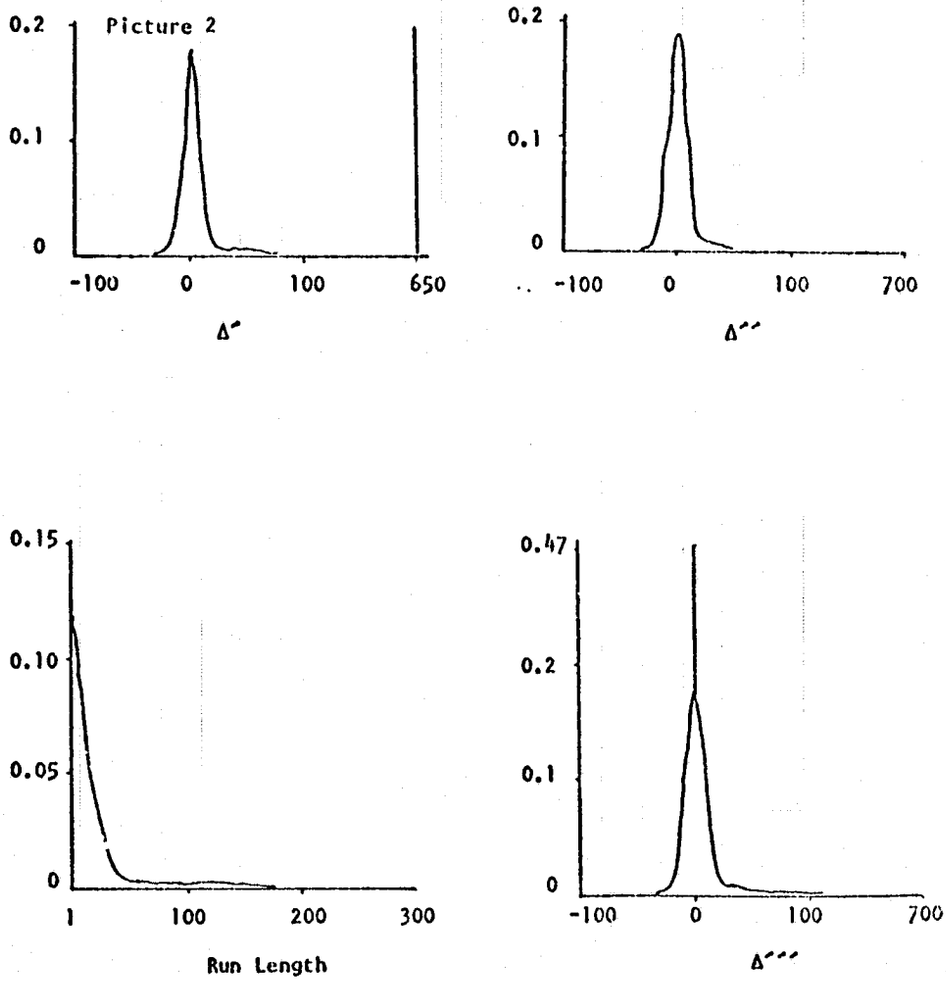
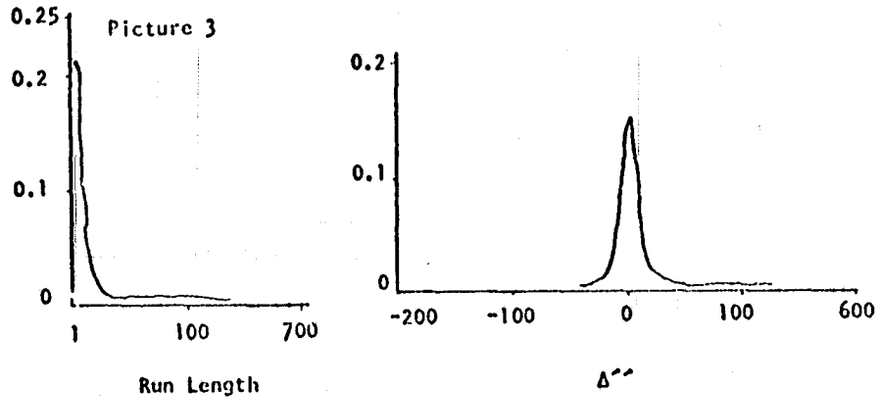


Figure 4.4 Envelope of the Probability Distribution of Indices of Picture 2



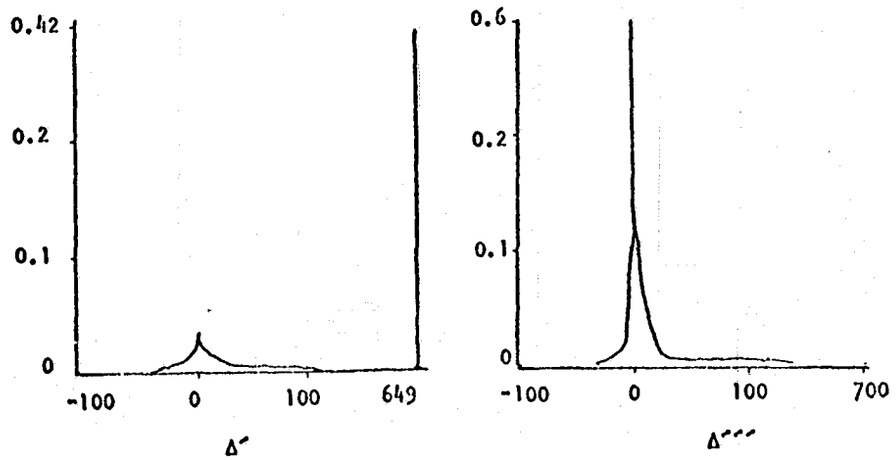


Figure 4.5 Envelope of the Probability Distribution of Indices of Picture 3

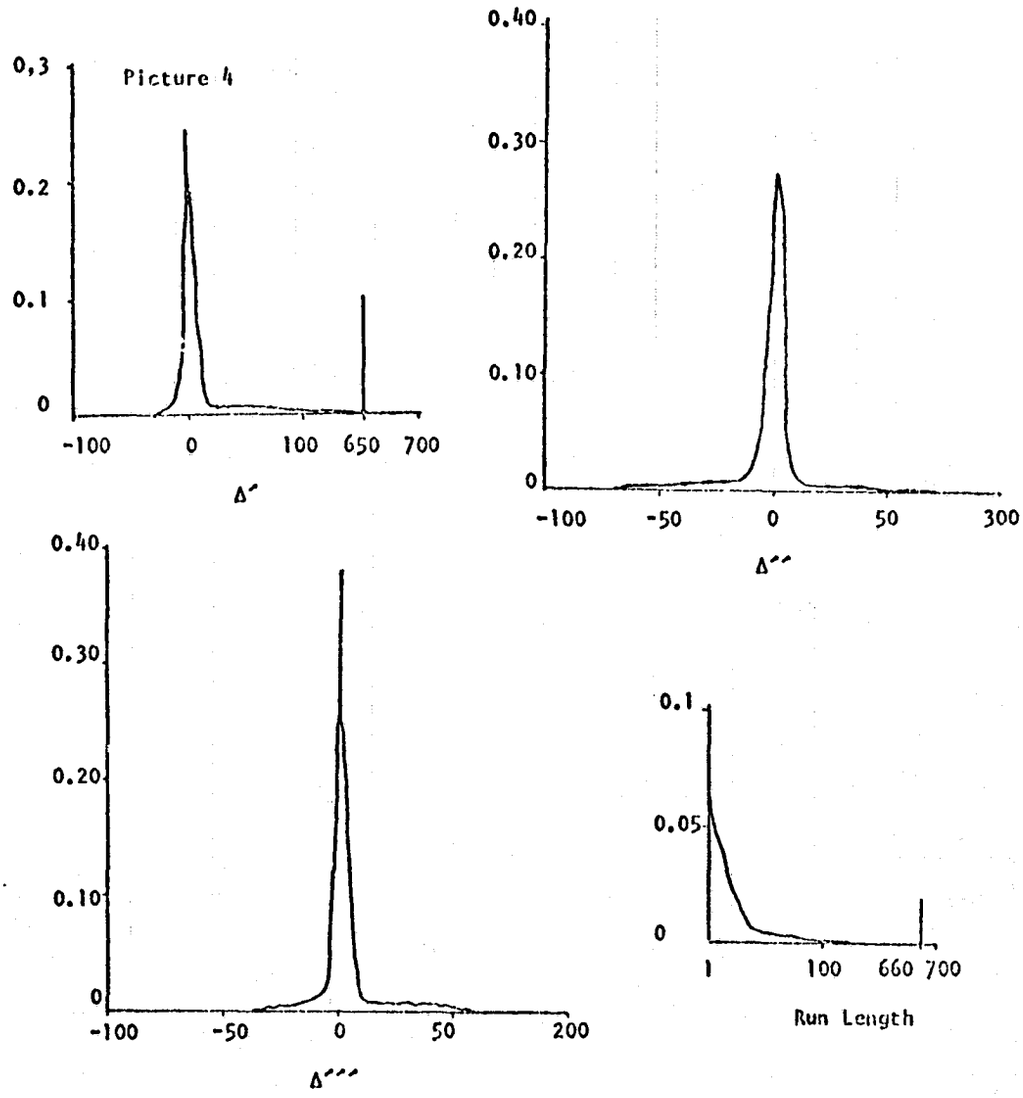


Figure 4.6 Envelope of the Probability Distribution of Indices of Picture 4

of Δ''', however, there is no such spike presented in the curve of Δ''. From the entropy point of view, this is rather important. The entropy for Δ''' will be less than that of Δ'' based on the characteristics of the entropy function. Furthermore, most of the derived data as shown in these figures has the same characteristics in the probability distribution which will make the design of efficient source codes a lot easier. Note that, in the probability distribution of Δ', there is always an impulse located at the end of the abscissa. This represents the probability of the figure terminations of the entire picture.

The figures also imply that it will be more efficient to apply different codes for different indices if the system is allowed to implement several options of codes in the encoder.

4.3 A Mathematical Model for the Derived Data

The probability density functions of run lengths and indices as shown in section 4.2, are exponentially distributed with parameter λ. The curves for the probability density function of Δ'' and Δ''' are roughly symmetric with respect to a dominant spike which is located at the origin of the horizontal axis.

Let the probability density function of Δ be

$$P(X) = \lambda\, e^{-\lambda X} \tag{4.1}$$

where λ is the parameter, and the integer X is the absolute value of Δ.

The probability of X to be between i + 1 and i is

$$P_i = \text{prob}\ \{ i < X < i+1 \} = 2 \int_{X=i}^{X=i+1} P(X)\, dX$$

$$= 2(i - e^{-\lambda})\, e^{-\lambda i} \tag{4.2}$$

Define $a = e^{-\lambda}$ $\quad |a| \leq 1$

Equation (4.2) can be written as

$$P_i = 2(1-a)\, a^i, \text{ for } i = 0,1,2,... \tag{4.3}$$

The entropy per source symbol, Δ, is

$$H = -\sum_i P_i \log P_i \tag{4.4}$$

From (4.4)

$$\begin{aligned} H &= -\{2(1-a)\log_2 2(1-a) + 2(1-a)\,a \log_2 2(1-a)\,a + ...\} \\ &= -\{\log_2[2(1-a)]^{2(1-a)} + \log_2[2(1-a)a]^{2(1-a)a} + ...\} \\ &= -\{\log_2[2(1-a)]^{2(1-a)}\,[2(1-a)a]^{2(1-a)a} ...\} \\ &= -\{\log_2[2(1-a)]^{2(1-a)(1+a+a^2+a^3+...)}\, a^{2(1-a)a+4(1-a)a^2+...}\} \end{aligned} \tag{4.5}$$

Since $2(1-a)(1+a+a^2+a^3+ ...) = (1-a)\frac{2}{1-a} = 2$

and $\quad 2(1-a)a+4(1-a)\,a^2+6(1-a)\,a^3 + ...$

$$= 2(1-a)a[1+2a + 3a^2 + ...]$$

$$= 2(1-a)a\ \frac{1}{(1-a)^2} = \frac{2a}{1-a}$$

Thus,

$$H = -\log_2[2(1-a)]^2\ a^{\frac{2a}{1-a}}]$$

$$= -2[\log_2 2 + \log_2(1-a)] - \frac{2a}{1-a}\log_2 a$$

$$= -2 + 2\log_2\frac{1}{1-a} + \frac{2a}{1-a}\log_2\frac{1}{a} \tag{4.6}$$

To find the upperbound for the entropy function H, let us consider a positive constant λ. $\lambda \neq 0$

Since $\ell n\ (1+\lambda) \leq \lambda$

i.e. $e^{-\lambda}(1+\lambda) \leq 1$

or $\lambda e^{-\lambda} \leq 1-e^{-\lambda}$ (4.7)

From (4.7)

$$\frac{\lambda e^{-\lambda}}{1 - e^{-\lambda}} \leq 1 \tag{4.8}$$

Again, let $a = e^{-\lambda}$ then $\ell n\ a^{-1} = \lambda$

Equation (4.8) can be written as

$$\frac{2a}{1-a}\ \ell n\ (\frac{1}{a}) \leq 2 \tag{4.9}$$

or $$(\frac{1}{a})^{\frac{2a}{1-a}} \leq e^2 \tag{4.10}$$

Multiplying (4.10) by $(\frac{1}{1-a})^2$

$$(\frac{1}{a})^{\frac{2a}{1-a}}\ (\frac{1}{1-a})^2 \leq (\frac{e}{1-a})^2 \tag{4.11}$$

Taking logarithm of base 2 on both sides of eq. (4.11)

we have

$$(\frac{2a}{1-a})\ \log_2(\frac{1}{a}) + 2\ \log_2 \frac{1}{1-a} \leq 2\ \log_2(\frac{e}{1-a}) \tag{4.12}$$

From eq. (4.12) and (4.6)

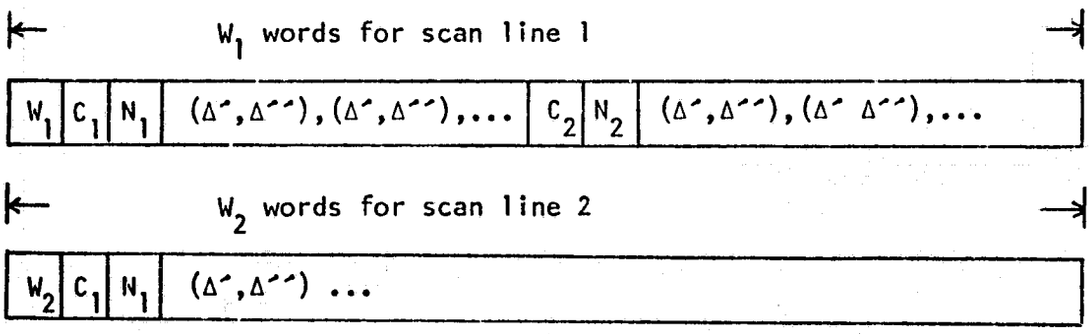
we conclude that

$$H \leq 2(\ \log_2(\frac{e}{1-a}) - 1) \tag{4.13}$$

4.4 Compressed Data Format and Block Packing

As the redundancy of the images was removed or reduced by the redundancy remover and encoded with the adequate variable length codes, the bit string has to be packed in an efficient format such that the transmitted data can be decoded correctly at the receiver. Furthermore, the compressed data format protects the data from the serious error propagation and confines the effect of the errors to the minimum extent such that the reproduction of the image can still be obtained with minimum error. In fact, the derived data from the data compression is usually very sensitive to the errors in the sense that if some of the derived data was corrupted, the original picture can not be faithfully reproduced. In other words, one single error in the derived data of a scan line may corrupt the rest of the scan line which will displace the figures in the picture.

We present a compressed data format for the derived data of PDQ or DDC in Figure 4.7.



W_i = word count

C_i = class code for class i

N_i = number of index pairs

Figure 4.7 Compressed Data Format for PDQ or DDC

The number of bits for W_i depends on the size of the image and the complexity of the figures. The length of W_i has to remain the same through all the scan lines. We can use the natural code as the class code since it will be more convenient to use a fixed length code to encode the class number. The chance of errors will be reduced. The number of the index pairs, N_i, is used to ensure the number of transmitted index pairs which is very helpful in the decoding process.

To initiate the new starting figures in a scan line, we pack the index pairs in a format so that no special code word for the initiation of new starting figures is necessary.

Let us consider Figure 4.8 as an example.

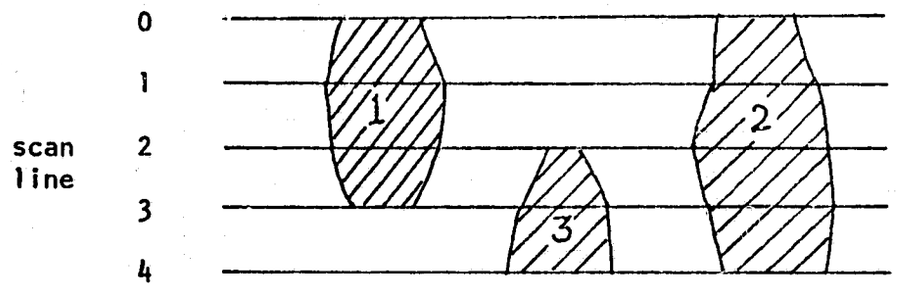


Figure 4.8 An Example of Three Figures in a Picture

We note that there is a new starting figure in scan line 2. Figure 1 terminated on scan line 3. When the indices of the figures were found, they are arranged by the sequential order that index pairs for the old figures were packed first according to the sequence that occurred in the scan line, followed by the index pairs of the new figures.

Thus the index pairs for the four scan lines are arranged as follows:

1. $(\Delta_1', \Delta_1'')(\Delta_2', \Delta_2'')$

2. $(\Delta_1', \Delta_1'')(\Delta_2', \Delta_2'')(\Delta_3', \Delta_3'')$

3. $(\Delta_1', \Delta_1'')(\Delta_2', \Delta_2'')(\Delta_3', \Delta_3'')$

4. $(\Delta_2', \Delta_2'')(\Delta_3', \Delta_3'')$

Note that in the scan line 2, Δ_3' denotes the starting column number, while Δ_3'' denotes the width of figure 3. In scan line 3, Δ_1' represents the termination code (i.e., n+1, if there are n samples per scan line), and Δ_1'' represents the width of the merging figure 1.

The number of index pairs varies from line to line. The indices are encoded by the variable length code and packed into the compressed data format. Thus, the length of the compressed data format for each scan line also varies from line to line.

In order to transmit the data string packed in the compressed data format at a constant bit rate, we have to divide the data string into fixed length blocks. Each block consists of fixed number of words. And the number of bits in a word depends on the computer. Through the help of the word count packed in front of the sequence of index pairs for each scan line, the start and end of the scan line still can be recognized at the receiving end.

Once the complete information of a scan line was received at the receiver, the decoding process can be done as follows:

Step (1) Read the number from word count, W_i.

Step (2) Decode the class code and the number of figures.

Step (3) Decode the index pairs.

Step (4) Update the column number and width for old figures.

Step (5) Locate the new starting column number and width for the new figures occurring in the same scan line.

Step (6) Repeat Step (2) to (5) if a multi-class picture is concerned.

Step (7) Reconstruct the scan line with appropriate symbols.

Step (8) Repeat Step (1) to (7) until the last scan line is reached.

Thus, an error free reconstructed picture can be obtained provided that the channel noise was not introduced in the data transmission.

CHAPTER 5

DATA COMPRESSION ON SATELLITE IMAGERY

The image taken by the remote sensing satellite is usually a two-dimensional array of sampled data. Each element of the array is a picture element which is quantized into m-bits of data. Thus, each picture element assumes one of 2^m grey levels. To transmit a picture of size $\ell x n$, we have to transfer $mx\ell xn$ bits per channel. For most recent multispectral satellite data, 7 bits per picture element is used. Those pictures with better signal to noise performance were compressed into 6 bits per picture element by a nonlinear scale. For example, pictures in channel 1 to 3 of the four-channel LANDSAT pictures were compressed before they are transmitted [58]. When the transmitted image is received at the ground station, it is requantized into 7 bits of data and one zero per picture element through an algorithmic scale.

With multisoectral satellite images, the total number of binary digits needed to be transmitted is great. The data will be transmitted at 120 mega bits per second, which is the rate designed for new LANDSAT-D THERMATIC MAPPER. Only a few ground stations are presently able to handle such a high transmission rate. In order to make those ground stations with lower receiving capacity accessible to the system, some efficient data compression schemes are expected to compress the data such that the same amount of information can be transferred to the ground at a lower transmission rate in the same period of time used by

the high transmission rate. This will eanble the satellite to transmit the data at several different rates simultaneously. A satisfactory representaiton of the digital image (e.g. satellite images) requires more binary bits than the entropy of the image. Fortunately, the redundancy presented in the imagery data can often be reduced through various coding schemes. This chapter deals with those promising coding techniques which are best suited to our goal based on the practical point of view and low-cost principle. We also assume a general application of the received images. Therefore, those techniques which deal with the pictures after classification is done are not considered.

5.1 Definition of Our Goal

As the information extracted from remotely sensed satellite images is shown to be very valuable and important for the investigation of man's environments and estimation of the earth's natural resources, many nations in the world have become more interested in participating in the program of a near real time data acquisition system. A high data transmission rate would be necessary to transfer the enormous data bulk from the satellite. The LANDSAT-D Thermatic Mapper Satellite which is planned to be launched in 1980 will operate at the rate of 120 mega bits per second. It can be expected that the cost of a high quality ground station able to receive such a high bit rate would be rather expensive. Those nations which cannot afford the cost will be unable to access the system directly.

Based on the assumption that the application of the received satellite data is for a general purpose, our investigation is destined to provide three data transmission rates directly from the satellite such

that the ground stations with receiving capability of 120, 30 and 20 mega bits per second are all able to receive the data directly from the satellite. In other words, data compression ratios of 4 and 6 are desired to reduce the rate from 120 mega bits to 30 or 20 mega bits, respectively.

A certain amount of distortion is introduced during the picture coding in addition to the channel noise in the picture transmission system. Here, we are mainly concerned with the performance of the various coding schemes for the digitized data. Therefore, we defined a fidelity criterion which is used as a measure of the reconstructed picture quality when compared to the original picture excluding the channel noise.

Suppose that the original picture consists of a n x m array of picture elements X_{ij}, where i = 1,2,...,n, j = 1,2,...,m. Each picture element X_{ij} is K binary bit word corresponding to one of the 2^K possible grey levels. The encoder compresses the original picture of nxmxK bits to a fewer number of bits. The decoder interprets the received coded messages and reconstructs the picture consisting of nxm array of picture elements Y_{ij}, where i = 1,2,...,n, j = 1,2,...,m.

The root mean square error between the original and reconstructed picture is

$$e_{rms} = \frac{1}{n \cdot m} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} (X_{ij} - Y_{ij})^2 \right]^{\frac{1}{2}} \qquad (5.1)$$

We will use this fidelity criterion to evaluate the performance of picture compression techniques discussed in this chapter. However, the human vision system has a characteristic that two reconstructed

pictures having the same amount of rms error may appear to be quite different to human observers. Indeed, the human vision system is more sensitive to errors in dark areas of the picture than errors in light areas. Therefore, we would not suggest to employ human observers in judging the rms errors in various reconstructed pictures.

5.2 Logarithmic Quantizing

Recall that a satisfactory picture usually requires 7 or more bits per picture elements. Since the amount of the image data needed to be transferred is enormous, it will be more economical to compress the image by utilizing a nonlinear quantization scale and decoding the received message to reconstruct a picture with 7 bits per picture element.

Since the distribution of the digitized sample data is nonuniform, and its distribution varies from one spectrum band to another. In order to compress the data from 7 bits into 6 bits per picture element, it is reasonable to adopt a nonuniform quantizer that increases the size of intervals which are unlikely to be occupied, and reducing the size of those whose usage is highly probable. This will result in a least rms error.

Indeed, this first level of data compression has been used on LANDSAT satellite images. The decompression of 6-bit picture into 7-bit data was done at a NASA ground station. Now we will demonstrate how the logarithmic quantizing scale is developed.

Let the horizontal axis or X-axis represent the nonuniform intervals, and the vertical axis or Y-axis represent the uniform intervals with width one unit. On the X-axis, let the ith interval be

of width δ_i centered on the value X_i. Therefore, any sample amplitude lying in the range,

$$X_i - \frac{1}{2}\delta_i \leq X < X_i + \frac{1}{2}\delta_i \tag{5.1}$$

will be represented by the amplitude X_i. Thus, the instantaneous squared error is $(X - X_i)^2$. Let the probability density of the amplitude distribution of the sampled data be $P(X)$. Then the squared error due to signals falling within this interval is

$$\begin{aligned} E_i^2 &= \int_{X_i - \frac{1}{2}\delta_i}^{X_i + \frac{1}{2}\delta_i} (X - X_i)^2 P(X)\, dx \\ &\simeq P(X_i) \int_{-\frac{1}{2}\delta_i}^{\frac{1}{2}\delta_i} Y^2\, dY = \frac{1}{12} P(X_i)\, \delta_i^3 \end{aligned} \tag{5.2}$$

The probability that the sampled data falls within the ith interval is

$$P_i \triangleq \int_{X_i - \frac{1}{2}\delta_i}^{X_i + \frac{1}{2}\delta_i} P(X)\, dx \simeq P(X_i)\, \delta_i \tag{5.3}$$

From (5.2) and (5.3)

$$E_i^2 = \frac{i}{12} P_i \delta_i^2 \tag{5.4}$$

Let the curve on the XY plane which will map the uniform scale into non-uniform scale and vice versa, be defined as the companding curve, (it will be named as the logarithmic curve due to the logarithmic quantizing). The slope of a chord joining neighboring points on the companding curve is approximately the slope of a tangent to the companding curve at the mid-point of the interval.

$$\left(\frac{dx}{dy}\right)_{X=X_i} \approx \delta_i \tag{5.5}$$

Equation (5.4) is the noise power due to nonuniform quantizing. Substituting (5.5) into (4.5) and summing over all the intervals.

$$E^2 = \frac{1}{12} \sum_i P_i \delta_i^2 = \frac{1}{12} \sum_i P_i \left(\frac{dx}{dy}\right)^2_{X=X_i} \tag{5.6}$$

With many levels in the scale, the summation can be approximated by the integral

$$E^2 = \frac{1}{12} \int P(X) \left(\frac{dx}{dy}\right)^2 dX \tag{5.7}$$

The signal power in terms of the probability distribution $P(X)$ is

$$S^2 = \int P(X)\, X^2\, dX$$

Thus, the signal to noise ratio is

$$\frac{S^2}{E^2} = \frac{\int P(X)\, X^2\, dX}{\frac{1}{12}\int P(X) \left(\frac{dx}{dy}\right)^2 dX} \tag{5.8}$$

We see that, if for certain constant K.

$$\frac{dx}{dy} = K X \tag{5.9}$$

then S^2/E^2 is independent of $P(X)$, or

$$\frac{S^2}{E^2} = \frac{12}{K^2} \tag{5.10}$$

The companding rule which would give this is found by integrating Eq. (5.9)

$$y = K^{-1} \log X \quad \text{for } x > 0 \tag{5.11}$$

For a practical reason, Eq. (5.11) is usually modified into $= K^{-1} \log (1 + \mu X)$ which displace the origin of the XY plane to the intercept on the X axis.

Indeed, the logarithm quantizing was used widely in the speech processing especially in the telephone. For further development in this subject refer to reference [59].

Generally speaking, the advantage of the nonuniform quantizer over the uniform quantizer is that it has more quantization levels in the region which is more likely to occur. This, in turn will reduce the mean square error.

5.3 DPCM

DPCM systems are based on an invention by Cutler [36]. In his invention, the quantized difference between successive sample values were transmitted rather than the sample values themselves. Oliver [60], Kretzmer [28], and Harrison [29] had proposed that the application of the linear predictor in the feedback communication system will reduce the redundancy in the data, and lower the required power in highly periodic signals, or equivalently, achieve some level of data compression.

A simple block diagram of the DPCM system is shown in Fig. 5.1. X(t) is the signal sensed by the sensor implemented on board the satellite. The sample selector provide the function of selecting the picture elements to be processed. This situation occurs when it is necessary to skip some of the picture elements in order to achieve the required data compression ratio.

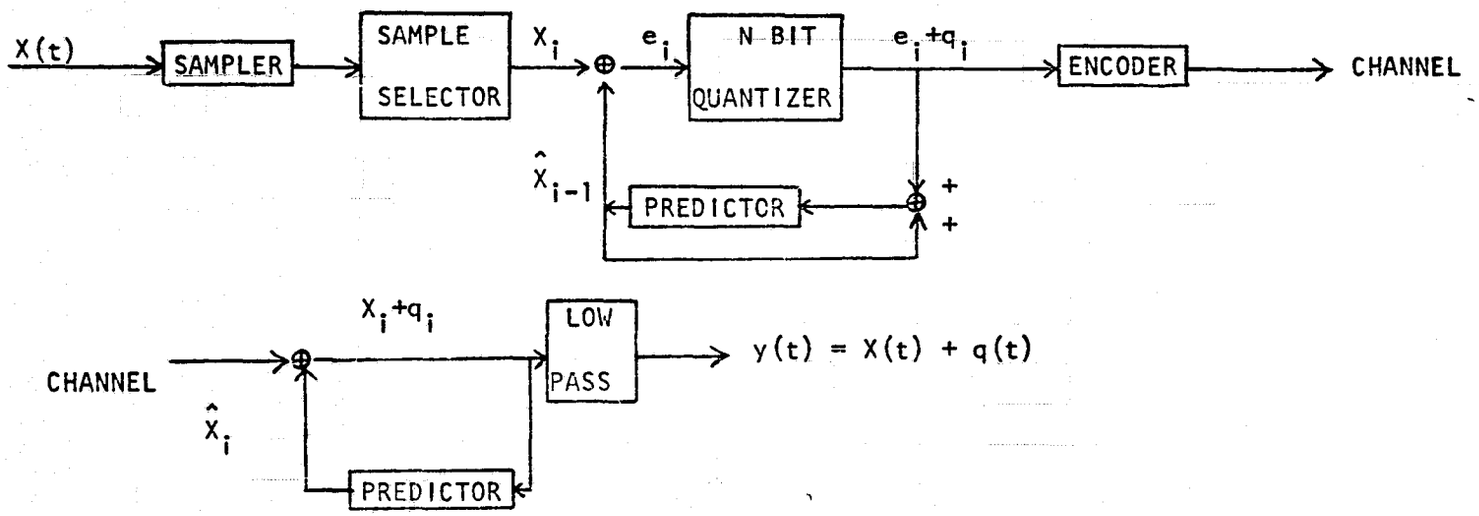


Figure 5.1 Block Diagram of a Modified DPCM System With Sample Selector

Most pictures contain two types of redundancy in picture elements. One of them is the redundancy due to the high correlation between picture elements. It can be removed by the differential pulse code modulation system. T. Kailath [61] said in his paper that every signal can be conceived as having a predictable part which contains no information and a nonpredictable part which contains all the information. DPCM discards the predictable part and encodes only the nonpredictable part and reconstructs the predictable part at the receiver.

The second type of redundancy is due to the nonuniform probability density of the nonpredictable part of the signal. We can remove this type of redundancy by entropy coding using a variable length code. However, in order to do the entropy coding, the probability density function of the data has to be found which may require a buffer to store the signal, thus, some delay would be necessary.

5.3.1 Linear Predictor

Let a stationary signal $X(t)$ with zero mean and variance σ^2 be sampled at the times $t_1, t_2, \ldots, t_n \ldots$ and let the sample values be $X_1, X_2, \ldots X_n, \ldots$ respectively.

A linear estimator of the next sample value X_0 based on the previous n sample values $X_1, X_2, \ldots X_n$, is defined to be

$$\hat{X}_0 = C_1X_1 + C_2X_2 + \ldots + C_nX_n.$$

For simplicity, we assume that C's and X's are real numbers and integers respectively. A linear predictor provides the estimate $\hat{X}_0$, and transmits the difference (or error)

$$e_0 = X_0 - \hat{X}_0$$

$\hat{X}_0$ is defined to be the best estimate of X_0 for which the expected value of the squared error is minimized.

The optimum choices of the coefficients C's which satisfy the minimum mean square error criterion can be found by taking the partial derivatives of $E[(X_0 - \hat{X}_0)^2]$ with respect to each one of the C's and setting the equation to zero, we are ready to have

$$E[(X_0 - (C_1X_1 + C_2X_2 + \ldots + C_nX_n)\ X_i] = 0 \qquad (5.12)$$

or

$$E[(X_0 - \hat{X}_0)\ X_i] = 0 \qquad i = 1,2,\ldots n \qquad (5.13)$$

Let $R_{ij} = E[X_iX_j]$ be the covariance of X_i and X_j then the condition for the optimum linear estimate is

$$R_{0i} = C_1R_{1i} + C_2R_{2i} + \ldots + C_nR_{ni} \qquad i = 1,2,\ldots n.$$

The sequence $\{e_i\}$ is less correlated and has smaller variance than the sequence $\{X_i\}$ from the fact that

$$\sigma_e^2 = E[(X_0 - \hat{X}_0)^2] = E[(X_0 - \hat{X}_0)\ X_0] \qquad (5.14)$$

or

$$\sigma_e^2 = R_{00} - (C_1R_{01} + \ldots C_nR_{0n}) \qquad (5.15)$$

If the sequence $\{X_i\}$ is a first order Markoff sequence then only one previous sample is needed to be used in finding the best estimate of X_0.

5.3.2 DPCM with Natural Code

The difference between the actual sampled value and the linear estimate is quantized into one of 2^N levels (N bits) either uniformly or nonuniformly. The qunatizer represents each value of e_i by the

nearest one of 2^N quantizing levels. Then, a natural code is used to encode each quantizing level with N binary bits. A buffer may be necessary to store the coded bit string, then transmit at a constant rate over a channel. The received quantizing levels are used to reconstruct the picture. The difference between e_i and the quantizing level used to represent it is called the quantizing error, q_i. Thus, the messages being transmitted are the sequence $\{e_i + q_i\}$.

Our goal is to achieve the data compression ratios of four and six under the constraint of minimum root mean square error. In other words, compressed pictures with average 1.5 bits and 1 bit per picture elements are desired. The natural code provides a convenient encoder which maps each quantizing level into a fixed length code word (N bits per code word if a N-bit quantizer is used). Suppose that a picture processed by the regular DPCM is transmitted with 6 bits per picture element. A modified DPCM as shown in Fig. 5.1 is designed to provide the desired compression ratios. With the natural code, the data compression ratio is fixed once the n-bit quantizer is determined no matter what quantizing scale is applied.

We have simulated the SKYLAB IV pictures on the computer by the schemes discussed later in this section. Figs. 5.2 to 5.6 are the original SKYLAB IV pictures which were taken in January of 1974. These pictures provide information about the snow cover on the ground. For comparison, Fig. 5.2 will be used in computer simulation for various schemes.

The histograms of these pictures are shown in Figs. 5.7 to 5.11. Figs. 5.12 to 5.16 show the histograms of the differences between the

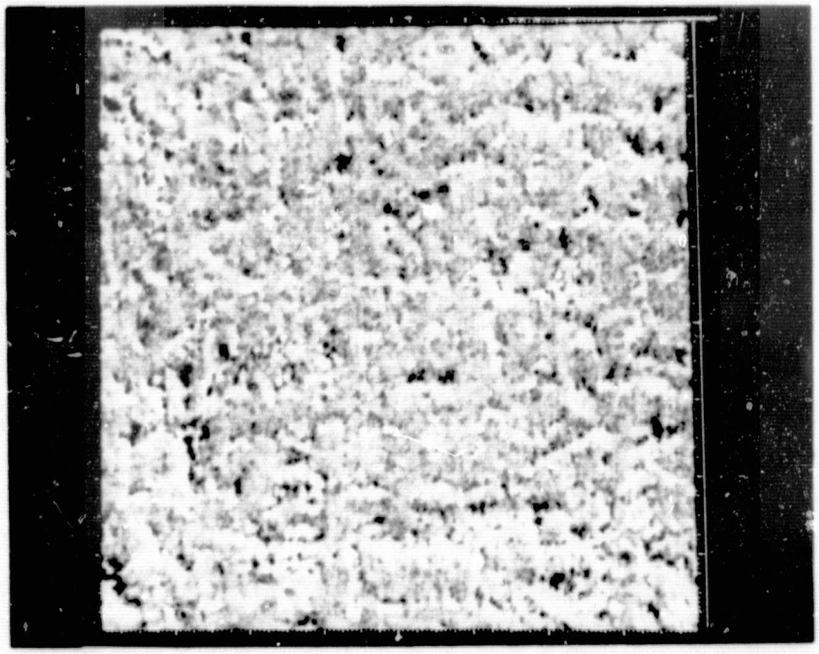


Figure 5.2 Simulated LANDSAT-D Thematic Mapper (0.52-0.61 μm)
From SKYLAB IV Pictures

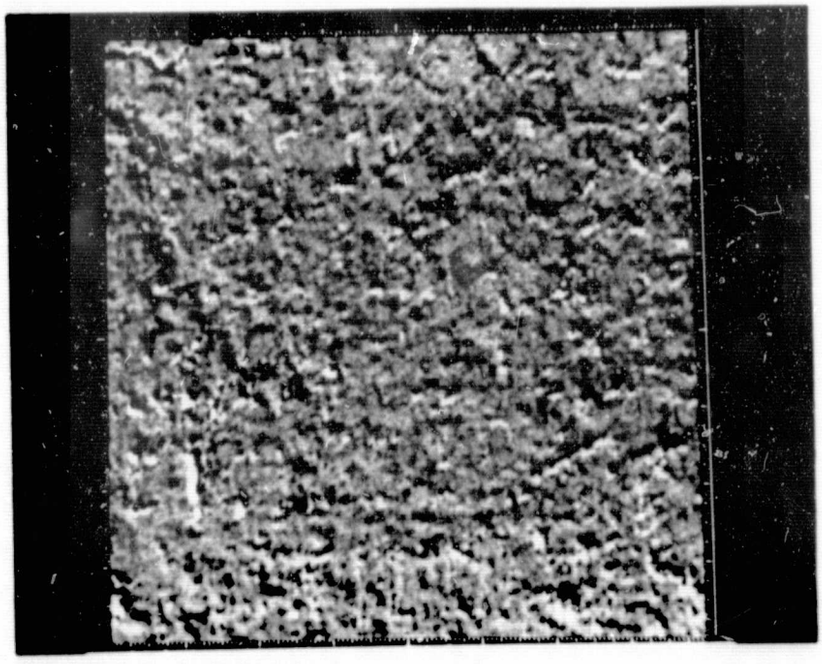


Figure 5.3 Simulated LANDSAT-D Thermatic Mapper (0.68-0.76 μm)
From SKYLAB IV Pictures

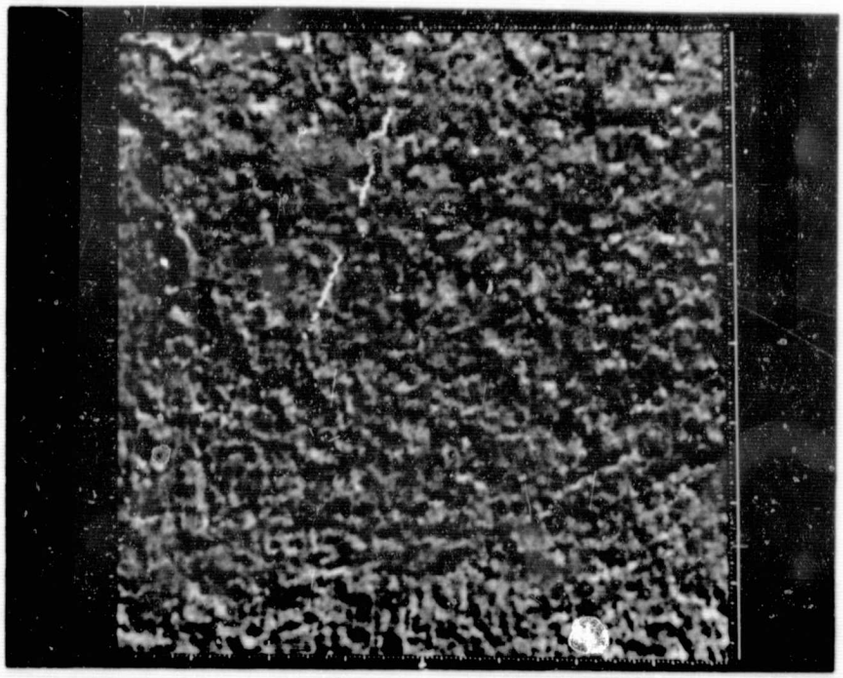


Figure 5.4 Simulated LANDSAT-D Thermatic Mapper (0.78-0.88 μm) From SKYLAB IV Picture

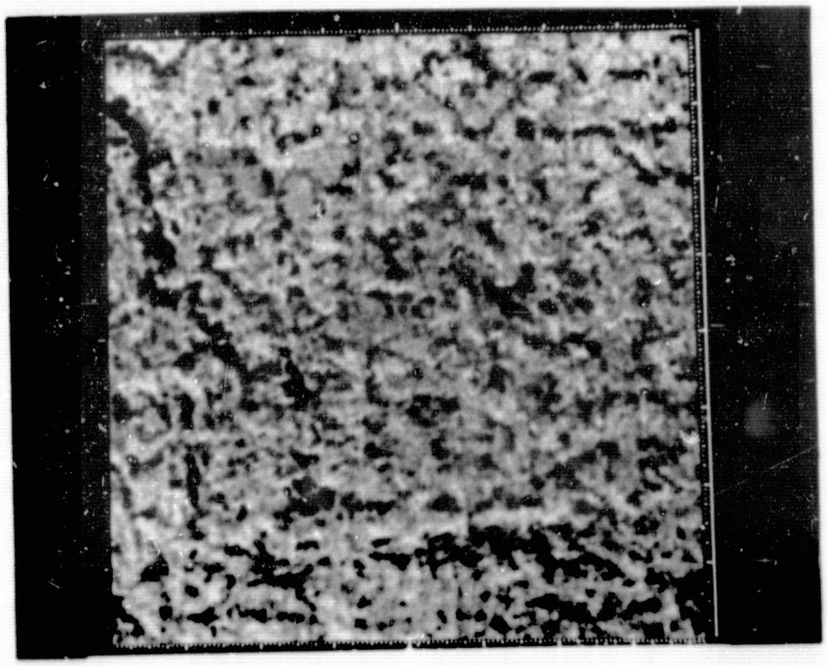


Figure 5.5 Simulated LANDSAT-D Thermatic Mapper (1.55-1.75 μm)
From SKYLAB IV Pictures

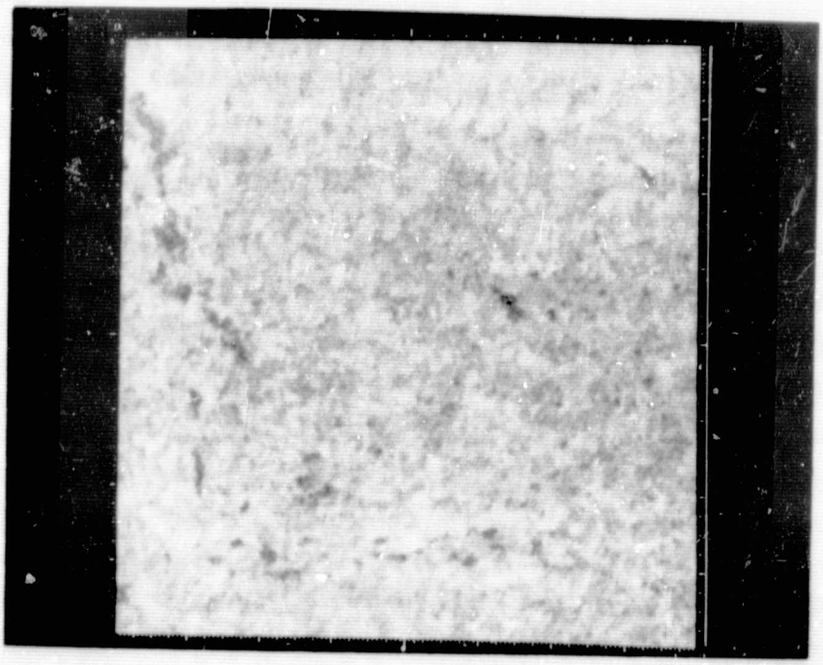


Figure 5.6 Simulated LANDSAT-D Thermatic Mapper (10.20-12.50 μm)
From SKYLAB IV Pictures

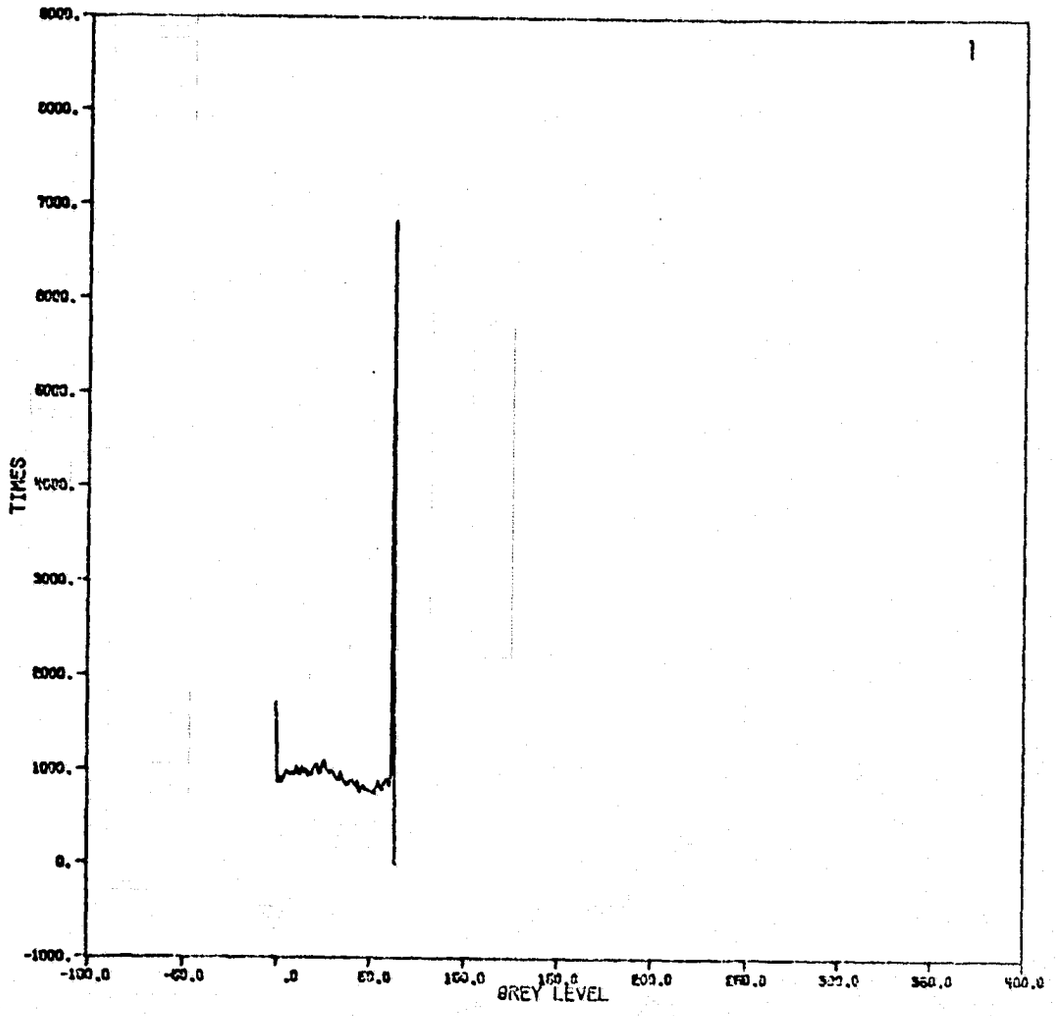


Figure 5.7 Histogram of simulated LANDSAT-D Thematic Mapper (0.52-0.61 μm)

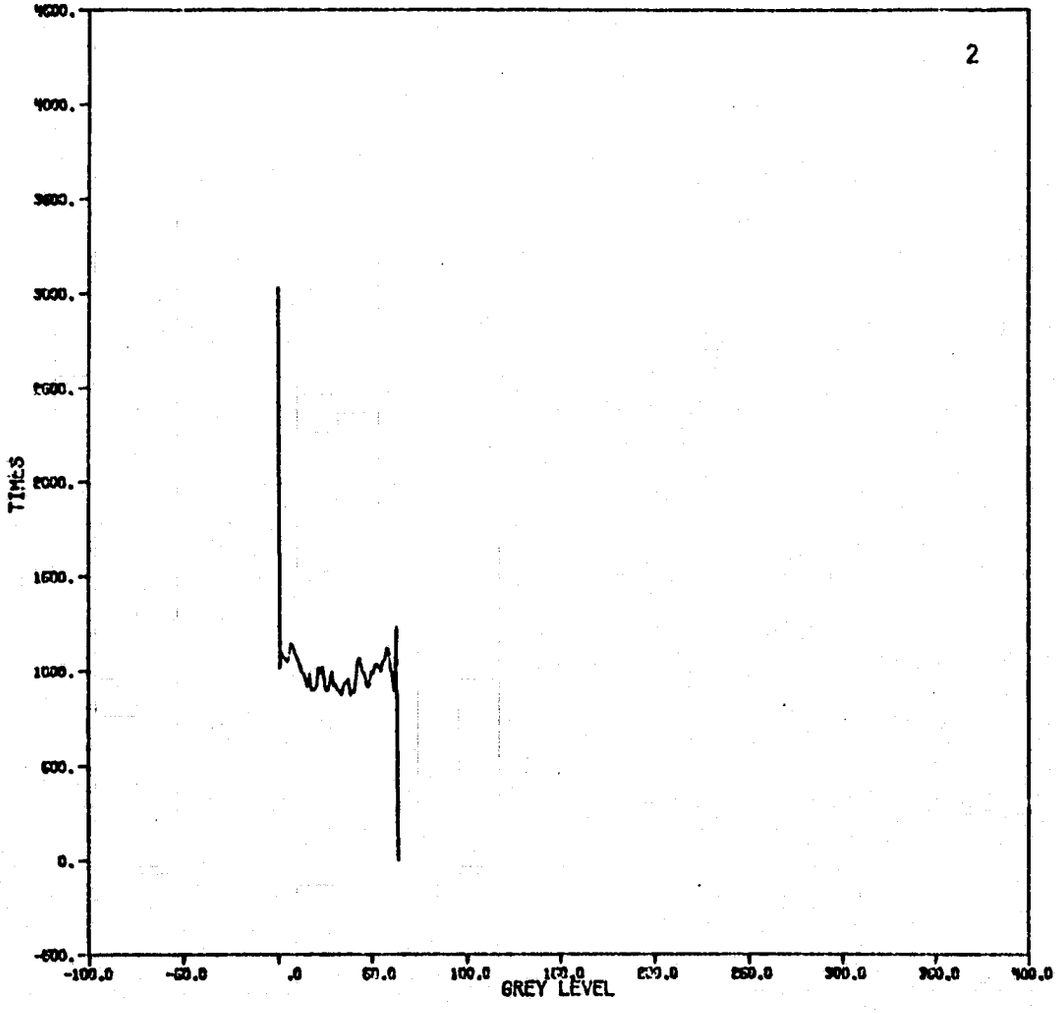


Figure 5.8 Histogram of Simulated LANDSAT-D Thematic Mapper (0.68-0.76 μm)

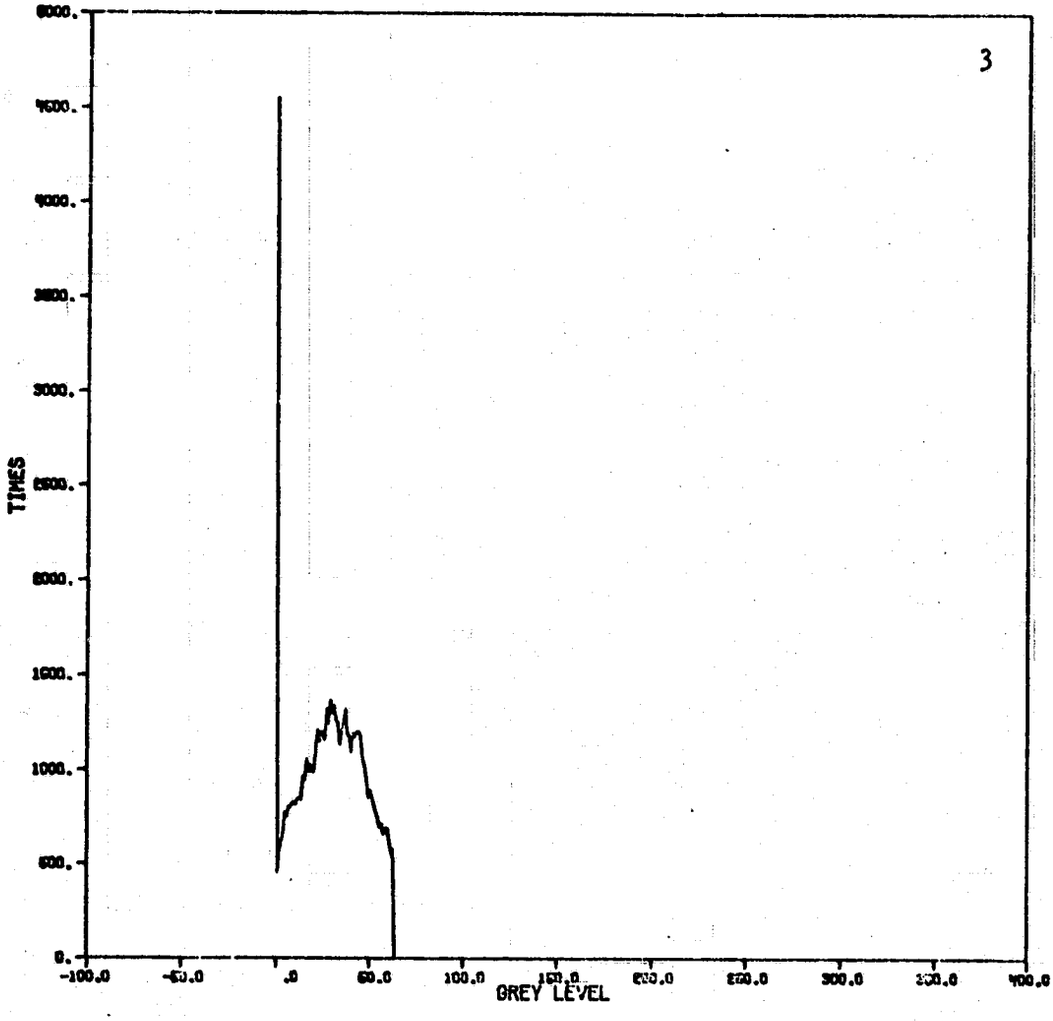


Figure 5.9 Histogram of Simulated LANDSAT-D Thematic Mapper (0.78-0.88 μm)

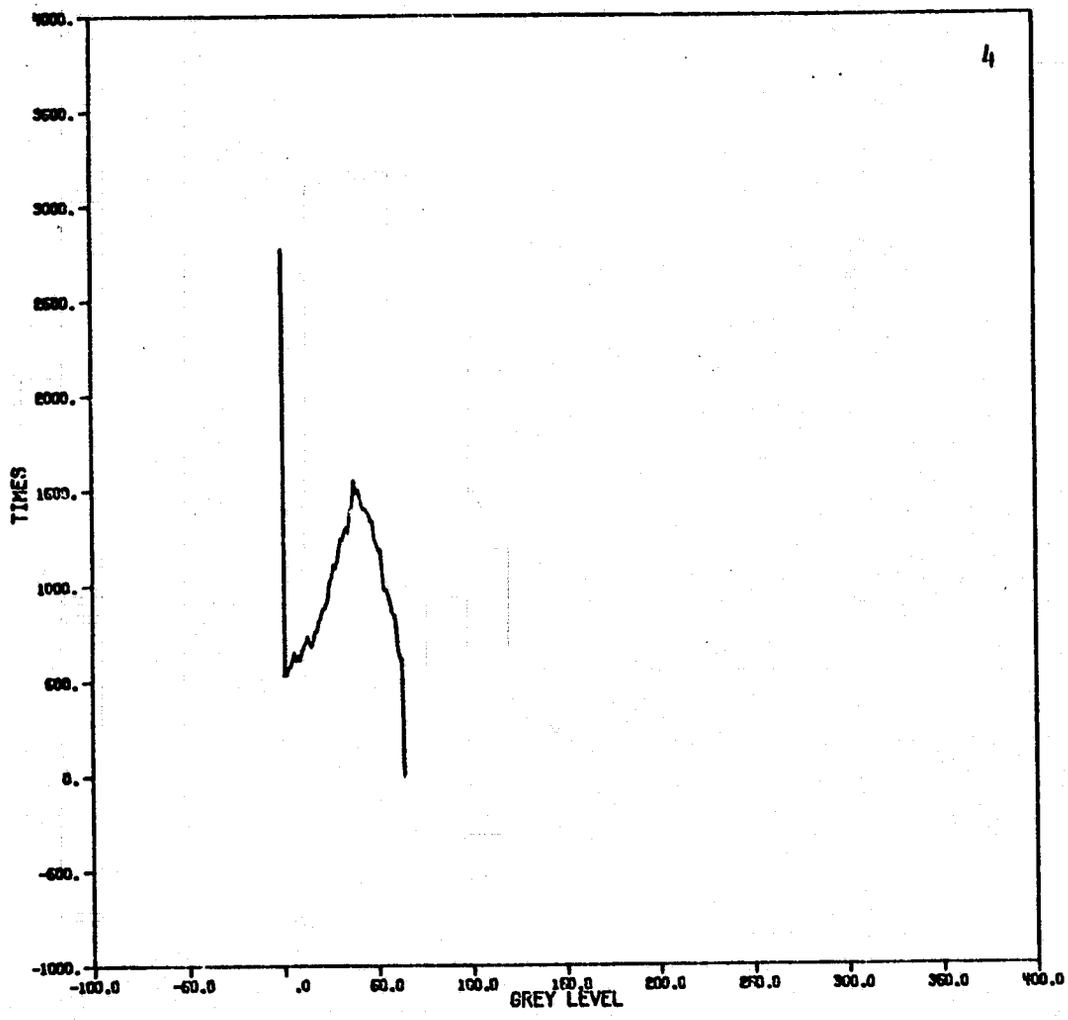


Figure 5.10 Histogram of Simulated LANDSAT-D Thematic Mapper (1.55-1.75 μm)

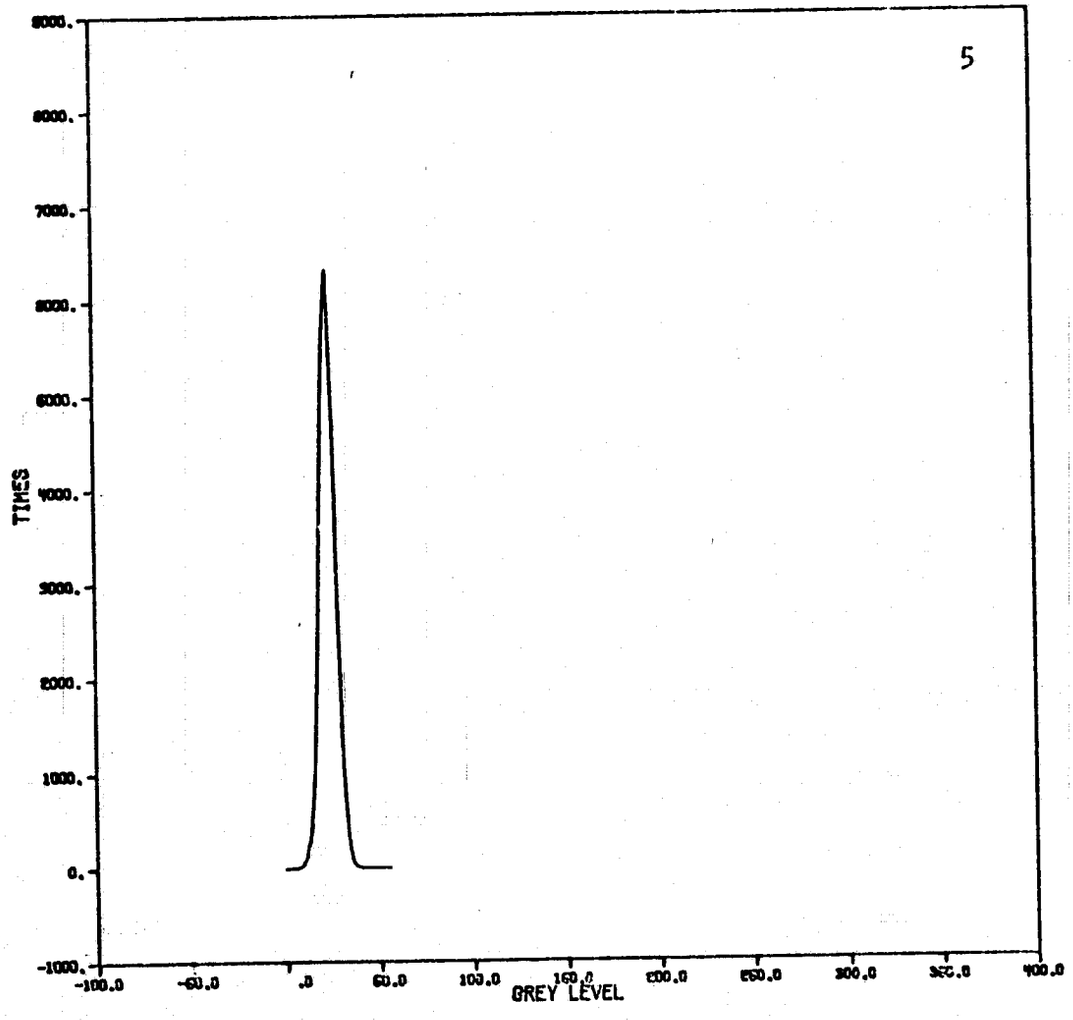


Figure 5.11 Histogram of Simulated LANDSAT-D Thematic Mapper (10.20-12.50 μm)

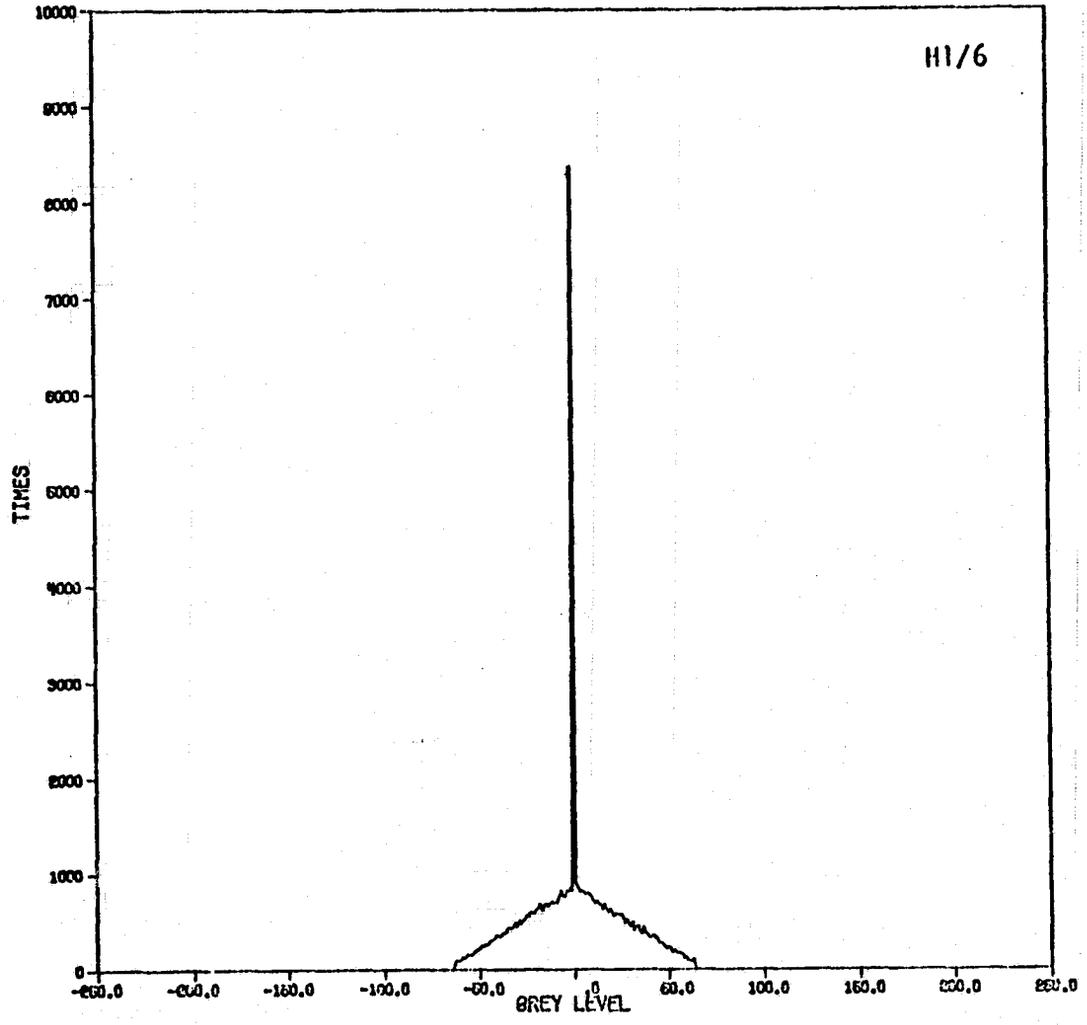


Figure5.12 Histogram of the Differences of the Neighboring Pels for Simulated LAND SAT-D Thematic Mapper(0.52∿0.61 μm)

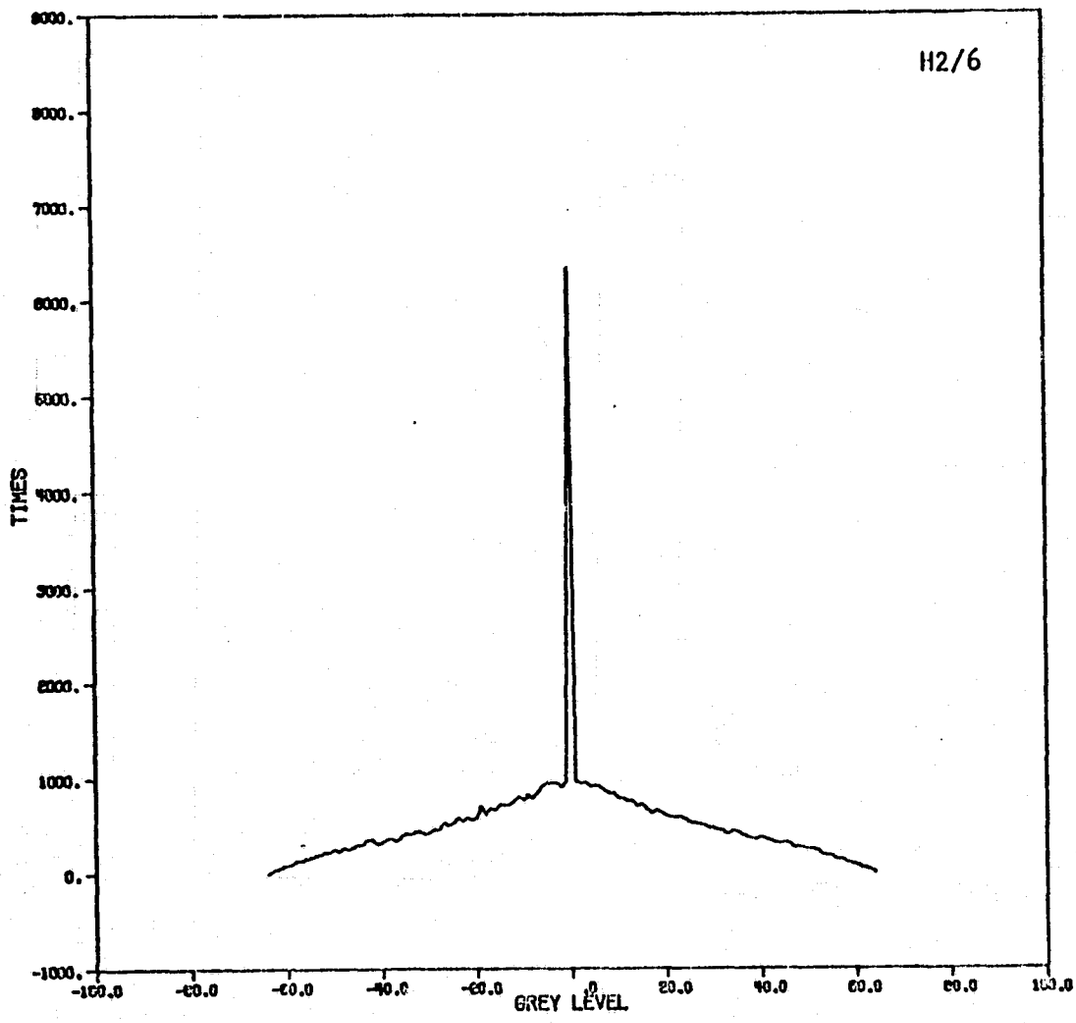


Figure5.13 Histogram of the Differences of the Neighboring Pels for Simulated LAND SAT-D Thematic Mapper(0.68∿0.76 μm)

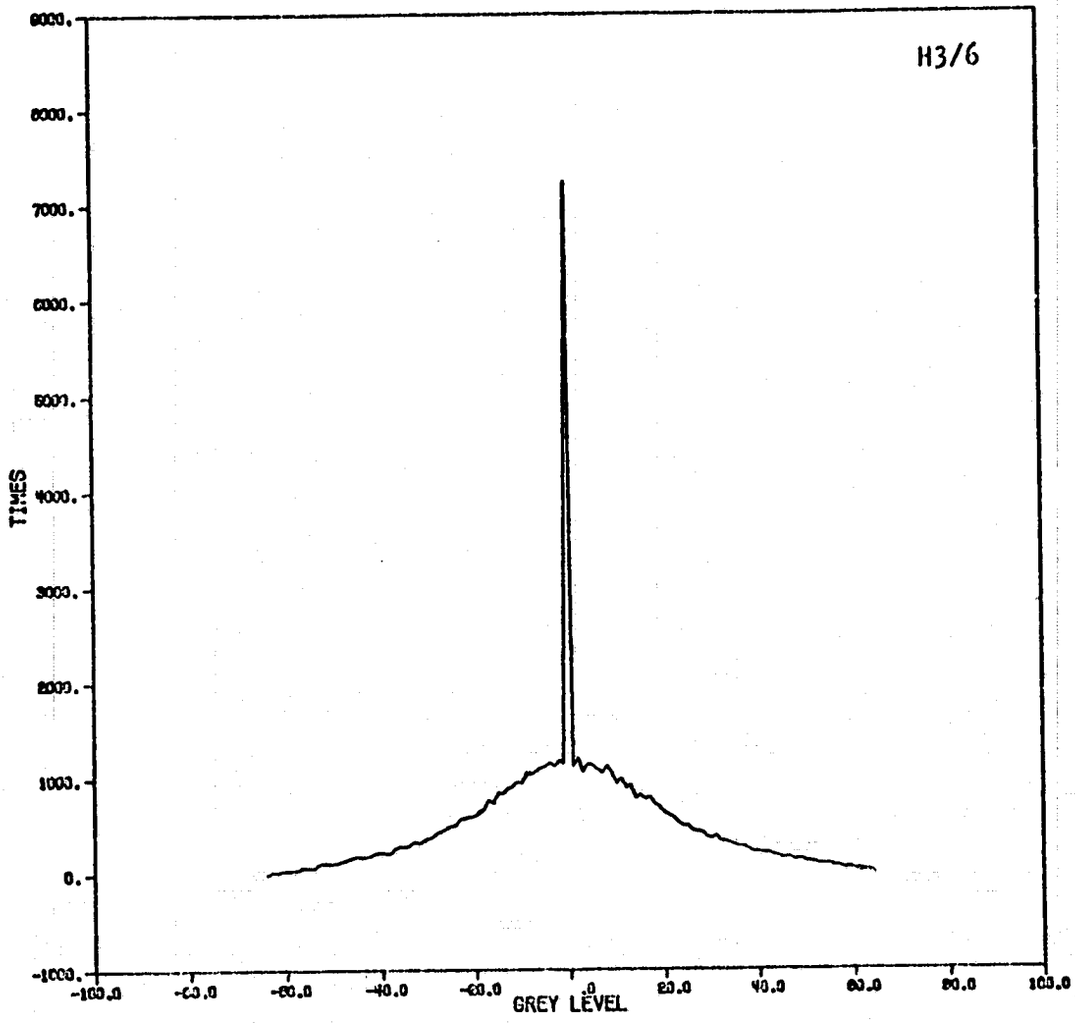


Figure5.14 Histogram of the Differences of the Neighboring pels for Simulated LAND SAT-D Thematic Mapper(0.78∿0.88 μm)

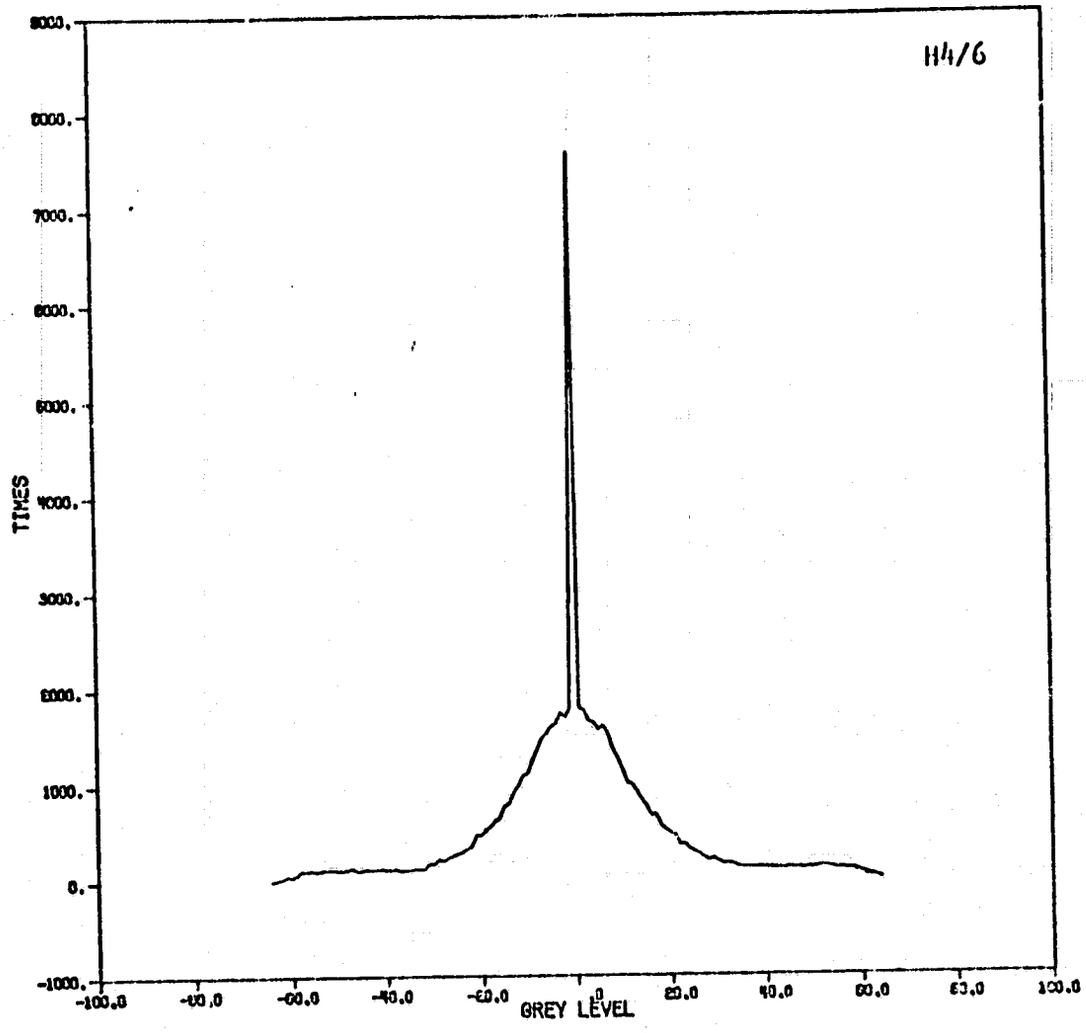


Figure5.15 Histogram of the Differences of the Neighboring pels for Simulated LAND SAT-D Thematic Mapper(1.55~1.75 μm)

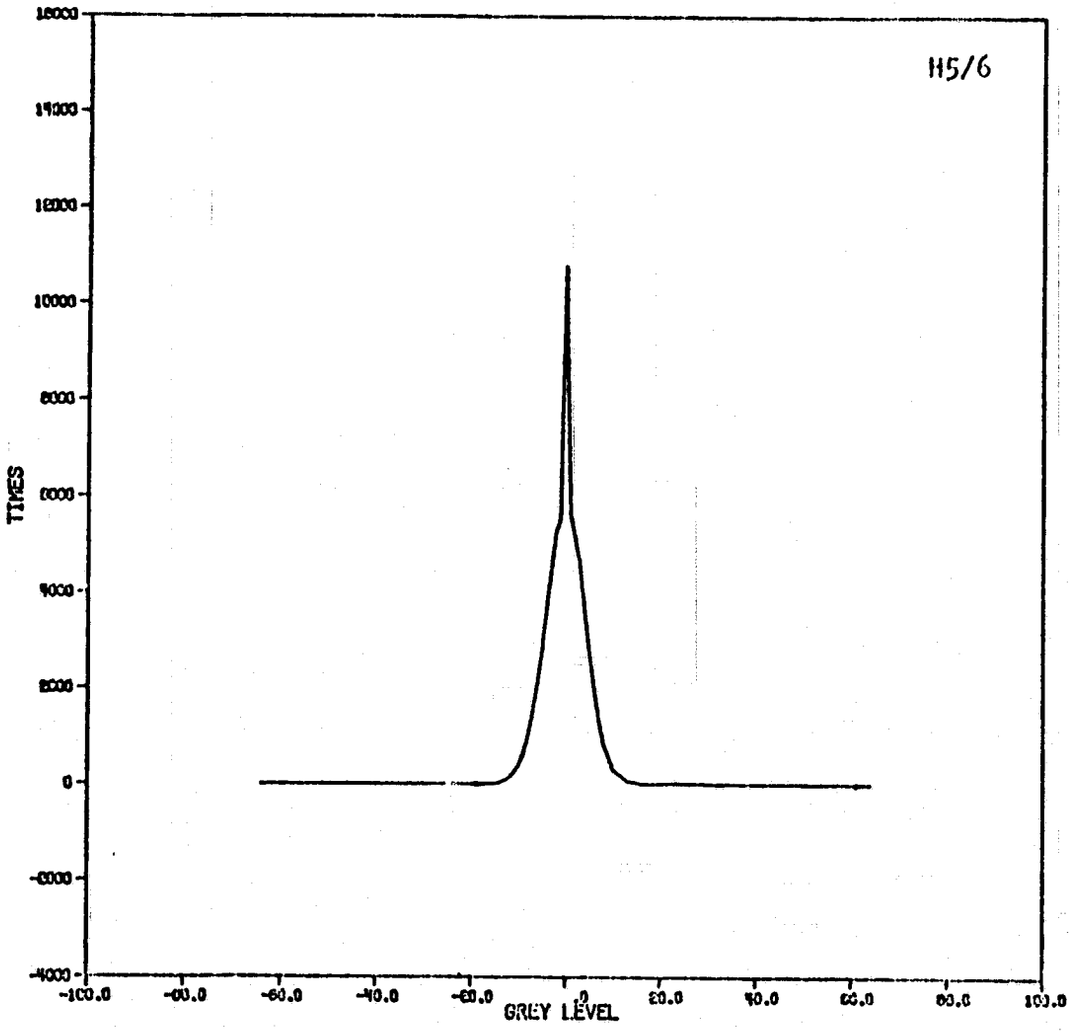


Figure5.16 Histogram of the Differences of the Neighboring pels for Simulated LAND SAT-D Thematic Mapper(10.20~12.50 μm)

neighboring picture elements (i.e., $X_i - X_{i-1}$). Some statistics of the SKYLAB IV pictures were presented in Table 5.1 through 5.2.

From the histograms of $X_i - X_{i-1}$, we can see that the probability of the difference between neighboring picture elements which ranged from -10 to 10 is rather high. To minimize the rms error due to quantization, an optimum nonuniform quantizer with unequally spaced levels will be the best candidate. Some sophisticated methods suggested by Max [37] are very attractive. However, they are still too complicated in practical operations. Indeed, our simulations showed that DPCM with an optimum uniform quantizer did provide reasonably good results. Although they are slightly inferior to the DPCM system with a nonuniform quantizer. In order to provide almost real time transmission, and to eliminate the time delay due to calculation of the data statistics, and nonuniform quantizing, a DPCM with optimum uniform quantizing will be a better system.

The optimum uniform quantizer will map the data into one of a finite set $\{V_1,\ldots V_M\}$ of integers:
In other words

$$U_j < X_i \leq U_{j+1} \rightarrow Y_i = V_j \quad (j=1,\ldots M) \tag{5.16}$$

The set $\{U_j\}$ is called the transition levels, and the set $\{V_j\}$ is the quantization level. We defined $U_{j+1} - U_j = r$ where r is an integer. The quantization value V_j is assigned to

$$V_j = U_j + K \cdot r \tag{5.17}$$

where K is a real number with magnitude less than 1.
Then

$$V_{j+1} - V_j = U_{j+1} + K \cdot r - (U_j + K \cdot r) = r \tag{5.18}$$

Table 5.1 Statistics of the Difference of Neighboring Pixels of SKYLAB IV Images

Channel	Mean	Standard Deviation
4	0	12.182
6	0	11.782
7	0	8.180
11	0	6.08
13	0	1.03

Table 5.2 Statistics of Some SKYLAB IV Images

Channel	Mean	Standard Deviation
4	31	13.01
6	17	9.28
7	9	6.05
11	9.1	4.01
13	20.07	1.5

In the modified DPCM, a sample selector is used to choose the picture elements to be processed by the DPCM system when skipping some picture elements is desired. In general, most of the multispectral satellite images are over sampled by a factor of 1.5 to 1. This will enable us to skip some of the picture elements without causing too much rms error.

We have developed four compression schemes which utilized the modified DPCM system with natural code. Those skipped picture elements were reconstructed by linear interpolation method.

DN 1: Every other picture elements were processed by the modified DPCM system with a 3-bit quantizer. The skipping factor is 2/1 and the data compression ratio achieved is 4.

DN 2: A one-bit quantizer is used with the DPCM system. The transmitted signal only initiates the sign of the error sequence $\{e_i\}$. In fact, this is a delta modulation system. The skipping factor is 1/1 and compression ratio, R, is 6. The rms error of the reconstructed picture depends on the choice of Kr, as shown in Eq. (5.17). If the successive picture elements do not change rapidly and the sampling rate is high enough, this scheme may operate reasonably well. In general it does not seem to be adequate to apply this scheme to any picture with which we are dealing.

DN 3: Apply every other picture elements to the modified DPCM system with a two-bit quantizer. Thus, the skipping factor is 2/1, and R = 6. In this scheme more quantization levels are provided to reduce the rms quantization error.

DN 4: A modified DPCM with a 4-bit quantizer is used. The picture is divided into blocks, each block is a 2x2 array of picture elements. Only the top left picture element of each block is processed. The skipping factor is 4/1. A data compression ratio of 6 is achieved.

For comparison, we present the performance of these four schemes with uniform and nonuniform quantizer in Table 5.3 and 5.4 respectively. The results were obtained by simulating compression schemes on the SKYLAB IV pictures. Notice that skipping more picture elements during the processing will enable us to use a quantizer with more quantization levels, but it also has a drawback that some more error may occur during the interpolation to obtain the skipped picture elements. In general, the extent of skipping factor really depends on the sampling rate of the picture. The reconstructed SKYLAB IV pictures using a uniform quantizer were presented in Figures 5.17 through 5.20.

5.3.3 DPCM With Variable Length Code

To make the best use of a variable length code, we have to compute the histogram or the probability density function of the error sequence $\{e_i\}$. The variable length code is optimized in a way that the error message, e_i, with higher probability of occurrence is mapped to shorter code words, while rarely occurred error messages are assigned to longer code words. Suppose that a N-bit quantizer is used in the DPCM system. The possible range of the error sequence would be divided into 2^N transition levels with nonuniform space. Meanwhile we also have 2^N quantization levels. C_1 code which was discussed before in previous chapter will be used as a variable length code in our computer simulation.

Table 5.3 The Performance of DPCM With Natural Code and Uniform Quantization Where N = No. of Bits for Quantizer R = Data Compression Ratio

Scheme	N	R	rms Error	Skipping Factor
DN 1	3 bits	4	10.52	2/1
DN 2	1 bit	6	10.44	1/1
DN 3	2 bits	6	10.915	2/1
DN 4	4 bits	6	11.01	4/1

Table 5.4 The Performance of DPCM With Natural Code and Nonuniform Quantization Where N = No. of Bits for Quantizer R = Data Compression Ratio

Scheme	N	R	rms Error	Skipping Factor
DN 1	3 bits	4	9.3231	2/1
DN 2	1 bit	6	10.44	1/1
DN 3	2 bits	6	10.04	2/1
DN 4	4 bits	6	10.00	4/1

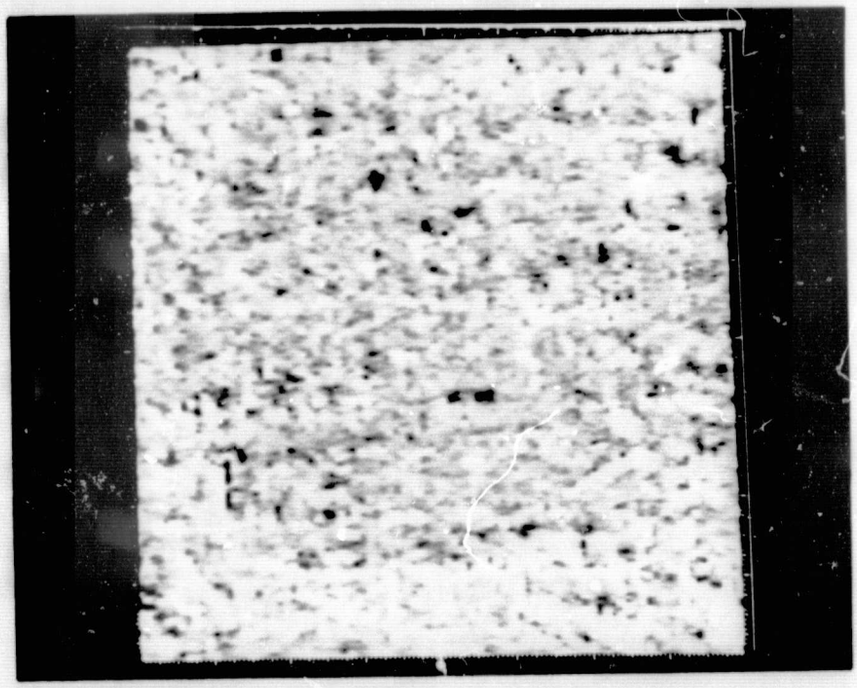


Figure 5.17 Reconstructed Picture From DN 1 Compression Scheme

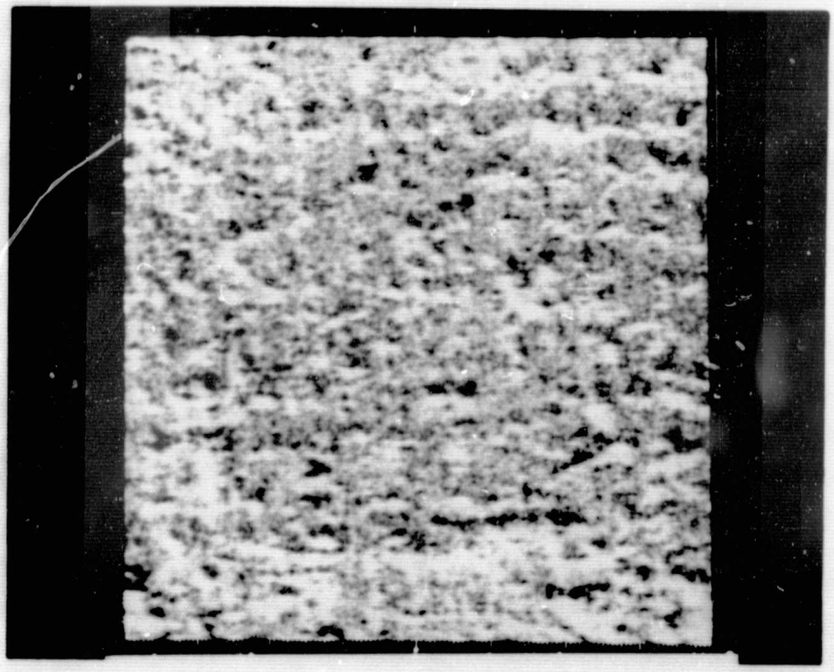


Figure 5.18 Reconstructed Picture From DN 2 Compression Scheme

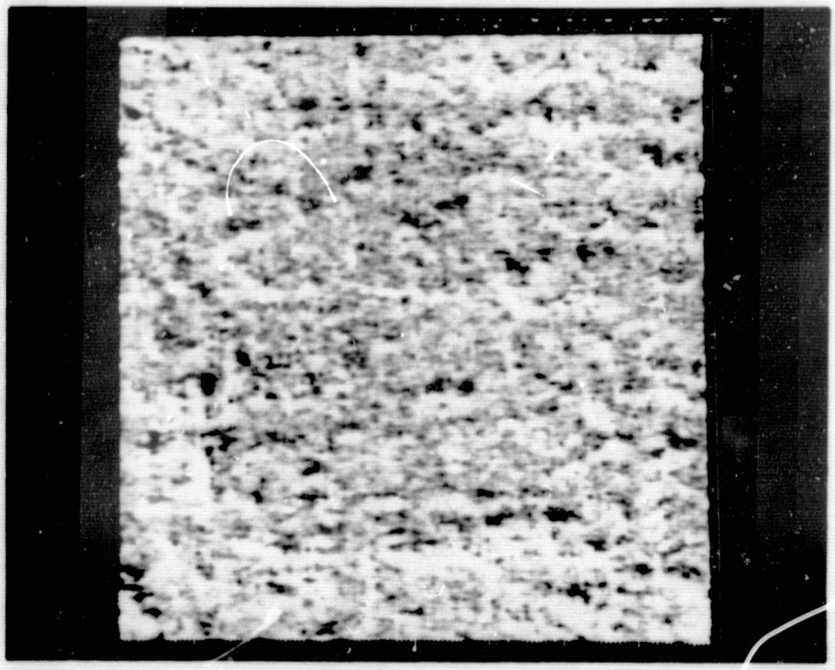


Figure 5.19 Reconstructed Picture From DN 3 Compression Scheme

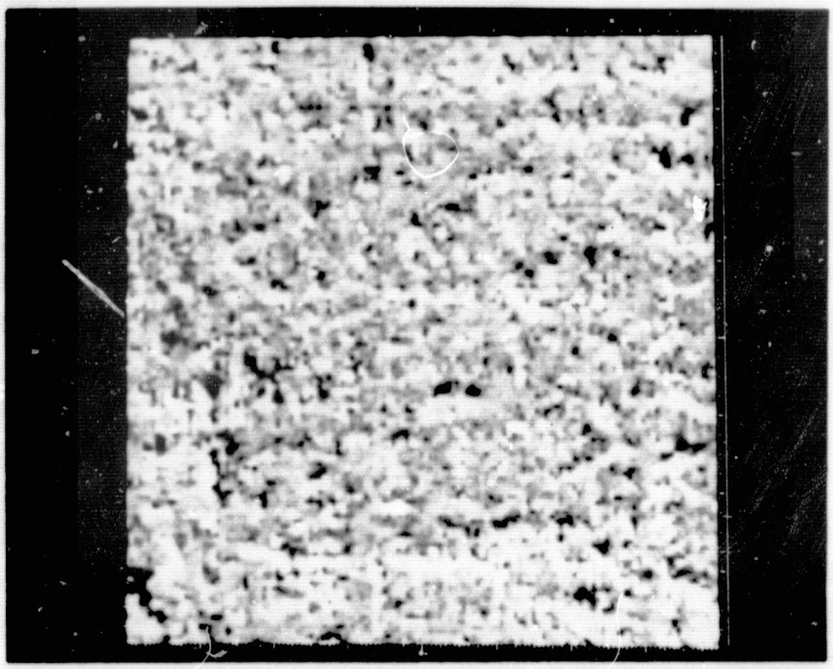


Figure 5.20 Reconstructed Picture From DN 4 Compression Scheme

Among all the code words of C_1 code, there are only two code words of length two bits, which are the shortest. Thus, the performance of 2-bit and 1-bit quantizer with C_1 code will definitely be inferior to the natural code. Also, the length of the intervals which are mapped into these two short code words will be an important factor in the overall data compression ratio. That is to say, assigning two long transition levels to two shortest code words will provide better data compression ratio. However, it aiso will cause a higher quantization error and rms error in the reconstructed picture.

The probability density functions for the error sequence $\{e_i\}$ shown in Figs. 5.12 to 5.16 are somewhat symmetric. They may be approximated by the exponential density function:

$$p(e) = \frac{1}{2\sqrt{\sigma}} \exp\left(- \frac{\sqrt{2}}{\sigma} |e| \right) \tag{5.19}$$

where σ^2 is the variance of the error sequence.

Since the probability density function is different for each picture, the best we can do is to choose some representative density function and match the quantizer to it.

One simple way to obtain the proper quantization and transition levels for minimizing the rms quantizing noise is to form a function, f(e), such that when e takes some value in the range of V to -V, f assumes a proper value. Thus, a table is formed. We can divide the range of -V to V into 2^N transition levels through the help of this table. The quantization value, V_i, between U_i and U_{i-1} in the transition axis can be assigned as

$$V_i = \frac{1}{2} (U_i + U_{i-1}) \tag{5.20}$$

A paper by Smith [62] showed that when the probability density function of the signal to be quantized is that of Eq. (5.19), the function f(e) is given by

$$f(e) = -\frac{V}{m}\ln\left[1 - \frac{e}{V}(1-\exp(-m))\right] \qquad 0 \leq e \tag{5.21}$$

$$f(-e) = f(e) \tag{5.22}$$

where $m = \sqrt{2}\ V/3\sigma$

This scheme is rather simple compared to those more sophisticated method such as Max's optimum quantizer. A table of the Smith nonuniform quantization is shown in Appendix.

We have designed the following schemes in which the modified DPCM system utilizing a nonuniform quantizer and C_1 encoder.

DB 1: A 3-bit quantizer is used with the modified DPCM system. The picture is first divided into blocks of 2x2 arrays of picture elements. Only the top left pels of each block are processed. The skipping factor is 4/1. The system is expected to achieve R=6 or better.

DB 2: This scheme is the same as DB 1 except a 2-bit quantizer is used in DB 2. The skipping factor is 4/1. We expect this scheme to achieve slightly higher data compression than DB 1.

DB 3: Apply the modified DPCM system with a 3-bit quantizer to every other picture elements of each scan line. The skipping factor is 2/1. This scheme is expected to achieve R=4 or better.

DB 4: A modified DPCM system with a 4-bit quantizer is applied to every other picture elements. The skipping factor is 2/1. It is expected to achieve R=4 or better, and the quantizer noise would be reduced compared to that of DB3.

The performance of these four schemes is shown in Table 5.5. The reconstructed SKYLAB IV pictures were presented in Figs. 5.21 through 5.24. From the simulation results, we found that the variable length code (C_1 code) did provide better reconstructed pictures by using more quantization levels and a nonuniform quantizer. However, it also has a drawback that the computation to accumulate the data statistics for the error sequence $\{e_i\}$ and the nonuniform quantization may cause some time delay.

It is really hard to draw a conclusion that the application of the variable length code will be better than the natural code, while implementing these compression techniques in the satellite to provide multirate data transmission. The most important factors in the decision making are the extent of time delays (due to accumulation of data statistics for the error sequence) and the buffer size required to store the data during the processing. Our simulations on the SKYLAB IV picture showed that for a picture of size 256x256, it will take 200 to 240 cpu seconds to obtain the error sequence statistics. In general, most of the SKYLAB IV pictures are of size 1194x2240 or larger, and it will require more than two hours of cpu time to find the data statistics for the error sequence which is not adequate for a real time data transmission system.

5.4 Hybrid Transform Coding

Various techniques have been developed to process the images with highly correlated picture elements to generate a set of nearly uncorrelated data. Among them, the DPCM system which was discussed in section 5.3 and the transform coding [63 , 19] and other schemes are

Table 5.5 The Performance of Modified DPCM With Variable Length Code (C_1 code)

Scheme	N	R	rms Error	Skipping Factor
DB 1	3 bits	6.8994	10.3923	4/1
DB 2	2 bits	7.1580	11.001	4/1
DB 3	3 bits	5.1341	10.1332	2/1
DB 4	4 bits	4.2939	9.099	2/1

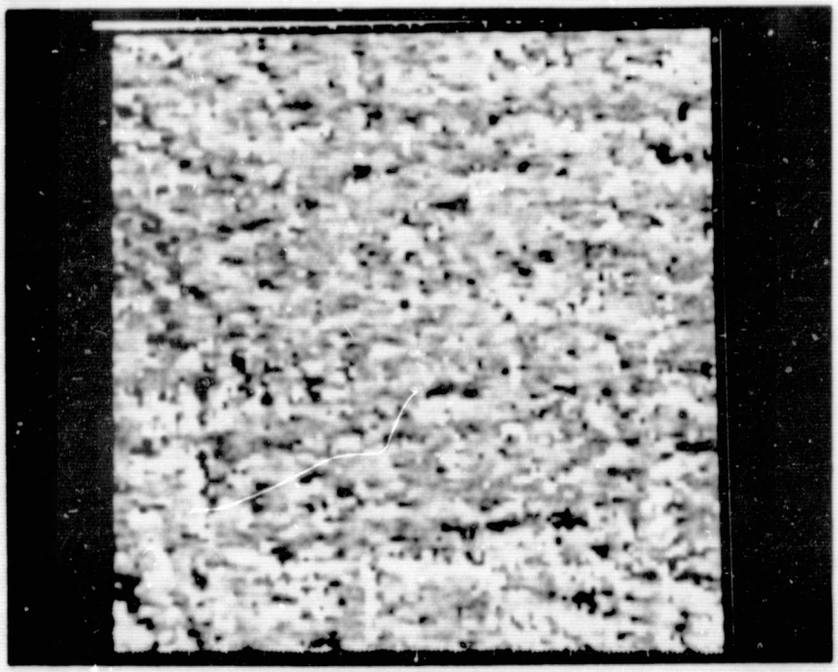


Figure 5.21 Reconstructed Picture From DB 1 Compression Scheme

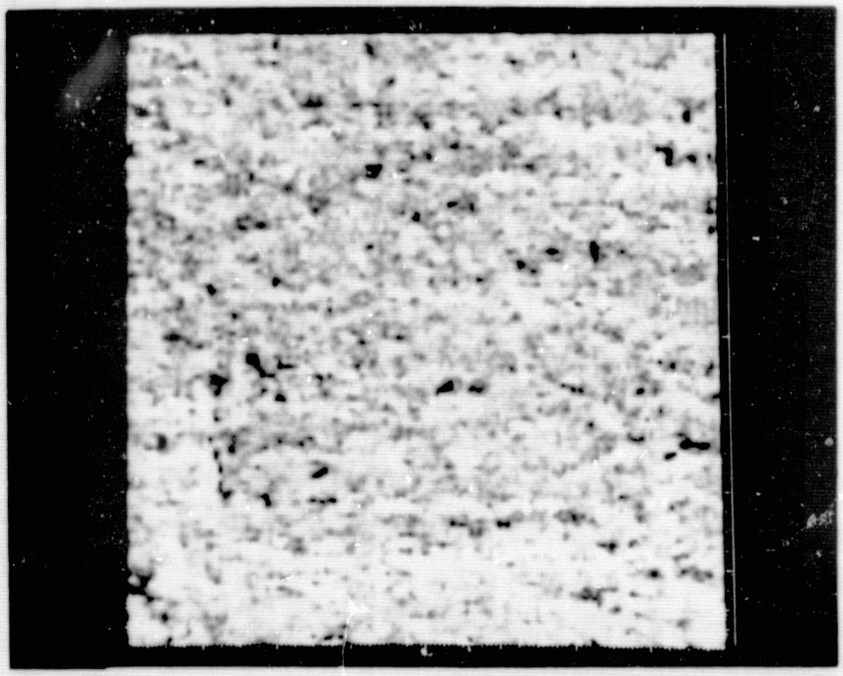


Figure 5.22 Reconstructed Picture From DB 2 Compression Scheme

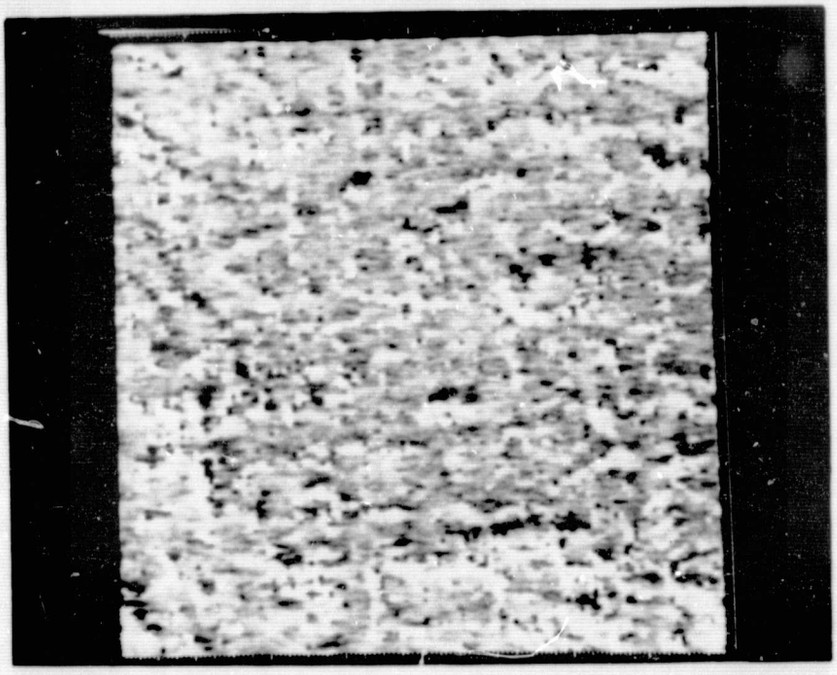


Figure 5.23 Reconstructed Picture from DB 3 Compression Scheme

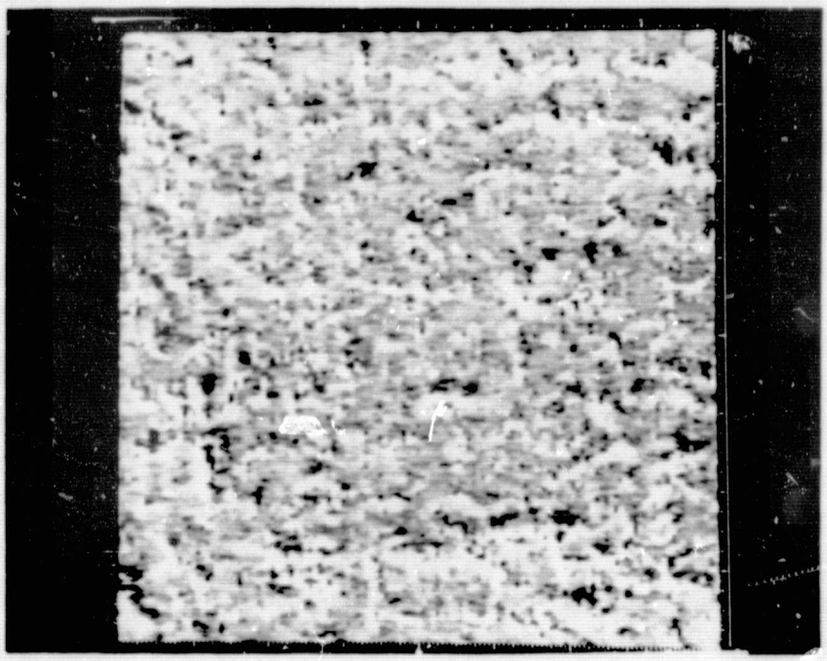


Figure 5.24 Reconstructed Picture From DB 4 Compression Scheme

known to be very efficient. Studies on the DPCM and the transform coding had indicated that each scheme has its own advantage and some limitations. For example, the transform coding provides excellent performance at lower bit rate, the coding degradation resulted from this scheme is less objectionable to the human vision system and less sensitive to the channel noise. On the other hand, the DPCM system achieves a better coding performance when operated at a higher bit rate, and it requires the minimum equipment complexity. Furthermore, the memory or buffer size needed for DPCM is very small. The disadvantage of the DPCM system is its sensitivity to the picture statistics while the nonlinear quantizing and variable length code are concerned, in addition to its error propagation when the channel noise is introduced during the transmission.

Hybrid coding was first introduced by A. Habibi [64] to combine the transform coding with DPCM system to absorb the attractive features of both schemes and avoid the limitations of each system.

Suppose that a picture is an array of MxN picture elements. We divide each scan line of picture elements into sub-blocks. Each block consists of n picture elements where N is an integer multiple of n. Each of these blocks was processed by the cosine transformation to obtain n coefficients.

Let the picture elements of each sub-block of a scan line be denoted as $U(\ell)$, where $\ell = 0,\ldots,n-1$. Then, the cosine transform of $U(\ell)$ is defined as

$$C(0) = \frac{\sqrt{2}}{n} \sum_{\ell=0}^{n-1} U(\ell) \qquad (5.23)$$

$$C(K) = \frac{2}{n} \sum_{\ell=0}^{n-1} U(\ell) \cos \frac{(2\ell+1)K\pi}{2n} \quad K=1,2,\ldots n-1 \qquad (5.24)$$

where C(K) is the Kth cosine transform coefficient. Eq. (5.24) can be rewritten as:

$$C(K) = \frac{2}{n} \text{Re} \{ e^{-(iK\pi)/2n} \sum_{\ell=0}^{2n-1} U(\ell) W^{K\ell} \} \qquad (5.25)$$

where $W = e^{-i2\pi/2n}$ $\quad i = \sqrt{-1}$

and $U(\ell) = 0$ for $\quad \ell = n, n+1, \ldots 2n-1$

Note that Re $\{\cdot\}$ represents the real part of the term enclosed. From Eq. (5.25), it is seen that all n transform coefficients can be obtained by fast Fourier transform (FFT), which reduces the computation efforts and cpu time.

The picture elements within each sub-block are highly correlated with the picture elements in the corresponding sub-block on the next scan line. We can expect that the coefficients generated by the cosine transform to be also highly correlated, because we do the same transformations to all the sub-blocks. A DPCM system is used to operate on each column of the transformed coefficients to remove the redundancy in the vertical direction.

To encode the coefficient differences more efficiently, we use the block quantization which assigns a different number of bits to encode the different coefficient differences based on the statistics of the coefficient differences. In short, the cosine transform is applied in the horizontal direction followed by the DPCM in the vertical direction.

Some computer simulation results using the hybrid transform coding are presented in Figs. 5.25 through 5.26, corresponding to the following schemes:

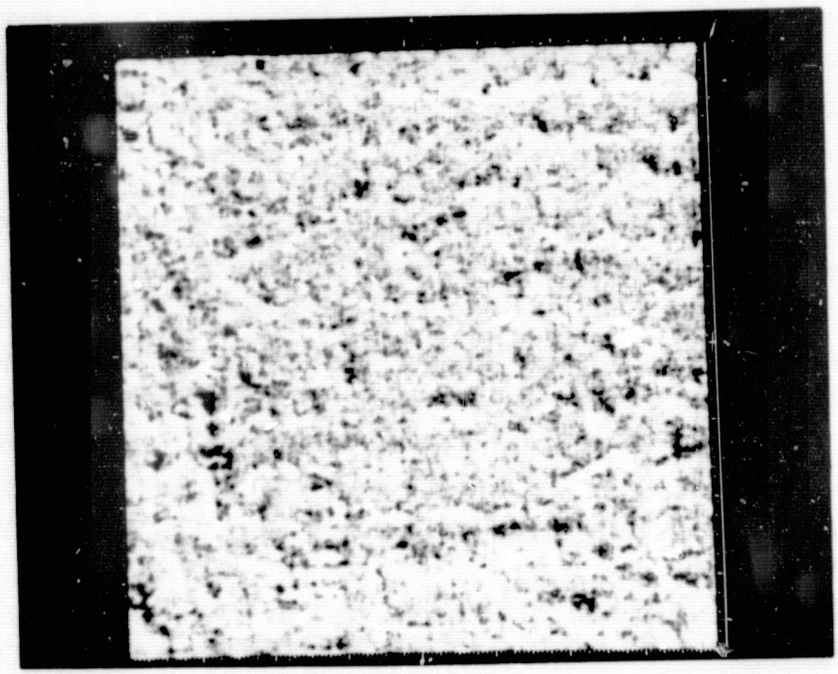


Figure 5.25 Reconstructed Picture From H 1 Compression Scheme

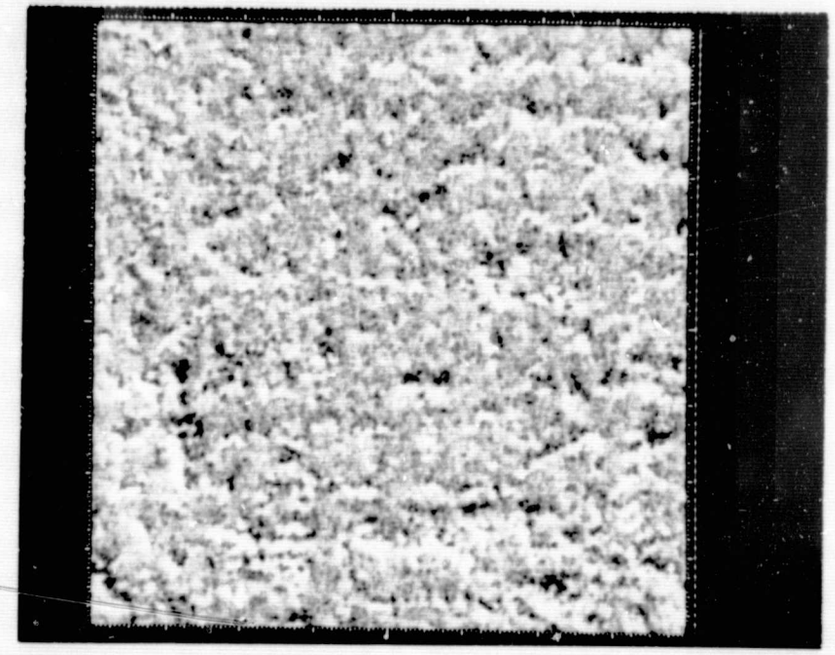


Figure 5.26 Reconstructed Picture From H 2 Compression Scheme

H1: The bit allocation is 1 bit per picture element in average. The data compression ratio is 6.

H2: The bit allocation is 1.5 bits per picture element in average. The data compression ratio is 4.

The bit allocation for the coefficient differences is shown in Table 5.6. The performance of the hybrid schemes is shown in Table 5.7. It is obvious that the distortion is not objectionable to the human observer, and the degradation is very smooth throughout the whole picture. There is no artificial square boundaries in the reconstructed pictures. However, this scheme is a little time consuming due to many operations required. The time delay of the hybrid technique should be taken into account to see whether the time delay can be tolerated.

5.5 Linear Interpolation Technique

We recall that the picture elements of a digitized image are highly correlated with the neighboring picture elements. The probability that a picture element is of the same grey level as its neighbors is rather high. A simple and easy to operate data compression technique is to take advantage of this picture element correlation. The key idea is to skip some picture elements uniformly to provide a proper skip factor.

First of all, we divide the entire picture into blocks of size nxm, and transmit only the top left picture element of each block with 6 bits/pel. Thus, a data compression ratio of nxm is achieved (which is the same as the skipping factor in this case). As we know that the satellite images were oversampled in the ratio of 1.5:1, the magnitude of n or m should not exceed 2, otherwise the rms error of the reconstructed picture will become more significant. Our computer simulation also verified this argument.

Table 5.6 Bit Allocation for Coefficient Differences

Coefficients	1.5-Bit Picture	1-Bit Picture
1	2	2
2	2	2
3	2	2
4	2	2
5	2	2
6	2	2
7	2	2
8	2	2
9	2	1
10	2	1
11	2	1
12	2	1
13	2	1
14	2	1
15	2	1
16	2	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	1	1
27	1	0
28	1	0
29	1	0
30	1	0
31	1	0
32	1	0

Table 5.7 The Performance of the Hybrid Transform Coding Where R = Data Compression Ratio

Scheme	N	R	rms Error
H 1	1 bit/pel	6	6.89
H 2	1.5 bit/pel	4	5.91

Two data compression techniques based on the linear interpolation are designed to provide compression ratios of 4 and 6 respectively.

L 1: The picture is divided into blocks. Each block is a 2x2 array of picture elements. Only the top left picture elements of each block are processed and transmitted with 6 bits/pel. Therefore, a data compression ratio of 4 is obtained. The reconstruction algorithm would be well explained through the help of the following example:

A	B	C	D
E	F	G	H
I	J	K	L

Suppose that A, C, I, and K were the transmitted picture elements. The algorithm is to calculate the skipped picture elements by the following equations:

$$\begin{aligned} B &= \frac{1}{2}\ (A+C) \\ J &= \frac{1}{2}\ (I+K) \\ E &= \frac{1}{2}\ (A+I) \\ G &= \frac{1}{2}\ (C+K) \\ F &= \frac{1}{4}\ (B+E+G+J) \end{aligned} \qquad (5.26)$$

The procedure is repeated for all the blocks of the entire pictures.

L 2: Same as L 1 except the block size is a 2x3 array of picture elements. The data compression ratio is 6. The reconstruction algorithm is as follows:

In the example, if A, D, I, and L were the transmitted picture elements, the skipped picture elements can be found through the following calculations:

$$B = \frac{2}{3} A + \frac{1}{3} D$$

$$J = \frac{2}{3} I + \frac{1}{3} L$$

$$C = \frac{1}{3} A + \frac{2}{3} D$$

$$K = \frac{1}{3} I + \frac{2}{3} L \qquad (5.27)$$

$$E = \frac{1}{2} (A+I)$$

$$H = \frac{1}{2} (D+L)$$

$$F = \frac{2}{3} E + \frac{1}{3} H$$

$$G = \frac{1}{3} E + \frac{2}{3} H$$

The procedure is repeated for all the blocks of the entire picture.

We present the reconstructed pictures of Fig. 5.2 based on schemes L 1 and L 2 in Figs. 5.27 and 5.28. Their performances were shown in Table 5.8.

Since only simple operations were involved in the linear interpolation technique, the equipment complexity for this technique is minimized. In fact, a switching circuit device can be used to serve as a sample selector to provide appropriate skipping factor.

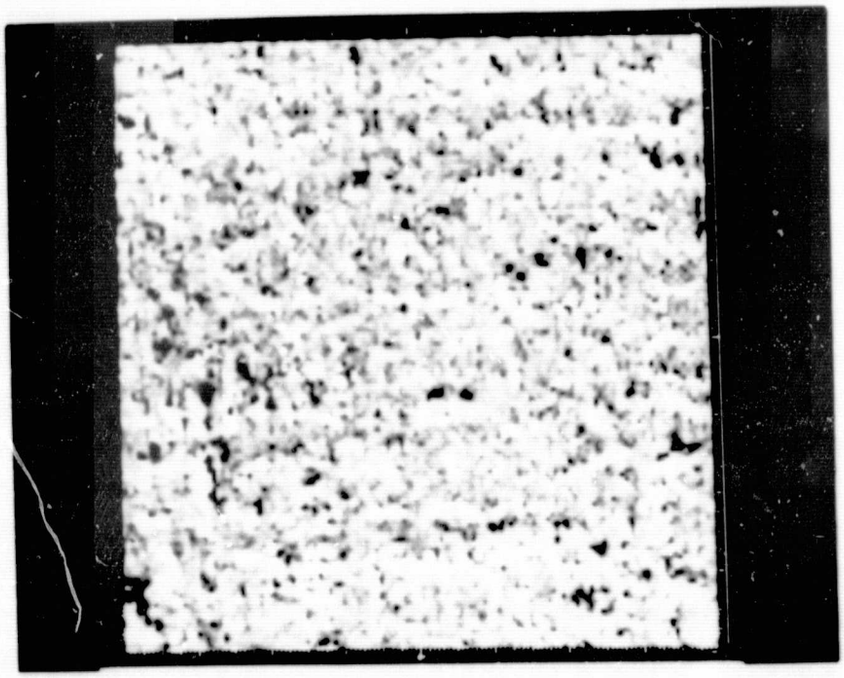


Figure 5.27 Reconstructed Picture From L 1 Compression Scheme

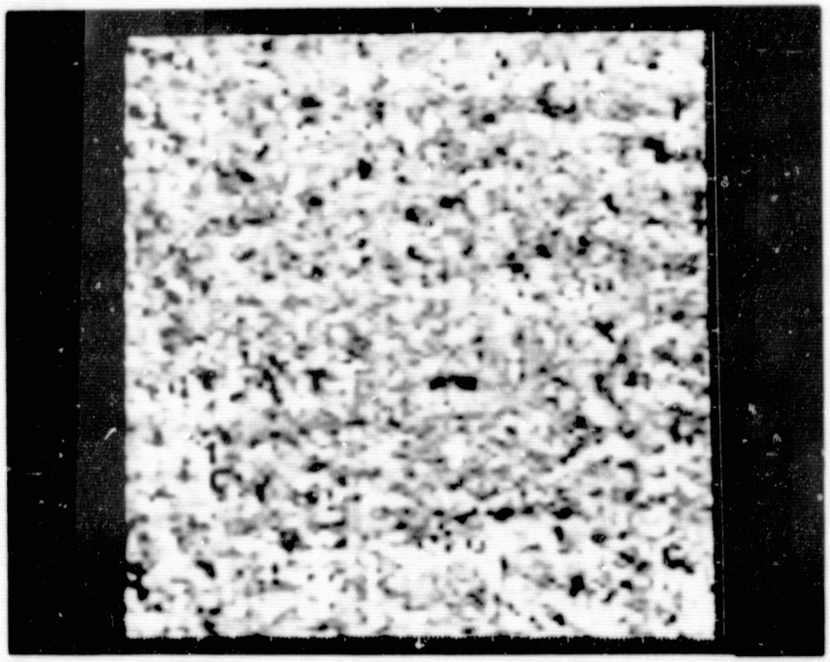


Figure 5.28 Reconstructed Picture From L 2 Compression Scheme

Table 5.8 The Performance of Linear Interpolation Technique Where R = Data Compression Ratio

Scheme	N	R	rms Error
L 1	6 bit/pel	4	9.79
L 2	6 bit/pel	6	11.27

As shown in Fig. 5.29, a block diagram of the multi-rate data transmission system will operate at 120, 20, and 30 mega bits per second. Here the sample selector played an important role in this system. Note that certain spectral band may have poor signal to noise performance. For example, channel 4 of LANDSAT (previously called ERTS) images has poor signal to noise performance, thus, 6-bit log scale quantizer is bypassed.

In fact, this is a very efficient and low cost technique, which provides reasonably good performance. Time delays and computational efforts are minimized. However, the rms error in the reconstructed pictures by this technique is slightly higher, especially in L 2. The overall performance of the linear interpolation scheme make it a promising candidate of a multi-rate data transmission system.

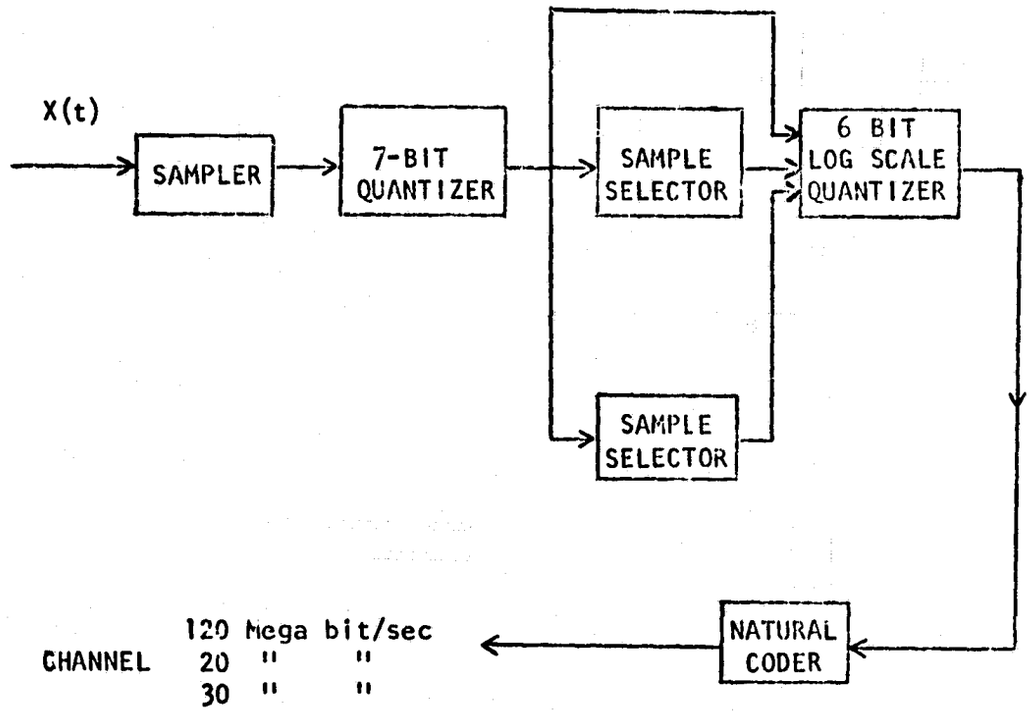


Fig. 5.29 Block Diagram of the Multi-rate Data Transmission System While the Linear Interpolation Algorithm is Used

CHAPTER 6

CONCLUSION

6.1 Summary of Results

This work was primarily concerned with the investigation of an efficient, almost real time, data compression system for multispectral satellite pictures and for binary or multi-level pictures obtained from the classification of satellite picture. Due to the different characteristics of these pictures, our investigation was divided into two parts.

The first part of this research dealt with efficient data compression schemes for binary and multi-level pictures. We have modified the predictive differential quantizing (PDQ) method to ease the design of the compressed data format. Furthermore, an improved data compression scheme called double delta coding (DDC) was introduced. DDC was shown, by computer simulation, to be more efficient than PDQ for those pictures where PDQ was better than run length coding (RLC). Both DDC and PDQ can be improved by applying background skipping technique. The simulation results showed that the application of background skipping on PDQ and DDC significantly reduced the processing time and achieved higher data compression.

The derived data, which is defined to be the output of the redundancy remover, should be encoded by the appropriate source codes before being transmitted over the channel. We have investigated the prefix and non-prefix source codes. Both of these are variable length

codes that can take advantage of the nonuniform distribution of the derived data. The extension code was formulated by combining the prefix code and the extension blocks. This class of codes operated rather efficiently, and required very little storage space. Moreover, the size of the extension block can be adjusted based on the practical situation. The code word of the non-prefix code consists of blocks of fixed length binary digits.

In general, the complexity of the objects in the picture played an important role in data compression. The pictures taken during the flooding period are more complicated than those taken in normal seasons. To facilitate the design of the data compression system, we can predetermine the coding scheme for each type of picture. Our simulations showed that run length coding is more efficient than others for those flooded pictures; however, double delta coding is the most efficient scheme for pictures taken during normal seasons. As the complexity of the picture and the number of grey levels are increased, none of the error-free coding schemes we have investigated can provide efficient data compression.

The second part of this work is concerned with data compression techniques which would provide data compression ratios of four and six for the continuous-tone satellite pictures. Since the complexity of the satellite pictures are usually very high, any error free compression technique can not guarantee the desired data compression ratios. Thus, a certain amount of distortion resulting from the data compression is inevitable. To evaluate the performance of the compression schemes, the root mean square error was used as the error criterion.

The DPCM system is a very efficient data compression system with minimal design complexity and high operation speed. We have investigated the modified DPCM system which used a sample selector to skip some picture elements. To keep the rms error as small as possible, we found that it was better to limit the maximal skipping factor to two in each direction. With fixed length codes, the uniform quantizer will be easier and faster to operate. On the other hand, the nonuniform quantizer will provide better performance when the variable length code is employed. Furthermore, the variable length code enables us to choose a quantizer with more quantization levels to achieve smaller rms error.

The hybrid transform coding system, which combines the DPCM system with the transformation technique, takes advantage of both schemes. The reconstructed picture of the hybrid scheme is less objectionable to human observers and the resulting rms error was minimal among all the compression schemes we have investigated. However, it is the most time consuming scheme and has the highest cost of implementation.

The linear interpolation technique is the simplest compression scheme both in implementation and processing. For the satellite pictures which were oversampled at the ratio of 1.5:1, a skipping factor of two in each direction is reasonable. Thus, the distortion resulting from the L2 scheme with a skipping factor of six is more significant than the L1 scheme with a skipping factor of four. In general, the linear interpolation technique is an efficient and low cost compression scheme whose performance is reasonably good.

In conclusion, the selection of a compression scheme for a multirate data transmission system depends on several factors such as the rms error, the computational cost, time delay, and the complexity of implementation.

6.2 Suggestion for Future Research

The compressed data format is designed to pack the encoded derived data into a compact format and to enable the receiver to identify the data string for each scan line such that the number of objects in each scan line can be decoded correctly. However, the compressed data format did not offer the protection and error detection capability to each individual encoded index (i.e., Δ's or run lengths). When the channel is noisy and the transmitted indices are corrupted, the objects in each scan line will be displaced causing a degradation in the reconstructed picture.

One possible way to solve this problem is to perform channel coding before the encoded data are transmitted such that the errors occurring in the encoded indices can be detected at the receiver. However, it will decrease the data compression ratio. Another possible way is to investigate an encoder which can be used as both source and channel coder. Some researchers have reached preliminary results in this open area of research; however, those methods are very limited and can not be applied to practical systems.

LIST OF REFERENCES

1. Proceedings of the Seventh International Symposium on Remote Sensing of Environment, University of Michigan, May, 1971.

2. K. S. Fu, D. A. Landgrebe, and T. Phillips, "Information Processing of Remotely Sensed Agricultural Data," Proc. IEEE, April 1969.

3. D. A. Landgrebe, "Machine Processing for Remotely Acquired Data," LARS Inf. Notes 031573, LARS, Purdue University, 1973.

4. K. S. Fu, Syntactic Methods in Pattern Recognition, New York, Academic Press, 1974.

5. S. Tsuji and R. Fujiwara, "Linguistic Segmentation of Scenes Into Regions," Second International Joint Conference on Pattern Recognition, Copenhagen, Denmark, August, 1974.

6. R. M. Haralick, K. Shanmugan, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. on System, Man, and Cybernetics, Vol. SMC-3, No. 6, November, 1973.

7. N. J. Nilsson, Learning Machine, McGraw-Hill, New York, 1965.

8. L. C. Wilkins and P. A. Wintz, "Studies on Data Compression," TR-EE 70-17, Part I and II, Purdue University, September 1970.

9. J. N. Gupta and P. A. Wintz, "Multi-Image Modeling," TR-EE 74-24, Purdue University, September 1974.

10. P. J. Ready, P. A. Wintz, and D. A. Landgrebe, "A Linear Transformation for Data Compression and Feature Selection in Multi-Spectral Imagery," LARS Inf. Notes 072071, LARS, Purdue University, February 1971.

11. P. H. Swain, T. V. Robertson, and A. G. Wacker, "Comparison of the Divergence and B-Distance in Feature Selection," LARS Inf. Notes 020871, LARS, Purdue University, February 1971.

12. P. H. Swain, "A Basis for Remote Sensing Data Analysis," LARS Inf. Notes 111572, LARS, Purdue University, 1972.

13. E. H. Ruspini, "Numerical Methods for Fuzz Clustering," Information Science, Vol. 2, 1970.

14. F. J. Rohlf, "Adaptive Hierachical Clustering Scheme," System Zool., Vol. 18, 1970.

15. E. H. Ruspini, "A New Approach to Clustering," Journal of Information and Control, Vol. 15, pp. 22032, 1969.

16. J. C. Lindenlaub, "Guide to Multispectral Data Analysis Using LARS," LARS Inf. Notes 062873, LARS, Purdue University, 1973.

17. W. G. Eppler, "An Improved Version of the Table Look-up Algorithm for Pattern Recognition," Ninth Symposium on Remote Sensing of Environment, November 1973.

18. L. E. Franks, "A Model for the Random Video Process," Bell Sys. Tech. J., 45, pp. 609-630, April 1966.

19. A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Trans. on Comm. Tech., Vol. COM-19, February 1971.

20. A. Habibi, "Two Dimensional Bayesian Estimate of Images," Proc. of IEEE, Vol. 60, July 1972.

21. E. Wong, "Recent Progress in Stochastic Processes - A Survey," IEEE Trans. on Infor. Theory, Vol. IT-19, pp. 262-275, July 1968.

22. J. Wood, "Two Dimensional Discrete Markovian Field," IEEE Trans. on Infor. Theory, Vol. IT-18, March 1972.

23. J. Capon, "A Probabilistic Model for Run Length Coding of Pictures," IRE Trans. on Inf. Theory, Vol. IT-5, pp. 157-163, December 1959.

24. T. S. Huang and A. B. S. Hussain, "Facsimile Coding by Skipping White," IEEE Trans. on Comm., Vol. COM-23, No. 12, pp. 1452-1460, December 1975.

25. T. S. Huang, "Run Length Coding and Its Extension," Proc. Symp. Picture Bandwidth Compression, April 1969.

26. D. Spenser and T. S. Huang, "Bit-Plane Encoding," Computer Processing in Communications, Symp. Proc., Polytechnic Inst., Brooklyn, Brooklyn, N.Y., pp. 101-120, April 1969.

27. B. Lippel, "Experiments with A New Message Format for Digital Representation of Photographs," Computer Processing in Comm., Symp. Proc., Polytechnic Inst. Brooklyn, Brooklyn, N.Y., pp. 121-139, 1969.

28. E. R. Kretzmer, "Statistics of Television Signals," Bell System Tech. J., Vol. 31, pp. 751-763, July 1952.

29. C. W. Harrison, "Experiments with Linear Prediction in Television," Bell System Tech. J., Vol. 31, pp. 764-783, July 1952.

30. W. F. Schreiber, "The Measurement of Third Order Probability Distribution of Television Signals," IRE Trans. on Inf. Theory, Vol. IT-2, September 1956.

31. W. A. Youngblood, "Picture Progress," MIT/RLE, 1968.

32. J. E. Cunninghan, "Image Correction - Transmission Experiments," MIT/RLE, 1963.

33. D. Graham, "Image Transmission by Two Dimensional Contour Coding," Proc. IEEE, Vol. 55, March 1967.

34. W. F. Schreiber, "The Mathematical Foundation of the Synthetic Highs System," MIT/RLE, January 1963.

35. D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proc. of IRE, Vol. 40, September 1952.

36. C. C. Cutler, "Differential Quantization of Communication Signals," Patent 2 605 361, July 29, 1952.

37. J. Max, "Quantizing for Minimum Distortion," IRE Trans. on Inf. Theory, Vol. IT-6, pp. 7-12, March 1960.

38. P. A. Wintz, "Transform Picture Coding," Proc. IEEE, Vol. 60, July 1972.

39. W. K. Pratt and H. C. Andrews, "Transform Image Coding," USCEE Rep. 387, University of Southern California, March 1970.

40. J. Pearl, H. C. Andrews, and W. K. Pratt, "Performance Measures for Transform Data Coding," IEEE Trans. on Communication, Vol. COM-20, June 1972.

41. H. P. Krammer and M. W. Mathews, "A Linear Coding for Transmitting a Set of Correlated Signals," IRE Trans. on Inf. Theory, Vol. IT-2, September 1956.

42. L. C. Wilkins and P. A. Wintz, "Bibliography on Data Compression, Picture Properties, and Picture Coding," IEEE Trans. on Inf. Theory, Vol. IT-17, No. 2, March 1971.

43. T. Huang, W. F. Schreiber, and O. J. Tretiak, "Image Processing," Proc. of IEEE, Vol. 59, No. 11, November 1971.

44. B. Hunt, "Digital Image Processing," Proc. of IEEE, Vol. 63, No. 4, April 1975.

45. D. J. Connor, et. al., "Intraframe Coding for Picture Transmission," IEEE Proc., pp. 779-791, July 1972.

46. H. W. Gschwind, Design of Digital Computer, Springer-Verlag, New York, 1967.

47. J. J. Stiffler, Theory of Synchronous Communication, Prentice Hall, Englewood Cliffs, New Jersey 1971.

48. R. G. Gallger, Information Theory and Reliable Communication, John Wiley, New York 1968.

49. P. H. Chen and P. Wintz, "Boundary Detection for U.S. Army Corps of Engineers, Flood Map," Imagery Analysis and Modeling Quarterly Progress Report for ARPA, Purdue University, West Lafayette, IN, May 1975.

50. J. R. Duan and P. A. Wintz, "Information Preserving Coding for Multispectral Scanner Data," TR-EE 74-15, Purdue University, April 1974.

51. M. R. Anderberg, Cluster Analysis for Application, Academic Press, New York, 1973.

52. Wozencraf and Jacobs, Principles of Communication Engineering, John Wiley and Sons, Inc., New York, 1965.

53. R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, Inc., New York, 1973.

54. Peter G. Neumann, "Self-Synchronizing Sequential Coding with Low Redundancy," Bell System Tech. J., Vol. 50, No. 3, March 1971.

55. A. A. Sardinas and G. W. Patterson, "A Necessary and Sufficient Condition for the Unique Decomposition of the Coded Messages," IRE Convention Record, Part 8, pp. 104, 1953.

56. T. Huang, H. Meyr, and H. G. Rosdolsky, "Optimal Run Length Codes," IEEE Trans. on Communications, Vol. COM-22, No. 6, June 1974.

57. J. Ziv, "On the Coding of Data Sources with Unknown Statistics," MM 70-1213-1, January 1970.

58. V. L. Thomas, "Generation and Physical Characteristics of ERTS MSS System Corrected Computer Compatible Tape," GSFC, Greenbelt, Maryland, X-563-73-206, July 1973.

59. K. W. Cattermole, Principle of Pulse Code Modulation, London, ILIFFE Books, Lts., London, 1969.

60. B. N. Oliver, "Efficient Coding," Bell System Tech. J., pp. 724-750, Vol. 31, July 1952.

61. T. Kailath, "The Innovations Approach to Detection and Estimation Theory," Proc. IEEE, Vol. 58, pp. 680-695, May 1970.

62. B. Smith, "Instantaneous Companding of Quantized Signals," Bell System Tech. J., Vol. 36, pp. 653-709, May 1957.

63. H. C. Andrews, J. Kane, and W. K. Pratt, "Hadamard Transform Image Coding," Proc. IEEE, Vol. 57, pp. 58-68, January 1969.

64. A. Habibi, "Hybrid Coding of Pictorial Data," IEEE Trans. on Communication, Vol. COM-22, pp. 614-623, May 1974.

APPENDIX

Smith's Nonuniform Quantization

When the probability density function of the signal can be approximated by the exponential density function

$$P(e) = \frac{1}{2\sqrt{\sigma}} \exp\left(-\frac{\sqrt{2}}{\sigma}|e|\right)$$

where σ^2 is the variance of the signal e

The function f(e) is formed such that when e takes some value in the range of V to -V, f assumes a proper value.

$$f(e) = -\frac{V}{m} \ln\left[1 - \frac{e}{V}(1 - \exp(-m))\right]$$

$$f(e) = f(-e)$$

where $m = \sqrt{2}\, V/3\sigma$ For V = 63, σ = 12.18, we present a list of tables as follows:

SMITH NONUNIFORM QUANTIZATION

E	F(E)	E	F(E)
0	0	32	15.4046
1	.3771	33	16.0331
2	.7597	34	16.8005
3	1.1481	35	17.5279
4	1.5424	36	18.2762
5	1.9428	37	19.0470
6	2.3495	38	19.8414
7	2.7628	39	20.6610
8	3.1827	40	21.5075
9	3.6096	41	22.3826
10	4.0437	42	23.2884
11	4.4852	43	24.2272
12	4.9343	44	25.2013
13	5.3914	45	26.2136
14	5.8568	46	27.2672
15	6.3306	47	28.3656
16	6.8133	48	29.5128
17	7.3053	49	30.7133
18	7.8067	50	31.9723
19	8.3181	51	33.2958
20	8.8398	52	34.6907
21	9.3723	53	36.1653
22	9.9159	54	37.7292
23	10.4713	55	39.3939
24	11.0388	56	41.1732
25	11.6191	57	43.0842
26	12.2128	58	45.1479
27	12.8204	59	47.3909
28	13.4426	60	49.8472
29	14.0802	61	52.5619
30	14.7339	62	55.5956
31	15.4046	63	59.0334