Semiannual Status Report

NAG 5-1612

Submitted to
Instrument Division
Engineering Directorate
Goddard Space Flight Center
Greenbelt, MD 20771
Attn: Pen-Shu Yeh

K. Sayood, S. Nori, and A. Araj

Department of Electrical Engineering
University of Nebraska-Lincoln
Lincoln, Nebraska 68588-0511

Period: December 15, 1993 - June 15, 1994

1 Introduction

During this six month period our work concentrated on three, somewhat different areas. We looked at and developed a number of error concealment schemes for use in a variety of video coding environments. This work is described in an accompanying (draft) Masters thesis. In the thesis we describe application of this techniques to the MPEG video coding scheme. We felt that the unique frame ordering approach used in the MPEG scheme would be a challenge to any error concealment/error recovery technique.

We continued with our work in the Vector Quantization area. The work on recursively indexed Vector Quantization will be reported in a PhD dissertation during the current period. We have also developed a new type of Vector Quantizer, which we call a *Scan Predictive Vector Quantization.* The Scan Predictive VQ was tested on data processed at Goddard to approximate Landsat 7 HRMSI resolution, and compared favorably with existing VQ techniques. A paper describing this work is included with this report. The paper has been submitted to *IEEE Transactions on Image Processing.*

The third area is concerned more with reconstruction than compression. While there is a variety of efficient lossless image compression schemes, they all have a common property that they use past data to encode future data. This is done either via taking differences, context modeling or by building dictionaries. When encoding large images, this common property becomes a common flaw. When the user wishes to decode just a portion of the image, the requirement that the past history be available forces the decoding of a significantly larger portion of the image than desired by the user. Even with intelligent partitioning of the image dataset, the number of pixels decoded may be four times the number of pixels requested. We have developed an adaptive scanning strategy, which can be used with any lossless compression scheme, and which lowers the additional number of pixels to be decoded to about 7% of the number of pixels requested! A paper describing these results is included in this report. This work will be reported at the 1994 International Geoscience and Remote Sensing Symposium.

During this period, the following paper appeared in print

" Coding of Color-Mapped Images," (with A.C. Hadenfeldt), *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, pp. 534-541, May 1994.

A copy of the paper is included with this report.

Also during this period, the following paper was accepted for publication

"A Constrained Joint Source Channel Coder Design," (with F. Liu and J.D. Gibson), to appear in *IEEE Journal on Selected Areas of Communication.*

Error Concealment

Reconstruction of Packet Video After Packet Loss

by

Sekhar Nori

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Engineering

Major: Electrical Engineering

Under the Supervision of Professor Khalid Sayood

Lincoln, Nebraska

May, 1994

Contents

List of Figures

List of Tables

Abstract

In packet switched networks, even with error correction protection, packet loss is unavoidable. Hence a method of error recovery, which takes the characteristics of the video signals into account, is required. This process, known as Error Concealment, attempts to recover or reconstruct the missing blocks from the structured picture data. A method of error concealment based on the motion estimation is proposed in this paper. This method makes use of the fact that most of the frames look alike (excepting during the shifts) and hence uses the past as well as the future information to reconstruct the missing information in a frame (in addition to the information from the same frame).

The underlying assumptions in this method of error concealment are

- pixels in the image are much smaller than any of the important details
- most of the pixels' neighbors represent the same structure.

Chapter 1

Introduction

For multimedia communication and information services, the evolution of asynchronous transfer mode (ATM) networks based on packet switching represents the flexibility and freedom in maintaining the quality of these services Packet switched networks were originally invented for carrying burst-type data as it was uneconomical to use continuously connected circuit. In conventional circuit switched connections a dedicated path is established and a bandwidth is assigned in advance. Quality of service would degrade if the channel capacity were to exceed, while the excess channel capacity would be wasted if the output rate of the source were less than the channel capacity. Packet video is a relatively new field and has attracted a lot of attention. As image information representing detail, motion, etc. varies, variable bit rate (VBR) coding tailored for packet switched networks can be utilized to maintain constant image quality. Also by channel sharing among multiple video

sources, transmission efficiency could be improved.

The flexibility of packet switching provides new opportunities for video communication and at the same time it also presents new challenges. The problems inherent in this scenario are packet loss and packet delay. The former can be due to random bit errors in the packet destination address and heavy traffic at certain nodes in ATM network, while the latter is caused by holding of packet at any of the switching nodes until a slot is open, resulting in a differential transmission delay between packets. The differential delay causes problems in timing relationship between video generation and reconstruction for display. Hence it is necessary to incorporate error correction protection into coding techniques compatible with packet video. A simple method to incorporate error protection scheme is to generate parity packets containing parity bits generated from the information packets. A lost packet could hence be recovered by initiating error protection at the receiver. This scheme however increases the rate and hence contributes to packet loss. It has been observed however, that the error correction ability of the error protection system more than makes up for the packet loss introduced by the increased rate.

In packet switched networks, even with error correction protection, packet loss is unavoidable and a correction method is required for video packet transmission, which takes characteristics of video signals and video coding into account. In this method the receiver detects the damaged picture caused by the lost packet and performs error concealment. This thesis proposes an error concealment method for

video packets lost during transmission.

Chapter 2 describes various image compression techniques. Broadly these are classified into spatial domain techniques and frequency domain techniques. Advantages and disadvantages of each of these techniques are discussed in this chapter.

Chapter 3 describes various components of a basic video codec suitable for packet video. It also briefly describes how error correction protection can be incorporated into the coding algorithm.

Chapter 4 presents various error concealment and reconstruction algorithms. A new algorithm for error concealment based on estimation motion from frames both in the past and the future is presented. The performance of this method together with the results and its limitations on two motion sequences are presented.

Chapter 5 presents the conclusion.

Chapter 2

Understanding Image Compression

The goal of data compression of images is to reduce transmission and storage costs. To achieve compression it is necessary to consider representations beyond simple analog to digital conversion of image data. Several other factors such as high correlation between adjacent pixels have to be taken into consideration while compressing images (spatial domain compression). In addition, correlation between adjacent frames, has to be taken into consideration while compressing motion sequences. In this chapter we discuss various image coding techniques mostly applicable to moving images.

2.1 Understanding Digital Images

It has been known for quite some time that a wide spectrum of colors can be generated from a set of three primaries: red, yellow and blue. Television displays generate colors by mixing lights of the additive primaries. The color space obtained through combining the three colors can be determined by drawing a triangle on a special color chart with each of the base colors as an endpoint. This classic color chart was established by Commission Internationale de L'Eclairage (CIE).

One of the special concepts introduced by 1931 CIE chart was the isolation of luminance (brightness) from chrominance (hue). Using the guidelines of CIE the National Television System Committee (NTSC) defined picture transmission in the form of luminance and chrominance components. The new color space was labeled YIQ, where the Y stood for the luminance component while the I and the Q stood for the in-phase component and quadrature component of chrominance respectively.

In Europe two television standards later emerged, Phase-alternation-line (PAL) format and Séquentiel couleurà and mémoire (SECAM) format, both with identical color space, YUV. The difference between the PAL/SECAM YUV color space and the NTSC YIQ color space is a 33 degree rotation in UV space. The YUV format as well as the YIQ format concentrates most of the image information into the luminance and less into the chrominance. The result is that each of the individual components can be coded individually without much loss of efficiency.

2.1.1 File Formats

There are two formats for PAL-style input: CIF, representing an input file of 352×288 for the luminance and 176×144 for the chrominance components; and QCIF, representing an input file of 176×144 for the luminance and 88×72 for the chrominance components. For NTSC images the most common input style is the CIF-style which represents an input file of 352×240 for the luminance and 176×120 for each of the chrominance components.

2.2 Intra Frame Processing

2.2.1 Spatial Domain Methods For Image Compression

In this section we consider digital coding techniques that operate on the data in the spatial domain.

Pulse Code Modulation (PCM)

In pulse code modulation the incoming video signal is sampled and quantized. Hence it is just a digital representation of the original analog signal. This method of coding does not consider the inter-pixel correlation while coding the image sequences.

Predictive Coding

In PCM, successive inputs to the quantizer are treated independently, so there is no exploitation of the significant redundancy present in images.

The philosophy behind predictive coding is to remove redundancy between successive samples of input data and to quantize only the new information. The most common example of a predictive coding system is Differential Pulse Code Modulation (DPCM). In DPCM, the difference between successive samples is quantized and transmitted as opposed to other coding schemes where the original samples are quantized. This scheme works well for images since there is a lot of correlation between adjacent pixels [1].

The basic equations describing DPCM are, (see Figure 2.1)

$$d(n) = x(n) - \hat{x}(n) \tag{2.1}$$

$$u(n) = d(n) - q(n) \tag{2.2}$$

$$\text{and } y(n) = \hat{x}(n) + v(n) \tag{2.3}$$

where $y(n)$ is the DPCM approximation to coder input $x(n)$, $d(n)$ is the prediction error, $q(n)$ is the quantization error, $u(n)$ is the quantized prediction error, and v(n) is the quantized prediction error which may have been corrupted by channel noise [2].

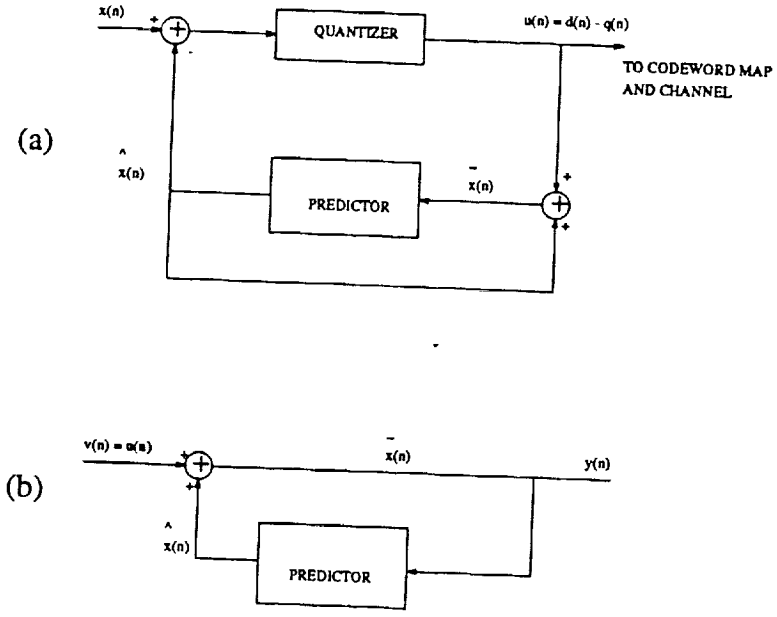


Figure 2.1: Block diagram of DPCM: (a) Coder and (b) Decoder

2.2.2 Transform Coding

An alternative to predictive coding is transform coding. Discrete Cosine Transform (DCT) is the most commonly used transform coding technique in which square subregions in the image are processed with a discrete cosine transform. Conceptually a one dimensional DCT can be thought of as taking the Fourier Transform of an infinite sequence (see Figure 2.2). For a spatial image $f(x, y)$ the two-dimensional discrete cosine transform $F(u, v)$ is given by

$$F(u,v) = \frac{4C(u)C(v))}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos \frac{(2x+1)\pi u}{2N} \cos \frac{(2y+1)\pi v}{2N} \tag{2.4}$$

The inverse transform is given by

$$f(u,v) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N} \tag{2.5}$$

where

$$C(w) = \begin{cases} 1/\sqrt{(2)} & \text{for } w = 0 \\ 1 & \text{for } w = 1, 2, \cdots, N-1 \end{cases}$$

where N is the width of the image block, the range for u and v is from 0 to N-1.

The reason behind using DCT is that for correlated or low frequency sources the DCT tends to concentrate the energy into a very few coefficients. The coefficients containing most of the energy can be used to approximate the source output. Images tend to have large regions of low spatial frequency. This makes the DCT a very useful transform to use with images.

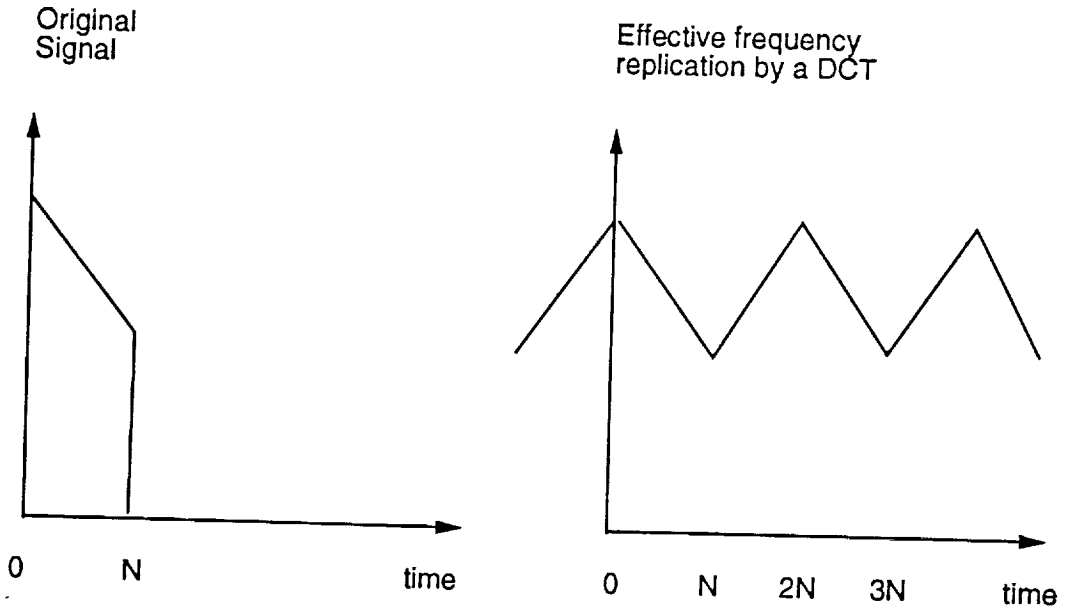


Figure 2.2: The DCT transform

DCT-Based Compression

Figures 2.3 and 2.4 show the key processing steps which are the heart of the DCT-based models of operation. These figures illustrate the compression of a grayscale image. As can be seen from the Figure 2.3 each of the 8 X 8 blocks of the image makes its way through each of the processing steps and gets compressed. Color image compression can be thought of as multiple grayscale images being compressed entirely (i.e., all the componenets) or one at a time.

The DCT is related to the Discrete Fourier Transform (DFT) [3]. Each of the NXN block is a N^2 point discrete signal. The FDCT takes such a signal as its input and decomposes into N^2 orthogonal basis signals. The DCT coefficient values can thus be regarded as the relative amount of the 2D spatial frequencies contained in the N^2-point input signal. The coefficient with zero frequency in both dimensions is called the "DC coefficient" and the remaining coefficients are called the "AC coefficients".

At the decoder the IDCT reverses this processing step. It takes N^2 DCT coefficients and reconstructs the $N \times N$ block. In principle, the DCT introduces no loss to the source image samples; it merely transforms them to a domain in which they can be more efficiently encoded.

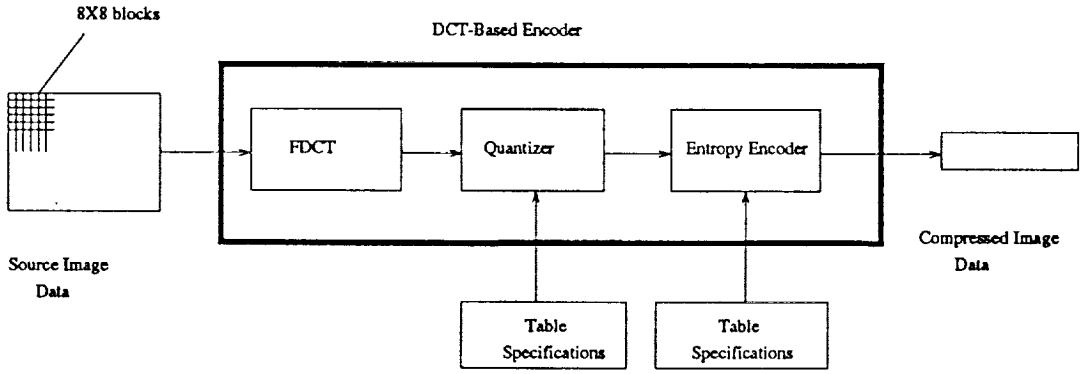


Figure 2.3: DCT-Based Encoder Processing steps

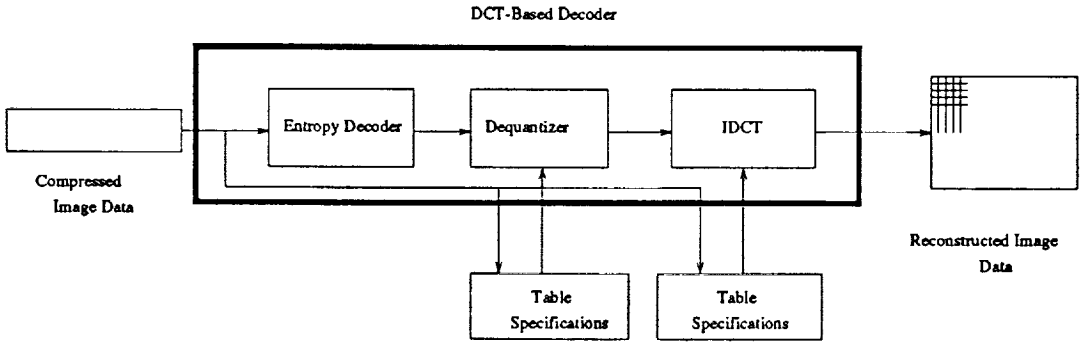


Figure 2.4: DCT-Based Decoder Processing steps

2.3 Inter Frame Processing

The amount of compression possible by spatial processing alone is limited. A very high degree of temporal correlation exists whenever there is little motion in the scene. Even if there is movement, high correlation may still exist depending on the spatial characteristics of the image.

2.3.1 Motion Compensation Estimation

In any temporal compression scheme the signal is compressed by first predicting how the next frame will appear and then sending the difference between the prediction and the actual image. A reasonable prediction would be the previous frame. This sort of temporal differential encoding is very similar to Differential Pulse Code Modulation (DPCM) and performs very well when the motion between adjacent frames is insignificant. If there is significant motion however this scheme would perform worse than if the next frame had simply been coded by itself.

Motion compensation and estimation is a process of improving the performance of any temporal compression scheme when motion occurs. In this procedure, displacement needs to be calculated between the previous frame and the present frame. If this information is known at the decoder site, then the previous frame can be shifted or displaced in order to obtain a more accurate prediction of the next frame that has yet to be transmitted. Motion displacement could be generated on a frame, partial frame or a pixel basis. Motion vectors (displacement) are generally calculated

on a partial frame basis (with the area of the portion chosen to equal a superblock) since, it would be too expensive (to calculate the motion vector at the encoder and provide the information to the decoder) on a pixel basis while, it is not very useful to generate a single motion vector for an entire frame. The dimensions of the super block vary from implementation to implementation.

The process of displacing portions of a previous frame in order to predict the next frame is shown in figure 2.5.

2.4 Rate Buffer Control

Since the output rate for channel transmission is fixed while the data is variable length huffman coded, it is necessary to rate buffer control the output. This rate buffer is normally implemented as a one frame FIFO (First In First Out) after the huffman coder.

The FIFO input rate is continuously monitored and the quantization level is adjusted to prevent buffer overflow or underflow. As the quantization level is decreases the block length increases and the FIFO input rate increases while an increase in quantization level causes the block length to decrease and hence the FIFO input rate to decrease.

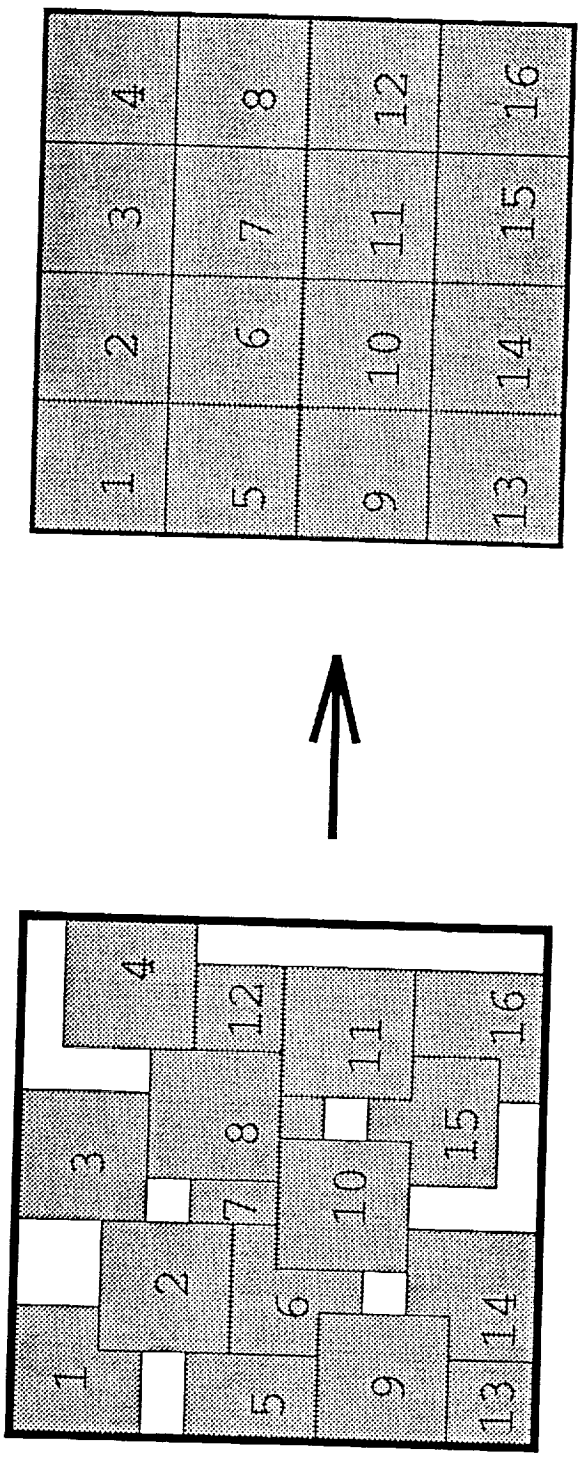


Figure 2.5: Using Motion Compensation To Predict Next Frame

Chapter 3

Designing A Video Codec

3.1 A Basic Video Coder

A basic video coder has five stages: a motion compensation stage, a transformation stage, a lossy quantization stage, and two lossless coding stages. The motion compensation like DPCM takes the difference between the present image and the previous image if they are alike. The transform concentrates the information in a few coefficients, the quantizer is responsible for selecting the high energy DCT coefficients. The two coding stages compress the data close to their symbol entropy. This coding stage is considered lossy since the reconstructed image is not exactly the same as the original image (due to the quantizer). Lossless coders (without quantization stage) have been found to achieve very poor compression.

The frame work of a basic video codec is given in the Figure 3.1

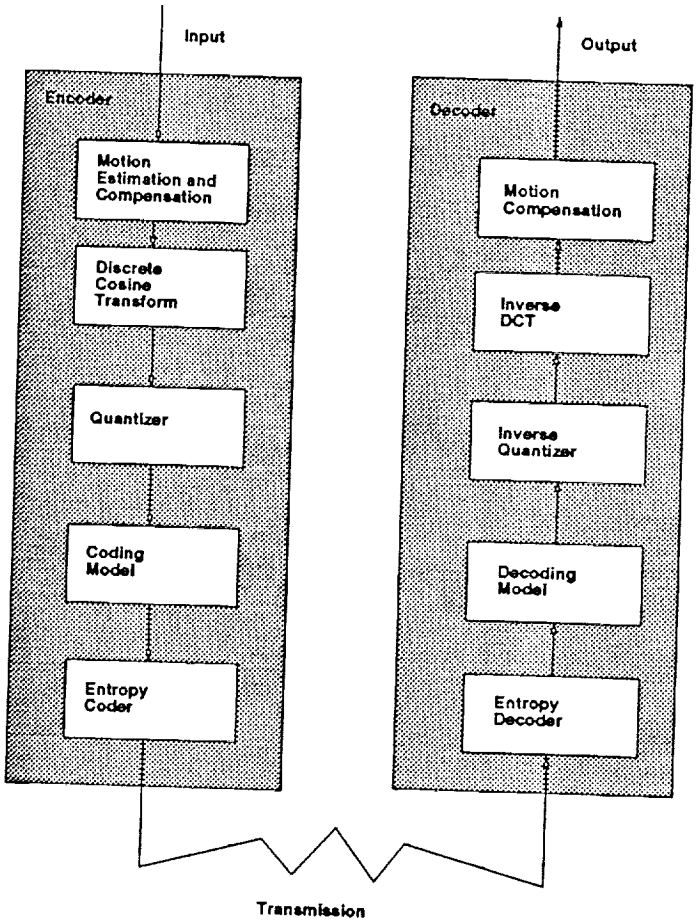


Figure 3.1: Block diagram of Video Codec: (a) Coder and (b) Decoder

3.1.1 Motion Compensation Estimation

Since most frames in an image sequence look alike (excepting during the shifts due to movement) the difference between th e blocks are coded rather than the blocks themselves. The motion compensation model for a basic codec is shown in the Figure 3.2

The motion compensation model (shown in the Figure 3.2) separates the motion sequences into three different types of frames: intraframes, which are coded without any prediction; forward predicted frames, which are predicted based on the intraframes; bidirectionally predicted frames , which are predicted based on either the intraframes or the forward predicted frames.

3.1.2 Transform Stage

The transform stage is used to concentrate the energy into a few coefficients of the block. The image is normally divided into small blocks to simplify the complexity of this stage. The transform method chosen by CCITT is the two dimensional 8 by 8 DCT. The formula for two dimensional 8 by 8 DCT can be written as

$$F(u,v) = (1/4)C(u)C(v)\sum_{i=0}^{7}\sum_{j=0}^{7} f(i,j)\cos\frac{(2i+1)\pi u}{16}\cos\frac{(2j+1)\pi v}{16} \qquad (3.1)$$

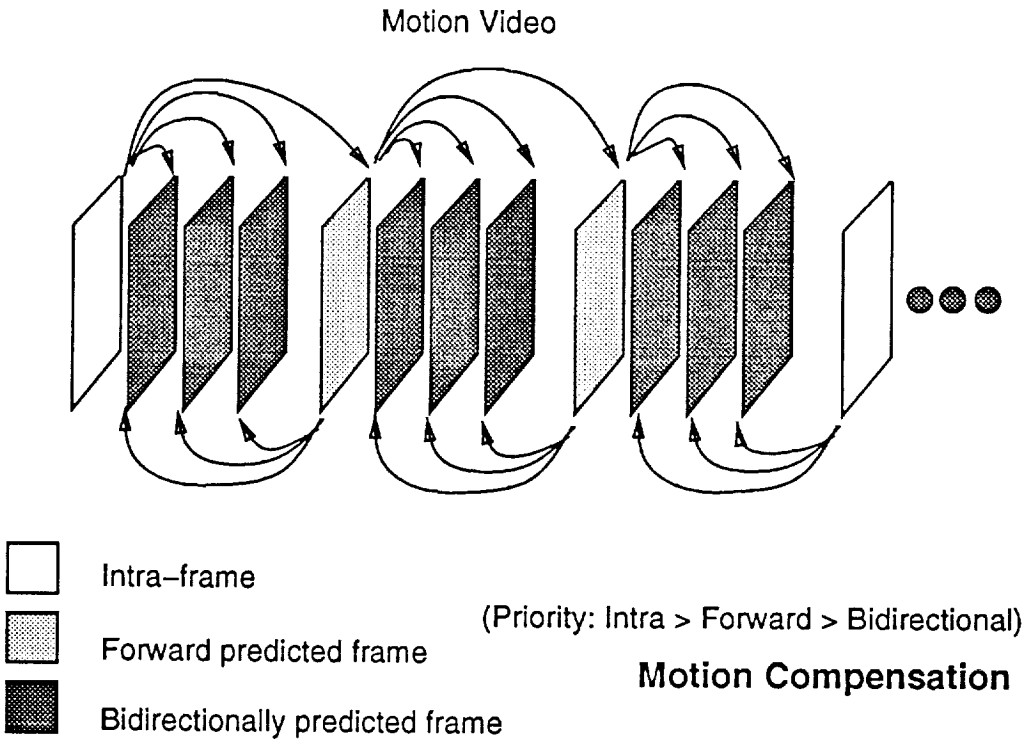


Figure 3.2: The Motion Compensation Model For A Simple Codec

where

$$C(x) = \begin{cases} 1/\sqrt{(2)} & \text{for } w = 0 \\ 1 & \text{for } w = 1, 2, \cdots, 7 \end{cases}$$

The output of 8 by 8 DCT is in such a way that the average value of the block (DC coefficient) is in the upper left corner. Progression from left to right represents the increasing number of vertical edges, while progression from top to bottom represents increasing number of horizontal edges.

The inverse 2D DCT can be written as

$$f(u,v) = \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)C(v)F(u,v) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \qquad (3.2)$$

3.1.3 Quantization

The DCT coefficients are quantized to increase the number of zero valued coefficients. The DCT blocks are quantized with the DC and the AC terms separately. Quantization is the lossy stage in the coding scheme. The image quality deteriorates if the quantization is too coarse, while useless bits coding noise have to be spent if the quantization is too fine.

3.1.4 Coefficient Scanning

The quantized DCT coefficients are arranged in a zig-zag pattern (see Figure 3.3). Zig-zag pattern scanning arranges the DCT coefficients in ascending frequency order. The assumption behind zig-zag pattern scanning of DCT coefficients is that the

lower frequency components tend to have higher values than the higher frequency components. In images the high frequency DCT coefficients are normally zero. Hence, zig-zag pattern scanning helps in accumulating zeroes towards the end of the block and helps in reduction of transmitted coefficients.

The DC coefficients are encoded by the number of significant bits followed by the bits themselves, while the AC coefficients are encoded based on the number of zeroes before the next non-zero coefficient.

The inverse run-length coder translates the coded stream into either a DC coefficient or a run-length followed by an AC coefficient. The zero coefficients (based on the run length)) are appended into the buffer followed by the non-zero AC coefficients.

3.1.5 Entropy Coding

The final processing step for a basic video codec is entropy coding. This step achieves additional (lossless) compression based on the statistical characteristics of the quantized DCT coefficients. The most commonly used entropy coding scheme is Huffman coding. To compress data symbols, the Huffman coder creates shorter codes for frequently occurring symbols and longer codes for occasionally occurring symbols.

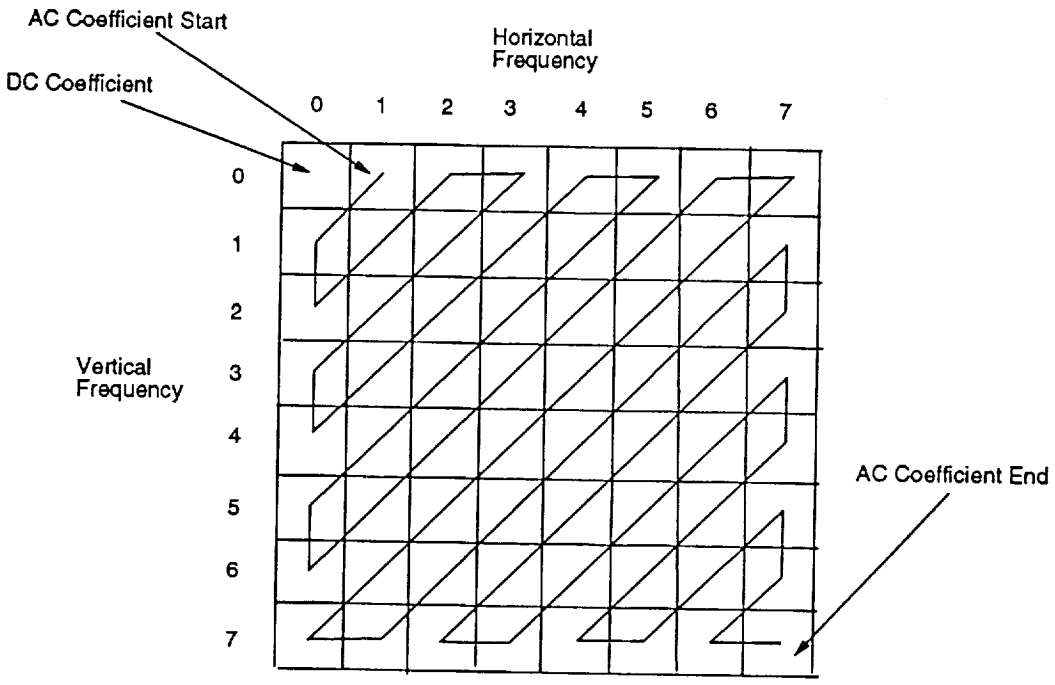


Figure 3.3: The DCT transform

3.2 Error Correction Protection

The problems inherent in the packet video are packet loss and packet delay. It is therefore necessary for coding techniques compatible with packet video to consider these problems.

A simple technique to protect the information packets is to add parity packets to the existing information packets. A lost or delayed packet could then be recovered by initiating error correction. This scheme however has a disadvantage in that it increases the number of packets transmitted and hence contributes to the loss of packets. However a good error correction scheme is suppose to more than make up for the loss of packets due to increased rate.

A single error correcting (7,4) hamming code was implemented and incorporated with the video coder to protect the information packets. The decoder was modified to perform error correction only when one packet was lost. This scheme was found to work well due to following two reasons

- the probability of losing a single packet is higher than the probability of losing more than one packet. Hence most of the time only a single packet is lost (even after taking the increased rate into consideration)
- it does not corrupt the correct packets in the process of recovering the lost packets.

Chapter 4

Error Concealment

In packet switched networks, even with error correction protection, packet loss is unavoidable. Hence a method of error recovery, which takes the characteristics of video signals into account, is required. This process, known as error concealment, attempts to recover or reconstruct the missing blocks from the structured picture data. In this chapter various methods of error concealment are discussed. The underlying assumptions in all these methods are

- pixels in the image are much smaller than any of the important details
- most of the pixels' neighbors represent the same structure.

A method of error concealment based on the motion estimation is proposed and the results are presented.

4.1 Intra Frame Processing

4.1.1 Block Averaging

This method of error concealment involves replacing the missing block in the frame by the average of the surrounding blocks. The basic assumption in this method of error concealment is that the neighboring blocks represent the same structure. The process of averaging the surrounding blocks to replace the missing block can be done either in the spatial domain (spatial averaging) or in the frequency domain (spectral averaging) . Both, spatial averaging and spectral averaging, perform well when the missing blocks are not at the edges. The figures (see Figures 4.1, 4.2) shows a frame (Susie sequence) obtained after performing Spectral and spatial error concealment on a frame in which 1 % of the blocks were randomly thrown away.

4.2 Inter Frame Processing

4.2.1 Block Replacement

In this method the missing blocks in a frame are replaced by the blocks, in the corresponding location, from the previous frame (see Figure 4.5). This method would work well if there were not much motion between the two successive frames. Figures 4.6, 4.7 show two successive frames without significant motion between them. The missing blocks in the Figure 4.7 were filled by the blocks from the preceding frame (Figure 4.6).

Figure 4.1: Spatial Error Concealment (Susie Sequence)

Figure 4.2: Spectral Error Concealment (Susie Sequence)

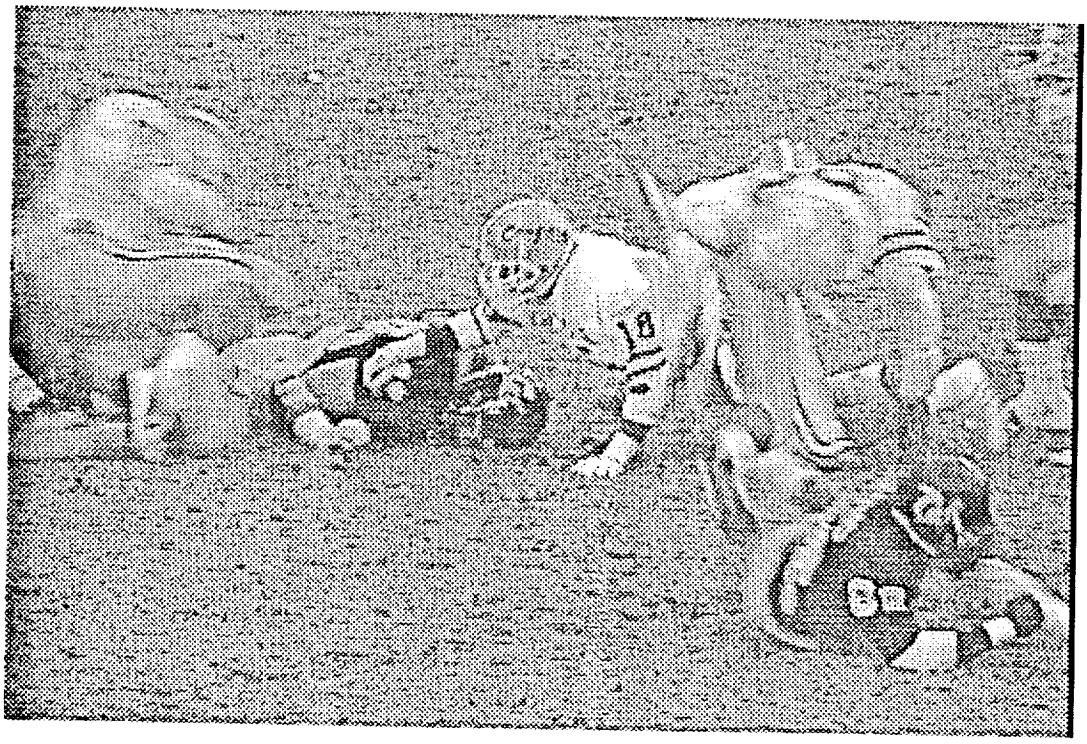


Figure 4.3: Spatial Error Concealment (Football Sequence)

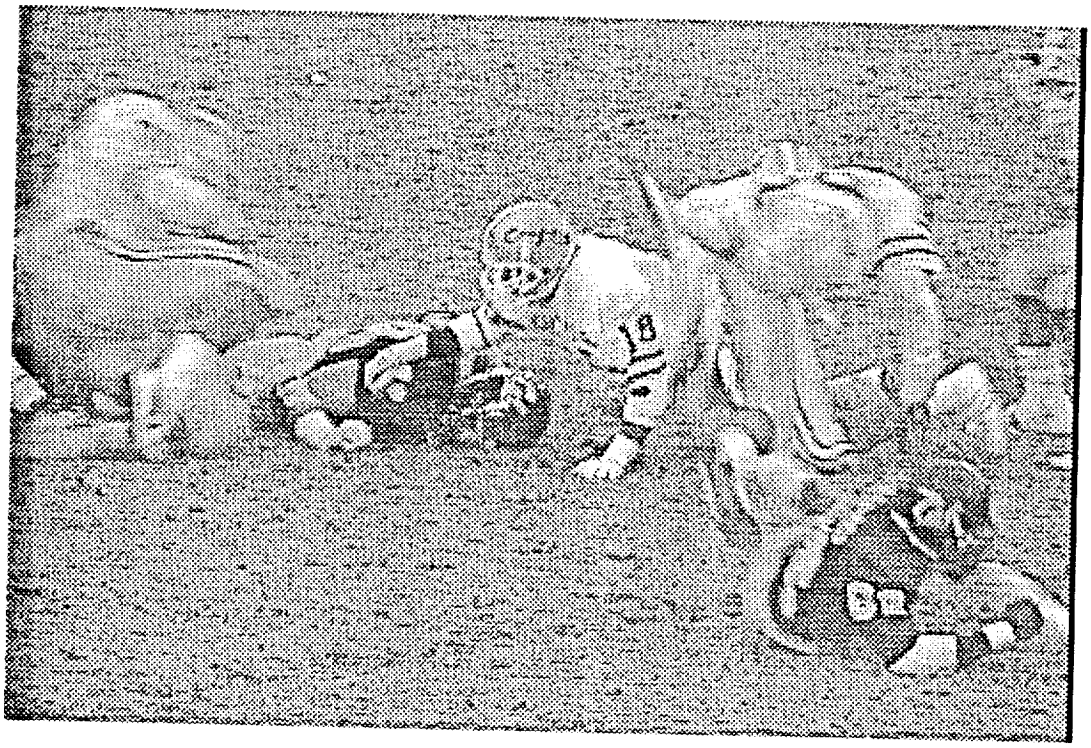


Figure 4.4: Spectral Error Concealment (Football Sequence)

The missing blocks in Figure 4.9 were filled by the corresponding blocks from the previous frame (see Figure 4.8). This is an example where there is a significant motion between two successive frames.

4.3 Error Concealment Model Based on Motion Estimation

Since the frames of a video coded sequence are motion compensated, it is necessary to consider the type of the frame before performing error concealment. The motion compensation model for the video coded sequence is shown in Figure 3.2. This model divides a motion sequence into three different types of frames which are intraframes, forward predicted frames and bidirectionally predicted frames (refer to section 3.2).

The flowchart of the model (see Figure 4.10) explains the various steps involved in the error concealment for different types of frames (I, B or P). This model takes advantage of the fact that most frames in a motion sequence look alike, and uses the information from the previous and/or next frames to fill up the missing blocks in the frame (present). This model estimates the motion of the present frame (frames with the blocks missing) with the previous and/or the next frame (depending on the type of the frame) and compares it with a threshold (T) (to take care of the scene changes or significant motion).

Intraframes:

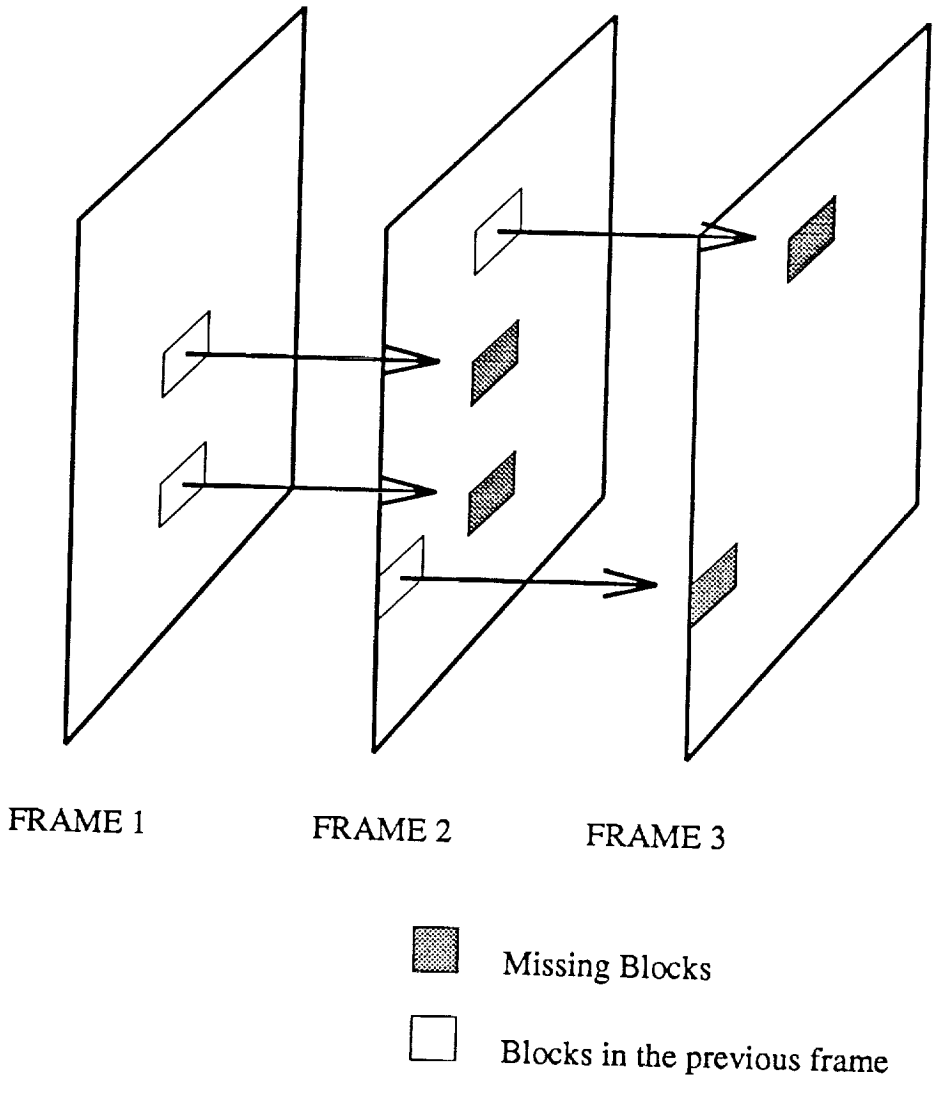


Figure 4.5: Replacing Blocks From Previous Frame

Figure 4.6: Replacing Blocks From Previous Frame (no significant motion between frames): Previous Frame

Figure 4.7: Replacing Blocks From Previous Frame (no significant motion between frames): Present Frame

Figure 4.8: Replacing Blocks From Previous Frame (significant motion between frames): Previous Frame

Figure 4.9: Replacing Blocks From Previous Frame (significant motion between frames): Present Frame

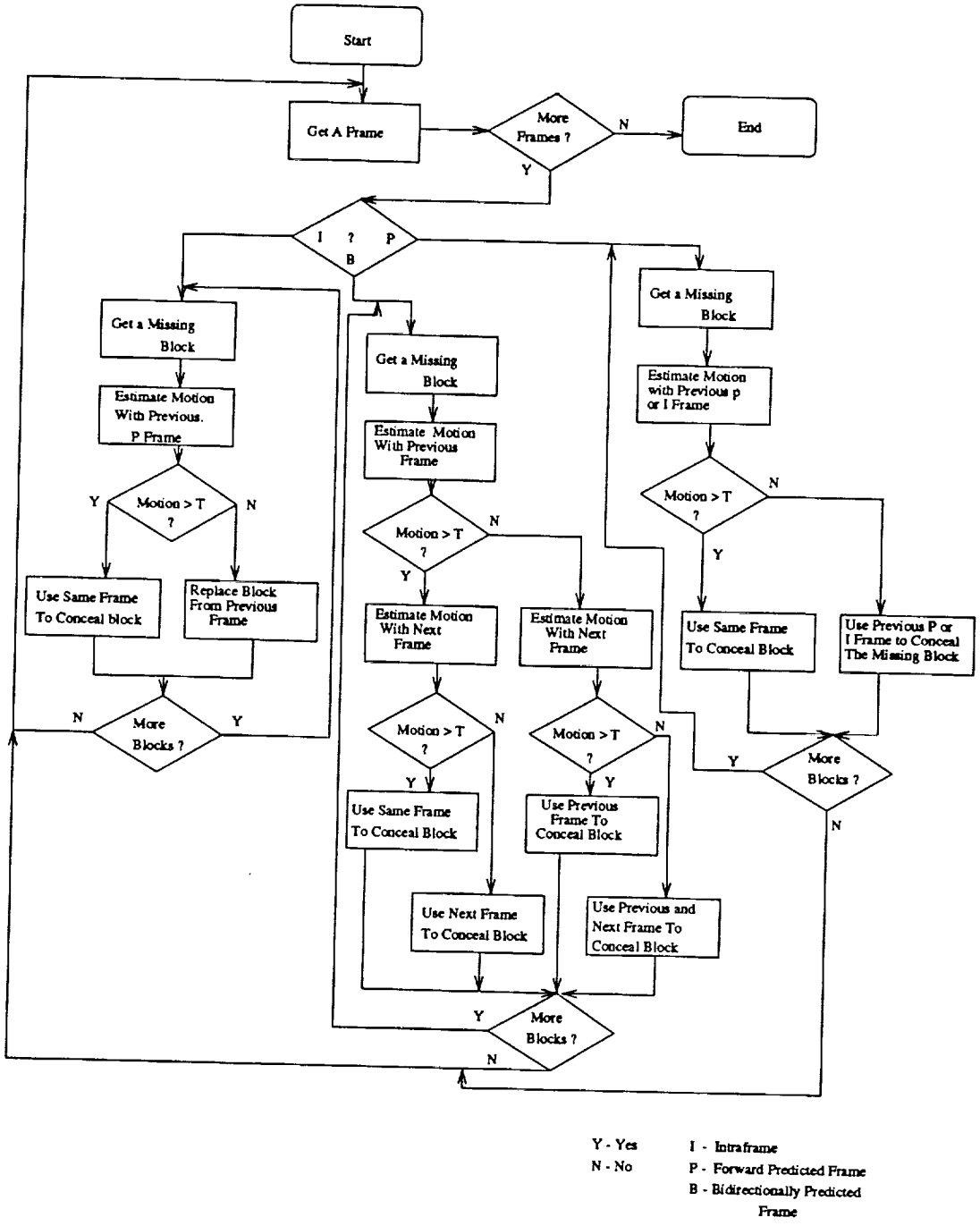


Figure 4.10: Flowchart of Error Concealment Model

The intraframes are error corrected before performing error correction on the forward predicted and bidirectionally predicted frames. These frames were reconstructed by estimating the motion with the previous forward predicted frame.

Forward Predicted Frames:

The forward predicted frames are error reconstructed using the information from the previous intraframe. Again motion is estimated between the two frames before replacing the missing blocks.

Bidirectionally Predicted Frames:

To reconstruct the missing blocks in the bidirectionally predicted frames the information in both the adjacent frames was used. Motion between the present frame and the previous frame and the present frame and the next frame was estimated. Missing blocks in the present frame was then replaced by performing a frequency domain interpolation between the blocks obtained from the previous and the next frame.

The process of motion estimation involves taking the four blocks surrounding a missing block and moving through a predefined search space (starting with the same location) in the previous and/or the next frame and comparing the error against a threshold (T). The threshold (T) was found to depend on the activity in the block and variance (of the blocks surrounding the missing block) was found to be a good estimate of the activity in the region.

The following Figures show the first twenty four frames of (original and recon-

structed) susie and football sequences (in which 10% of the packets were lost during transmission).

The PSNR for the first twenty four frames before and after reconstruction for susie and football sequences is shown in Figures 4.11 to 4.26. A significant increase in PSNR values was observed with error concealment as can be observed from the graphs for the first twenty four frames. The artifacts due to error concealment were less observable in football sequence than the susie sequnce (even though the loss of packets is roughly the same) because of the fast motion of the objects. All the sequences used in this chapter are contained in an accompanying video tape for subjective evaluation.

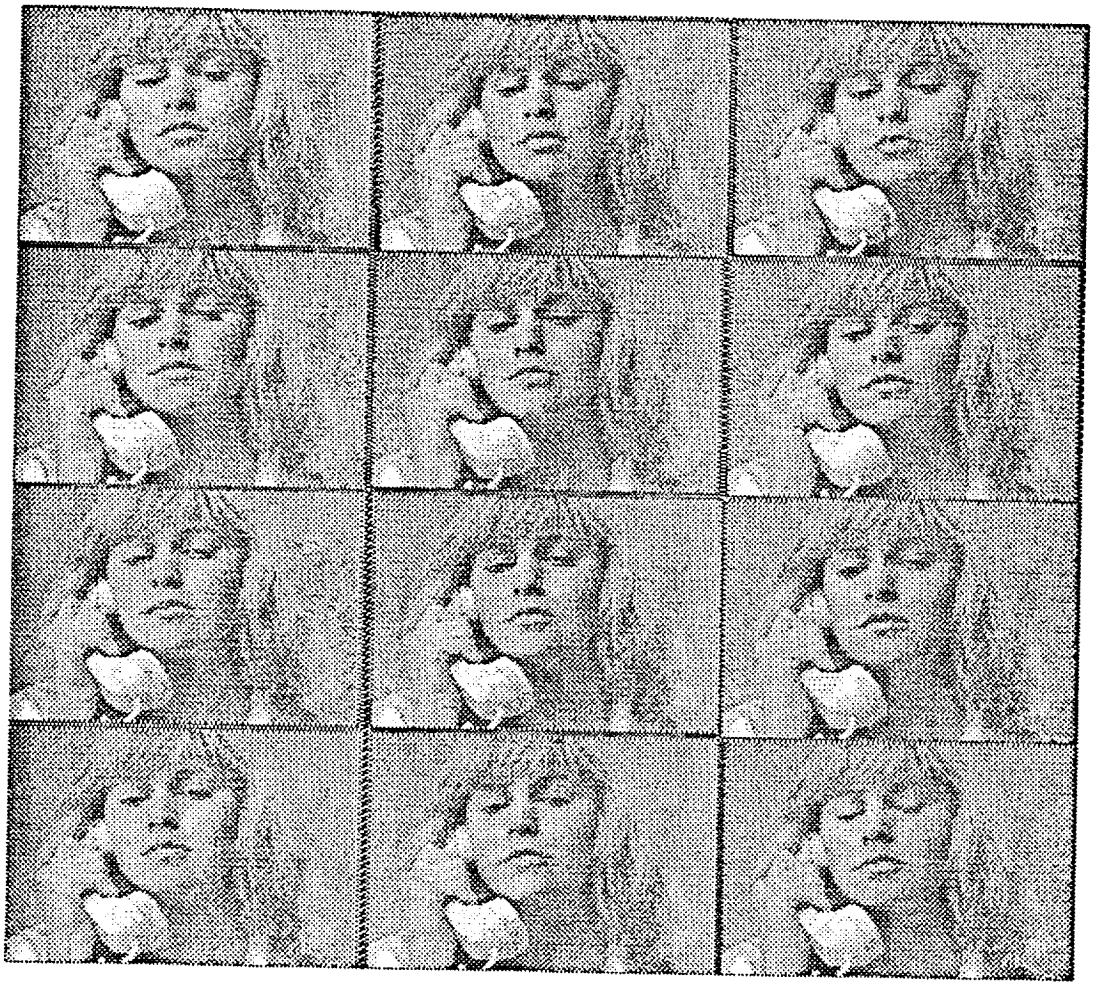


Figure 4.11: Frames 1:12(Original Susie Sequence)

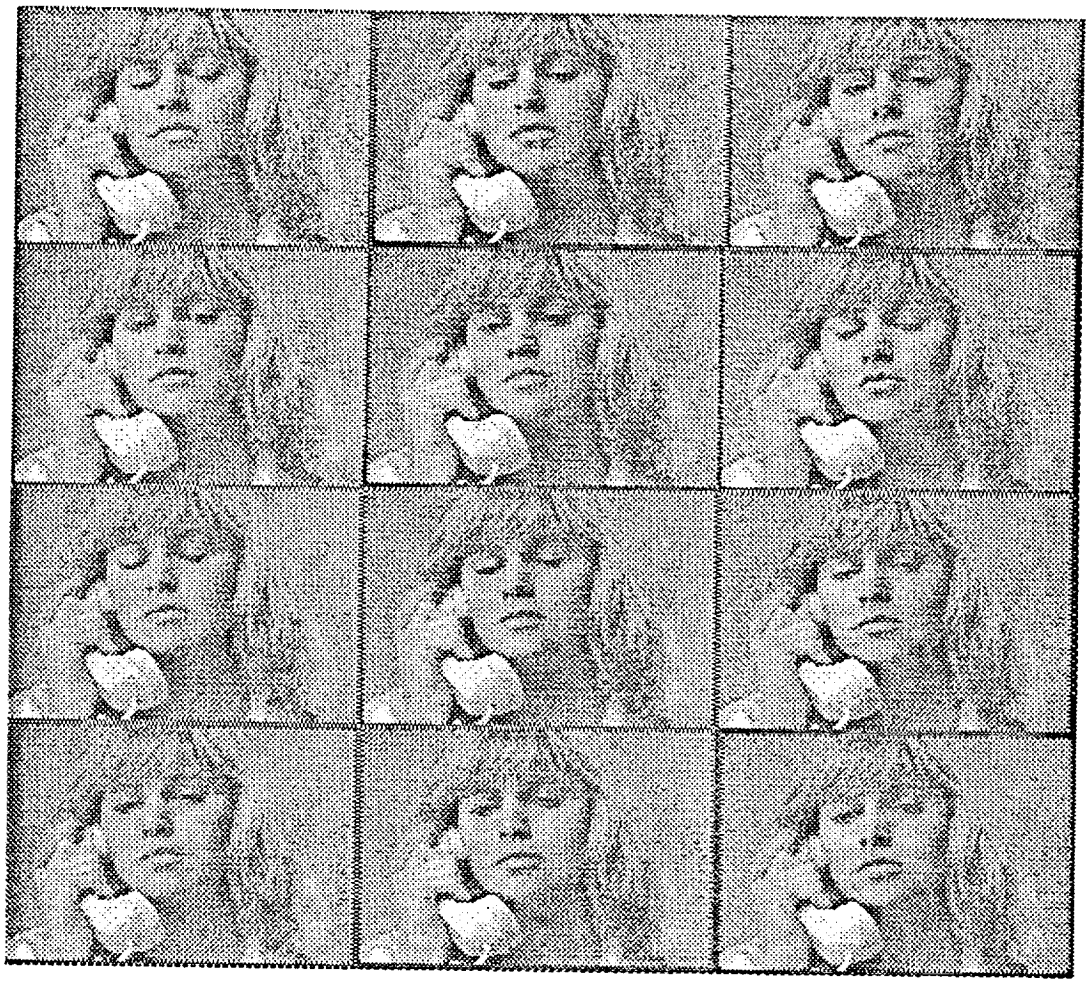


Figure 4.12: Frames 13:24(Original Susie Sequence)

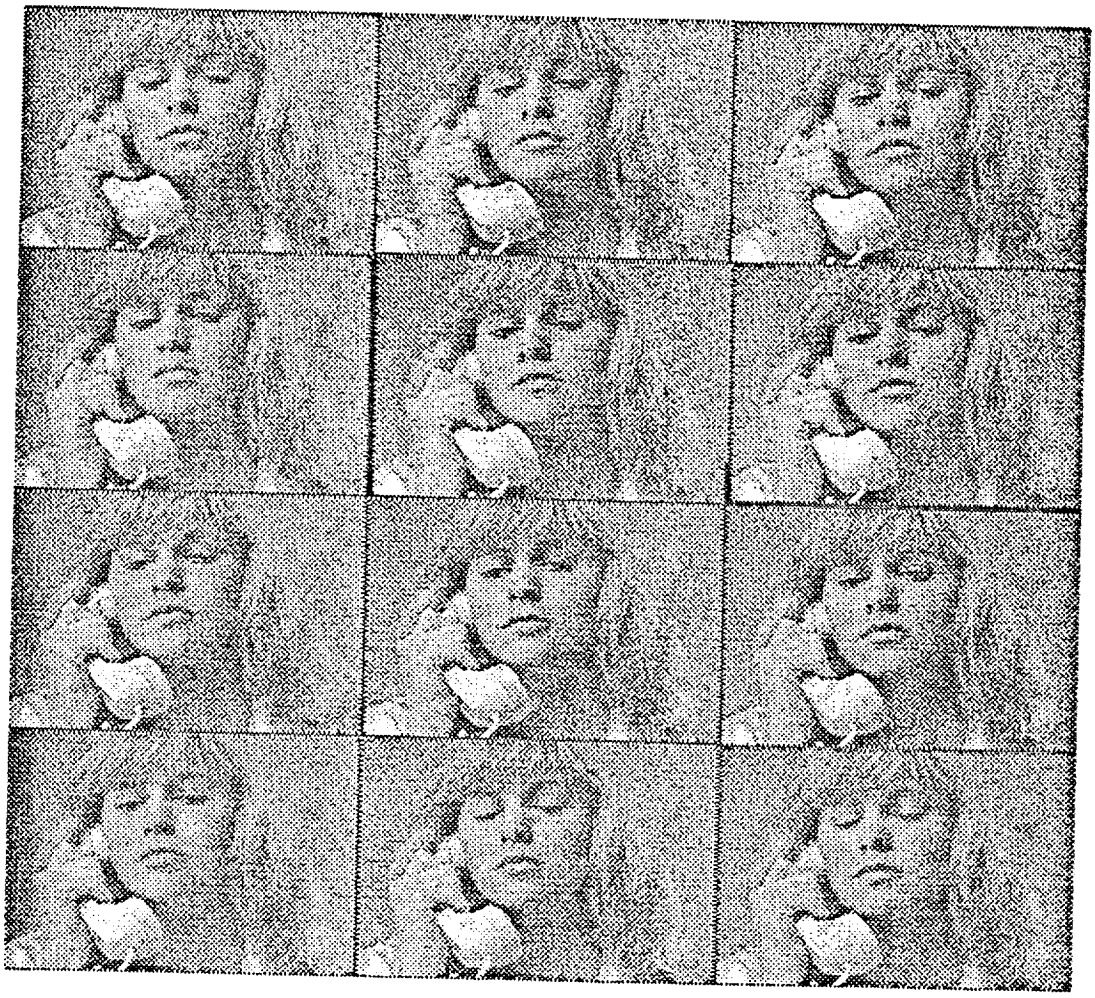


Figure 4.13: Frames 1:12(Reconstructed Susie Sequence)

Figure 4.14: Frames 13:24(Reconstructed Susie Sequence)

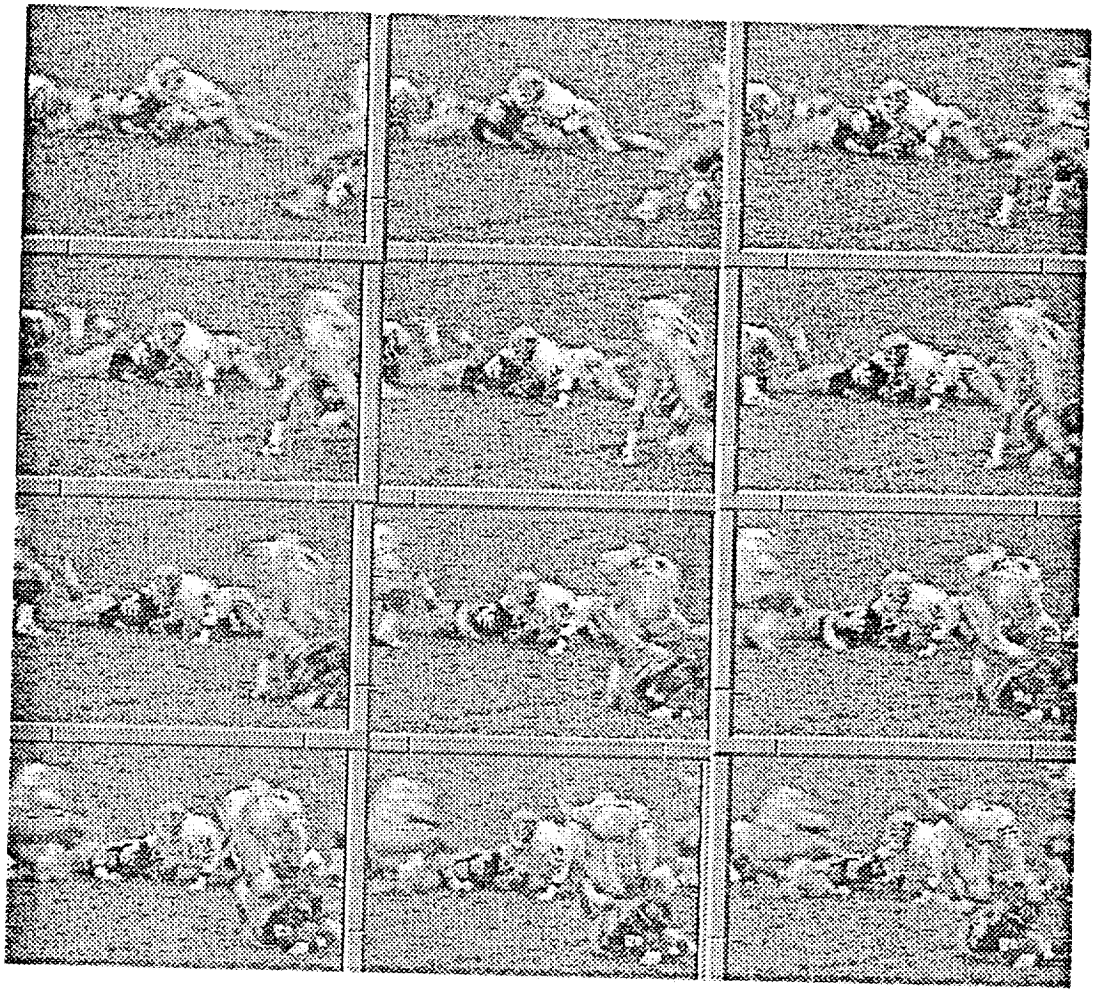


Figure 4.15: Frames 1:12(Original Football Sequence)

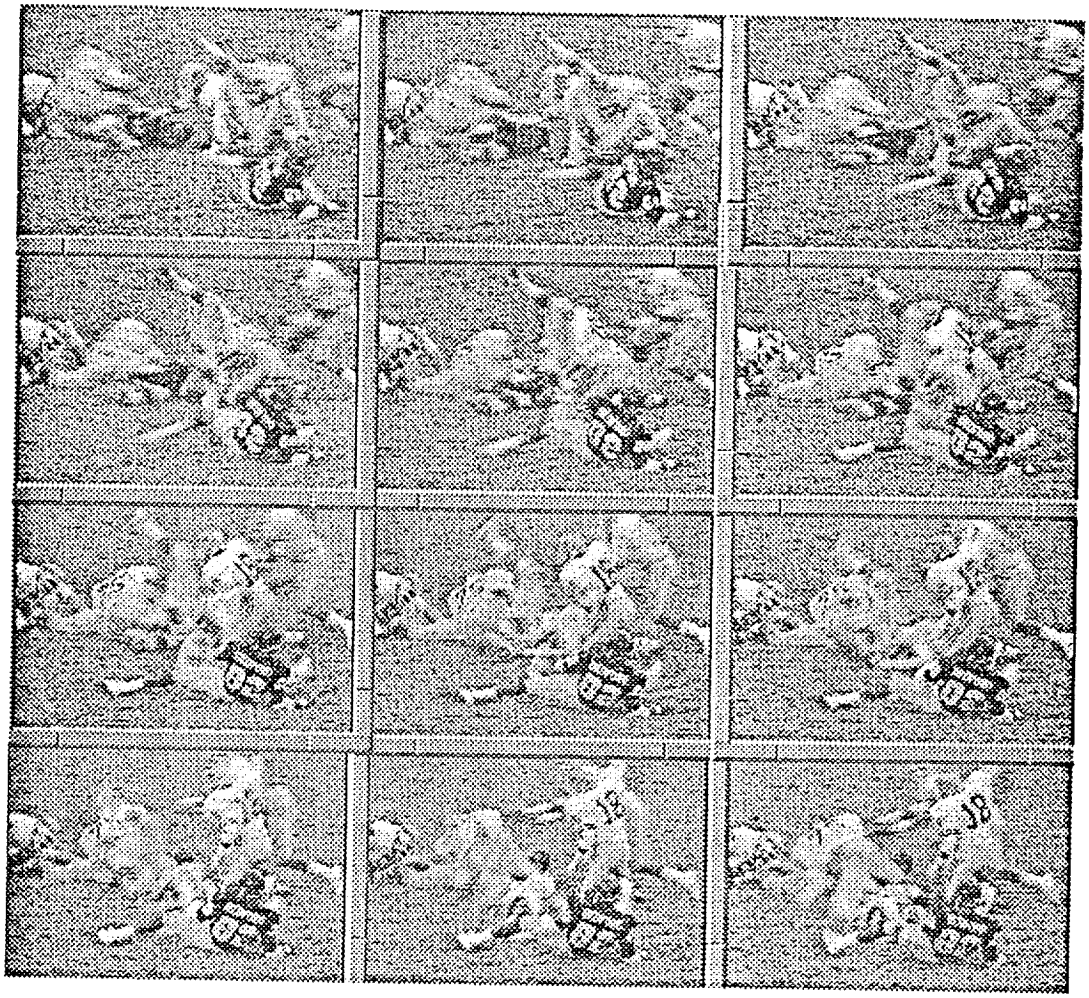


Figure 4.16: Frames 13:24(Original Football Sequence)

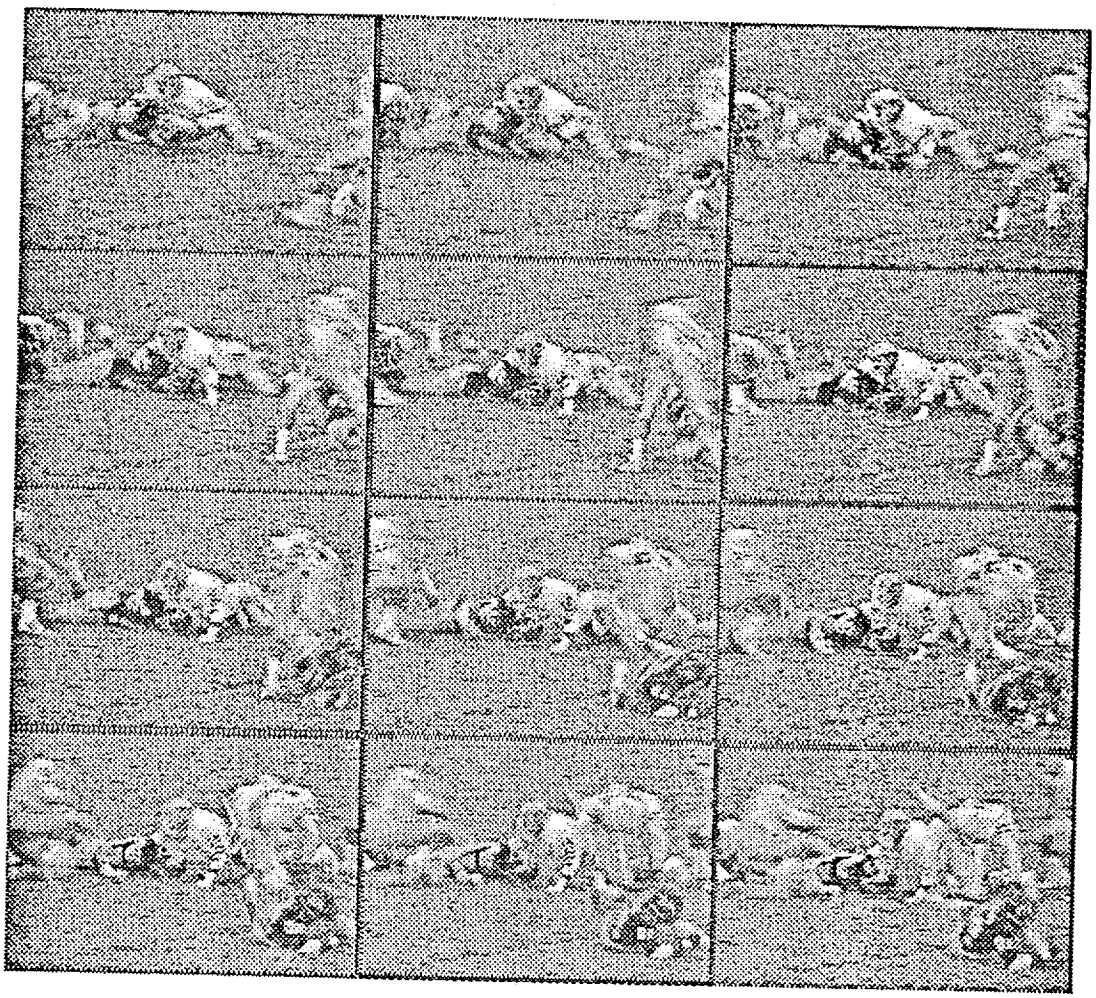


Figure 4.17: Frames 1:12(Reconstructed Football Sequence)

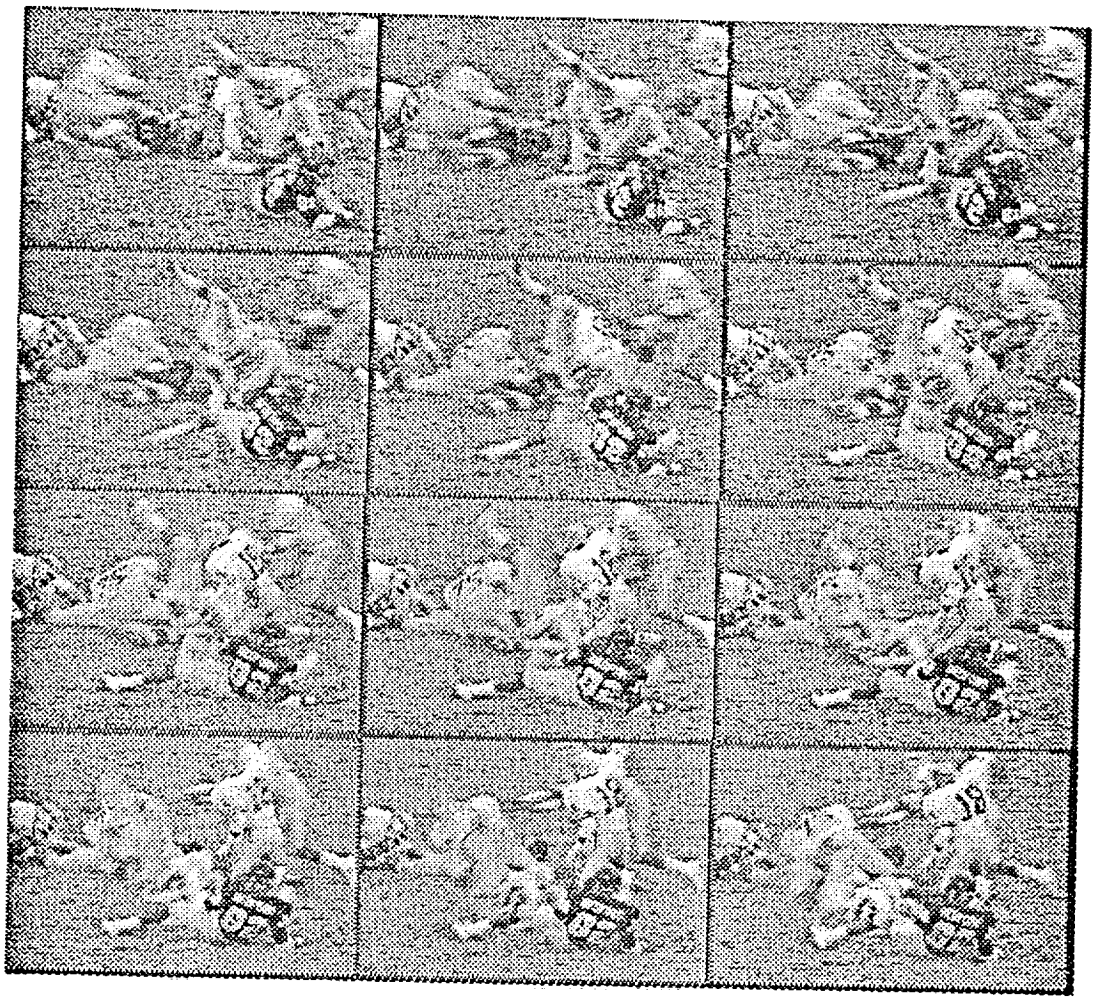


Figure 4.18: Frames 13:24(Reconstructed Football Sequence)

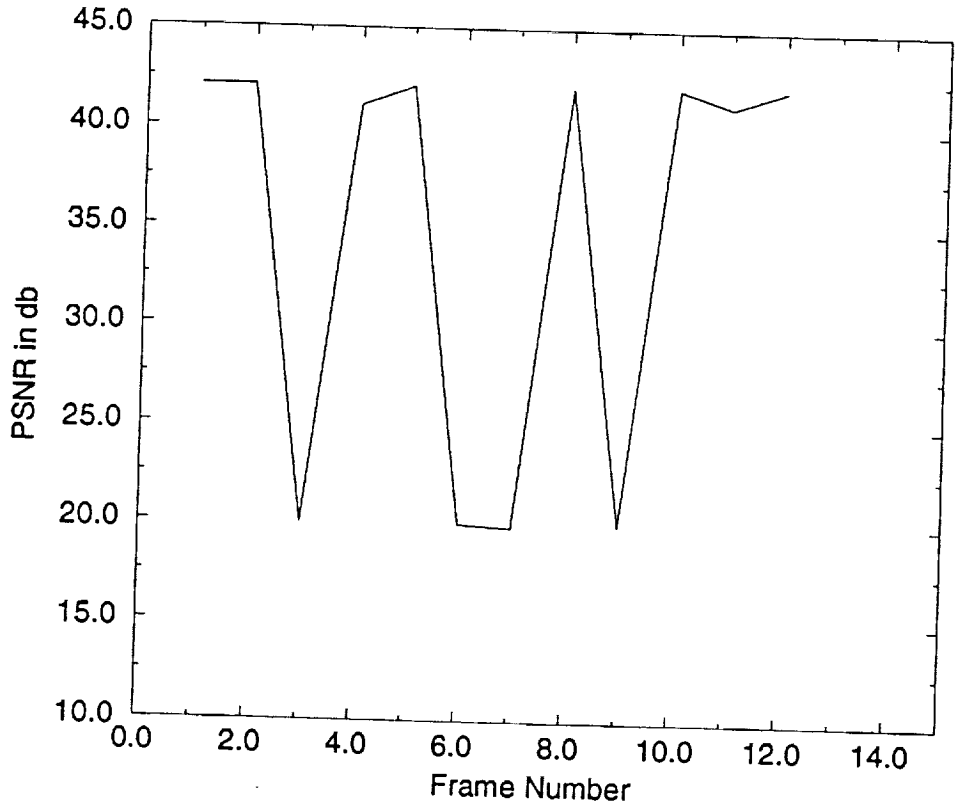


Figure 4.19: PSNR of 1:12 frames(Susie Sequence) before Error Concealment

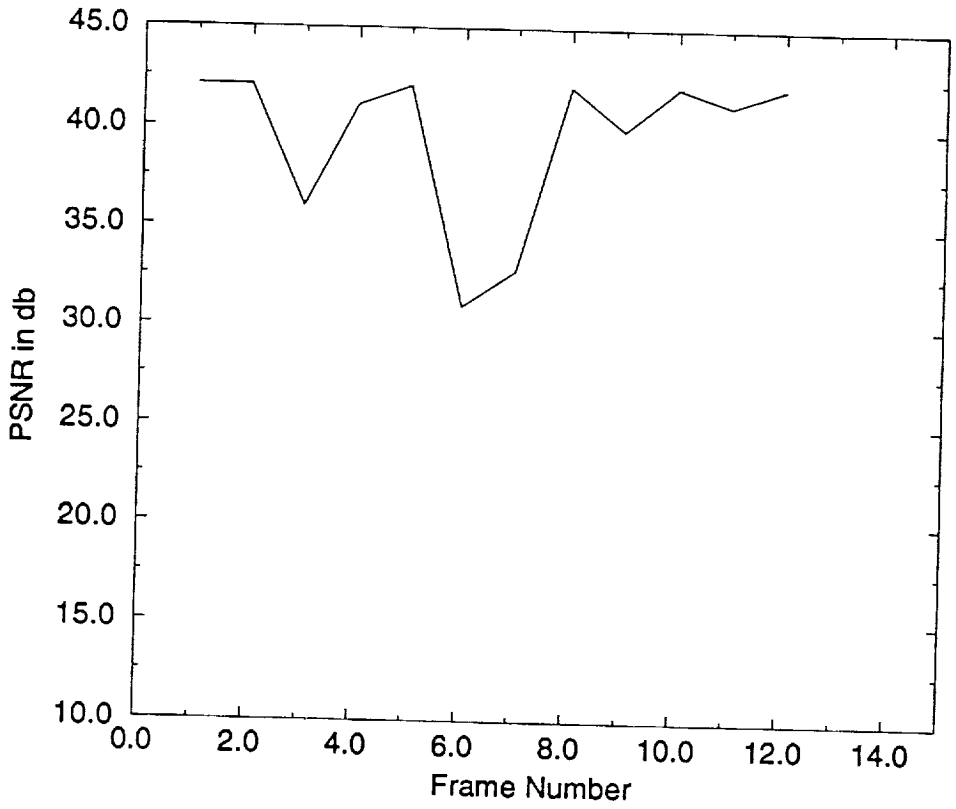


Figure 4.20: PSNR of 1:12 frames(Susie Sequence) after Error Concealment

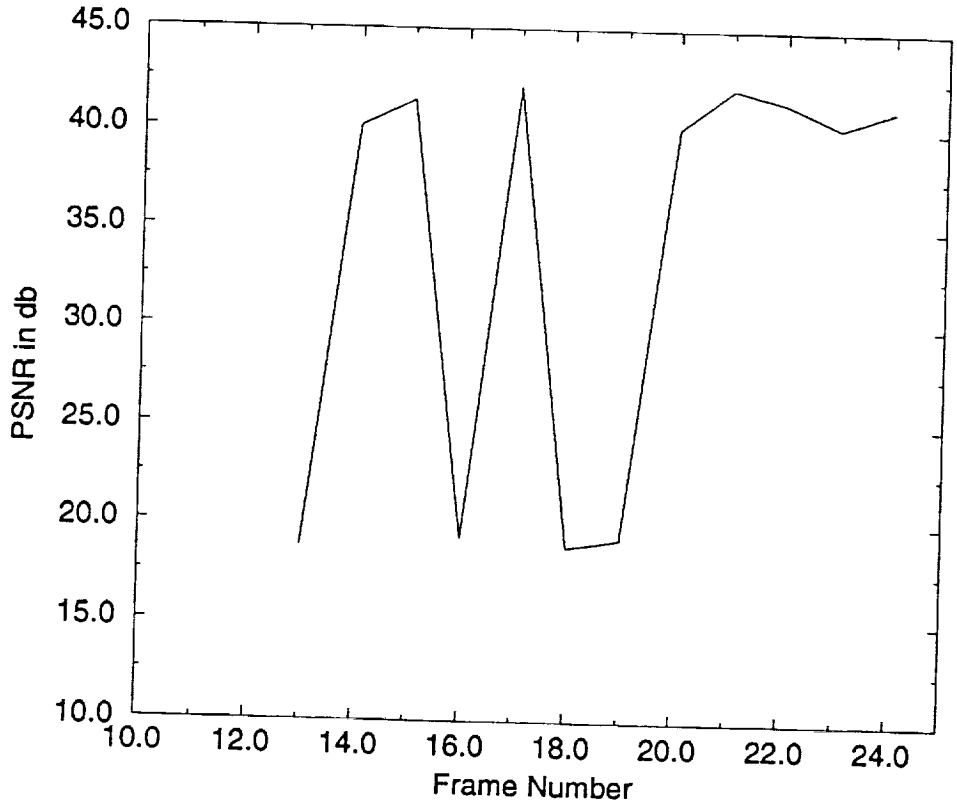


Figure 4.21: PSNR of 13:24 frames(Susie Sequence) before Error Concealment

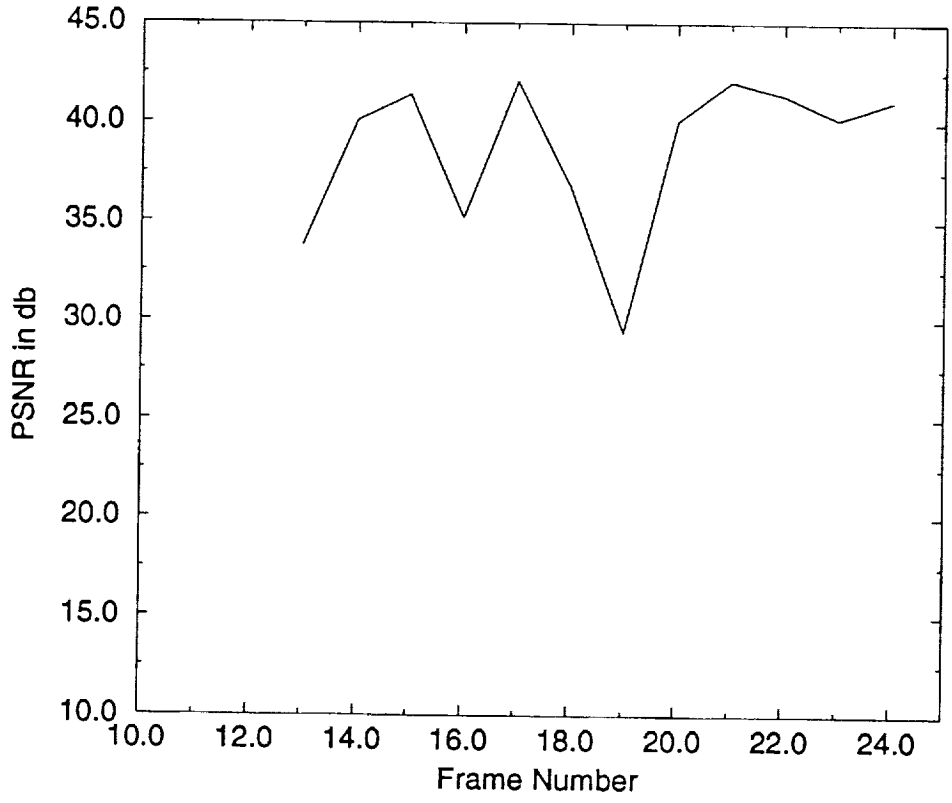


Figure 4.22: PSNR of 13:24 frames(Susie Sequence) after Error Concealment

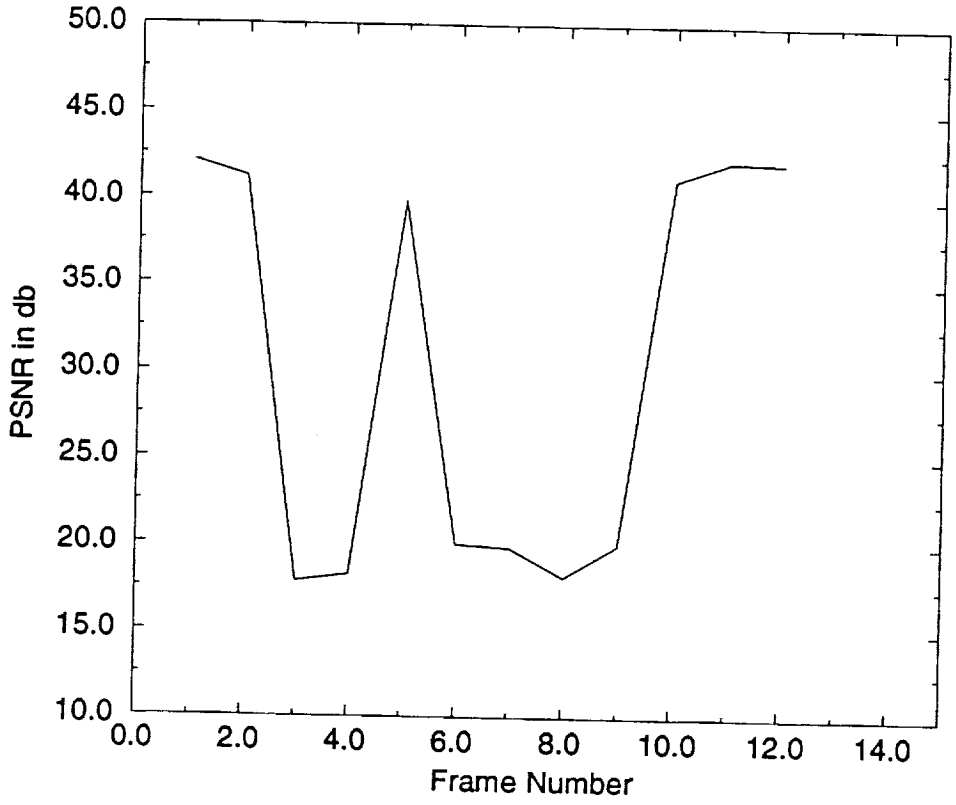


Figure 4.23: PSNR of 1:12 frames(Football Sequence) before Error Concealment

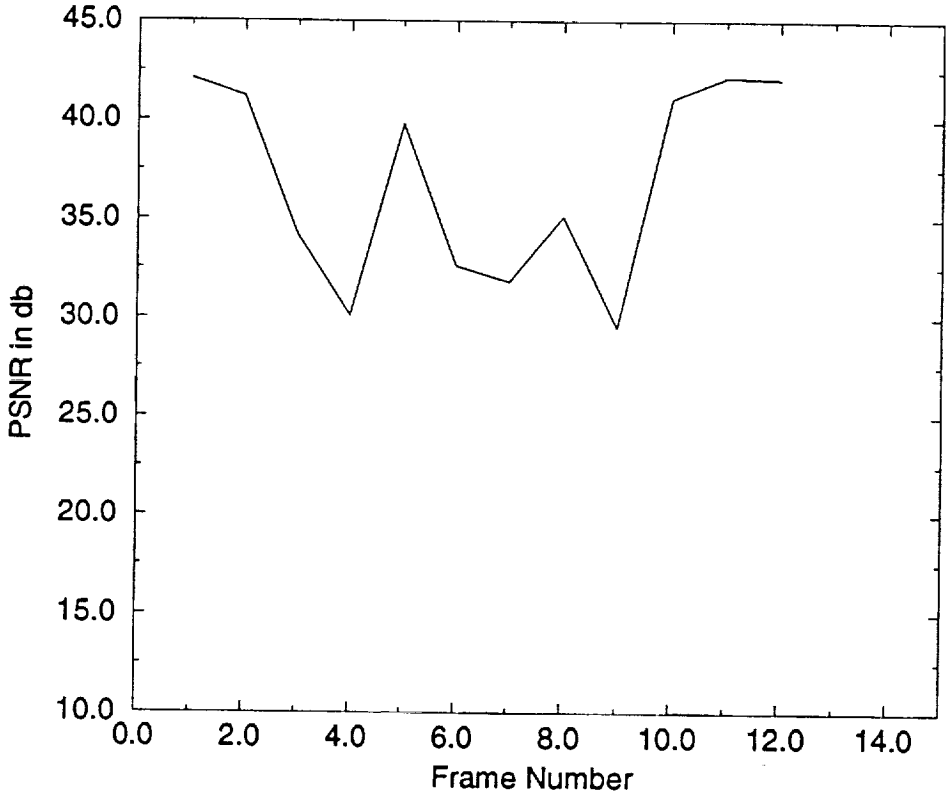


Figure 4.24: PSNR of 1:12 frames(Football Sequence) after Error Concealment

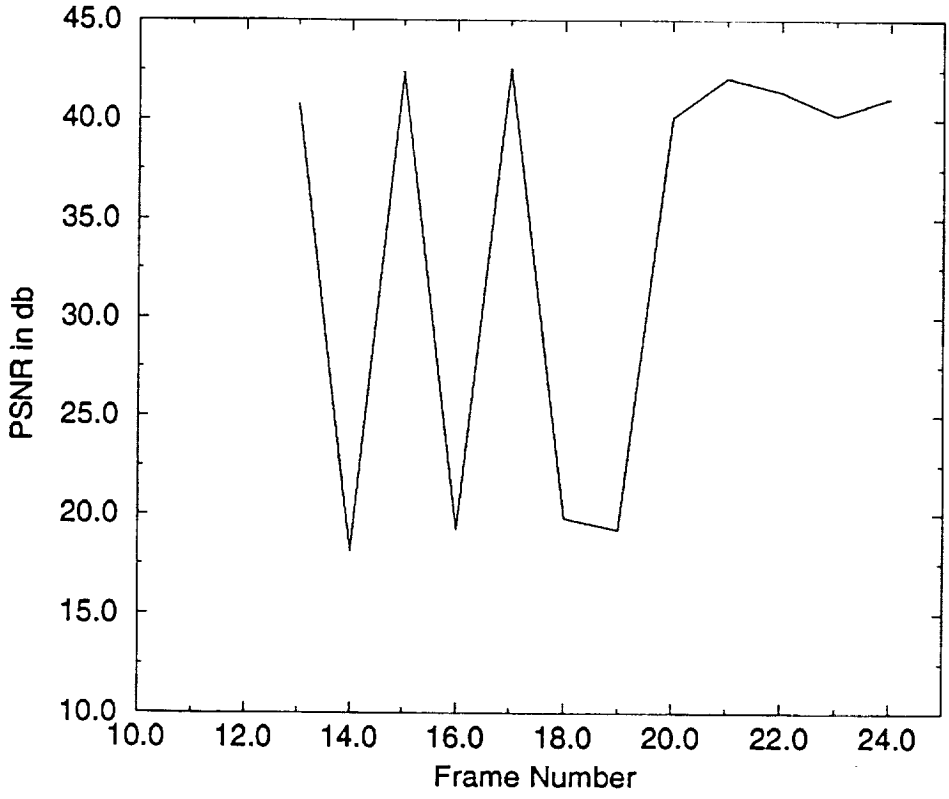


Figure 4.25: PSNR of 13:24 frames(Football Sequence) before Error Concealment

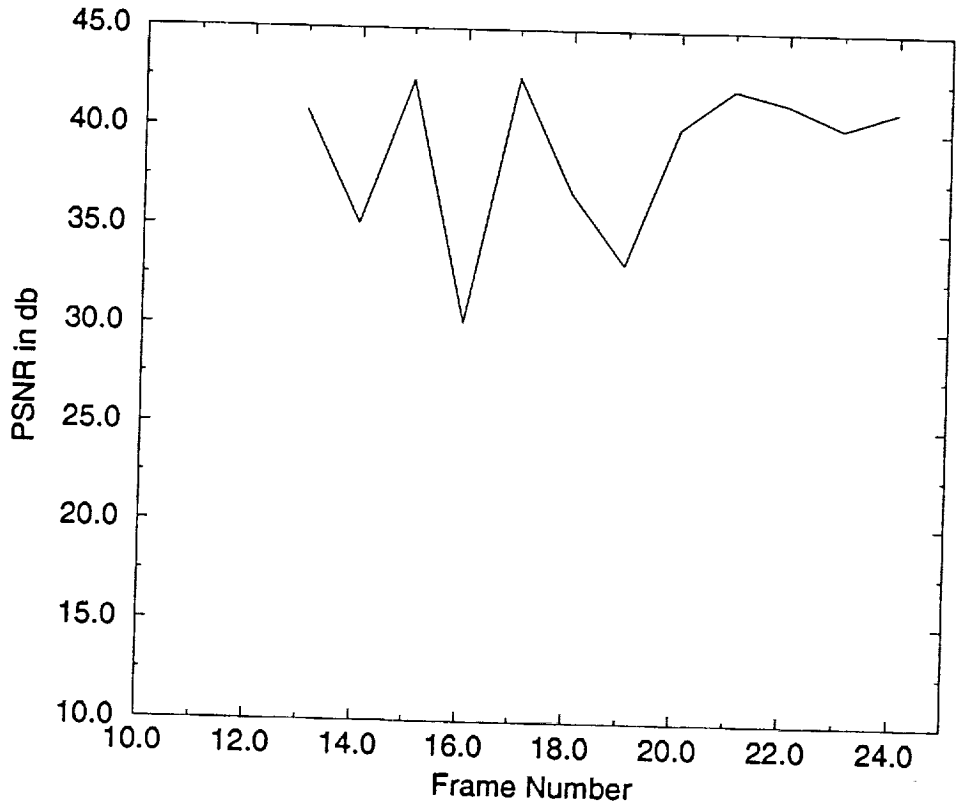


Figure 4.26: PSNR of 13:24 frames(Football Sequence) after Error Concealment

Chapter 5

Conclusion

Packet loss in packet switched networks is inevitable even with error correction as was mentioned before. Hence there is necessity for error concealment at the receiver. In this thesis a method of error concealment based on motion estimation is proposed. This method takes advantage of the fact that most frames look alike and hence uses the information from the previous and the next frames (where possible) to conceal the errors (missing packets) by estimating the motion. The method proposed was implemented together with a standard MPEG video codec and tested for its performance on two motion sequences (Susie and Football). For both the sequences a significant increase in PSNR was observed with error concealment.

An interesting aspect of the proposed method is its response at scene cuts. It is obvious that the motion compensation of the previous frame in such cases is meaningless as the contents of the previous frame and the current frame are entirely

different. This method however does work better than other error concealment methods (which use only the past information to perform error concealment [4]).

In some frames (particularly where the motion was complicated) this method resulted in degradation of picture quality (as can be observed on the accompanying video). This however can be attributed more to the poor performance of the motion estimation algorithm and the incorrect motion compensation than to the error concealment procedure.

Bibliography

[1] Netravali A.N. "On quantizers for DPCM coding of picture signals". *IEEE Trans. Inf. Theory IT-23*, April 1977.

[2] N.S Jayanth Peter Noll. *Digital Coding Of WaveForms.* PRENTICE-HALL, INC., Englewood Cliffs, NewJersey 07632, 1984.

[3] Gregory K. Wallace. "The JPEG Still Picture Compression Standard". *IEEE Transactions on Consumer Electronics*, Dec Submitted in 1991.

[4] Ghanbari M. and Serfidis V. "Cell-Loss Concealment in ATM Video Codecs". *IEEE Transactions on Circuits and Systems*, June 1993.

[5] Karlsson G. and Vitterli M. "Sub-band coding of video signals for packet switched networks". *Proc. SPIE Visual Commun. Image Processing II*, Oct 1987.

[6] Ghanbari M. "Two-layer coding of video signals for VBR networks". *IEEE J. Select. Areas Commun.*, June 1989.

[7] Morrison G. and Beaumont D. "Two-layer video coding for ATM networks". *Signal Processing: Image Commun.*, June 1991.

[8] Ghanbari M. and Serfidis V. "Heirarchical motion estimation using texture analysis". *IEEE Int. Conf. on Image Processing*, April 1992.

[9] Pirsch P. Mursman H.G and Grallert H.J. "Advances in picture coding". *Proc. IEEE*, April 1985.

[10] Wada M. "Selective recovery of packet loss using error concealment". *IEEE J. Select. Areas Commun.*, June 1989.

[11] Ghanbari M. "An adapted H.261 two layer video codec for ATM networks". *Proc. 3rd Int. Packet Video Workshop*, March 1990.

[12] Nomura M. and Ohta N. "Layered packet-loss protection for variable rate video coding using DCT". *II Intl. Workshop on Packet Video, Torino, Italy*, Sept 1988.

SCAN VQ

N15

Scan Predictive Vector Quantization of Multi-spectral Images *

Nasir D. Memon† Khalid Sayood‡

Abstract

Conventional VQ based techniques partition an image into non-overlapping blocks which are then raster scanned and quantized. Image blocks that contain an edge, that is, an abrupt change in intensity, result in high frequency vectors. The coarse representation of such vectors leads to visually annoying degradations in the reconstructed image. We present a solution to the edge degradation problem. Our approach reduces the number of vectors with abrupt intensity variations by using an appropriate scan to partition an image into vectors. The manner in which vectors are extracted from the image will influence the performance of quantization schemes that are subsequently applied to the vectors. Therefore, it is necessary to examine the problem of optimally scanning the image in order to minimize the number of vectors with abrupt intensity variations. In this paper we address this question and give a novel technique for extracting vectors from an image which when quantized give much better results than if the vectors were obtained in the standard manner. We show how our techniques can be used to enhance the performance of vector quantization of multi-spectral data sets. Comparisons with standard techniques are presented and shown to give substantial improvements.

Edics IP 1.1

*K. Sayood wishes to acknowledge the support of Goddard Space Flight Center for this work

†Department of Computer Science, P.O. Box 70, Arkansas State University, AR 72467, (501)-972-3090, email: memon@quapaw.astate.edu, FAX: 501-972-3950

‡Department of Electrical Engineering, University of Nebraska, Lincoln, NE 68588-0123, (402)-472-6688, email: ksayood@eecomm.unl.edu

1 Introduction

Vector Quantization (VQ) has been found to be an efficient image compression technique due to its ability to approximate patterns in the source output [6]. Conventional VQ based techniques partition an image into non-overlapping blocks which are then raster scanned and quantized. The codebooks are constructed from averages of similar patterns in a training set. Because of the way they are obtained the codebook vectors tend to be smooth. Even when explicit efforts are made to include high frequency vectors such as edge vectors, the number of such entries are limited by the relatively small size of the codebook [15]. Therefore the likelihood of finding a good approximation to a smooth vector is much more than that of finding a good match to a high frequency or edge vector. While most blocks in an image do not contain edges, edges are perceptually very important, and coarse representations of these blocks can lead to a substantial degradation in perceptual quality. Furthermore, edges are very important in a number of image processing applications, such as classification and pattern recognition. Edge degradation can adversely effect such applications.

Various solutions have been proposed to improve edge reproduction. Some of the more well-known ones are Classified VQ, Finite-State VQ, and Predictive VQ [1]. In this paper we present an alternative solution to the edge degradation problem. Instead of trying to increase the number of codebook entries which more closely match the high frequency patterns in the input to the Vector Quantizer, our approach is to try and *reduce* the number of high frequency vectors at the VQ input. Our approach minimizes the number of vectors with abrupt intensity variations by using an an appropriate scan to partition an image into vectors.

In Figure 1 (left) we show a 5×5 segment taken from the Lena image plotted as a surface, with the vertical axis representing the intensity value and horizontal axes representing spatial co-ordinates. The segment contains an edge, that is an an abrupt change in intensity, as we move from left to right. In Figure 1 (right) we show two different vectors that were obtained by raster scanning the block. If we scan the block row by row, going from left to right,

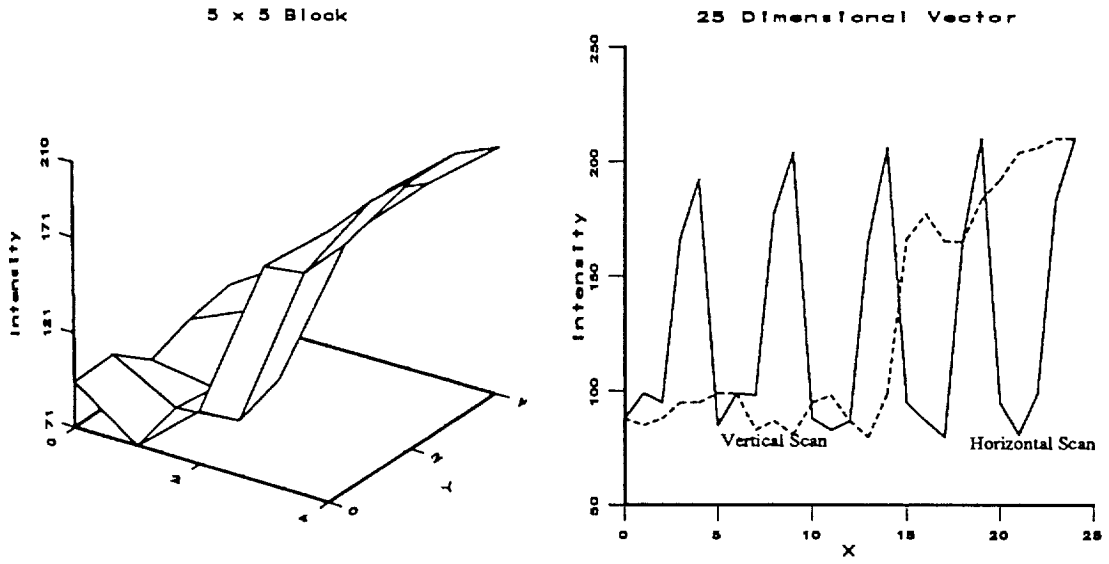


Figure 1: An image block (left) and two different vectors formed by scanning the block (right).

we obtain the vector plotted using a solid line. The vector formed in this manner has 4 peaks, each representing a transition from the end of one row to the beginning of another. Alternatively, if we scan the block column by column, going from top to bottom, we obtain the vector shown in the same plot in dotted line. This vector is much smoother than the vector obtained using the left to right scan, and is much more likely to have a codebook entry 'close' to it. We can see that the manner in which vectors are formed from a block influences the nature of the resulting vectors, which in turn can influence the fidelity of their representation. This argument, can in fact be extended to the entire image.

Therefore, we would like to find a systematic method for scanning the image which would result in vectors that are smooth in some sense, and therefore more likely to find close matches. In this paper we address this question and give a novel technique for extracting vectors from an image which when quantized give much better results compared to vectors obtained by standard techniques.

The idea of scanning an image in an 'efficient manner' in order to improve performance

of a compression scheme is certainly not new. In terms of practical image compression techniques, work to date has concentrated on scanning the image using fixed scans that exploit the inherent two-dimensional relationships that are present in image data. In fact, the (perhaps) earliest work investigating alternative scanning techniques was done by Wyner with the objective of scrambling video signals in order to protect against eavesdropping [18, 19]. Later, Matias and Shamir [10] showed that using pseudo-random space filling curves for scrambling actually results in reducing bandwidth required for transmission.

By far the most popular of such special scanning techniques have been the discrete approximations of Hilbert and Peano space filling curves. In fact, Ziv and Lempel have shown that the problem of optimally compressing n–dimensional data can be reduced to that of optimally compressing a 1-dimensional string by using a discrete approximation of a Hilbert space filling curve [9]. However, their optimality result is asymptotic and the scheme they propose is not practically applicable to gray scale images. Nevertheless, the Hilbert scan has been effectively used to enhance the performance of a variety of image compression techniques. In [16] a Hilbert scan is used to rearrange pixels prior to vector quantization. An image compression technique based on a wavelet transform of vectors extracted by performing Hilbert/Peano like scans is reported in [2]. Yang et. al. [20] use Peano scanning along with fractal coding to compress still images. Cole [4] has used Peano and Hilbert scans for data compaction of raster graphics.

In the next section we review some ideas from previous work and briefly introduce the notion of scan models. In section three we show how scan models can be used to enhance the performance of vector quantization of multi-spectral data sets. We present comparisons of the proposed technique with standard techniques and show that the proposed techniques compare favorably. Finally, we conclude in section five with a brief summary and pointers to future work.

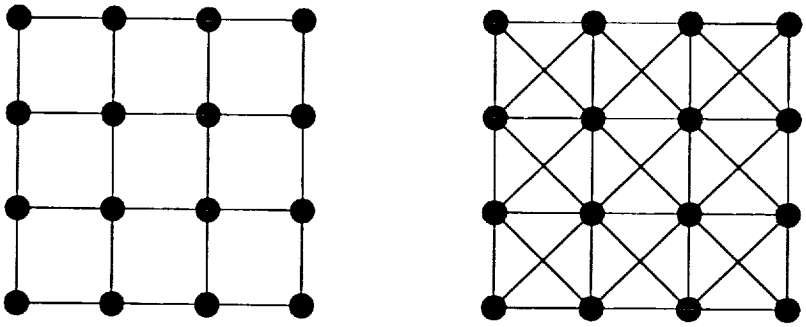


Figure 2: Adjacency graphs A_4 and A_8 for the 4 and 8 connected model respectively.

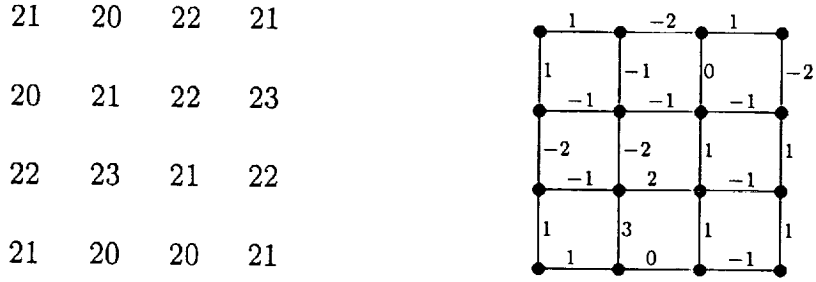


Figure 3: A digital image and its difference graph

2 Scan Models

In this section we review scan models and related concepts from previous work [14]. We consider a *digital image* P, to be an $M \times N$ array of integers such that $0 \leq P[m,n] < L-1$ for $0 \leq m < M$ and $0 \leq n < N$. The notion of 'adjacency' between pixels in an image is often based on the 4-neighborhood model or the 8-neighborhood model, the adjacency graphs of which, A_4 and A_8, are shown in Figure 2. An image P induces a *weighting function* on the edges of an adjacency graph if we assign the weight on an edge to be the difference in intensity values of the two pixels corresponding to the vertices incident upon the edge. We call the weighted version of an adjacency graph, induced by an image P to be the *difference graph* of P. An image and its difference graph using the 4-neighborhood model are shown in Figure 3.

Given an adjacency graph, we call any spanning tree of the graph, a *scan model.* A scan model specifies an order for traversing the pixels of an image, for the given neighborhood

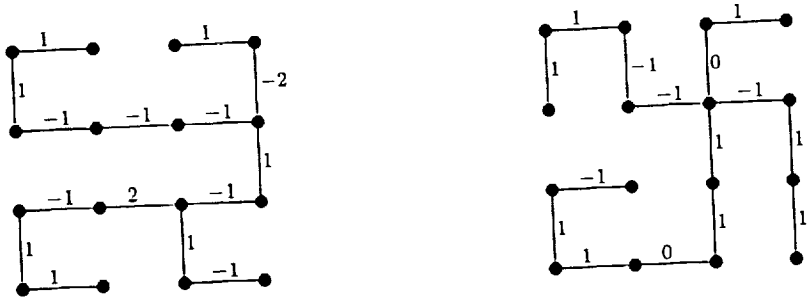


Figure 4: Two prediction trees for the image in Figure 1

scheme. Standard traversal schemes like the raster scan and the Hilbert scan [9] are special cases of a scan model. A scan model can also be viewed as a non-causal prediction model for an image. For example, the scan model on the left in figure 4 specifies that the prediction for pixel $(1, 2)$ should be the intensity value of its neighbor on its left; and the right neighbor of pixel $(1, 3)$ is to be used as a prediction for its value and so on. In a similar manner the prediction scheme for each pixel is specified. In this paper, we use the first interpretation, that is, we view a scan model as specifying a traversal of an image.

If we are to use scan models for image processing tasks, then we would be interested in a model that is optimal with respect to some objective function that depends upon the specific application at hand. In [14] we look at a few objective functions and investigate algorithms for constructing optimal models. What is interesting is that given our formulation, the problems related to finding good models can be abstracted as graph problems. That is, problems which involve constructing a spanning tree of the difference graph with desired properties.

A natural objective function to minimize is the sum of absolute weights on the edges corresponding to a scan model, which we call a MAW scan model. A MAW scan minimizes the absolute sum of differences between successive pixels in the scan. Computing a MAW scan involves finding a minimum weight spanning tree of the difference graph, after the original weights are replaced by their absolute value. A minimum weight spanning tree of a weighted graph can be computed in time $O(MN \log\log(MN))$ [3]. In our case, since the

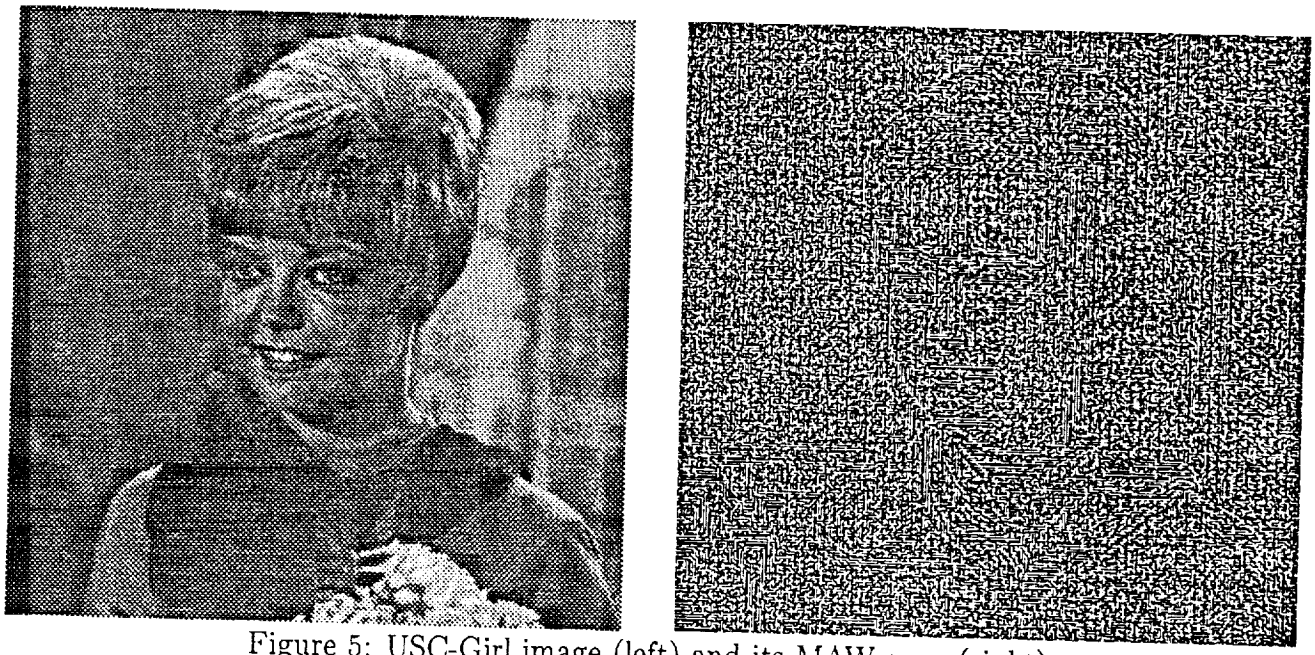


Figure 5: USC-Girl image (left) and its MAW scan (right).

graph is *sparse*, a minimum weight spanning tree can be computed in time $O(MN \log^* MN)$ [1] [5], which for all practical purposes amounts to $O(MN)$.

An MAW scan that was constructed for the USC-Girl image is shown in Figure 5. It can be seen that unlike statistical models like context based models and linear models, scan models are essentially 'structural' in nature. They capture the essential two-dimensional structure inherent in an image. Hence, they could be potentially of use in a variety of image processing tasks. In previous work we have investigated their application to lossless compression of still images [13] and multi-spectral image data [12], and in lossy plus lossless image compression [11]. In the rest of this paper we investigate their application to partitioning an image into vectors prior to quantization.

[1] $\log^* n = \min\{i \geq 0 : \log^i n \leq 1\}$; $\log^i$ is defined by $\log^1 n = \log n$ and $\log^{i+1} n = \log(\log^i n)$

Image	Blocks	MAW Scan
USC-Girl	136	48
Couple	133	42
Girl-1	96	20
Girl-2	216	38
House	330	67
Tree	638	227

Table 1: Comparison of variance vectors from blocks and MAW scan

3 Using an MAW scan model to form vectors

As we said before, in conventional VQ techniques image blocks that contain an edge result in high detail vectors, which usually result in visually annoying degradations in the reconstructed image. In this section we apply the notion of a scan model to address the edge degradation problem. Our approach minimizes the number of vectors with abrupt intensity variations by using an MAW scan to partition an image into vectors.

An MAW scan by definition minimizes differences between successively scanned pixels. Hence vectors formed by taking k successive pixel values along an MAW scan will be highly correlated and can be clustered and quantized with lesser distortion. In order to test this hypothesis we performed the following experiment on a standard test set of 256×256 RGB images taken from the USC database. We first partitioned the image into 4×4 blocks and computed the variance of each block. The mean of the variance values, rounded to the nearest integer is shown in the first column of table 1. We then computed an MAW scan for the image and formed vectors of dimension 16 by performing a *depth-first* traversal of the MAW tree. The variance of each vector was computed and the mean value is shown in the second column of table 1. It can be seen that vectors obtained from an MAW scan contain much less activity than those formed from $k \times k$ non-overlapping blocks.

In order to test our second hypothesis that for a fixed bit rate, vectors formed from MAW scans can be quantized with lesser distortion as compared to $k \times k$ blocks we per-

Image	Block VQ		Scan VQ	
	SNR	PSNR	SNR	PSNR
USC-Girl	21.02	30.35	22.79	32.09
Couple	16.33	31.72	19.26	34.61
Girl-1	27.93	32.99	31.61	36.63
Girl-2	27.93	32.99	31.61	36.63
House	26.37	31.31	27.50	32.44
Tree	21.21	26.02	22.87	27.69

Table 2: Comparison of SNR and PSNR for Block VQ and Scan VQ.

formed another experiment. Here, we first used the Generalized Lloyd's algorithm (GLA) for generating a codebook of size 256 for each of the test set of images. Since the test images were color images represented in the RGB domain, we took the green band for the our experiments. Vectors were formed by raster scanning 4×4 blocks. Column 2 and 3 of table 2 show the SNR and PSNR values obtained by encoding each of the images with its local codebook.

We next used an MAW scan to form vectors which were then clustered by the same Generalized Lloyd's algorithm to form a codebook of the same size and dimension. Columns 4 and 5 of table 2 show the SNR and PSNR values obtained by encoding each of the images by its own local codebook. We see a significant increase in SNR and PSNR obtained when vectors formed by using an MAW scan are quantized. We would like to point out that the images have each been encoded by using local codebooks, that is a codebook generated from the image itself. This would not be done in practice but our intention in this section was only to demonstrate the validity of our approach.

The problem with using scan models for forming vectors prior to quantization is that an optimal scan model will vary from image to image. Hence an encoding of the scan has to accompany an encoding of the image. Unfortunately, due to the large number of possible scans, the cost of encoding a scan is usually more than 1.5 bits per pixel [14]. Hence our approach can not be used for single frame images in a straight forward manner. For multi-

spectral data sets however, the cost of encoding a scan can be avoided by making use of spectral correlations. In the next section we show how this can be done.

4 Scan Predictive VQ

Compression is generally achieved by removing inherent redundancies present in data. In the case of a multi-spectral data set, there are two sources of redundancy - *spatial redundancy* and *spectral redundancy*. By spatial redundancy we mean correlations among spatially adjacent pixels in the same spectral band. By spectral redundancy we mean correlations among pixels that have approximately the same spatial location but are in adjacent spectral bands. While spatial correlation is adequately exploited by standard Vector Quantization (VQ) techniques, variations of VQ that exploit spectral correlations have started emerging only recently.

Gupta and Gersho [8] have recently proposed a paradigm for vector quantization of multi-spectral data called *Feature Predictive Vector Quantization*. They point out that the conventional approach to vector quantization of multi-spectral data by forming a vector that spans spectrally adjacent blocks leads to high complexity with reasonable block sizes. For example, if we take a block size of 4×4 and form a vector X by concatenating blocks X_1 and X_2 from two spectrally adjacent bands leads to a vector $X = (X_1, X_2)$ of dimension 32. A bit rate of 0.5 bits per pixel then requires a codebook of size 65,536. In order to alleviate this problem they extract a reduced dimensionality feature vector U from X and transmit a quantized version of U from which an estimate $\hat{X}$ of X is formed by the receiver. In this section we present *Scan Predictive VQ*, a compression technique for multi-spectral data sets that is based on the notion of scan models. The scheme retains a manageable complexity in terms of vector dimension and at the same time effectively exploits spectral correlations.

Correlations between spectral bands in a multi-spectral data set are a result of the fact that the bands are imaging the same physical structures. Thus while pixel values in neighboring bands may be very different, the relationships between a pixel and its neighbors may

Image	Blocks	MAW Scan	Predictive Scan
USC-Girl	136	48	73
Couple	133	42	58
Girl-1	96	20	24
Girl-2	216	38	69
House	330	67	198
Tree	638	227	364

Table 3: Comparison of variance vectors from blocks, MAW scan and Predictive scan

be very similar in adjoining spectral bands. This relationship information is captured well in by a scan model. In fact, experiments have shown that an MAW scan of one band effectively models the image in a spectrally adjacent band [12]. The third column of table 3 shows the mean variance for vectors formed by using the MAW scan of the red band on the green band of the test images.[2] The first two columns are the same as table 1. It can be seen that although the variance is not as low as that obtained by using an optimal scan, using the optimal scan of the previous band to extract vectors does lead to highly correlated vectors as compared to using non-overlapping blocks. Similar results were obtained on the other bands.

Hence, given a multi-spectral image, the first image in the sequence can be compressed and transmitted by any conventional method and subsequent to that we can use the optimal model of the $((k-1)^{th})$ previous image on the (k^{th}) current image in the sequence in order to form vectors of the required dimension. These vectors can then be quantized by any of the vector quantization techniques described in the literature. This approach gives us a simple and efficient backward adaptive technique that exploits both spatial and spectral correlations. By backward adaptive we mean an adaptive technique in which both the transmitter and receiver are in possession of the information necessary for adaptation. This happens when the output of the transmitter (which is also available to the receiver), is used

[2] A color image in the RGB domain is essentially a multi-spectral image formed by three sensors responding to a narrow band of wavelengths centered around 700 nm (red), 546.1 nm (green) and 435.8 nm (blue) respectively.

for future adaptation. This has the advantage of obviating any necessity for transmission of additional or 'side' information. However, as the current information can only be used for future adaptations, there is necessarily a delay in the adaptation process. We call this technique *Scan Predictive Vector Quantization (SPVQ)*. Note that there is no cost incurred in encoding the scan, since it is being constructed from the previous image.

The codebook for Scan Predictive VQ is designed by using a GLA-like algorithm. The design can be done using either an *open-loop* or *closed-loop approach* [7]. In the open-loop approach, vectors from the current band, k are extracted by using the MAW scan of the original band $k-1$ image. These vectors are then clustered by the generalized Lloyd's algorithm to obtain a codebook of the desired size. In practice since the original image is not available to the receiver, reordering with an MAW scan is only possible with respect to the reconstructed image of the previous band. Hence the codebook is not optimal for the actual data being used. However, if the resulting reconstructed image is of sufficiently high quality, then it would be very close to the original and the codebook should give close to optimal quality.

The codebook can also be constructed by using a *closed-loop* approach. In such an approach the codebook vectors from the current band are obtained by using an MAW scan of the reconstructed image in the previous band. We can see that in the closed-loop design process, the training sequence of vectors changes with every iteration and hence convergence to a local minimum is not guaranteed. However, it has been observed in practice that the closed-loop technique gives substantial improvement over the open-loop technique [6]. Our experience has been consistent with this observation and hence in the rest of this paper we report results only for the closed-loop design technique.

In table 4, we give results obtained from two test images, California and Moffet, shown in figure 6[3] for different types of VQ. In each case, the codebook was generated by using the

[3]Thematic Mapper simulator data, acquired on NASA C-130 aircraft. Original data processed by P.-S. Yeh at NASA GSFC to approximate Landsat7 HRMSI resolution. The California image was taken over the San Luis Reservoir and has a ground resolution of 5.7m. The Moffet image has a ground resolution of 4.8m

Band	Block VQ		Scan Predictive VQ	
	SNR	PSNR	SNR	PSNR
1	22.80	27.75	-	-
2	24.24	30.58	25.66	31.90
3	22.90	29.41	23.44	29.95
4	23.94	29.99	24.37	30.42
5	24.50	27.92	25.21	28.63
6	24.43	30.49	25.78	31.84
7	21.57	29.73	23.47	31.63
8	22.08	25.16	25.49	28.57
Average	**23.38**	**29.02**	**24.77**	**30.42**

Table 4: Comparison of SNR and PSNR for California Image.

Moffet image and encoding results are presented for the California image. The Moffet image was 350×512 and the California image 232×512. Both images had 8 spectral bands.

First we used the Generalized Lloyd's algorithm to construct a codebook of size 1024 for vectors of size 16 that were formed by raster scanning 4×4 blocks of the Moffet image. The California image was then encoded by using full search VQ. Columns 2 and 3 give the SNR and PSNR values obtained. Experiments were also performed by forming a vector that spanned across two adjacent 2×4 blocks. This, however did not lead to any improvements in SNR and PSNR values and often resulted in poorer performance. This leads us to conclude that forming vectors by scanning two spectrally adjacent image blocks is not an effective technique for capturing spectral correlations.

Next we designed a codebook of size 1024 from 16 dimensional vectors obtained from a depth-first traversal along an MAW scan of the previous band for bands 2 to 8 of the Moffet image, by using the closed-loop technique presented above. Bands 2 to 8 of the California image were then encoded by using this codebook and quantizing vectors obtained by a depth-first traversal of the reconstructed image of the previous band. The first band of the California image was encoded by using the JPEG standard [17]. The particular implementation of JPEG used in this work was a public domain implementation provided

Image	Bpp for	Block VQ		SPVQ	
	Red	Green	Blue	Green	Blue
USC-Girl	0.79	28.4	28.0	30.2	31.8
Couple	0.78	27.1	27.4	31.8	31.2
Girl 1	0.46	31.9	31.9	35.1	34.6
Girl 2	0.73	29.5	30.8	31.6	32.2
Average	**0.69**	**29.2**	**29.5**	**32.2**	**32.5**

Table 5: Comparison of PSNR Values for test images.

by the *Independent JPEG Group*. This implementation provides an input parameter Q, that controls the quality and bit rate of the compressed image. A value of 50 was used for Q for a bit rate of 1.20 bits per pixel. This gives us an average rate of 0.69 bits per pixel for the entire image. Columns 4 and 5 show SNR and PSNR obtained. We see that an improvement of more than 1.5 db is obtained on an average. In figure 7 we show the reconstructed band 5 of the California obtained by using conventional VQ and also the one obtained by using the closed-loop technique. We see that the edge artifacts in the image obtained by SPVQ are considerably reduced. Also, note from table 4 that band 5 is where the smallest gain in SNR/PSNR is obtained by SPVQ as compared to the other bands.

We also repeated our experiments for the RGB images listed in table 1. Here, we constructed codebook of various sizes from four images using the closed-loop design technique. A different set of images was then Vector Quantized with this codebook. In table 5 we show the PSNR values obtained with a codebook size of 4096 and vector dimension 16 for a few images none of which were a part of the training set. Also, the first band (red band) for both the images was encoded using the JPEG standard with Q = 50. The resulting bit rate for the red band is shown in column 1 of table 5. The PSNR values for the green and blue band are shown in columns 3 to 6. We see an improvement of 3 db can be obtained by using an appropriate scan to form vectors. The reconstructed USC-Girl image obtained from SPVQ and block VQ are shown in figure 5. In the image obtained by block VQ we clearly see the staircase effect, especially near the shoulders. The image obtained by SPVQ on the other

hand has no such artifacts. The original image appears in figure 5.

At this point we would like to make a couple of points. First, note that the first band in the sequence has to be encoded by a conventional coding scheme. Hence, the quality of the first image can potentially effect the performance of scan predictive VQ on the second band and the quality of reconstructed image for the second band influences the quality of the third band etc. Our experiments seem to indicate that results obtained are robust with respect to the quality of the first image in the sequence. As long as the first image is of reasonable quality, the quality of subsequent images, on an average, remain unaffected. Here by reasonable quality, we mean for example, a Q factor of 50 or greater when using the JPEG standard.

Second, we would also like to note that better PSNR values at comparable bit rates have been reported in the literature with sophisticated enhancements to the basic VQ technique. However, all such enhancements can easily be incorporated into the scheme presented here. We have deliberately used simple codebook generation, organization and search techniques so that a proper estimate of the gains made by alternative techniques of forming vectors can be obtained.

5 Conclusions and Future Work

We have seen that scan models can be used to develop a simple and effective solution to the problem of edge degradation encountered during vector quantization of a sequence of correlated images, like multi-spectral images, 3-D medical images or a video sequence. An MAW scan by definition minimizes differences between successively scanned pixels. If we have a sequence of correlated images, then our experiments have shown that an MAW scan of one image in the sequence effectively models the next image in the sequence. Using an MAW scan of the previous image to extract vectors from the current image and quantizing these vectors by usual vector quantization techniques leads to significant improvements in performance over conventional block VQ techniques. Besides simple VQ, in future work we

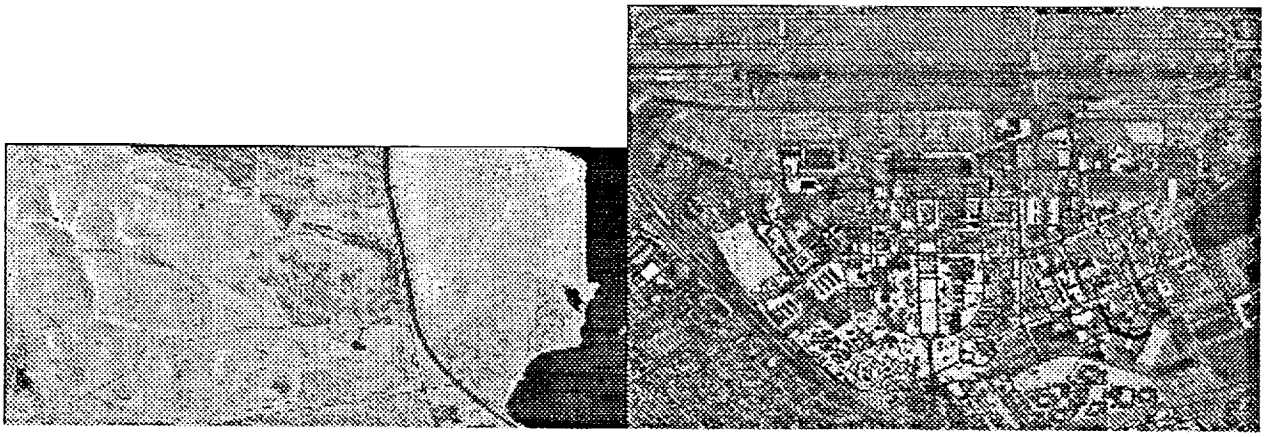


Figure 6: Original California (left) and Moffet (right) images, band 5.

Figure 7: California image (band 5) at 0.75 bpp after Block VQ (left) and SPVQ (right).

Figure 8: Girl-1 image (green band) at 0.75 bpp after Block VQ (left) and SPVQ (right).

N15

Accessing Portions of Losslessly Compressed Multiband Data

Marlys Remus and Khalid Sayood
Department of Electrical Engineering
University of Nebraska - Lincoln
fax: (402) 472-4732
email: marlys@eecomm.unl.edu

ABSTRACT

In this paper, we address the problem of accessing portions of multiband data which has been losslessly compressed. An approach that uses the fractal property of some well known space filling curves to provide access to portions of a losslessly compressed data set is described. This approach reduces the average amount of decompression necessary to access any portion of the data set, thereby reducing the amount of time required to access the compressed data. Various tradeoffs exist which will be discussed with practical examples.

INTRODUCTION

NASA's Mission to Planet Earth[1] will result in an enormous increase in the amount of data that will need to be archived. This has lead to an increased interest in not only more efficient lossless compression techniques, be also in faster methods of accessing losslessly compressed data. In particular, accessing portions of large multiband data sets.

There are several efficient lossless compression techniques currently available. For instance, predictive coding schemes use previous data points to generate a prediction for the current data value. The prediction error is then losslessly coded. Context based algorithms use the neighboring data values to determine the best encoding scheme. Dictionary schemes build a library of previously encountered patterns which can be used to encode patterns yet to be encountered. All of these techniques have one thing in common. They use information from previous data to encode future data values. Therefore, it is necessary to start decoding at the beginning of the data set. For example, if one needed to access a 128x128 section of band 7, of a 512x512, 7 band data set, it may be necessary to decode the entire data set.

In this paper, we present an algorithm which limits the amount of decoding necessary to access any portion of a compressed data set. Thereby, providing the user with convenient access to the data.

ENCODING ALGORITHM

The goal of this approach is to reduce the amount of past information needed to access any particular data value in the set. This will in turn reduce the amount of decoding necessary to retrieve any portion of the data set. This was carried out by partitioning the data set into smaller subsets and then losslessly encoding the subsets. Two different methods of scanning the data subsets were used.

Partitioning the Data Set

The data set is partitioned into three dimensional subsets. For instance, a 512x512, 7 band data set maybe partitioned into 128x128, 1 band subsets. Figure 1 illustrates this partitioning.

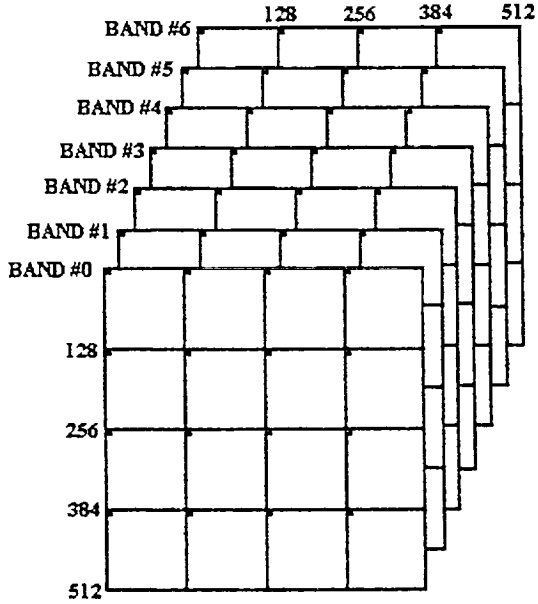


Figure 1. Example of partitioned data set.

The first data value in each subset is encoded using full resolution. The subsequent data values can then be encoded using any lossless compression technique. For this particular work we have used a simple predictive coding approach in which the difference between neighboring data values are Huffman coded.

The location of the first data value in the compressed file is kept in a code book. This allows the user to open the compressed file and advance the file pointer to the beginning of any particular subset and begin decoding at that point instead of starting at the beginning of the file.

The encoding algorithm was tested on a Landsat - TM 512x512, 7 band data set. The results for various partitions are given in Table 1. As shown, the compressed file size and the additional code book requirements increased as the size of the partitioned subsets decreased. This indicates a trade off between storage requirements and the ability to access small portions of data quickly.

Scanning the Subsets

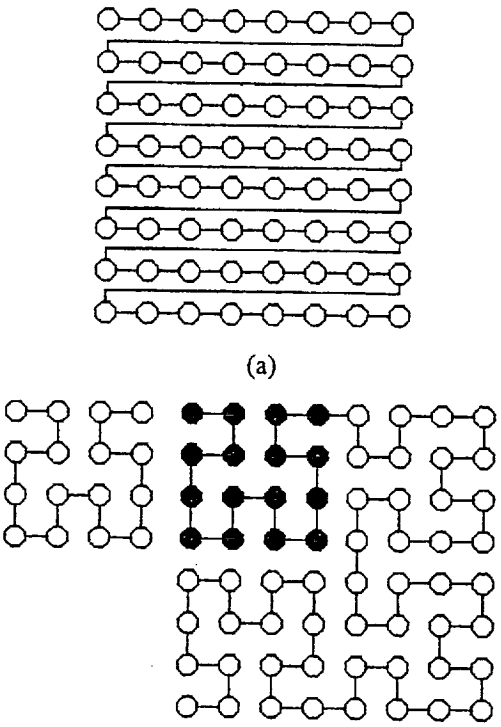
Two scanning patterns were used in testing the encoding algorithm. First, each band in each subset was scanned sequentially using a simple raster scan. Second, the Hilbert scanning pattern was used to scan each band in the subset sequentially.

[1] This work was supported in part by the NASA Goddard Space Flight Center under grany NAG5-1612.

Table 1. Test results for proposed encoder algorithm using various subset sizes.

Data Subset Description	Size of Data Subset (bytes)	Compressed File Size (bytes)	Additional Code Book Requirements (bytes)
512 rows, 512 columns, 7 bands	1835008	971895	0
12 rows, 512 columns, 3 bands	786432	971895	8
12 rows, 512 columns, 2 bands	524288	971895	12
512 rows, 512 columns, 1 band	262144	971895	24
56 rows, 256 columns, 3 bands	196608	974040	44
56 rows, 256 columns, 2 bands	131072	974040	60
256 rows, 256 columns, 1 band	65536	974040	108
28 rows, 128 columns, 3 bands	49152	978028	188
28 rows, 128 columns, 2 bands	32768	978028	252
128 rows, 128 columns, 1 band	16384	978028	444
4 rows, 64 columns, 3 bands	12288	985164	764
4 rows, 64 columns, 2 bands	8192	985164	1020
64 rows, 64 columns, 1 band	4096	985164	1788

he raster scan, shown in Figure 2a, had the advantage of being easily implemented on any size subset. On the other hand, The Hilbert scan, shown in Figure 2b, had the advantage of allo ng fast access to smaller portions of data at the cost of ncr sed complexity and constraints on the subset size.



(a)

(b)

gure 2. Raster scanning pattern (a) and the Hilbert scanning pattern (b) for an 8x8 block.

The Hilbert scan pattern for a 4x4 block is shown in Figure 2b. his pattern is then rotated by 90 and 180 degrees to form the 8x8 ock shown in Figure 2b. The 8x8 block is then rotated 270 and 3 degrees to form a 16x16 block. This process is repeated until l the data points in each band have been scanned. Thereby, ubdividing the data points into even smaller sub blocks. Notice that each rotation of the scanning pattern doubled the sub block size. Therefore, it is necessary to limit the subset size to a power of two.

DECODING ALGORITHM

The decoding algorithm was set up to allow the user to retrieve any desired data subset by specifying the coordinates of the first data value, number of rows, columns and bands in the desired subset. The algorithm then searched the code book for the start of the encoded subsets which contained the requested data. The algorithm did not place any restrictions on the size of the desired data subset.

The decoding algorithm was tested on a Landsat - TM, 512x512, 7 band data set after it had been compressed into subsets of 128x128, 1 band using both the raster and Hilbert scanning patterns. Requests for data were assumed to be uniformly distributed over the entire data set.

A set of uniformly distributed request were generated using a random number generator to select the coordinates of the first data value. For simplicity, the size of the requested set was held constant at 256x256, 1 band. The number of data points decoded to retrieve the requested sets were calculated . The results of this test are given in Table 2. As shown, both scanning patterns produced approximately the same results. Therefore, if the assumption that the requests for data would be uniformly distributed over the entire data set was correct, the choice of scanning pattern would make very little difference.

Table 2. Results of uniformly distributed requests

Scanning Method	Average Number of Data Points Decoded
Hilbert scan	123100
Raster scan	122522

However, if the request for data are assumed to be centered around some point of interest in the data set, the assumption of a uniform distribution is incorrect and the choice of scanning patterns may be important.

A more accurate model of the requests may be a normal distribution around the point of interest. A 'smart' encoding algorithm was developed to test this theory.

'SMART' ENCODING ALGORITHM

The 'smart' encoding algorithm starts by encoding the data set using the method just described with one added feature. Each partitioned subset is subdivided into sub blocks and as requests for the data are processed, a frequency count of the number of times each sub block is accessed is kept.

After a sufficient number of requests have been processed, the data set is re-encoded and each subset is scanned with the pattern which provides the most efficient access based on past requests. A number which identifies the scanning pattern is encoded in the compressed file at the beginning of each subset followed by the first data value in the subset. Four different sets of 'smart' scanning patterns were developed.

In order to simulate past requests, a set of 1000 blocks whose first pixel location was normally distributed around the point 128,128 with a variance of 128 was generated. For simplicity, the requested blocksize was held constant at 256x256, 1 band.

Once the data set had been re-encoded, each of the four 'smart' scanning patterns were tested by calculating the number of data points that needed to be decoded to retrieve a test set of normally distributed requests. The test requests were normally distributed around the point 128x128 with a variance of 256. The results of the four 'smart' scanning patterns are given in Table 3.

Notice that there is an increase in storage requirements of approximately 16 bytes per subset. This is due to the need to store the frequency counts.

Pattern #1

The first set of 'smart' scanning patterns, shown in Figure 3, uses the Hilbert scan with the processing order of the first sub blocks determined by the frequency count. There were a possibility of eight different patterns, therefore, three bits where used at the beginning of each subset to identify which pattern was being used.

The results shown in Table 3, indicate that approximately 20 thousand fewer data points were decoded using this method when compared to the original Hilbert or raster scan.

Pattern #2

The second set of 'smart' scanning patterns, shown in Figure 4, uses the raster scan with the processing order of the rows and columns determined by the frequency count. There were a possibility of only four different patterns, therefore, only two bits were used at the beginning of each subset to identify which pattern was used.

The results shown in Table 3, indicate that these scanning patterns provided a savings of approximately 12 thousand data points over the scanning patterns used in the first set.

Pattern #3

Frequencies for the corner second sub blocks where calculated for the third set of 'smart' scanning patterns. The scanning patterns used the Hilbert scan with the processing order of the second sub blocks determined by the frequency count. There were 16 possible scanning patterns, the eight used in the first set and the eight additional patterns shown in Figure 5. Therefore, it was necessary to use four bits at the beginning of each subset to identity which pattern was used to encode the subset.

The results shown in Table 3 indicate only a slight improvement over the patterns used in the previous method.

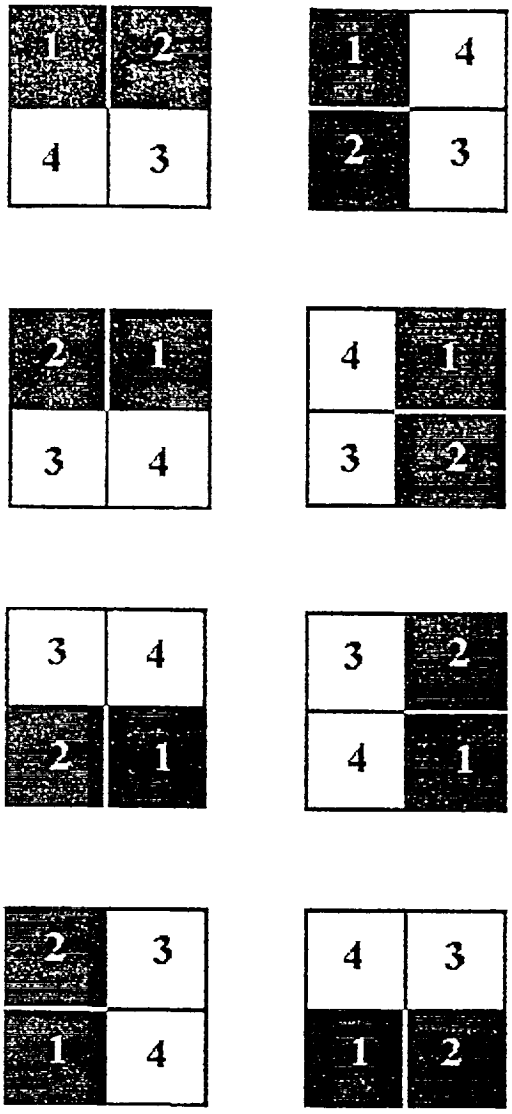


Figure 3. Hilbert scan with procession order of first sub blocks determined by frequency count.

Pattern #4

A close examination of the previous three methods revealed that certain types of requested where handled more efficiently by the Hilbert scan and others by the raster scan, as shown in Figure 7. Therefore, the fourth set of 'smart' scanning patterns used a combination of the two patterns. The scanning patterns used consisted of the four patterns shown in Figure 4 and four new patterns which started at each corner and scanned the subset as shown in Figure 6. There was a total of eight different patterns, therefore, only three bits were used at the beginning of each subset to identify which pattern had been used to encode the subset.

The results shown in Table 3 indicate that this set of scanning patterns produced a significant decrease in the number of data points decoded to retrieve the requested data sets.

CONCLUSION

An approach to decompressing portions of losslessly compressed multi band data was presented. The proposed approach first partitioned the data set into three dimensional subsets and encoded the first data value with full resolution. The location of the first data value in the compressed file was then added to the code book. Therefore, any subset could be accessed by opening the compressed file and advancing the file pointer to the location of the first data value and begin decoding there opposed to starting at the beginning of the compressed file.

The algorithm was first tested by assuming that requests for the data would be uniformly distributed over the entire data set. Both the Hilbert and raster scanning patterns were used to encode the data set. Both scanning patterns produced approximately the same results.

The request were then assumed to be normally distributed around a particular point of interest in the data set and a 'smart' algorithm was developed to select the scanning pattern which provided the most efficient access to the data, based on previous requests. This 'smart' algorithm used properties of both the Hilbert and raster scanning patterns and provided significantly better results.

Table 3. Results of 'smart' scanning patterns

Scanning Method	Average Number of Data Points Decoded	Additional Storage Requirements per Partitioned Subset
Original Hilbert scan.	122345	4 Bytes
Original raster scan.	123020	4Bytes
Hilbert scan with processing order of first sub blocks determined by frequency count.	101012	20 Bytes, 3 Bits
Raster scan with procession order of rows and columns determined by frequency count.	89400	20 Bytes, 2 Bits
Hilbert scan with processing order of second sub blocks determined by frequency count.	88069	20 Bytes, 4 Bits
Combination of Hilbert and raster scans with processing order of sub blocks, rows and columns determined by frequency count.	70291	20 Bytes, 3 Bits

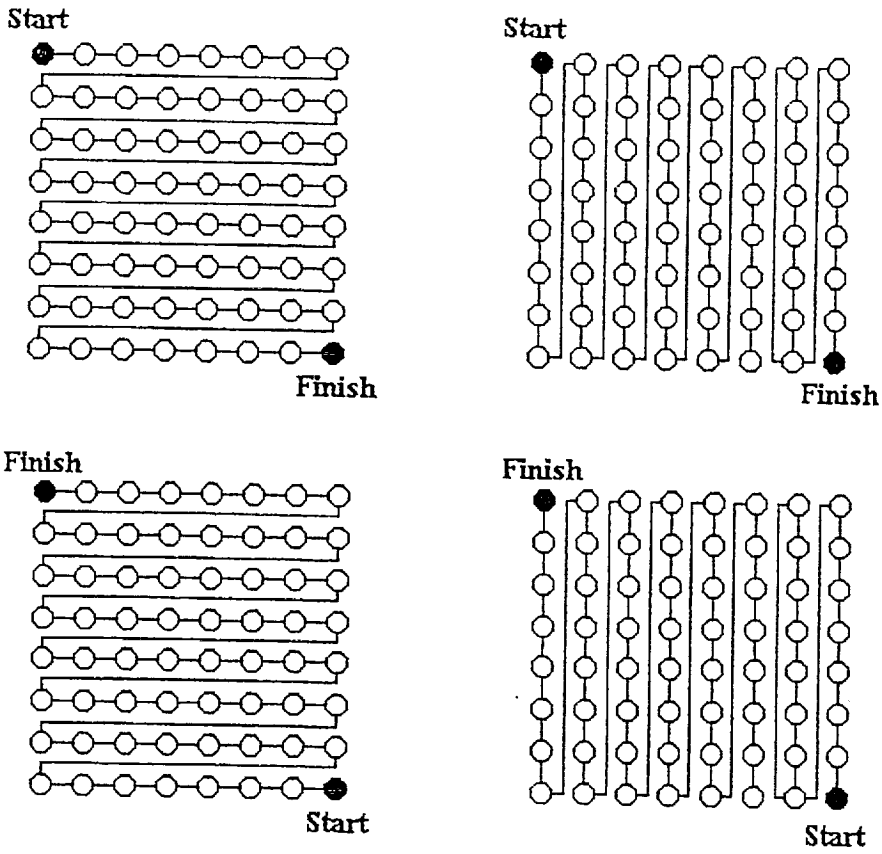
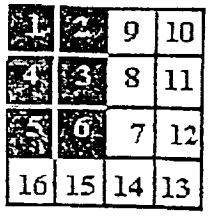
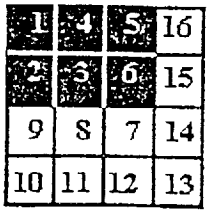
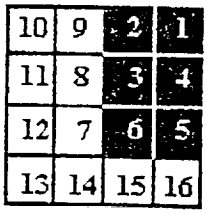
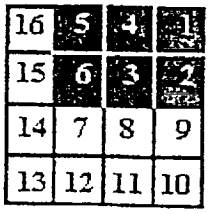


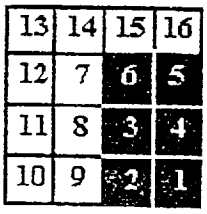
Figure 4. Raster scan with processing order of rows and columns determined by frequency count.

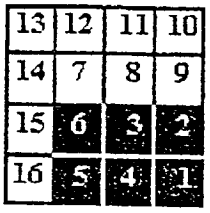


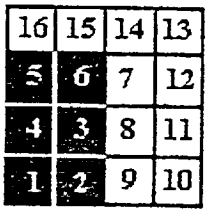












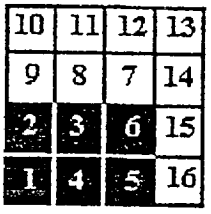
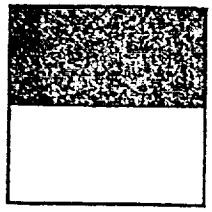
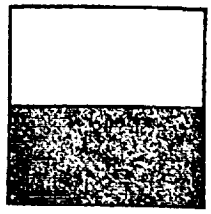
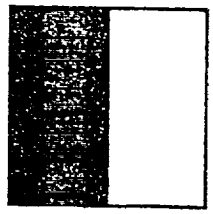
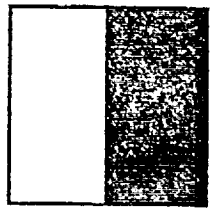


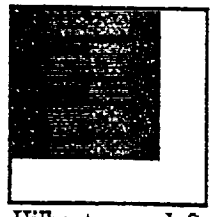
Figure5. Hilbert scan with the processing order of the second sub blocks determined by the frequency count.

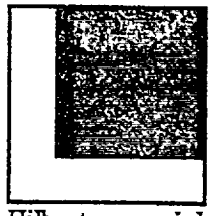


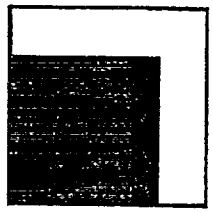












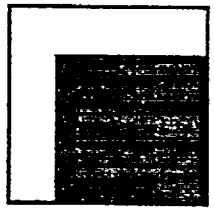


Figure 7. Type of request in subsets and the scanning pattern which provides the best results.

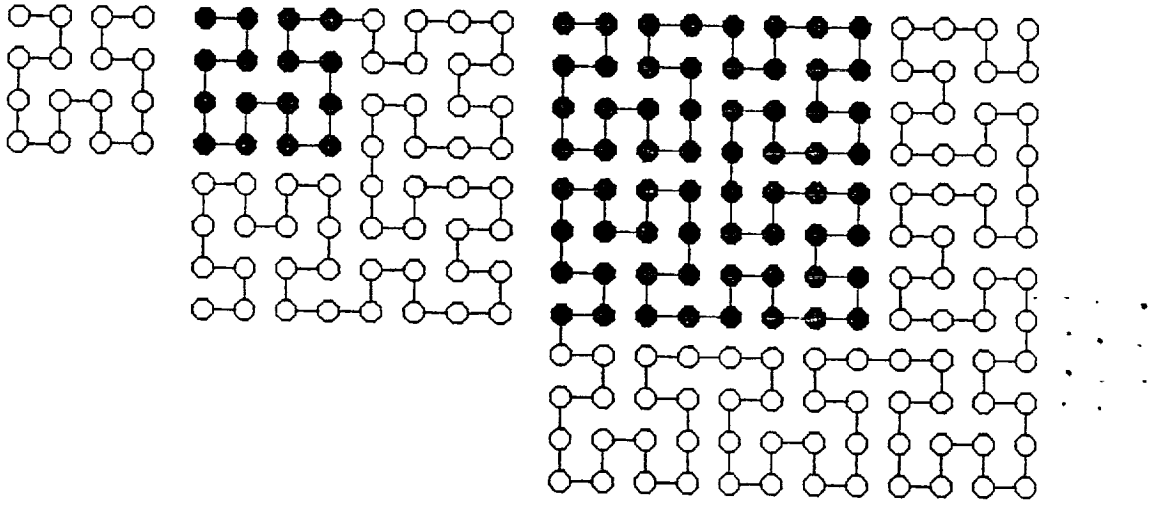


Figure 6. Example of Hilbert scan with 4x4 building blocks.

Compression of Color-Mapped Images

Andrew C. Hadenfeldt, *Member, IEEE*, and Khalid Sayood, *Member, IEEE*

Abstract—Multispectral data is often displayed and stored as a color-mapped or pseudo-color image. Pseudo-color is also used to enhance features in a single-band image. The use of pseudo-color tends to rearrange the structure in the image in such a way as to prevent efficient compression. This structure can be restored by sorting the color maps. Restoration of the structure increases the efficiency of lossless compression and permits the use of lossy compression algorithms. The latter benefit is especially useful for many progressive transmission algorithms.

I. Introduction

MULTISPECTRAL remotely sensed data are often displayed as composite images on color monitors. These composite images are generated by treating three spectral bands of the multispectral dataset as the red, green, and blue planes of an RGB image. If the pixels in each plane are represented by 8 b, each pixel in the composite image is represented by 24 b, allowing a total of 2^{24} colors to be displayed. More expensive systems may use more than 24 b/pixel. A disadvantage of the full-color display is the large amount of memory required to represent an image. This memory must be quickly accessible to allow real-time updating of the CRT, making full-color image displays costly. Also, the images involved require large amounts of storage space, whether in display memory or on a mass-storage device. A less expensive solution is needed.

Many commercial image processing and geographic information systems (GIS) use a pseudo-color or color-mapped frame buffer. The values stored in memory are used as indexes into a 24-b table, the color map. Each entry in the color map consists of 8-b values for the red, green, and blue portions of the pixel. The color-mapped system allows the display of a small number of colors at a time, 2^8 for the system shown in the figure, which can be selected from a larger set of colors (2^{24} for this example). By careful selection of the colors in the color map, a large variety of images can be displayed, often with quality approaching that of a full-color display system. The color map is obtained through a quantization process, the goal of which is to select the most "representative" 256 colors from the available colors. The color-map generation algorithms do not attempt to put the color-map entries in any particular order. Unfortunately, this lack of order makes compression more difficult.

Perhaps the most useful trait of image data used in image compression is the pixel-to-pixel correlation. For an achromatic image, this means that the integer pixel values (which describe the intensities of the pixels) will be numerically similar for spatially adjacent pixels. For a full-color image, a similar condition exists for adjacent pixels on individual color planes. In a color-mapped image, the values stored in the pixel array are no longer directly related to the pixel intensity (or the magnitude of one of the color components). Two color indexes that are numerically adjacent (close) may point to two very different colors. Hence, the correlation between pixels appears to be lost. This makes compression of these images very difficult. With the arrival of instruments that generate images with higher spatial resolution, and significantly more spectral bands, compression of these images has become an important problem.

For most compression algorithms to work there has to be some correlation structure in the data. The structure in the color-mapped image still exists, but only via the color map. Therefore for compression, this structure has to be reintroduced into the image. We show that the reintroduction of structure can be accomplished by sorting the color map. In this paper we study the sorting of color maps and show how the resulting structure can be used in both the lossless and lossy compression of images.

The sorting procedure is described in the following section and the lossless and lossy compression results are presented in Sections III and IV, respectively.

II. Color-Map Sorting

Sorting the color map can be done to satisfy one of two possible goals. The first is the desire to restore the correlation among the pixels to allow them to be efficiently coded, i.e., a reduction in the differential entropy. The second goal is to allow the introduction of small errors in the color index values, such as those resulting from quantization, without a large reduction in the subjective image quality. Even in this latter case, the desire for entropy reduction is implied since that is the purpose of quantization. These two goals conflict somewhat since the sensitivity of the eye to color errors is dependent on many things. It will be useful to find a solution that satisfies both of these goals to some degree.

Color-map sorting is a combinatorial optimization

Manuscript received January 3, 1994. This work was supported in part by the NASA Goddard Space Flight Center under Grant NAG 5-1612 and by the NASA Lewis Research Center under Grant NAG 3-806.

A. C. Hadenfeldt is with the University of Nebraska Medical Center, Omaha, NE 68198.

K. Sayood is with the Department of Electrical Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588.

IEEE Log Number 9402214.

problem. Treating the K color-map entries as vectors, the problem is defined as follows. Given a set of vectors $\{a_1, a_2, \cdots, a_K\}$ in a three-dimensional vector space and a distance measure $d(i, j)$ defined between any two vectors a_i and a_j, find an ordering function $L(k)$ that minimizes the total distance D:

$$D = \sum_{k=1}^{K-1} d[L(k), L(k+1)]. \quad (1)$$

The ordering function L is constrained to be a permutation of the sequence of integers $\{1, \cdots, K\}$. Another possibility results when the list of color-map entries is considered as a ring structure. That is, the color-map entry specified by $L(K)$ is now considered to be adjacent to the entry specified by $L(1)$. In this case, an additional term of $d[L(K), L(1)]$ is added to the distance formula D.

The sorting problem is similar to the well-known traveling salesman problem, and is identical if the color map is considered as a ring structure. As such, the problem is known to be *NP*-complete [1], and the number of possible orderings to consider is $1/2[(K-1)!]$ [6]. Algorithms exist that can solve the problem exactly [2], [6]; however, these algorithms are computationally feasible only for K not greater than about 20. Efficient algorithms for locating a local minimum exist [6] for $K \leq 145$. For large color maps such as $K = 256$, another approach is necessary. Two techniques were tested. The first is a "greedy" technique, discussed in Section II-A. The second involves an algorithm that has performed well in practice, a technique known as *simulated annealing*. Simulated annealing was chosen as the sorting method for the color maps in this paper and is described in more detail in Section II-B.

To complete the problem definition above, the distance metric d must be determined. There are several possibilities depending on the color space used. For the present paper, the distance metric was chosen to be an (unweighted) Euclidean distance, and different color spaces were investigated. Three color spaces were selected: the NTSC RGB space, the CIE $L^*a^*b^*$ space, and the CIE $L^*u^*v^*$ space. The NTSC RGB space was chosen since it corresponds to the primary colors of the original images. Color spaces that can be linearly transformed to the NTSC RGB space were not considered, since the use of the unweighted Euclidean distance measure would give similar results for such a color space. The two CIE color spaces were selected since they provide a means to measure perceptual color differences.

A. Greedy Sorting Algorithm

The first color-map sorting algorithm investigated was a greedy algorithm. As the name implies, this is simply a "take what you can get" approach. The algorithm begins by selecting a starting node (i.e., a color vector). From this node, proceed to a node that has not yet been visited by selecting the path with the least cost. For the color-map sorting problem, the "cost" is the distance between the colors, the function $d(i, j)$ defined in (1). The algorithm proceeds until all nodes (colors) have been visited. To avoid any penalties due to the choice of the starting node, a path was formed starting at each of the 256 nodes. From the 256 paths, the path of least cost was then selected.

Tests using the greedy sorting algorithm indicated that it was not very successful in sorting the color maps. The resultant images were still quite sensitive to small errors in the color indexes, and had high differential entropies. The simulated annealing algorithm in the next section was able to provide better results in both categories.

B. Sorting Using Simulated Annealing

Simulated annealing [1], [7] is a stochastic technique for combinatorial minimization. The basis for the technique comes from thermodynamics and observations concerning the properties of materials as they are cooled. The technique described in this section is based on the implementation in [7].

In the traveling salesman problem, the goal is to visit each city only once and return to the original city with a minimum path cost. Similarly, solving the color-map sorting problem involves selecting each color only once while minimizing the sum of the distances between the colors. To find a solution using simulated annealing, an initial path through the nodes (cities, colors) is chosen, and its cost computed. The algorithm then proceeds as follows:

1) Select an initial temperature T and a cooling factor α.

2) Choose a temporary new path by perturbing the current path (see below), and compute the change in path cost $\Delta E = E_{new} - E_{old}$. If $\Delta E \leq 0$, accept the new path.

3) If $\Delta E > 0$, randomly decide whether or not to accept the path. Generate a random number r from a uniform distribution in the range [0, 1), and accept the new path if $r < \exp(-\Delta E/T)$.

4) Continue to perturb the path at the current temperature for I iterations. Then, "cool" the system by the cooling factor $T_{new} = \alpha T_{old}$. Continue iterating using the new temperature.

5) Terminate the algorithm when no path changes are accepted at a particular temperature.

The decision-making process is known as the *Metropolis algorithm*. Note that the decision process will allow some changes to the path that increase its cost. This makes it possible for the simulated annealing method to avoid easily being trapped in a local minimum of the cost function. Hence, the algorithm is less sensitive to the initial path choice. Aarts and Korst [1] show that if certain conditions are satisfied, the simulated annealing technique can asymptotically converge to a global optimum. Even in cases where it does not converge to the optimum, the method often provides high-quality solutions.

In the above description, initial values for T, α, and I need to be selected. Selection of these values requires some experimentation, although a few guidelines are pro-

vided in the references. In this work, initial values of T ranged from 80 to 500, depending on the color space used. The cooling factor α was usually chosen as 0.9. The simulated annealing algorithm seemed to be most sensitive to the choice of this value, as values outside the range (0.85, 0.95) caused the cooling to occur too slowly or too quickly. The number of iterations per temperature I was chosen as 100 times the number of nodes (colors), or 25 600. However, to improve the execution speed of the algorithm an improvement suggested Press [7] was added, which causes the algorithm to proceed to the next temperature if (10) (number of nodes) = 2560 successful path changes are made at a given temperature.

Also, a method for perturbing the path must be selected. In this work, the perturbations were made using the suggestions of Lin [6], [7]. At each iteration, one of two possible changes to the path are made, chosen at random. The first is a *path transport*, which removes a segment of the current path and reinserts it at another point in the path. If we think of the color map as an array, this corresponds to moving a segment of the array to a different location in the array. The "hole" left in the array is filled up by sliding the components of the array down or up (depending on the new insert location). The location of the segment, its length, and the new insertion point are chosen at random. The second perturbation method, called *path reversal*, removes a segment of the current path and reinserts it at the same point in the path, but with the nodes in reverse order. The location and length of the segment are again, randomly chosen.

The algorithm outlined in the previous paragraphs formulates color-map sorting as a traveling salesman problem. This type of problem usually assumes a complete tour will be made (i.e., the salesman desires to return to the original city). Hence, the color map is assumed to have a ring-like structure. However, the simulated annealing technique can also be used if this is not the case, allowing the color map to be considered as a linear list structure. Experiments using both structures were conducted.

III. Color-Map Sorting and Lossless Compression

A measure that is used to describe the statistical properties of an image is its entropy. Entropy provides a measure of the randomness of a source, based on an assumed model for that source. It also provides an estimate of the number of bits per sample required to code the source. Treating the image as a memoryless source with an alphabet S containing R symbols, the zeroth-order entropy H_0 is defined as

$$H_0 = -\sum_{i=1}^{R} P(S_i) \log_2 P(S_i) \quad \text{bits} \qquad (2)$$

where $P(S_i)$ is the probability of occurrence of symbol S_i. If there is a correlation between adjacent pixels, another possibility is to consider a first-order model for the image. If the image is transmitted as a one-dimensional source in a row-by-row (or column-by-column) manner, a first-order differential entropy H_1 can be defined on an alphabet D consisting of the $2R - 1$ possible differences between the elements of alphabet S:

$$H_1 = -\sum_{j=1}^{2R-1} P(D_j) \log_2 P(D_j) \quad \text{bits.} \qquad (3)$$

TABLE I
Entropies of the Source Images

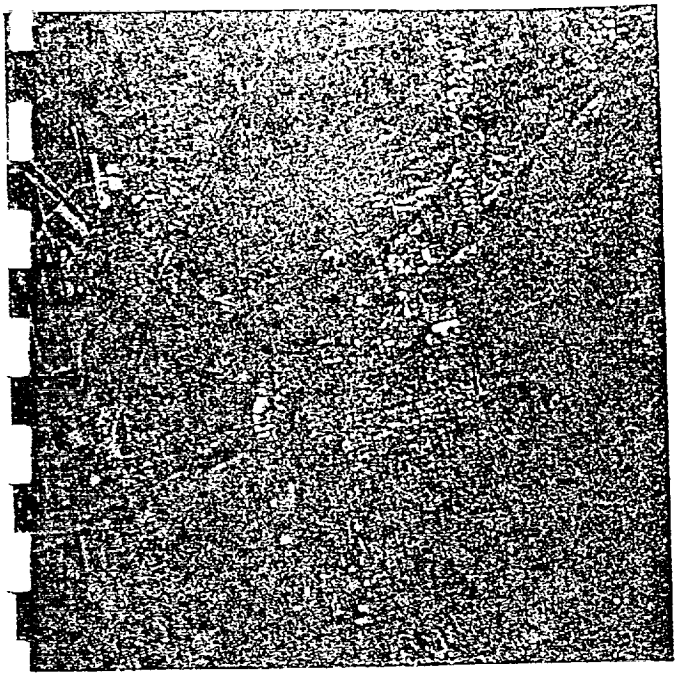
Image	H_0	H_1
Lena	7.617	7.413
Park	7.470	7.797
Omaha	7.242	7.165
Lincoln	5.916	6.674

These quantities were computed using the index arrays for four test images and are listed in Table I. The Lincoln and Omaha images were constructed from channels 2, 3, and 4 of a thematic mapper simulator (TMS) image and are shown in Fig. 1. The Lena and Park images are well-known standard images.

The color maps for these images were sorted using simulated annealing in the RGB and $L^*u^*v^*$ space. The effect of the sorting on the color map is shown in Fig. 2. Fig. 2(a) displays the colors of the 256 indexes of the color map for the Omaha image before sorting, while the colors of the 256 indexes after sorting are shown in Fig. 2(b). The effect of the sorting has been to make "neighboring" index values correspond to colors that are also close in a perceptual sense.

The numerical effect of sorting the color maps of the test images using simulated annealing are shown in Table II and Table III. Results for sorting the color map as a circular ring structure are shown in Table II, while the results of sorting the color map as a linear structure are shown in Table III. Values are given in the tables for the resulting first-order entropy and the final path cost (the distance measure D).

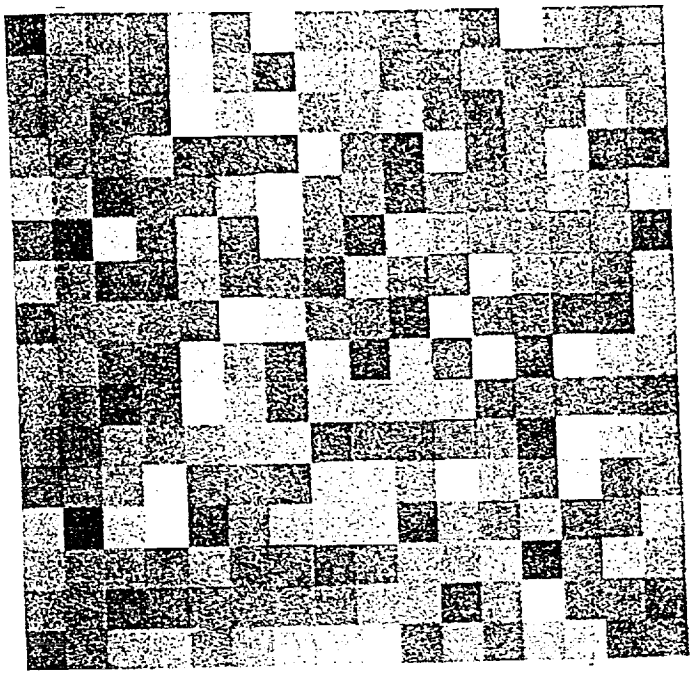
Note that the zeroth-order entropy H_0 is not changed by the sorting process, since permuting the color-map entries does not change the frequency of the occurrence of a particular color. The lower first-order entropies of the resultant images indicate that some of the spatial correlation between color indexes has been restored in each case. The sorting results for the NTSC RGB space show that sorting in this space yields good results, if entropy reduction (the first goal stated above) is the goal. However, the $L^*u^*v^*$ space sorting gives better results, with the added advantage that the perceptual differences between color-map entries have been considered. Hence, the resultant images from this sort should also be able to accept quantization errors while maintaining good subjective quality, the second goal stated previously. We examine this further in the next section. Comparing the results shown in Tables I–III, we can see that the sorting of the color map has resulted in a drop in entropy of 2 b/pixel for the Lena image



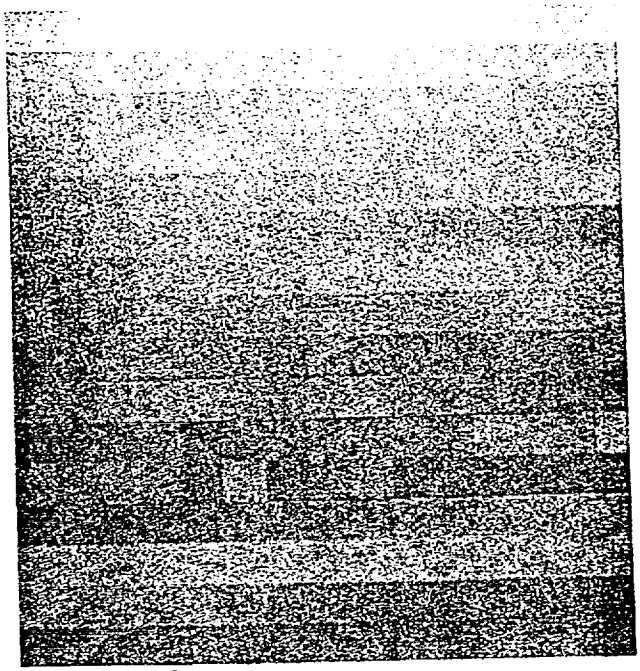
(a)

(b)

Fig. 1. (a) Lincoln image. (b) Omaha image.



(a)



(b)

Fig. 2. (a) Color map for Omaha image before sorting. (b) Color map for Omaha image after sorting.

TABLE II
RESULTANT IMAGES WITH CIRCULARLY SORTED COLOR MAPS

Image	RGB Space		L*a*b Space		L*u*v* Space		LZW (GIF)
Name	Cost	H_1	Cost	H_1	Cost	H_1	
Lena	13.53	5.641	857.80	5.627	208.49	5.490	6.43
Park	19.82	6.325	1609.46	6.350	310.41	6.218	6.88
Omaha	13.74	6.269	1063.82	6.303	363.21	6.178	7.03
Lincoln	13.42	5.513	1393.88	5.531	224.06	5.478	5.74

TABLE III
RESULTANT IMAGES WITH LINEARLY SORTED COLOR MAPS

Image	RGB Space		L*a*b Space		L*u*v* Space	
Name	Cost	H_1	Cost	H_1	Cost	H_1
Lena	11.68	5.575	847.29	5.933	200.31	5.512
Park	15.66	6.260	1509.25	6.775	292.29	6.546
Omaha	10.61	6.532	1004.99	6.554	283.66	6.199
Lincoln	10.61	5.774	1177.80	6.120	204.64	5.735

and 1–1.5 b/pixel for the other images. Entropy coding techniques such as Huffman coding and arithmetic coding permit the lossless encoding of data close to entropy. Therefore, we can treat the entropy figures as estimates of the coding rates. For 512 × 512 images a savings of 1–2 b/pixel translates to a savings of between 32 768 to 65 536 bytes/image. For a large database of images this could be a considerable saving. As many remote sensing applications require large repositories of images, using sorted color maps can lead to a significant reduction in storage requirements.

For comparison, the images were also compressed using the Lempel–Ziv algorithm used in the GIF format [3]. The numbers shown in Table II were obtained after removing the overhead included in all GIF files. The performance of the Lempel–Ziv scheme is between 0.25 and 1 b/pixel worse than the differential entropy of the sorted images.

IV. Color-Map Sorting and Lossy Compression

NASA's earth observing system (EOS) will result in even larger archives of remotely sensed images. In order for remote users to easily access these images, a "browse" facility that allows the user to quickly access low-resolution versions of the images is a necessity. Currently, in the global land information system (GLIS) the browse facility is implemented by only storing previously subsampled low-resolution versions of the images on-line. Browse features that allow on-line delivery of full resolution images can be implemented through the use of progressive transmission. In most progressive transmission schemes, images compressed using lossy compression techniques are first sent to the remote user. If the image is what the user is looking for, he or she can request that the image be refined by sending more information. For standard pseudo-color images, lossy compression (with even little loss) would result in the destruction of most of the features of the image. This is evident from the image in Fig. 5(a), where the three least-significant bits have been dropped from the image using unsorted color maps.

The sorting of the color map restores some perceptual structure to the color-map indexes in the sense that indexes close in numerical value are also close in some perceptual sense. Therefore, it should be possible to introduce errors into the indexes without destroying the image. To verify this hypothesis, we dropped the three least-significant bits of the $L^*u^*v^*$-sorted Omaha image. Good subjective results were obtained using quantization levels down to as low as 5 b/pixel from the 8-b original. Fig. 3 shows the result of quantizing the Omaha image to 5 b/pixel, before and after the color map has been sorted. Notice that the image in Fig. 3(a), which used the unsorted color maps, displays severe distortion obscuring most of the image, while the image in Fig. 3(b), which used the sorted color map, suffered only minimial degradation. Several caveats are in order here. While the distance between the 8-b indexes have more perceptual meaning after sorting, the sorted color-map image should not be assumed to have the same properties as an 8-b monochrome image. In some cases, if the distance between the original and reconstructed (compressed and decompressed) indexes is large enough, there might be a drastic change in color between those pixels in the original and reconstructed image, which would be immediately apparent. In the monochrome case large distances would correspond to changes in shading, which might be overlooked by the viewer. Also, if the original (noncolor-

(a)

(b)

Fig. 3. (a) Omaha image with unsorted color map quantized to 5 b. (b) Omaha image with sorted color map quantized to 5 b.

mapped) images are available, better compression performance would be obtained by compressing the original image than by compressing the (sorted) color-mapped image.

To see how well the sorted color-mapped images lend themselves to lossy compression we compress them using particular implementations of two popular lossy compression techniques, the discrete cosine transform (DCT) and differential pulse code modulation (DPCM).

A. DCT Coding of Color-Mapped Images

In the DCT approach the image is divided into $N \times N$ blocks (N is typically 8). The blocks are then transformed using the DCT basis set. In the transform domain most of the energy is compacted into a few coefficients. The coding resources (bits) are devoted to the coefficients with higher energy so a high-energy coefficient will be quantized with more bits, while a low-energy coefficient will be quantized with few or zero bits (i.e., discarded). At the receiver the quantized coefficients are transformed back to the spatial domain. The allocation of bits to the individual coefficients can be based on the average statistics of the image (or class of images) or on the characteristics of each individual block [5]. The latter approach is used in the recently approved JPEG standard for image compression [10].

In Fig. 4 we coded the Omaha image with the unsorted color map at 2 b/pixel using the JPEG algorithm.[1] The JPEG coding was applied to the index values leaving the color map intact. As can be seen from the Omaha image shown in the figure, the river is about the only thing still visible. It should be noted that for 8-b monochrome images, DCT coding at 2 b/pixel generally provides a reconstruction that is indistinguishable from the original.

In Fig. 5 we show the same image, this time with the sorted color map, coded at 2 and 1 b/pixel using the JPEG algorithm.

B. DPCM Coding of Color-Mapped Images

The DPCM system consists of two main blocks, the quantizer and the predictor (see Fig. 9). The predictor uses the correlation between samples of the waveform $s(k)$ to predict the next sample value. This predicted value is removed from the waveform at the transmitter and reintroduced at the receiver. The prediction error is quantized to one of a finite number of values that is coded and transmitted to the receiver and is denoted by $e_q(k)$. The difference between the prediction error and the quantized prediction error is called the quantization error, or the quantization noise. If the channel is error free, the reconstruction error at the receiver is simply the quantization error. To see this, note (Fig. 6) that the prediction error $e(k)$ is given by

$$e(k) = s(k) - p(k) \tag{4}$$

[1]The JPEG coded images were coded using software from the independent JPEG group.

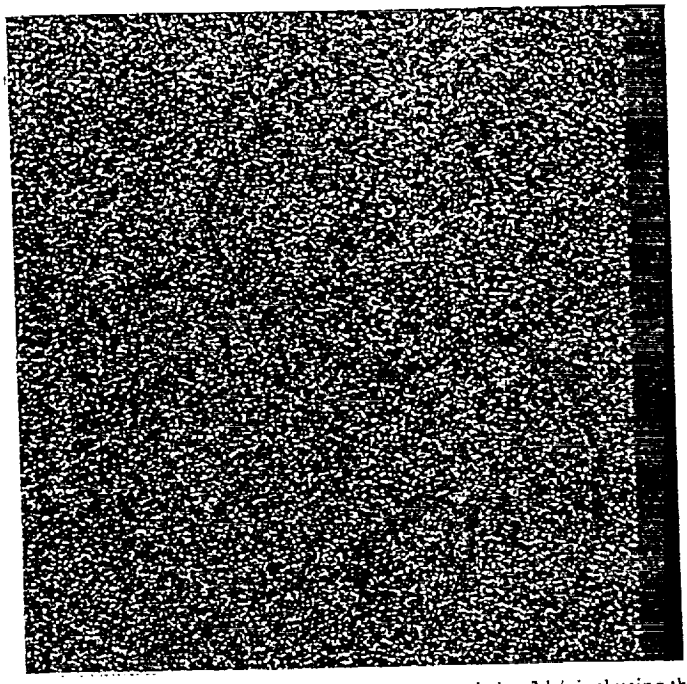


Fig. 4. Omaha image with unsorted color map coded at 2 b/pixel using the JPEG algorithm.

where $s(k)$ is original signal predicted by $p(k)$, which is given by

$$p(k) = \sum a_q \hat{s}(k - j) \tag{5}$$

$$\hat{s}(k) = e_q(k) + p(k). \tag{6}$$

Assuming an additive noise model, the quantized prediction error $e_q(k)$ can be represented as

$$e_q(k) = e(k) + n_q(k) \tag{7}$$

where $n_q(k)$ denotes the quantization noise. The quantized prediction error is coded and transmitted to the receiver. If the channel is noisy, this is received as $\tilde{e}_q(k)$, which is given by

$$\tilde{e}_q(k) = e_q(k) + n_c(k) \tag{8}$$

where $n_c(k)$ represents the channel noise. The output of the receiver $\tilde{s}(k)$ is thus given by

$$\tilde{s}(k) = \tilde{p}(k) + \tilde{e}_q(k) \tag{9}$$

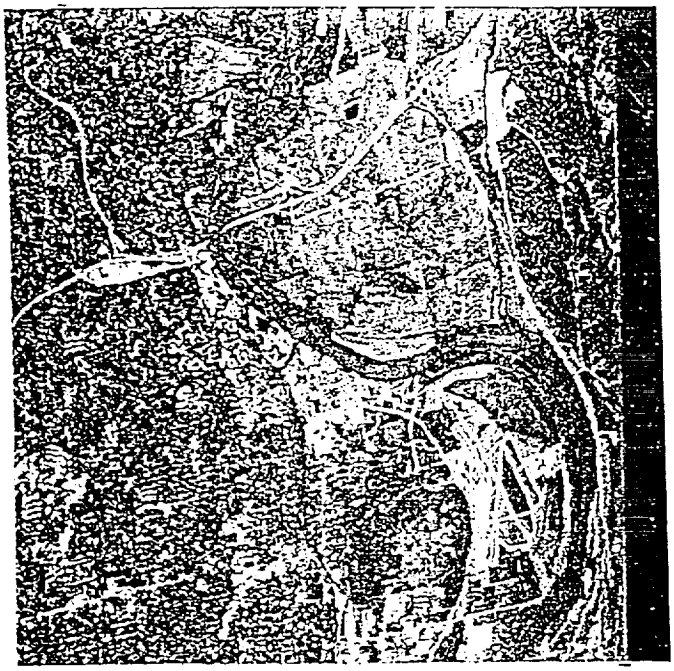
$$\tilde{p}(k) = p(k) + n_p(k) \tag{10}$$

the additional term $n_p(k)$ being the result of the introduction of channel noise into the prediction process. Using (5), (8), (9), and (11) in (10) we obtain

$$\tilde{s}(k) = s(k) + n_q(k) + n_c(k) + n_p(k). \tag{11}$$

If the channel is error free, the last two terms in (8) drop out and the difference between the original and the reconstructed signal is simply the quantization error.

When the prediction error is small, it falls into one of the inner levels of the quantizer, and the quantization noise is of a type referred to as granular noise. If the prediction error falls in one of the outer levels of the quantizer, the



(a)

(b)

Fig. 5. (a) Omaha image with sorted color map coded at 1 b/pixel using the JPEG algorithm. (b) Omaha image with sorted color map coded at 2 b pixel using the JPEG algorithm.

incurred quantization error is called overload noise. Granular noise is generally smaller in magnitude than the overload noise and is bounded by the size of the quantization interval. The overload noise, on the other hand, is essentially unbounded and can become very large depending on the size of the prediction error.

In the busy regions of images, especially edges, the prediction error is generally large, leading to large overload noise values. In monochrome images those noise values result in a blurred appearance around edges, which

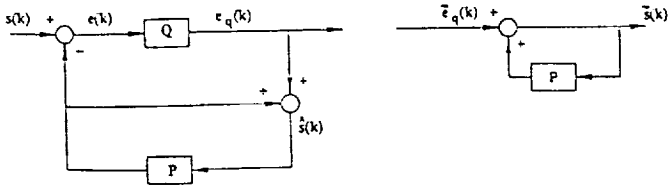


Fig. 6. Block diagram of a DPCM system.

may be acceptable for certain applications. However, in color-mapped images these noise values will result in splotches of different colors. The edge preserving DPCM (EPDPCM) system avoids this problem by the use of a recursively indexed quantizer [8], [9].

For a given quantizer stepsize Δ and a positive integer K, define x_l and x_h as follows:

$$x_l = -\left\lfloor \frac{K-1}{2} \right\rfloor \Delta$$

$$x_h = x_l + (K-1)\Delta \tag{12}$$

where $\lfloor x \rfloor$ is the largest integer not exceeding x. A recursively indexed quantizer of size K is a uniform quantizer with step size Δ (the uniform spacing both between the thresholds and between the output levels) and with x_l and x_h being its smallest and largest output levels (Q defined this way always has 0 as an output level). The quantization rule Q is given as follows. For a given input value x we have the following:

1) If x falls in the interval $[x_l + (\Delta/2), x_h - (\Delta/2)]$, then $Q(x)$ is the nearest output level.

2) If x is greater than $x_h - (\Delta/2)$, see if $x_1 \triangleq x - x_h \in [x_l + (\Delta/2), x_h - (\Delta/2)]$. If so, $Q(x) = [x_h, Q(x_1)]$. If not, form $x_2 = x - 2x_h$ and do the same as for x_1. This process continues until for some m, $x_m = x - mx_h$ falls in $[x_l(\Delta/2), x_h - (\Delta/2)]$, in which case x will be quantized into

$$Q(x) = (\underbrace{x_h, x_h \cdots x_h}_{m \text{ times}} Q(x_m)). \tag{13}$$

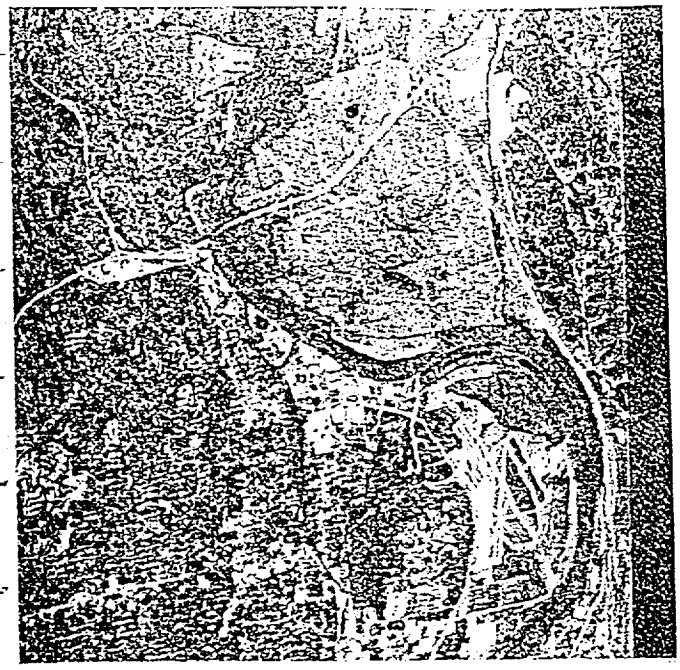
3) If x is smaller than $x_l + (\Delta/2)$, a similar procedure to this is used, i.e., $x_m = x - mx_l$ is formed so that it falls in $[x_l + (\Delta/2), x_h - (\Delta/2)]$, and is quantized to $[x_l, x_l, \cdots, x_l, Q(x_m)]$.

In summary, the quantizer operates in two modes: it operates in one mode when the input falls in the range (x_l, x_h), and another when the input falls outside of the specified range.

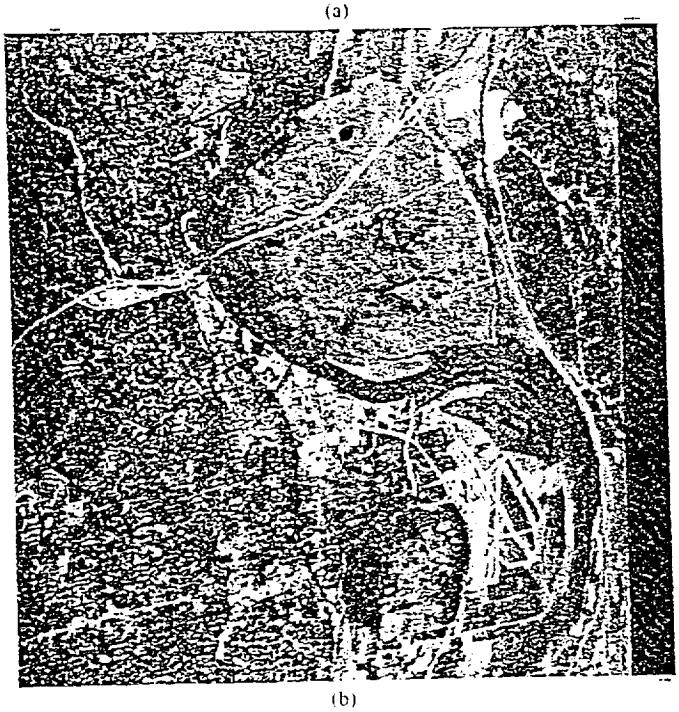
The magnitude of the quantization error is therefore always bounded by $\Delta/2$. This attribute makes it ideal for application to the coding of color-mapped images. Another advantage of the EPDPCM system is that as the quantizer output alphabet can be kept small without incurring overload error, the output is amenable to entropy coding.

Results using the EPDPCM system are shown in Fig. 7. The images in Fig. 7(a) and (b) were coded at a rate of 2 b/pixel and 1.37 b/pixel respectively.

The advantage of DPCM systems over transform cod-



(a)



(b)

Fig. 7. (a) Omaha image with sorted color map coded at 2 b/pixel using EPDPCM. (b) Omaha image with sorted color map coded at 1.37 b/pixel using EPDPCM.

ing systems is their low complexity and higher speed. However, the reconstruction quality obtained using transform coding systems is generally significantly higher than that of DPCM systems at a given rate. Comparing Fig. 7(a) and (b) with Fig. 5(a) and (b), this is obviously not the case for the sorted color-mapped images. In fact, the quality of the 2-b EPDPCM-coded image is actually somewhat higher than the 2-b DCT-coded image. Thus, using the EPDPCM system provides advantages both in terms of complexity and speed, and reconstruction quality.

V. Conclusion

In this paper we have shown that the use of sorted color maps makes color-mapped images of the type used by GIS amenable to both lossless and lossy compression. The sorting of the color maps can provide significant savings of resources for remote sensing image archives, while making lossy compression of color-mapped images possible. The latter fact allows for the use of progressive transmission schemes with pseudo-color and color-mapped images. For lossy compression, conventional wisdom dictates the use of DCT coding for most types of images. However, for color-mapped images, DPCM coding might be more advantageous.

Acknowledgment

The authors wish to acknowledge the Center for Advanced Land Management Information Technologies (CALMIT), University of Nebraska-Lincoln, for providing the Omaha and Lincoln images.

References

[1] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*. New York: Wiley, 1989.
[2] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton, NJ: Princeton Univ., 1962.
[3] CompuServe, Inc., *Graphics Interchange Format (GIF) Specification*. Columbus, OH: CompuServe, Inc., June 1987.
[4] J. D. Foley *et al.*, *Computer Graphics: Principles and Practice, Second Edition*. Reading, MA: Addison-Wesley, 1990.
[5] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
[6] S. Lin, "Computer solutions of the traveling salesman problem," *Bell Syst. Techn. J.*, pp. 2245–2269, Dec. 1965.
[7] W. H. Press *et al.*, *Numerical Recipes in C*. New York: Cambridge Univ., 1988.
[8] M. C. Rost and K. Sayood, "An edge preserving differential image coding scheme," *IEEE Trans. Image Process.*, vol. 1, pp. 250–256, Apr. 1992.
[9] K. Sayood and S. Na, "Recursively indexed quantization of memoryless sources," *IEEE Trans. Inform. Theory*, vol. 38, Nov. 1992.
[10] G. K. Wallace, "The JPEG still picture compression standard," *Commun. Assoc. Comput. Mach.*, vol. 34, no. 4, pp. 31–44.

Andrew C. Hadenfeldt (S'87–M'82) received the B.S. (cum laude) and M.S. degrees in electrical engineering from the University of Nebraska–Lincoln in 1989 and 1991, respectively. His main research interest was compression and progressive transmission of color images.

In 1992 he joined the Section of Cardiology, University of Nebraska Medical Center. He is currently working on the transmission and storage of medical images and medical information management.

Mr. Hadenfeldt is a member of Eta Kappa Nu and Tau Beta Pi.

Khalid Sayood received his undergraduate education at the Middle East Technical University, Ankara, Turkey, and the University of Rochester. He received the B.S. and M.S. degrees from the University of Rochester, Rochester, NY, and the Ph.D. Degree from Texas A&M University, College Station, TX, in 1977, 1979, and 1982, respectively, all in electrical engineering.

He joined the University of Nebraska–Lincoln in 1982, where he is a Professor with the Department of Electrical Engineering. His current research interests include data compression, joint source/channel coding, communication networks, and biomedical applications.

Dr. Sayood is a member of Eta Kappa Nu and Sigma Xi.