Improvement of MPEG-2 Compression by Position-Dependent Encoding

by

Eric Reed

B.S., Electrical Engineering
Drexel University, 1994

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the
Degree of

Master of Science
in Electrical Engineering

at the
Massachusetts Institute of Technology

February 1996

Signature of Author ____________________
Department of Electrical Engineering
February,1996

Certified by ____________________
Jae S. Lim
Professor of Electrical Engineering
Thesis Supervisor

Accepted by ____________________
Frederic R. Morgenthaler
Chairman, Committee on Graduate Study

Improvement of MPEG-2 Compression by Position-Dependent Encoding

by

Eric Reed

Submitted to the Department of Electrical Engineering and Computer Science on February 6, 1996 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Abstract

In typical video compression algorithms, the DCT is applied to the video, and the resulting DCT coefficients are quantized and encoded for transmission or storage. Most of the DCT coefficients are quantized to zero. Efficient encoding of the DCT coefficients is usually achieved by encoding the location and the amplitude of the nonzero coefficients. In typical MC-DCT compression algorithms, up to 90% of the available bit rate is used to encode the locations and the amplitudes of the nonzero DCT coefficients. Therefore, efficient encoding of the location and amplitude information is extremely important.

A novel approach to encoding of the location and amplitude information, joint position-dependent encoding, is being examined. Joint position-dependent encoding exploits the inherent differences in statistical properties of the runlengths and amplitudes as a function of position within the 8x8 DCT coefficient block. The bit rates using joint position-dependent encoding versus the MPEG-2 codebooks is compared.

Thesis Supervisor: Jae S. Lim

Title: Professor of Electrical Engineering

To family and friends
for their love and support.

Acknowledgments

I would like to thank my thesis supervisor, Professor Jae Lim, for his guidance, advice and helpful suggestions throughout my thesis work. I would like to thank everyone at the Advanced Telecommunications Research Program (ATRP) at MIT, all the graduate students and Cindy LeBlanc and Denise Rossetti, for their friendship, advice, and suggestions. Special thanks to David Baylon for reviewing the original manuscript and many helpful suggestions throughout this research. I would especially like to thank John Apostolopoulos, for sharing his knowledge, for his timely and thorough discussions on video compression, and many helpful suggestions throughout my thesis work.

I would like to thank my family and friends for their love, support, and encouragement. I would especially like to thank mom for all her sacrifices over many years to make this research opportunity possible.

This research has been sponsored by ATRP and the US Department of Defense. The current members include Ampex Corporation, General Instrument, Polaroid, Eastman Kodak, Capital Cities/ABC, IBM, and PBS.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past few decades, advancements in digital video technology have made it possible to use digital video compression for many telecommunication applications which include digital broadcast, multi-media and interactive video. In an effort to standardize compression of video and associated audio on digital storage media, the Moving Picture Experts Group (MPEG) standard was established by cooperative efforts of many organizations throughout the world. The MPEG standard utilizes modern sophisticated video compression methods including source adaptive processing, motion estimation/compensation, transform domain data representation, and statistical coding. The standard converts full motion video into a bitstream that can be delivered over existing computer and telecommunication networks.

One application that benefits from modern video compression technology is all-digital High Definition Television (HDTV). One standard HDTV format transmits 60 progressively scanned 720 x 1280 pixel frames every second. Since each pixel contains red, green, and blue color components with 8 bit representations, the uncompressed bit rate corresponds to over 1.3 billion bits per second. This enormous amount of data must be transmitted over a limited single channel bandwidth of 6 MHz for terrestrial broadcast. Considering that the proposed transmission scheme for the Grand Alliance (GA) HDTV system (US standard) uses an 8 level (3 bits/symbol) vestigial sideband (VSB) transmission system, symbols will be transmitted at a rate slightly greater than 10 Msymbols per second since only a little more than 5 MHz of the 6 MHz channel is actually usable. This results in a transmission rate slightly greater than 30 million bits per second

(Mbps). After apportioning bits for error correction, there remains only about 20 Mbps for the video, audio, and special services. In order to broadcast the standard HDTV format within a 6 MHz television channel, the raw data rate must be reduced to below 20 Mbps requiring a compression factor greater than 70. The GA HDTV system adopted MPEG-2 as a basis for the task of compressing the video. A detailed description of the GA HDTV system can be found in [8].

In many telecommunication applications, including HDTV, bandwidth is the major limiting factor. Therefore, it is advantageous to use as few bits as possible while keeping the video quality at an acceptable level. In the MPEG-2 compression system, the quantized DCT coefficients utilize between 80 and 90% of the bit rate. Therefore, improving the method of coding the quantized DCT coefficients will provide significant improvements to the overall system. While MPEG-2 fits the video and associated audio into a constrained medium, this thesis introduces a method to reduce the bit rate requirement even further by exploiting the different statistical properties of the quantized transform coefficients based on position and coefficient type. Hence, the coding scheme has been appropriately termed position-dependent encoding (PDE). Though the techniques discussed in this thesis are extendible to other digital compression standards as well, focus is placed on MPEG-2 for HDTV compression.

PDE was introduced in the past with significant savings in bit rate requirements. Tests were performed using a separate coding approach which assigns codewords to the runlengths and amplitudes separately. The GA system uses MPEG-2 which requires a joint coding approach where the runlength and the following amplitude are coded as one event. The average total decrease in bit rate using separate coding was 6.1 %. In these tests, the codebooks were trained from a given set of sequences and performance was measured by applying the exact set of sequences to the system. Obviously, these tests do not emulate the training and running in an actual coding environment and therefore may produce results that are optimistic. However, the results indicate that the approach has a great deal of potential and further attention should be given.

The overall goal of this thesis is to develop PDE codebooks using an MPEG-2 encoder and compare performance based on bit rates to the MPEG-2 variable length code (VLC) tables. During the experiments, we will address a number of questions that previous results left unanswered. First, tests should be performed in a manner that best emulates an actual coding environment. Therefore, this thesis examines PDE performance using sequences outside the training set. Since an MPEG-2 encoder is used in all testing, a joint coding approach is taken. Joint coding offers the benefit of exploiting the correlation among the runlengths and amplitudes and therefore may produce improvements over the separate coding approach. Lastly, previous tests do not include B-frames. Therefore, the effect of inserting B-frames into the frame structure is examined.

A video compression system can be described by three distinct but interrelated stages: representation, quantization, and codeword assignment (Figure 1-1).

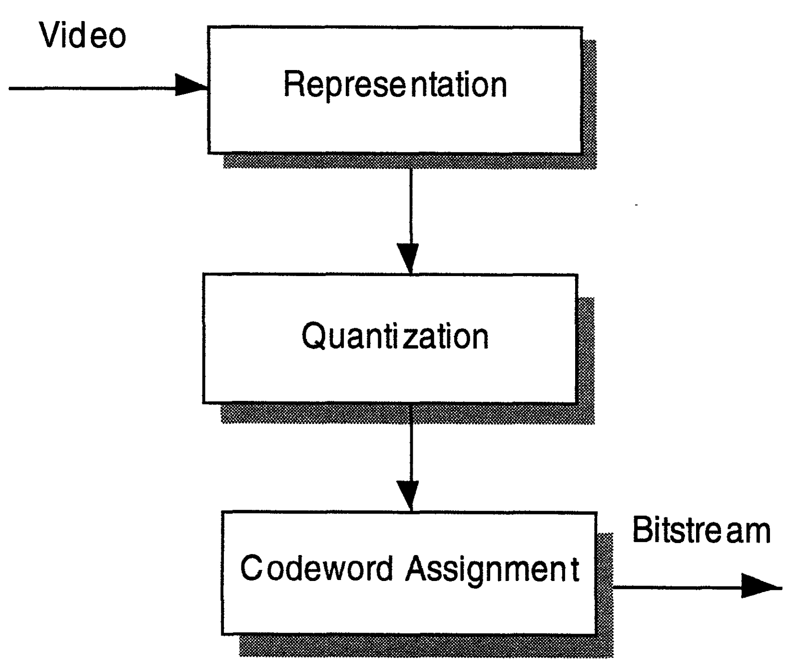


Figure 1-1 A typical video compression system.

The goal is to reduce the redundant and irrelevant information inherent in a typical video signal along the temporal, spatial and color space dimensions. Joint PDE provides a more efficient manner of assigning codewords. In order to understand the full benefits of this coding approach, one should have a deep understanding of digital video compression. Deep understanding of compression technology will enable one to realize the tradeoffs and take advantage of the large amount of parameters that affect the performance of joint PDE. Therefore, chapter 2 provides a more detailed discussion of MPEG-2 compression with emphasis placed on concepts affecting performance. The functions of each of the three stages will be described.

Chapter 3 discusses the issue of coding the quantized transform coefficients. Conventional coding approaches are introduced followed by a discussion on joint PDE. Practical considerations concerning PDE are discussed in Chapter 4. Chapter 5 discusses the experimental setup and presents the results of the joint PDE approach. Finally, Chapter 6 offers some concluding remarks and discusses potential applications and future work.

Chapter 2

Overview of MPEG-2 Compression

The MPEG-2 model is based on a layered structure described by a total of 6 layers: video sequence, group-of-pictures, picture, slice, macroblock, and block. The goal of the layered structure is to separate entities in the bitstream that are logically distinct, to prevent ambiguity and to aid in the decoding process. As each of the layers is described, important compression concepts are discussed.

The video sequence (base) layer is the highest level of the coded bitstream. At the beginning of each sequence, a sequence header is inserted into the bitstream consisting of a series of data elements. The header contains pertinent information needed for decoding such as the bit rate, buffer size, frame rate, frame size, pel aspect ratio, and quantization information. MPEG-2 syntax allows users to enter data into the bitstream for their specific applications. For example, users can define their own quantization matrices rather than using the MPEG-2 default matrices since the syntax allows entire matrices to be transmitted to the decoder. The quantization matrices play an important role in the MPEG-2 encoder which will be discussed later in the chapter. Also, MPEG-2 defines a set of profile and level indications which ensure interoperability with many areas of application. The profile is a defined subset of the bitstream syntax and the level is a defined set of constraints imposed on the parameters in the bitstream. In order to meet HDTV requirements, MPEG-2 operates at the main profile and high level. The profile and level parameters allow MPEG-2 to meet many desired bit rates and resolutions. In order to support tuning in, frequent repetitions of the video sequence parameters are inserted into the bitstream.

Each video sequence is divided into random access units which consist of a series of consecutive pictures (frames). In MPEG-2 terminology, this layer is defined as the Group-of-Pictures (GOP). The number of pictures in a GOP is arbitrary. Each GOP is required to begin with an intra picture, a picture that is processed directly from the original image without reference to any other pictures. Intra pictures serve as a reference for future pictures and thus initiate the prediction loop. Therefore, every picture within the GOP can be reconstructed from other pictures within the group. This allows random accessibility, editing and basic VCR functions within the compressed bitstream. At the start of each GOP, a GOP header is inserted into the bitstream consisting of timing information.

The frames within each GOP make up the picture layer. This layer is the primary coding unit and consists of a picture header followed by the picture data. The picture header is sent at the start of each picture allowing parameters to be changed on a frame by frame basis. The header provides temporal location information, the picture type, and motion vector constraints.

Color Space Conversion

Each pixel of the video contains red, green, and blue (RGB) color components. The human visual system has a higher sensitivity to the luminance compared to the color detail. Therefore, initial processing of each frame involves linearly transforming RGB into another color coordinate system. In MPEG-2, RGB is transformed to the YUV coordinate system which allows the properties of the human visual system to be exploited. Y represents the luminance or intensity of the image and is equivalent to a black-and-white image. U and V represent the chrominance components which contain the color detail of the image. The transformation reduces the correlation existing among RGB components, therefore allowing the Y component to be processed separately without significantly affecting the chrominance. The human visual system is less sensitive to detail in the chrominance compared to the luminance. As a result, the chrominance component can be subsampled or quantized more coarsely than the luminance without significantly affecting video quality. By using the YUV representation instead of RGB, significant bit rate reductions can be achieved. Each pixel represented by RGB requires 3 parameters. If each chrominance component is decimated by a factor of two both horizontally and vertically, then 1.5 parameters are needed to describe each pixel in the YUV representation. Therefore, the number of parameters to be coded is reduced by 50 %.

Each frame is further segmented into slices which can be defined arbitrarily. Typically, slices are defined as 16 full rows of resolution containing an integer number of 16 x 16 blocks. A slice header consisting of location information is inserted into the bitstream at the start of each slice. Slices add robustness into the compressed bitstream; for example, the intra DC predictors as well as motion vector predictors are reset at the beginning of each slice.

Each slice consists of a fixed number of macroblocks, which are 16 x 16 blocks that make up the motion compensation unit. Each macroblock consists of a section of the luminance component along with the corresponding chrominance components. Macroblocks share the same motion displacement since temporal processing is performed in this layer. In order to achieve proper motion rendition, a high frame rate is necessary, resulting in a great deal of temporal redundancy among adjacent frames. For example, the GA HDTV transmits up to 60 still frames every second, therefore adjacent frames often contain the same backgrounds and objects at different spatial locations. Therefore, reducing redundancy along the temporal dimension provides a significant amount of compression which is necessary for the transmission of high quality video at low bit rates.

Temporal Processing

Temporal processing involves a combination of estimating the motion between adjacent frames and then compensating for this motion. Motion estimation (ME) is referred to as the process of estimating the motion of objects within a video sequence. The process of compensating for the presence of motion is referred to as motion compensation (MC). The basic idea behind ME/MC is to make a prediction of the current frame from neighboring frames. The error in the prediction, referred to as the motion-compensated residual, is processed and transmitted thus eliminating much of the redundancy in the signal. Two methods of motion compensation are used by MPEG-2: MC-prediction and MC-interpolation. MC-prediction refers to the process of estimating motion from past frames while MC-interpolation estimates motion from a combination of past as well as future frames. As a result, the MPEG-2 frame structure includes intra coded (I), predictive-coded (P), and bidirectionally predictive-coded (B) frames. A typical MPEG-2 frame structure is shown in Figure 2-1.

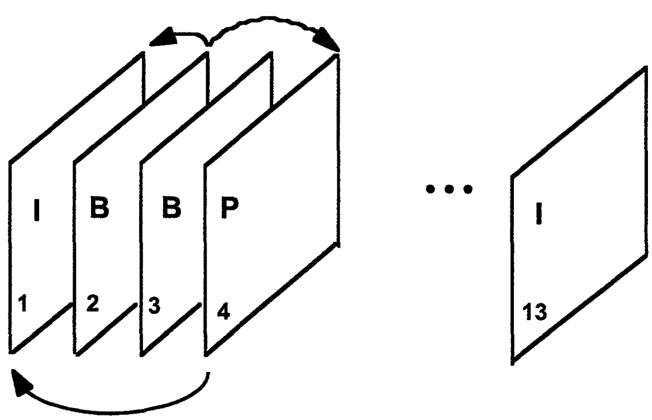


Figure 2-1 Typical MPEG-2 frame structure (N=12,M=3). N is the period of intra pictures or the GOP size. M-1 is the number of B-frames between a given pair of I- or P-frames.

The organization of the pictures is quite flexible and depends on the application. P-frames are predicted with reference to a past I- or P-frame where B-frames are predicted with reference to past and future I- and P-frames. Therefore, each macroblock in a B-frame can choose between intra coding or the three different types of prediction: forward, backward, or bidirectional (average of past and future frames). MC-interpolation provides the highest amount of compression due to very efficient prediction. It allows MPEG to deal with uncovered areas in cases of scene changes and provides better statistical properties since the effect of noise is reduced due to the averaging involved. However, inserting B-frames decreases the correlation between the reference frames thus making them harder to encode. I-frames provide only moderate compression since no temporal processing is performed but serve as a reference for random accessibility. In the case where motion compensation is used, the error of the prediction is encoded to ensure that degrading artifacts do not occur.

Spatial Processing

After temporal processing, a great deal of compression is achieved by reducing the redundancy along the spatial dimension. In most video, uniform regions exist where neighboring pixels have the same pixel values. Therefore, only one pixel is necessary to describe these regions. A popular spatial redundancy technique involves a linear transformation into another domain where most of the energy of the image lies within a small fraction of the transform coefficients. Coding and transmitting the most energetic coefficients will result in a high quality image with minimal distortion. MPEG has adopted the block Discrete Cosine Transform (DCT) for spatial processing. The DCT is attractive since the coefficients are real and fast algorithms (FFT) exist for efficient computation. The Discrete Fourier Transform (DFT) has inherent disadvantages such as complex coefficients and high frequency energy due to artificial discontinuities making it unfavorable for compression. Since the characteristics of typical video vary within each frame, frames are typically partitioned into 8 x 8 blocks which are independently transformed and processed. The benefits of partitioning the image into small blocks include a significant reduction in computational and memory requirements in addition to allowing spatial adaptive processing.

The block layer makes up the lowest layer in the MPEG syntax. The MPEG-2 standard supports three chrominance formats. For each format, macroblocks consists of four luminance blocks and a variable number of chrominance blocks. The 4:2:0 format contains two chrominance blocks for every four luminance blocks where the 4:2:2 and 4:4:4 formats contain four and eight chrominance blocks, respectively. The 4:2:0 format discussed earlier is used for applications where a significant amount of compression is necessary. The remaining chrominance formats are useful for applications requiring little compression such as routing video through a broadcast station before transmission.

Quantization and Codeword Assignment

Up to this point, the data has been manipulated into an elegant representation, but no compression has taken place. However, most of the perceptually important information has been compressed into a few pieces of information. Actual bit rate compression is achieved through quantization and codeword assignment. Quantization is performed to discretize the values of the DCT coefficients. A digital computer has a finite number of bits which limit the accuracy of a digital representation. An analog value can be represented digitally with a finite number of bits by defining a finite number of reconstruction or quantization levels. Quantization maps an analog value (DCT coefficient) to its quantized representation so that it can be described by a digital codeword. The analog transform coefficients can be individually quantized (scalar quantization) or they can be jointly quantized as a group (vector quantization). Vector quantization takes advantage of the statistical dependency among the elements at the cost of increased complexity. MPEG-2 uses scalar quantization where each of the coefficients are quantized separately.

In scalar quantization, each coefficient may be quantized with a uniform (linear) or nonuniform (nonlinear) quantizer. When quantizing the transform coefficients, it is beneficial to exploit the perceptual importance of the DCT coefficients by weighting the coefficients based on frequency and component type. Quantization is adapted to select the most important information in the coefficients while reducing less important information. This is accomplished by varying the stepsizes of the quantizer for each of the coefficients. Since the human visual system has a higher sensitivity to low frequency quantization noise, the low frequency coefficients are quantized more finely and the less important high frequency coefficients are quantized more coarsely. Since high detailed regions tend to mask noise making it difficult to see, the quantizer stepsize is increased for regions with high spatial activity. The method used by MPEG to achieve different stepsizes is to weight each coefficient based on its visual importance. The weighting factors have the effect of increasing the stepsize (coarser quantization) or decreasing the stepsize (finer quantization) in the appropriate regions of the block. Once the weighting matrix has been applied to the DCT coefficients, the normalized coefficients are quantized uniformly. Since intra DC coefficients carry the most important information, these coefficients are quantized separately with the highest precision. MPEG also introduces a dead zone for inter coefficients, so that more coefficients are quantized to zero to eliminate undesirable noise perturbations.

In order for the receiver to interpret the transmitted reconstruction levels, every possible output of the quantizer must be assigned a codeword before transmission. Codeword assignment converts the quantized coefficients into a digital bitstream for transmission. For messages with equal probabilities, uniform or fixed length codewords result in the optimal codeword assignment which has an average bit rate equal to the entropy, or information in the message. However, messages will typically have different probabilities of occurrence and therefore convey varying amounts of entropy. Since the goal is to maintain the lowest possible bit rate, variable-length codewords are introduced for messages with unequal probabilities of occurrence to take advantage of the statistical properties of the data. Messages with higher probabilities of occurrence may be assigned shorter length codewords while messages least likely to occur will be assigned longer codewords. Huffman coding, the entropy coding scheme used by MPEG-2, results in the lowest possible average bit rate that is uniquely decodable. The probability distributions that are necessary for creating the Huffman codebooks are usually obtained by collecting the relevant statistics from a set of video sequences (the training set).

Huffman coding reduces the average bit rate but it also produces a variable bit rate output from the encoder. In most broadcast scenarios, the channel has a fixed transmission rate and a variable bit rate is not tolerable. One method to hold the bit rate constant (used by MPEG) is to introduce a buffer which collects and holds the bits before transmission. Ideally, the buffer will be half full at all times. In order to prevent the buffer from possible overflow or underlow, a feedback mechanism is used to vary the amplitude resolution. For example, if the buffer begins to overflow, coefficients are quantized more coarsely until the buffer empties to a desirable level. If the buffer begins to underflow, coefficients are quantized more finely which would increase the bit rate.

For a description of general video compression techniques please refer to [1,3,9]. For a more in-depth discussion of video compression refer to [2]. For a more in-depth discussion of MPEG-2 refer to [4,6].

Chapter 3

Coding the Transform Coefficients

In order to design the Huffman codebooks discussed in chapter 2, a method that efficiently distinguishes all possible outputs of the quantizer must be developed. The most widely used method is runlength encoding approaches which is the focus of this chapter. Before introducing a joint PDE scheme, it is beneficial to become familiar with conventional runlength encoding approaches. Therefore, section 3.1 introduces a conventional runlength encoding approach used by MPEG-2. Section 3.2 illustrates the differences among the runlength and amplitude statistics based on position. The joint PDE approach is discussed in section 3.3. The chapter concludes with a discussion describing the differences among the DCT block types.

3.1 Conventional Coding

Since a majority of the bit rate is used for the quantized DCT coefficients, efficient coding methods are highly important. In typical video and image compression scenarios most of the DCT coefficients in a block are quantized to zero producing a sparse 8 x 8 matrix. An effective method that exploits the large number of zero coefficients is runlength encoding which involves encoding the location and amplitude of only the nonzero coefficients (selected coefficients). Since a great majority of the transform coefficients are quantized to zero, runlength encoding methods can achieve high compression. The quantized coefficients are ordered into a one-dimensional vector through a zigzag scanning of the block starting at DC and finishing at coefficient (7,7), as shown in Figure 3-1. The location information of the nonzero coefficients can be obtained by encoding the runs of zeros between consecutive nonzero coefficients. The first coefficient after a nonzero coefficient is considered the starting position of the appropriate runlength (i.e. runlength 0). The amplitude values of the nonzero coefficients are encoded along with their location information. MPEG-2 allows for an alternate scanning pattern in addition to that shown in Figure 3-1. However, all experiments presented in this thesis use the zigzag scan.

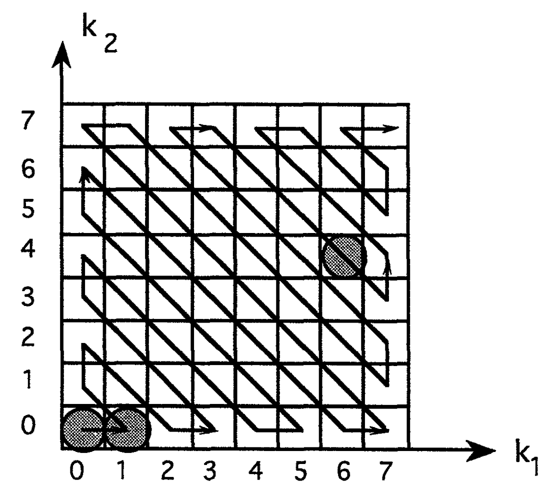


Figure 3-1 Zigzag scanning of the DCT block and ordering of the quantized coefficients.

Using a zigzag scan, the sequence of events to be encoded in Figure 3-1 for inter blocks are:

[1] runlength 0;

[2] amplitude of (0,0) coefficient;

[3] runlength 0;

[4] amplitude of (1,0) coefficient;

[5] runlength 50;

[6] amplitude of (6,4) coefficient; and

[7] EOB (End of Block).

The EOB event signifies that there are no more nonzero coefficients in this block. For intra blocks, the amplitude of the DC coefficient is coded separately due to its perceptual importance and is almost never zero. Therefore, if the block in Figure 3-1 were an intra block, the sequence of events to be encoded would be the following:

[1] amplitude of (0,0) coefficient;

[2] runlength 0;

[3] amplitude of (1,0) coefficient;

[4] runlength 50;

[5] amplitude of (6,4) coefficient; and

[6] EOB.

Separate and Joint Huffman Codebooks

The runlengths and amplitudes can be treated as separate events where one codebook is used to encode the runlengths and one codebook is used to encode the amplitudes. This approach is referred to as the *separate coding* of runlengths and amplitudes where both codebooks are one-dimensional.

On the other hand, a runlength and the following amplitude can be treated jointly as a single event, in which case only one codebook is needed. However, this codebook is two-dimensional. The advantage of the *joint coding* approach is that it exploits the correlation between a runlength and the following amplitude. MPEG standards require joint encoding of runlengths and amplitudes and therefore is the approach taken in this research. Refer to [5,7] for a thorough discussion of the separate coding approach.

Differences in Statistics and Motivation for PDE

Conventional approaches have proven to be highly effective. However, the bits are not being used in the most efficient manner. MPEG-2 uses 2 VLC tables in order to exploit the different statistics for intra and inter regions, but the statistics of the quantized transform coefficients also vary with frequency and component (luminance/chrominance) type. The MPEG tables do not exploit these inherent differences since the codebooks are designed to perform well everywhere. In other words, the codebooks are designed based on the most likely events out of all the nonzero coefficients. Recognizing the differences among coefficients and designing codebooks to exploit them can yield significant improvements on the performance of the encoder. The joint position-dependent encoding scheme that is to be discussed in the following sections exploits the differences in statistics of runlengths and amplitudes as a function of position and coefficient type.

3.2 Runlength and Amplitude Statistics

Unlike conventional runlength encoding approaches, position-dependent encoding exploits the differences in range and statistics of runlengths and amplitudes as a function of position in the block by introducing multiple codebooks based on the starting position of the runlength. This is illustrated in Figure 3-2.

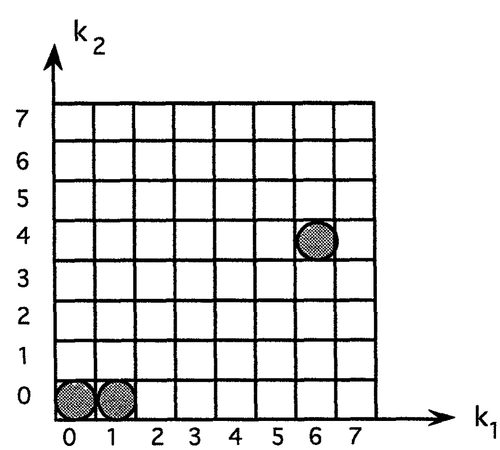


Figure 3-2 An example 8x8 block of quantized DCT coefficients. Non-zero coefficients are shaded. k_1 is the horizontal frequency, k_2 the vertical frequency; DC is in the bottom left corner.

As discussed in chapter 2, the human visual system has a higher sensitivity to low frequency quantization noise. For example, DC coefficient error results in mean value distortion for a block which exposes block boundaries. However, high frequency coefficient error will appear as noise or texture which is less annoying. Therefore, the low frequency coefficients are quantized with the highest precision and thus have the largest amplitude ranges. Since most video has a great deal of low frequency content to begin with, a majority of the signal energy lies in the low frequency coefficients.

From the above discussion, it is likely that a majority of the nonzero coefficients are concentrated in the low frequency region. Therefore, a nonzero coefficient in the low frequency region, such as (0,0) or (0,1), followed by a zero runlength has a comparably high probability of occurrence. For example, a nonzero high frequency coefficient, such as (6,4), will most likely be the last nonzero coefficient in the 8 x 8 block. Also, the majority of the nonzero high frequency coefficients will tend to have small amplitudes, whereas the nonzero low frequency coefficients will tend to have large as well as small amplitudes, depending on the block type. For example, coefficient (0,0) will most likely be large for intra regions (original image) and small for inter regions (MC-prediction error), but the coefficient at (6,4) is almost always small.

The range of the runlengths also depends on the starting position within the block. For example, the runlength range for a coefficient starting at position (1,0) is between 0 and 62 requiring a total of 6 bits to describe all runlength combinations. However, the runlength range starting at position (7,3) is only between 0 and 10 thus requiring 4 bits to describe all runlength combinations. The range of the amplitudes also depend on the position of the coefficient within the block since quantization matrices usually weigh coefficients differently. However, this depends on the user defined quantization matrices.

The codebooks designed for MPEG-2 assign the shortest codewords to the most likely events out of all the coefficients. If short codewords are assigned to the most likely events for each coefficient separately, we are able to exploit the differences mentioned above and thus reduce the bit rate. In order to exploit these differences, each coefficient within the 8 x 8 block should have its own codebook.

3.3 Joint Position-Dependent Encoding

In the joint position-dependent encoding scheme, each coefficient may have its own codebook for the joint amplitude-runlength event. Since a runlength and the following amplitude is treated as a single event, the correlation between amplitudes and the runlengths can be exploited when assigning codewords. From section 3-2, small runlengths followed by large or small amplitudes in the low frequency region and large runlengths followed by small amplitudes in the high frequency region are the most frequently occurring events. Figure 3-3 demonstrates the behavior of a sample collection of statistics for the joint amplitude-runlength event.

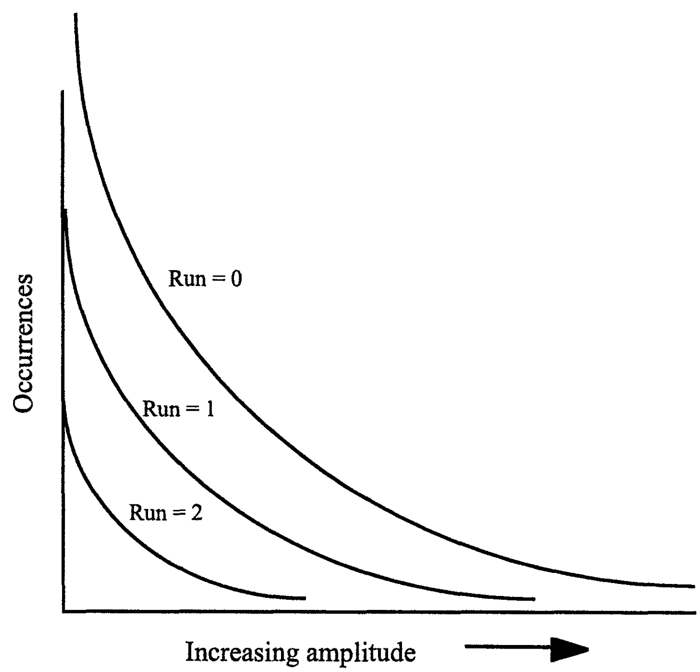


Figure 3-3 A representative collection of quantized DCT statistics. The diagram represents the statistics collected for an arbitrary position within the block (i.e. constant frequency).

It is worthwhile to look at Figure 3-3 more closely to show that the sample collection of statistics support the observations made earlier. The figure represents a collection of statistics for a given position, or one particular coefficient in the block. As we increase in runlengths while keeping the amplitude fixed, the probability of the joint amplitude-runlength event decreases. As we move toward higher runlengths, we are essentially increasing in frequency. Since these events are less likely to occur, this confirms the observation that small runlengths are more likely to occur in low frequency regions for a fixed position within the block. If we take a horizontal slice and travel from low to high frequency (right to left in the figure), this illustrates that amplitudes tend to decrease as we move into a high frequency region. Also, for a fixed runlength the probability of an event occurring decreases with increasing amplitude. Remember this makes sense since the figure does not include the intra DC statistics.

3.4 Block Type Differences

The runlength and amplitude statistics depend not only on the starting position within the block, but also on the block type. Blocks are distinguished depending on whether they are intra or inter encoded regions, or whether they represent the luminance or chrominance component. Inter blocks can be further segmented into P- and B-blocks. These block types may have unique properties that would benefit the encoder by assigning separate codebooks to each type.

For reasons discussed in section 3-2, the nonzero quantized DCT coefficients are concentrated in the low frequency region of the intra blocks. On the other hand, inter blocks represent the prediction error so they tend to have predominately small amplitudes that are sparsely spread throughout the block. Obviously, the amplitude and runlength statistics are significantly different for intra and inter blocks. MPEG-2 exploits these differences by defining two separate codebooks.

There are also differences that exist between the luminance and chrominance components. The human visual system has a reduced response to the chrominance components. Therefore, typical video compression systems can subsample the chrominance component by a factor of two along the horizontal and vertical dimensions without introducing noticeable (untolerable) distortion (4:2:0 subsampling format discussed in Chapter 2). This sampling results in four luminance blocks for every two chrominance blocks. Therefore, more of the bit rate is occupied by the luminance bits making it beneficial to exploit these differences in the runlength and amplitude statistics. In addition, the luminance and chrominance components may have different quantization matrices which contribute to the statistical differences.

Since more bits are assigned to P-frames over B-frames, the P-frames use up more of the bit rate per frame. In addition, B-frames provide better prediction compared to P-frames which results in different error statistics. Therefore, it may be beneficial to separate the statistics of the P- and B-frames.

These observations motivate introducing different sets of codebooks for each of the blocks types mentioned above: intra Y, intra UV, P Y, P UV, B Y, and B UV. This requires 382 total codebooks, one to represent each coefficient in each block type. Each inter block introduces 64 codebooks and each intra block introduce 63 codebooks since the DC coefficient is left alone. While it seems very beneficial to design separate codebooks, 382 codebooks require an enormous amount of memory and is highly impractical. Chapter 4 will address the practical considerations of joint PDE.

Chapter 4

Practical Considerations

4.1 Codebook Allocation

The MPEG-2 standard uses a total of 2 VLC tables to separate the differences between intra and inter regions. Each codebook has a total of 114 entries with the largest entry of 16 bits excluding the last bit that denotes the sign of the amplitude. Since joint PDE introduces multiple Huffman codebooks, memory requirements increase significantly. A total of 382 codebooks would be required for each coefficient to have its own, unique, codebook as discussed in Chapter 3. If each of these codebooks has the same size as the MPEG-2 codebooks, memory requirements increase by a factor of 192 resulting in a highly impractical scheme.

PDE could be made more practical by grouping coefficients that share similar runlength and amplitude statistics. Within each block, less important coefficients can be grouped together to share one codebook while the more important coefficients can maintain their own codebooks. Consequently, it is possible to reduce the number of codebooks to a more practical level without significantly reducing the joint PDE performance. For example, since the luminance takes up more of the bit rate, it makes sense to assign more codebooks to luminance blocks since there is a greater potential for overall gain. Also, since there are less nonzero high frequency coefficients, we can group more of these coefficients together with little effect on performance. Coefficient grouping also allows us to exploit any discrepancies in joint PDE performance by assigning more codebooks to block types where the highest performance is observed. This is illustrated in Chapter 5.3.

An example assignment of codebooks is shown in Figure 4-1. Coefficients that share codebooks have identical patterns.

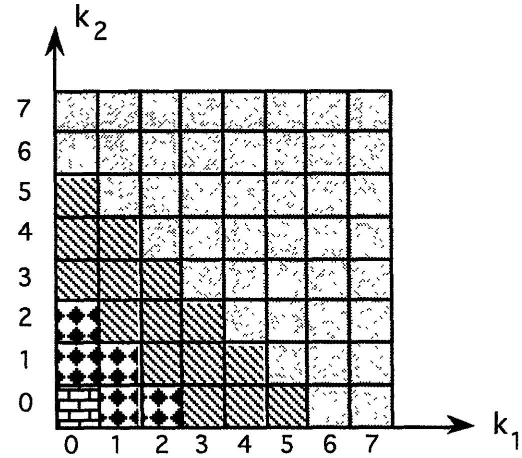


Figure 4-1 An example assignment of codebooks.

4.2 Escape Codes

In a joint runlength-amplitude encoding scheme, codebooks are two-dimensional resulting in a large number of possible events that need codewords. For example, a coefficient with a runlength range between 0 and 63 and an amplitude range between 0 and 1023 has a total of 65,472 possible events. Obviously, many of these events have an extremely low probability of occurrence (≈ 0 probability). Therefore, escape codes are introduced for events that are not likely to occur. With a joint PDE scheme, there are even more events within each codebook where the probability of occurrence is approximately zero. The MPEG-2 escape codeword format consists of a total of 24 bits: 6 bit escape code + 12 bits representing value + 6 bits representing run. This format can be preserved in the joint PDE codebooks. The difference lies in the fact that each PDE codebook can have its own, optimal, escape code. Also, it is appropriate to exploit the amplitude and runlength ranges for each codebook by assigning less bits to represent the runlengths and amplitudes within the escape codeword.

Limiting Codeword Length

There are many methods to introduce escape codes, for example, limiting the codeword length. Once each event is assigned a codeword based on Huffman coding, events that have a codeword length greater than some threshold will be escape coded. The aggregate probability of all events to be escape coded determines the escape code. Since this process changes the codeword assignment for even the non-escape-coded events, there may be additional events in the new codebook with codewords longer than the threshold. Therefore, this procedure is repeated until all non-escape coded events have codewords shorter or equal to the threshold length.

4.3 Estimating Events

In order to design a set of codebooks, statistics are collected from a representative set of video sequences (i.e. training set). Since many possible events may never occur in the training set that may occur in the actual test sequence, it is necessary to estimate the probability of these events. Figure 3-3 indicates that the statistics exhibit an exponential pattern for constant runlengths. One solution is to perform a least squares exponential fit. However, looking at the statistics more carefully, they follow an exponential pattern almost exactly for small amplitudes but become increasingly noisy with increasing amplitude and position. Also, for every set of statistics, there is a maximum amplitude (≈ 255) for which an event occurs that is significantly less than the maximum possible amplitude (1023) for a constant runlength. In other words, a majority of the high amplitude events never occur in training which have almost zero probability and therefore must be estimated. Based on these observations, the least squares fit is not appropriate in certain regions. Another method to estimate statistics involves using a one- or multi-dimensional moving average for regions following an exponential pattern. Regions where no pattern is evident (i.e. large amplitudes), which have nearly zero probability of occurrence, can be estimated using a uniform distribution model. Since the probabilities of occurrence are negligible, discrepancies from a uniform model will have little to no effect on performance. The latter method is used to estimate events in this research using a one-dimensional moving average and a uniform distribution model to estimate regions exhibiting no pattern.

Chapter 5

Experiments and Results

5.1 Experimental Setup

All experiments are based on an MPEG-2 encoder following Test Model 5. Since the focus of this thesis is the compression of HDTV signals, the encoder operates within the high level and main profile specifications. The details of the specification can be found in [6]. The frame structure for testing is shown in Figure 2-1 and is repeated here for convenience in Figure 5-1. All testing is based on a progressive scan along with 4:2:0 chrominance sampling. MPEG-2 default matrices are used for coefficient quantization which are defined in [6]. Runlengths are defined based on the zigzag scan pattern and the DCT coefficients are uniformly quantized for an amplitude range of 0 to 1023 (i.e. mquant ranges from 2 to 62).

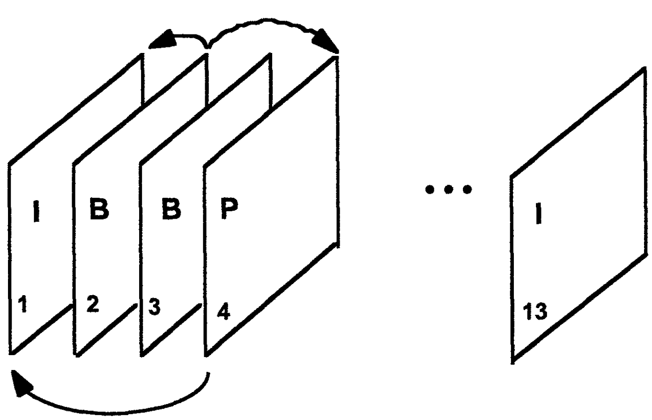


Figure 5-1 Actual MPEG-2 frame structure used for experiments (N=12,M=3).

The target bit rate was maintained at 0.34 bits / pixel using the rate control specified in Test Model 5. First, global buffer control is achieved by estimating the number of bits allocated for an entire GOP as *N*(bitrate / frame rate)* where N denotes the size of a GOP. After an entire picture is encoded, the discrepancy in the number of bits allocated and the actual number of bits needed are taken into account for the bit allocation of the next picture. Local control is achieved by assuming a uniform distribution model to estimate the bit allocation on a macroblock-by-macroblock basis. The deviation from a uniform distribution is taken into account for the bit allocation of the next macroblock. If more bits are used for the previous macroblock than anticipated, the current macroblock is quantized more coarsely effectively reducing the number of bits. The quantization parameter is also adjusted based on the local spatial activity which quantizes high frequency regions more coarsely. The most bits are allocated for I-frames and the least bits are allocated for B-frames. As can be seen from the bit rates actually achieved, the encoder had difficulty reducing the bit rate to 0.34 bits / pixel for most test sequences. One reason for this difficulty could be due to the frame structure used in experiments. In other words, for a reduction in bit rate down to 0.34 bits / pixel, a GOP consisting of 12 frames (N=12) may not be appropriate. Instead, a GOP consisting of 15 frames (N=15) would allow for more compression since an I-frame is inserted every 16 frames rather than every 13 frames. Compression could also be increased by inserting more B-frames into the frame structure.

Sequences and Codebooks

The quantized DCT coefficients are collected from a total of 14 video sequences. A total of 13 frames from each sequence are used for training. This results in one full prediction loop (GOP) plus one additional I-frame in the following GOP. Table 5-1 gives a list of the sequences used for training and testing.

Table 5-1 Summary of the sequences used in training and testing.

Sequence	*Sequence Name*	*Ver. Res.*	*Hor. Res.*	*Frame rate(Hz)*
1	football	720	1024	60
2	beer_truck	720	1024	60
3	tulipz	720	1024	60
4	tulipstxt	720	1024	60
5	picnic	720	1024	60
6	girl	512	512	60
7	mile471	720	1024	60
8	zoom_sign	720	1024	60
9	toytable	720	1024	60
10	raft	720	1280	60
11	traffic	880	1200	24
12	mall	880	1200	24
13	untouchables	832	2000	24
14	marcie	880	1200	24

For the following experiments, each test sequence in Table 5-1 has its own unique codebook. The codebooks for any particular sequence are trained based on the statistics of the 13 remaining test sequences. Statistics are collected for an allocation of 382 codebooks as discussed in section 3.4. In order to determine the benefit of exploiting the differences among P- and B-regions and measure performance for separate frame types, 382 codebooks are used to separate the statistics of the P- and B-blocks.

Calculations

The results in the following sections are presented as two separate sets. The first set compares only bits affected by PDE. The set marked 'TOTALS' measures the overall encoder performance using the entire bitstream. In order to make a fair comparison, bits unaffected by PDE are not included in the first set of results. These bits include intra DC, overhead, and the last bit of all nonzero coefficients used to denote sign. Obviously, this set of results will yield a higher gain in performance compared to the overall encoder performance. Since overall performance is of most practical importance, all graphs correspond to the overall results which incorporate the entire bitstream. The horizontal axis of each graph represents the test sequence number defined in Table 5-1 unless indicated otherwise.

Results are presented in bits per pixel which are calculated as the ratio of the total number of bits used to encode the entire test sequence to the total number of pixels in that sequence. Performance is measured in terms of the percentage decrease of the bit/pixel rate of the joint PDE tables over the MPEG-2 tables.

5.2 Preliminary Experiments

Preliminary experiments were performed on 10 of the 14 test sequences where each set of codebooks is trained from the other 13 test sequences. Since it is necessary to determine the optimal tradeoff between the number of codebooks used and the overall performance of joint PDE, the event corresponding to each coefficient having its own codebook is used as a measure of performance. Since MPEG-2 uses three separate frame types with each serving different purposes, it may be advantageous to allocate separate codebooks for I-, P-, and B-frames corresponding to a total of 382 codebooks. On the other hand, the total number of codebooks can be reduced to 254 if P- and B-blocks are grouped together to form one set of inter codebooks. These results illustrate how joint PDE performs for each frame type and will serve as a comparison between the 382 and 254 codebook case.

The results are presented in Table 5-2 and Figures 5-2 (a)-(f). Since we are comparing 382 codebooks to 254, we would expect to see an increase in performance for the 382 codebook case. However, results show that the 254 codebook case outperformed the 382 codebook case on 3 out of the 10 test sequences. In the other cases, performances are very similar will little or no gain using 382 codebooks. The reason for these unpredictable results is most likely due to the limited collection of statistics. In a practical setting, codebooks will be trained using thousands of frames of many representative sequences. However, these experiments were trained from a limited set of data: 13 frames of 13 test sequences. Therefore, it may be advantageous to separate codebooks for P- and B-frames in a practical setting. Since more bits are allocated for P-frames and prediction is more efficient for B-frames, it may be advantageous to exploit these characteristics.

From the table and figures, the largest gain using joint PDE is achieved with B-blocks with the worst overall performance found in I-blocks. The average decrease in bit rate for I-, P-, and B-frames is 7.1 %, 14.4 %, and 22.8%, respectively. This observation demonstrates a negative correlation between the number of bits allocated and the performance of the joint PDE scheme. From the figures, it is easy to see that the variance of the overall performance is much larger for the inter blocks compared to the intra blocks. This illustrates the importance of finding a collection of statistics that closely matches the test sequence.

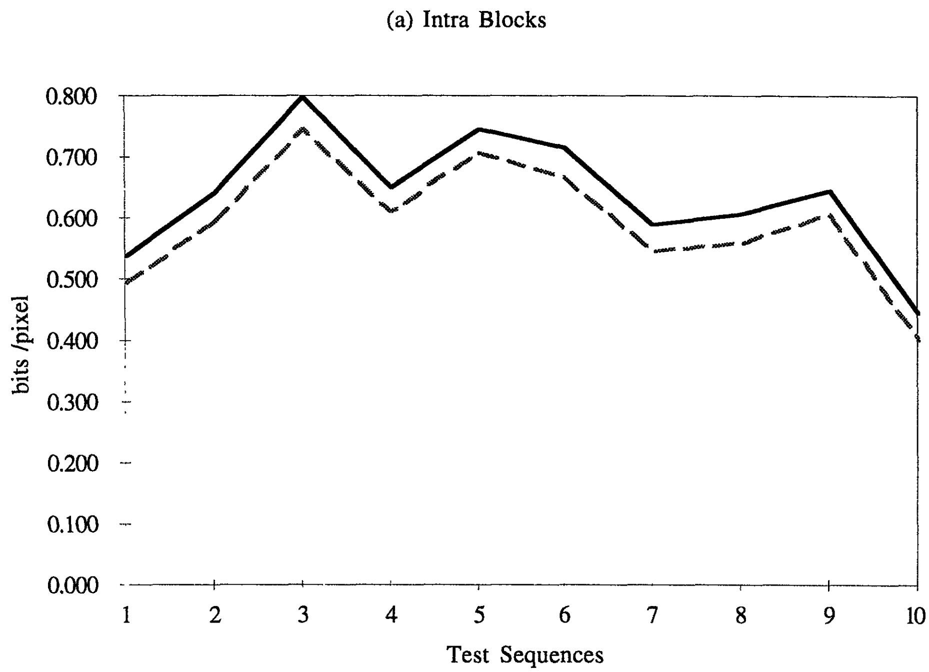
Since the collection of statistics used for experiments yields an insignificant difference in performance using 382 codebooks over 254, the remaining experiments combine the statistics of the P-and B-frames.

Table 5-2 Joint PDE vs. MPEG-2: Preliminary Experiments

			football (1)	beer_truck (2)	tulipz (3)	tulipstxt (4)	picnic (5)	girl (6)	mile471 (7)	zoom_sign (8)	toytable (9)	raft (10)	average
Source (Hz)			60	60	60	60	60	60	60	60	60	24	
Target bitrate (bits/pixel)			0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	
382 codebooks	Intra	MPEG-2 (bits/pixel)	0.396	0.473	0.593	0.452	0.545	0.516	0.433	0.452	0.464	0.302	0.463
		PDE (bits/pixel)	0.354	0.423	0.542	0.412	0.504	0.469	0.386	0.405	0.422	0.258	0.418
		improvement (%)	10.6	10.6	8.6	8.8	7.5	9.1	10.9	10.4	9.1	14.6	10.0
	Predicted	MPEG-2 (bits/pixel)	0.190	0.211	0.188	0.220	0.191	0.158	0.178	0.200	0.182	0.172	0.189
		PDE (bits/pixel)	0.142	0.164	0.176	0.204	0.176	0.146	0.134	0.144	0.140	0.152	0.158
		improvement (%)	25.3	22.3	6.4	7.3	7.9	7.6	24.7	28.0	23.1	11.4	16.4
	Bi-directional	MPEG-2 (bits/pixel)	0.043	0.060	0.069	0.122	0.050	0.072	0.051	0.074	0.063	0.036	0.064
		PDE (bits/pixel)	0.029	0.039	0.059	0.110	0.044	0.064	0.032	0.044	0.048	0.029	0.050
		improvement (%)	33.7	35.0	14.5	9.8	12.0	11.1	37.3	40.5	23.8	20.0	23.8
	Total	MPEG-2 (bits/pixel)	0.199	0.236	0.263	0.301	0.246	0.241	0.216	0.240	0.227	0.187	0.236
		PDE (bits/pixel)	0.160	0.189	0.238	0.275	0.224	0.217	0.172	0.183	0.190	0.160	0.201
		improvement (%)	19.8	19.9	9.5	8.6	8.9	10.0	20.4	23.8	16.3	14.6	15.2
254 codebooks	Inter	MPEG-2 (bits/pixel)	0.082	0.101	0.101	0.146	0.086	0.096	0.084	0.107	0.095	0.063	0.096
		PDE (bits/pixel)	0.058	0.073	0.091	0.135	0.078	0.087	0.057	0.071	0.073	0.056	0.078
		improvement (%)	29.3	27.7	9.9	7.5	9.3	9.4	32.1	33.6	23.2	11.7	19.4
	Total	MPEG-2 (bits/pixel)	0.199	0.236	0.263	0.301	0.246	0.241	0.216	0.240	0.227	0.187	0.236
		PDE (bits/pixel)	0.159	0.189	0.238	0.277	0.225	0.218	0.172	0.182	0.189	0.162	0.201
		improvement (%)	20.2	19.9	9.5	8.0	8.5	9.5	20.4	24.2	16.7	13.6	15.1

TOTALS - Includes all bits needed to encode sequence

			football (1)	beer_truck (2)	tulipz (3)	tulipstxt (4)	picnic (5)	girl (6)	mile471 (7)	zoom_sign (8)	toytable (9)	raft (10)	
382 codebooks	Intra	MPEG-2 (bits/pixel)	0.538	0.643	0.798	0.652	0.749	0.716	0.593	0.609	0.650	0.447	0.640
		PDE (bits/pixel)	0.497	0.594	0.748	0.613	0.708	0.669	0.547	0.563	0.609	0.404	0.595
		improvement (%)	7.6	7.6	6.3	6.0	5.5	6.6	7.8	7.6	6.3	9.6	7.1
	Predicted	MPEG-2 (bits/pixel)	0.212	0.240	0.223	0.265	0.226	0.186	0.200	0.224	0.207	0.202	0.219
		PDE (bits/pixel)	0.164	0.193	0.21	0.249	0.211	0.175	0.155	0.168	0.166	0.183	0.187
		improvement (%)	22.9	19.6	5.8	6.0	6.6	5.9	22.5	25.0	19.8	9.6	14.4
	Bi-directional	MPEG-2 (bits/pixel)	0.046	0.066	0.079	0.145	0.058	0.085	0.056	0.080	0.071	0.042	0.073
		PDE (bits/pixel)	0.032	0.045	0.069	0.133	0.044	0.076	0.037	0.049	0.057	0.035	0.058
		improvement (%)	30.4	31.8	12.7	8.3	24.1	10.6	33.9	38.8	19.7	17.6	22.8
	Total	MPEG-2 (bits/pixel)	0.348	0.362	0.395	0.461	0.383	0.381	0.357	0.355	0.372	0.350	0.376
		PDE (bits/pixel)	0.308	0.315	0.37	0.435	0.361	0.358	0.312	0.297	0.334	0.323	0.341
		improvement (%)	11.5	13.0	6.3	5.6	5.7	6.0	12.6	16.3	10.2	7.8	9.5
254 codebooks	Inter	MPEG-2 (bits/pixel)	0.090	0.113	0.118	0.174	0.102	0.112	0.092	0.119	0.108	0.074	0.110
		PDE (bits/pixel)	0.066	0.085	0.108	0.164	0.093	0.103	0.066	0.081	0.073	0.066	0.091
		improvement (%)	26.7	24.8	8.5	5.7	8.8	8.0	28.3	31.9	32.4	10.4	18.6
	Total	MPEG-2 (bits/pixel)	0.348	0.362	0.395	0.461	0.383	0.381	0.357	0.355	0.372	0.350	0.376
		PDE (bits/pixel)	0.308	0.314	0.371	0.437	0.362	0.359	0.312	0.296	0.334	0.325	0.342
		improvement (%)	11.5	13.3	6.1	5.2	5.5	5.8	12.6	16.6	10.2	7.3	9.4



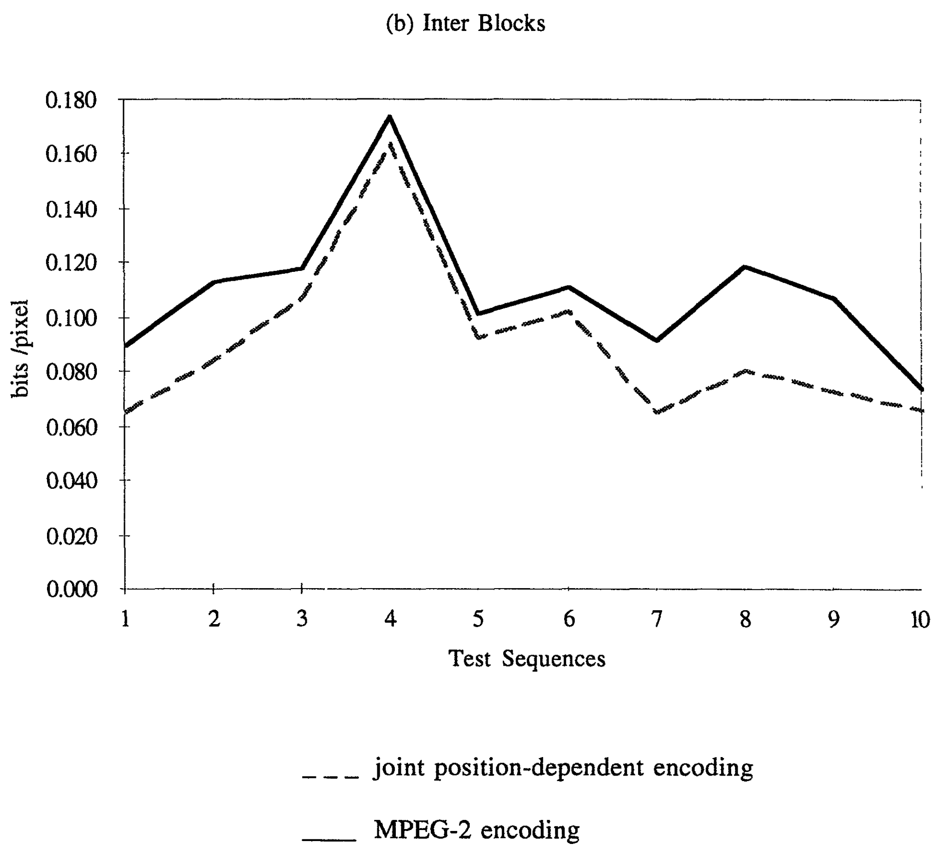


Figure 5-2 Joint PDE vs. MPEG-2 coding (254 codebooks): (a) Intra Blocks, (b) Inter Blocks.

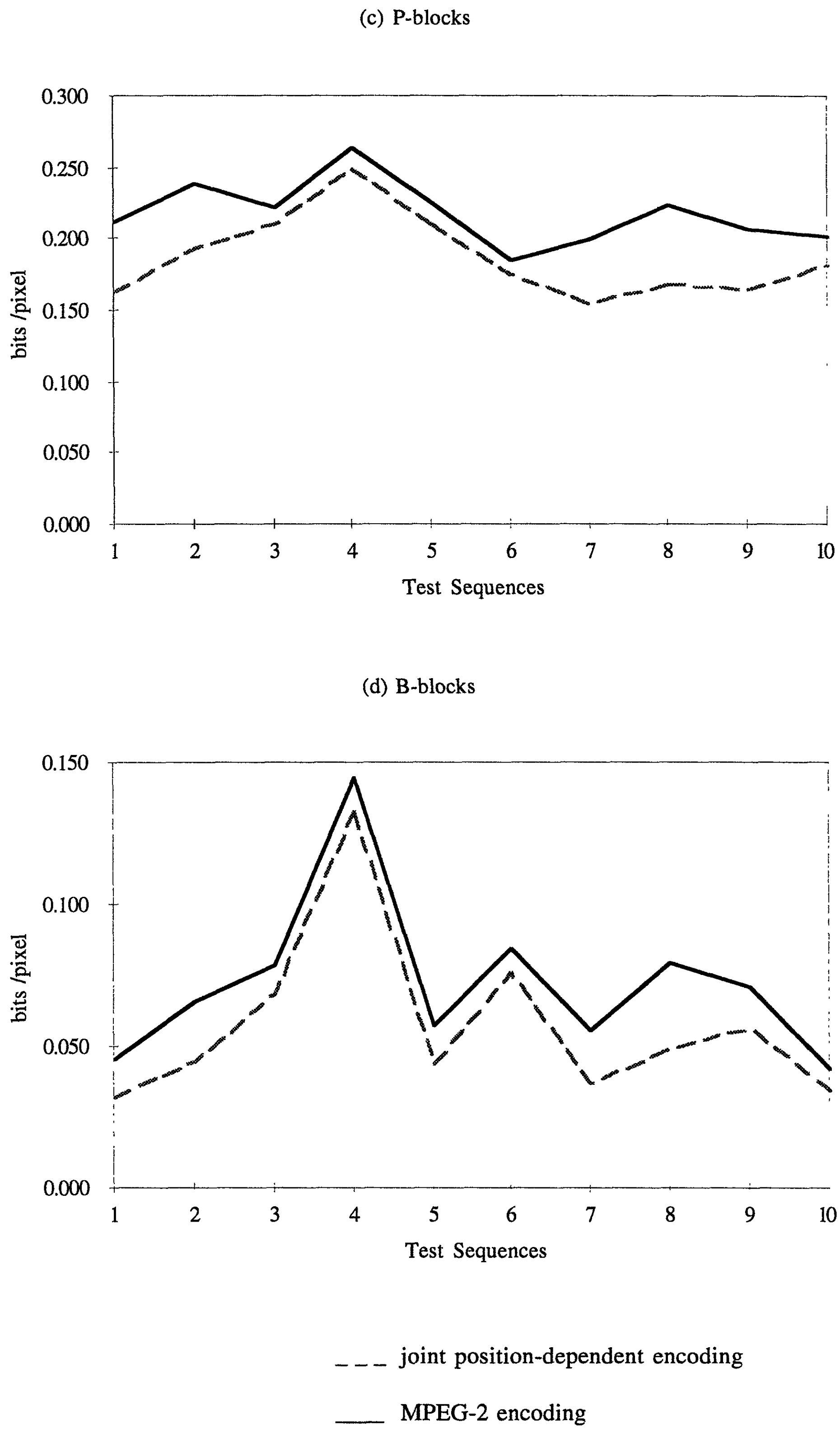
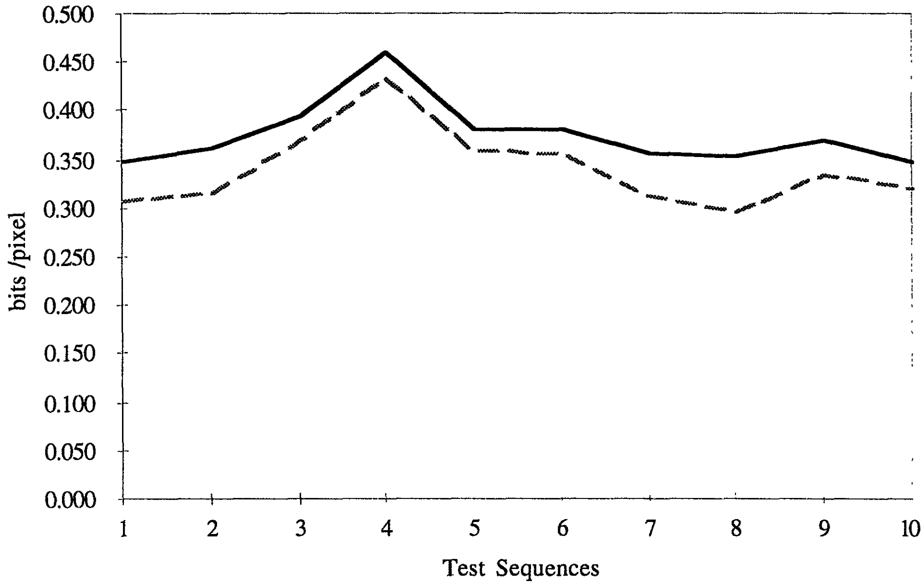
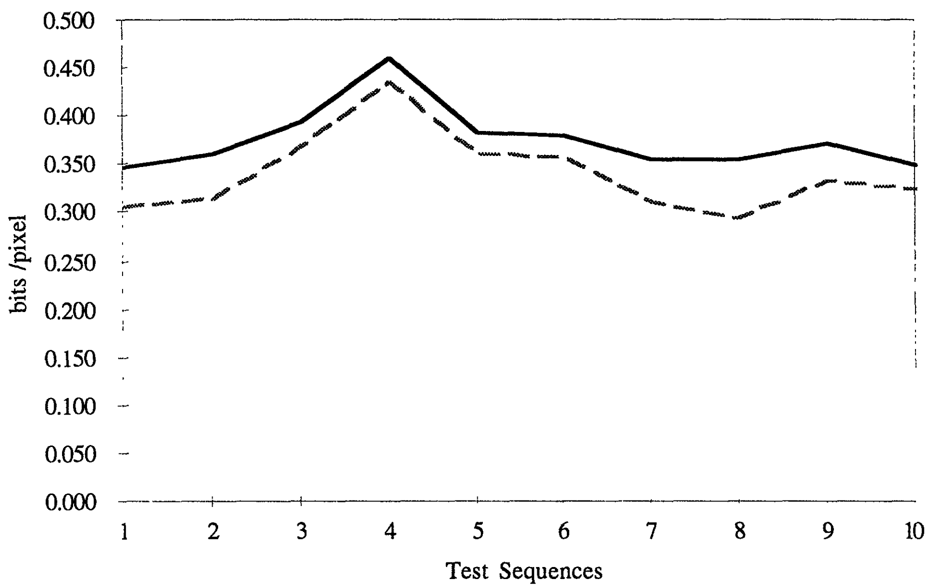


Figure 5-2 Joint PDE vs. MPEG-2 coding (384 codebooks): (c) P-Blocks, (d) B-Blocks.

(e) 382 Codebooks



(f) 254 Codebooks



_ _ _ joint position-dependent encoding

____ MPEG-2 encoding

Figure 5-2 Joint PDE vs. MPEG-2 coding: (e) Overall Performance, (f) Overall Performance.

5.3 Reducing the Number of Codebooks

In the previous section, each coefficient has its own codebook. Considering memory requirements, a scheme with many codebooks may be highly impractical. Therefore, it is imperative to decrease the number of codebooks by allowing coefficients share codebooks as discussed in section 4.1.

A decrease in the number of codebooks decreases the coding benefits of joint PDE. However, the number of codebooks can be reduced significantly without a significant degradation in performance. As a result, it is still possible to achieve most of the performance gain with a significant reduction in codebooks.

The results presented here use a total of 31 codebooks: 8 Intra Y, 3 Intra UV, 13 Inter Y, and 7 Inter UV. The exact pattern of the codebook selection is included in Appendix A. More codebooks have been assigned to inter blocks for two reasons. First, the inter blocks occupy more of the bit rate. Second, joint PDE performed better on average for the inter blocks compared to intra blocks from the preliminary results. Also, each intra block has one less codebook since intra DC coefficients are not coded using PDE while inter DC coefficients have their own codebooks.

Two separate experiments are performed. Experiment-1 corresponds to each test sequence having its own codebook where the test sequence statistics are not included in training. Experiment-2 corresponds to testing all 14 sequences to one codebook which is trained from a weighted average of all 14 test sequences. Since the statistics of the test sequence are included in training, experiment-2 results should provide slightly better results.

The results are summarized in Table 5-3 and Figures 5-3 (a)-(c). Reducing the total number of codebooks to 31, joint PDE still achieves an average decrease of 8 % in the bit rate compared to a 9.4 % decrease using 254 codebooks. The average decrease in bit rate for intra blocks is 7.1 % and the average decrease for inter blocks is 12.9 %.

The average percentage of improvement by including the test sequence statistics in training is 0.6 %. This shows that joint PDE performance is not greatly affected by using sequences outside the training set. These experiments demonstrate that joint PDE could be very useful in an actual encoding environment.

Comparing the results of this section to the 254 codebook case, the degradation in performance of going from 254 codebooks to 31 is approximately 1.4%. However this difference is not a fair comparison since the statistics in the previous case were weighted and estimated differently. In any case, the difference still illustrates that the degradation is small compared to the relaxed memory requirements.

Table 5-3 Joint PDE vs. MPEG-2: 31 Total Codebooks

Source (Hz)			60	60	60	60	60	60	60	60	60	60	24	24	24	24	
Target bitrate (bits/pixel)			0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	average

Sequence name (number)			football (1)	beer truck (2)	tulipz (3)	tulipstxt (4)	picnic (5)	girl (6)	mile471 (7)	zoom sign (8)	toytable (9)	raft (10)	traffic (11)	mall (12)	untouchables (13)	marcie (14)	
31 codebooks	Intra	MPEG-2 (bits/pixel)	0.396	0.473	0.593	0.452	0.545	0.516	0.433	0.452	0.464	0.302	0.369	0.393	0.401	0.309	0.469
		PDE - Exp-1 (bits/pixel)	0.347	0.423	0.564	0.412	0.515	0.481	0.382	0.400	0.427	0.262	0.326	0.358	0.354	0.278	0.425
		improvement (%)	12.4	10.6	4.8	8.9	5.5	6.7	11.7	11.6	8.0	13.3	11.6	8.9	11.7	10.0	10.4
	Inter	MPEG-2 (bits/pixel)	0.082	0.101	0.101	0.146	0.086	0.096	0.084	0.107	0.095	0.063	0.069	0.076	0.088	0.080	0.098
		PDE - Exp-1 (bits/pixel)	0.057	0.074	0.092	0.141	0.081	0.091	0.058	0.071	0.074	0.057	0.067	0.076	0.083	0.076	0.084
		improvement (%)	30.1	26.7	9.1	3.6	5.8	5.6	31.3	33.3	21.8	9.8	3.6	-0.5	5.7	5.0	14.7
	Total	MPEG-2 (bits/pixel)	0.199	0.236	0.263	0.301	0.246	0.241	0.216	0.240	0.227	0.187	0.198	0.202	0.211	0.206	0.244
		PDE - Exp-1 (bits/pixel)	0.156	0.190	0.245	0.284	0.231	0.226	0.171	0.182	0.192	0.164	0.181	0.193	0.193	0.189	0.215
		improvement (%)	21.5	19.5	7.0	5.6	6.1	6.2	20.9	24.3	15.2	12.1	8.4	4.7	8.6	8.3	13.0

TOTALS - Includes all bits needed to encode sequence

Sequence name (number)			football (1)	beer truck (2)	tulipz (3)	tulipstxt (4)	picnic (5)	girl (6)	mile471 (7)	zoom sign (8)	toytable (9)	raft (10)	traffic (11)	mall (12)	untouchables (13)	marcie (14)	average
31 codebooks	Intra	MPEG-2 (bits/pixel)	0.538	0.643	0.798	0.652	0.749	0.716	0.593	0.609	0.650	0.447	0.534	0.568	0.549	0.447	0.653
		PDE - Exp-1 (bits/pixel)	0.491	0.594	0.770	0.613	0.720	0.683	0.543	0.567	0.613	0.407	0.493	0.534	0.502	0.424	0.612
		improvement (%)	8.7	7.6	3.5	6.0	3.9	4.7	8.4	6.9	5.6	8.9	7.8	6.1	8.5	5.1	7.1
	Inter	MPEG-2 (bits/pixel)	0.090	0.113	0.118	0.174	0.102	0.112	0.092	0.119	0.108	0.074	0.082	0.092	0.104	0.095	0.113
		PDE - Exp-1 (bits/pixel)	0.066	0.086	0.109	0.169	0.096	0.107	0.067	0.083	0.088	0.068	0.080	0.092	0.099	0.090	0.100
		improvement (%)	26.7	23.9	7.9	2.7	5.4	4.4	26.8	30.3	19.0	8.8	2.9	-0.4	4.5	4.6	12.9
	Total	MPEG-2 (bits/pixel)	0.348	0.362	0.395	0.461	0.383	0.381	0.357	0.355	0.372	0.350	0.339	0.360	0.359	0.356	0.398
		PDE - Exp-1 (bits/pixel)	0.305	0.316	0.375	0.444	0.368	0.367	0.312	0.296	0.337	0.327	0.322	0.351	0.341	0.339	0.369
		PDE- Exp-2 (bits/pixel)	0.303	0.315	0.372	0.441	0.366	0.364	0.311	0.295	0.336	0.325	0.321	0.350	0.340	0.337	0.367
		Exp-1/Exp-2 (% change)	0.7	0.3	0.9	0.7	0.6	0.9	0.3	0.4	0.3	0.7	0.4	0.3	0.4	0.5	0.6
		improvement (%)	12.3	12.8	5.0	3.7	3.9	3.6	12.7	16.6	9.4	6.6	4.9	2.5	5.0	4.8	8.0

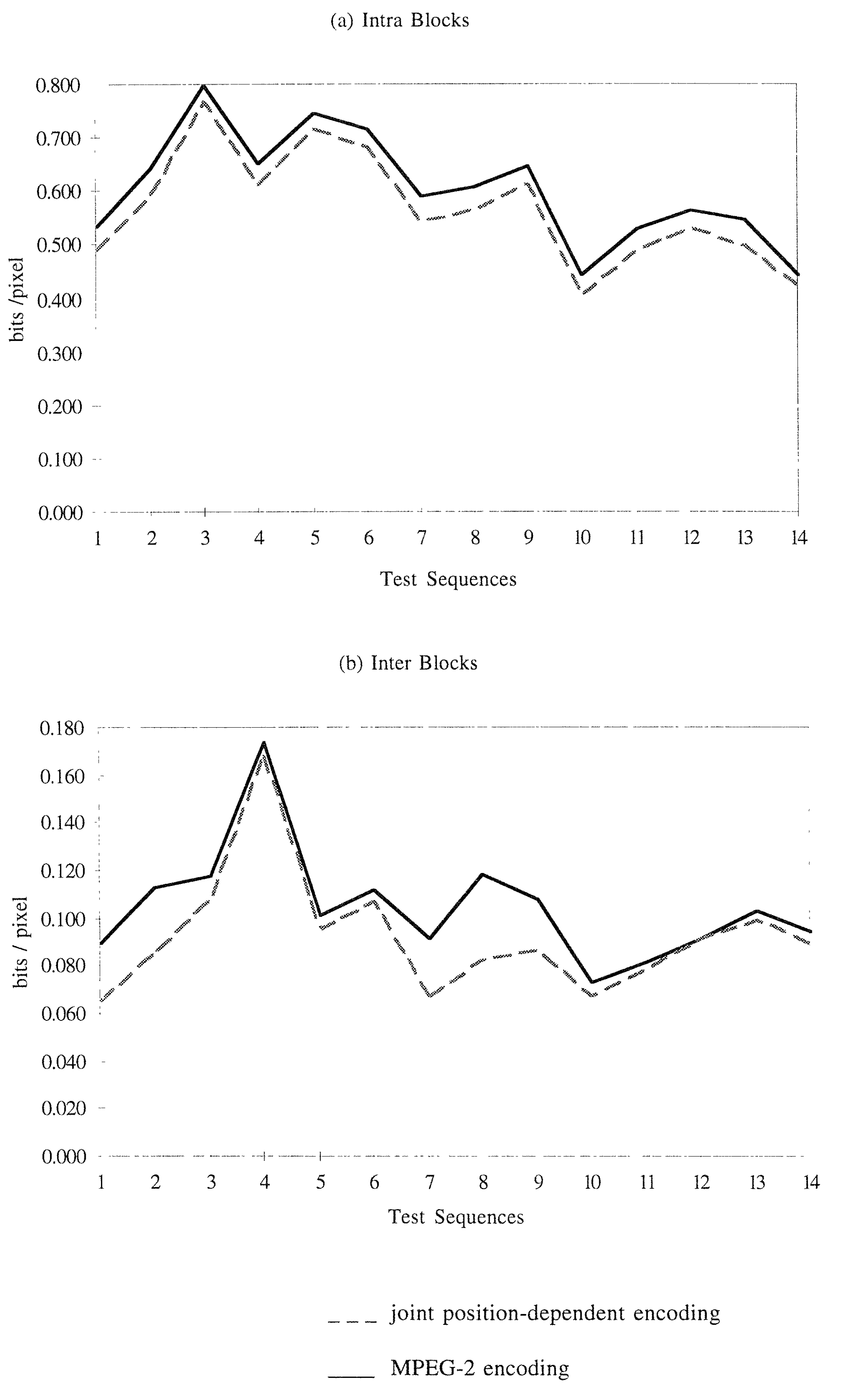


Figure 5-3 Joint PDE vs. MPEG-2 (31 total codebooks): (a) Intra Blocks, (b) Inter Blocks.

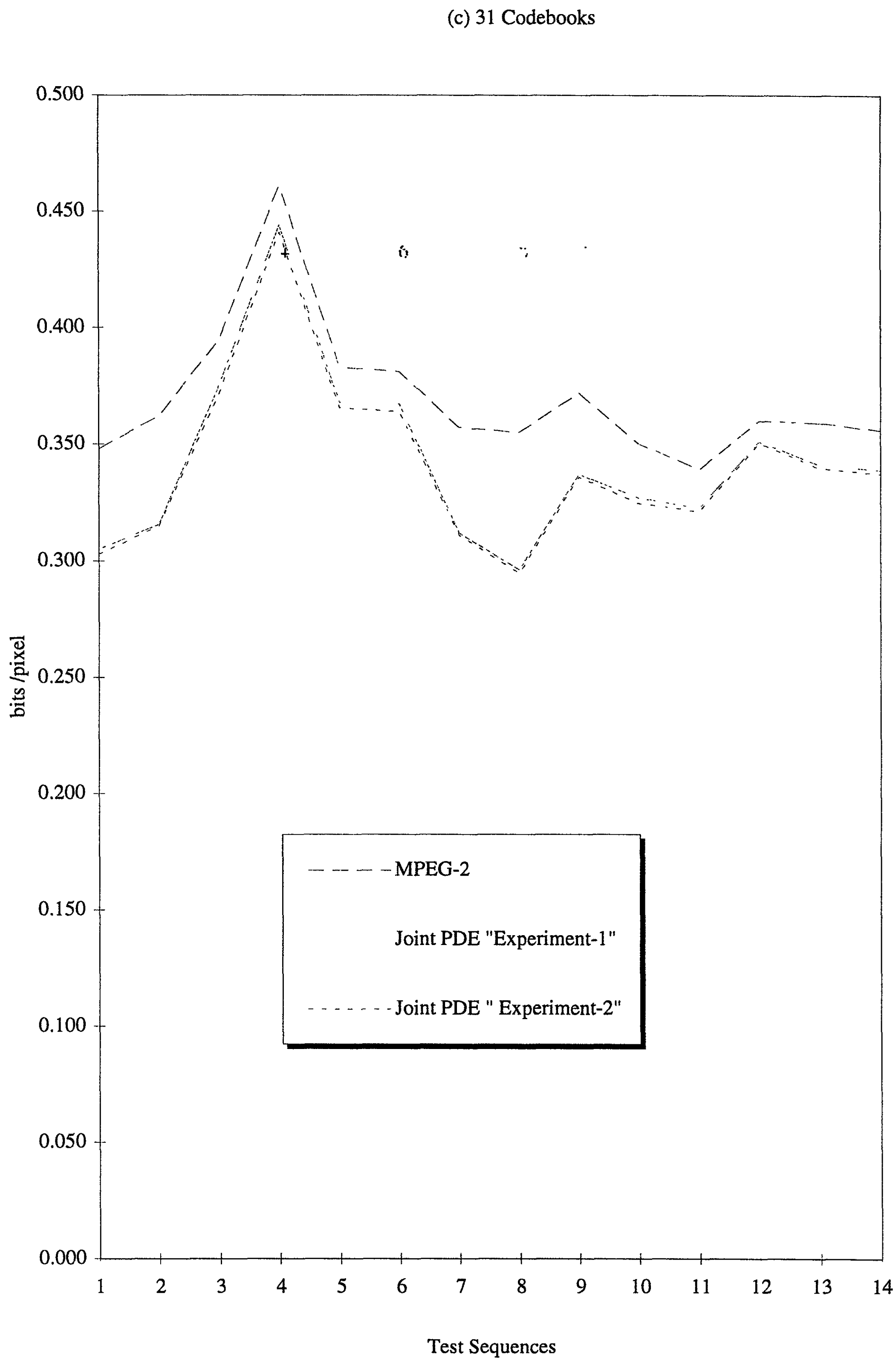


Figure 5-3 Joint PDE vs. MPEG-2 coding (31 total codebooks): (c) Overall performance.

5.4 Escape Codes Limiting the Codeword Length

In the previous section, we were able to decrease the bit rate by an average of 8.0 % using a total of 31 codebooks. However, the codebooks in section 5.3 are still impractical since the number of entries in each codebook is extremely large due to all possible runlength-amplitude events. Therefore, this experiment introduces escape codes to reduce the number of entries in each codebook. The total number of codebooks is fixed at 31 with the same allocation shown in Appendix A in order to make a fair comparison with the results of the previous section.

Escape codes that limit the codeword length to 13 are tested. Therefore all events that have a codeword length greater than 13 are escape coded. The results are summarized in Table 5-4 and Figures 5-4 (a)-(c). The overall percentage decrease in the bit rate is reduced from 8.0 % in the previous section down to only 7.6 %. By decreasing the average number of entries from tens of thousands to an average of 114 entries, there is only a 0.4 % degradation. The average decrease in bit rate for intra blocks is 6.7 % and the average decrease for inter blocks is 12.4 %. An example of one joint PDE codebook is provided in Appendix B.

Table 5-4 Joint PDE vs. MPEG-2: Escape Codes Limiting Codeword Length

Source (Hz)		60	60	60	60	60	60	60	60	60	60	24	24	24	24
Target bitrate (bits/pixel)		0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	average

Sequence name (number)			football (1)	beer truck (2)	tulips (3)	tulipstxt (4)	picnic (5)	girl (6)	mile471 (7)	zoom_sign (8)	toytable (9)	raft (10)	traffic (11)	mall (12)	untouchables (13)	marcie (14)	
31 codebooks	Intra	MPEG-2 (bits/pixel)	0.396	0.473	0.593	0.452	0.545	0.516	0.433	0.452	0.464	0.302	0.369	0.393	0.401	0.309	0.469
		PDE (bits/pixel)	0.353	0.426	0.569	0.416	0.522	0.488	0.384	0.401	0.423	0.263	0.327	0.359	0.359	0.280	0.428
		improvement (%)	11.0	10.0	4.1	8.0	4.3	5.4	11.3	11.2	8.9	12.8	11.2	8.6	10.4	9.4	9.7
	Inter	MPEG-2 (bits/pixel)	0.082	0.101	0.101	0.146	0.086	0.096	0.084	0.107	0.095	0.063	0.069	0.076	0.088	0.080	0.098
		PDE (bits/pixel)	0.057	0.074	0.092	0.141	0.081	0.093	0.058	0.072	0.075	0.057	0.067	0.076	0.084	0.076	0.085
		improvement (%)	30.1	26.8	8.6	3.1	5.5	3.4	31.1	33.2	21.5	9.6	3.0	-0.5	5.1	5.1	14.3
	Total	MPEG-2 (bits/pixel)	0.199	0.236	0.263	0.301	0.246	0.241	0.216	0.240	0.227	0.187	0.198	0.202	0.211	0.206	0.244
		PDE (bits/pixel)	0.158	0.191	0.246	0.286	0.234	0.230	0.172	0.182	0.193	0.165	0.182	0.193	0.195	0.190	0.217
		improvement (%)	20.7	19.1	6.3	5.0	5.1	4.4	20.6	24.1	14.8	11.7	8.0	4.5	7.7	7.7	12.3

TOTALS - Includes all bits needed to encode sequence

Sequence name (number)			football (1)	beer truck (2)	tulips (3)	tulipstxt (4)	picnic (5)	girl (6)	mile471 (7)	zoom_sign (8)	toytable (9)	raft (10)	traffic (11)	mall (12)	untouchables (13)	marcie (14)	average
31 codebooks	Intra	MPEG-2 (bits/pixel)	0.538	0.643	0.798	0.652	0.749	0.716	0.593	0.609	0.650	0.447	0.534	0.568	0.549	0.447	0.653
		PDE (bits/pixel)	0.496	0.597	0.774	0.617	0.726	0.690	0.545	0.560	0.616	0.409	0.494	0.535	0.507	0.419	0.614
		improvement (%)	7.8	7.2	3.0	5.4	3.0	3.7	8.1	8.1	5.2	8.6	7.5	5.8	7.6	6.2	6.7
	Inter	MPEG-2 (bits/pixel)	0.090	0.113	0.118	0.174	0.102	0.112	0.092	0.119	0.108	0.074	0.082	0.092	0.104	0.095	0.113
		PDE (bits/pixel)	0.067	0.087	0.109	0.170	0.097	0.109	0.068	0.083	0.088	0.068	0.080	0.092	0.100	0.090	0.101
		improvement (%)	26.1	23.3	7.2	2.2	4.9	2.4	26.6	30.3	18.7	8.5	2.4	-0.1	4.0	4.3	12.4
	Total	MPEG-2 (bits/pixel)	0.348	0.362	0.395	0.461	0.383	0.381	0.357	0.355	0.372	0.350	0.339	0.360	0.359	0.356	0.398
		PDE (bits/pixel)	0.306	0.317	0.379	0.446	0.370	0.371	0.313	0.297	0.338	0.328	0.323	0.352	0.343	0.340	0.371
		improvement (%)	11.9	12.5	4.1	3.3	3.3	2.6	12.5	16.5	9.1	6.4	4.7	2.4	4.5	4.4	7.6

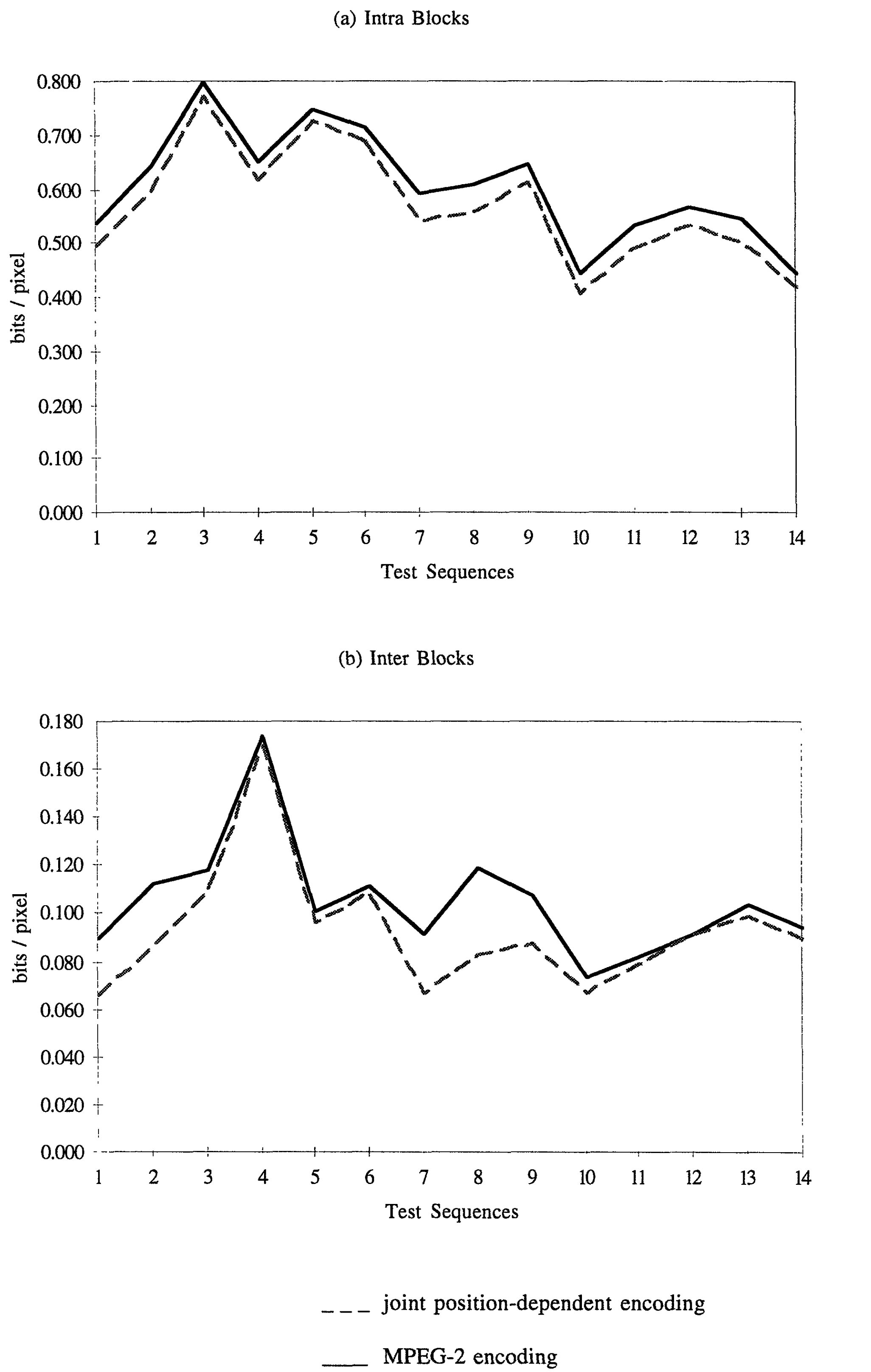


Figure 5-4 Escape codes limiting codeword length to 13: (a) Intra Blocks, (b) Inter Blocks.

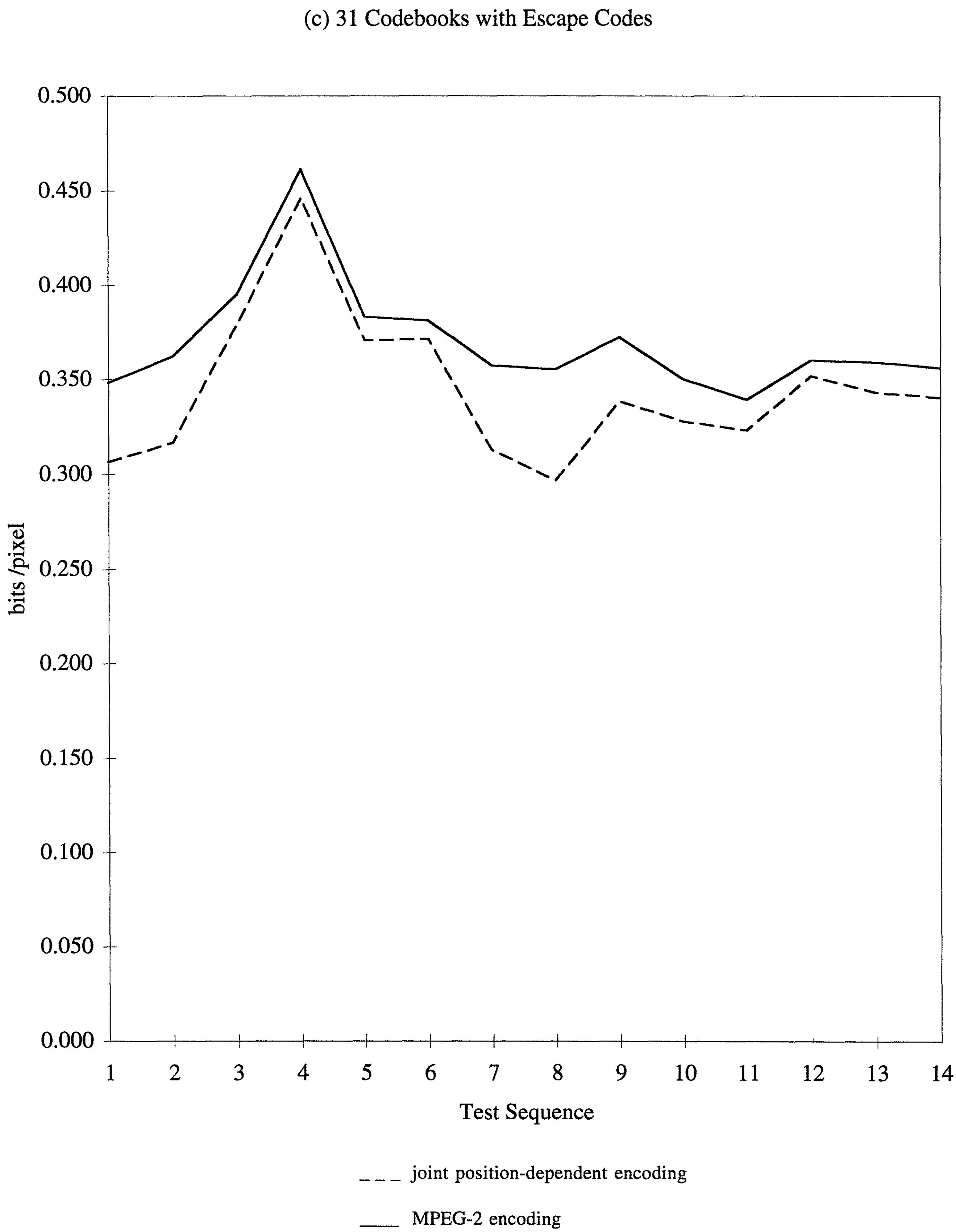


Figure 5-4 Escape codes limiting codeword length to 13: (c) Overall Performance.

5.5 Varying Codeword Length for Escape Codes

The experiment performed here investigates the performance of PDE with varying codeword length. This test is performed only for test sequence 7. The same codebook allocation from the previous sections is used while the codeword length varies from 13 down to 7. Figure 5-5 summarizes the results. MPEG-2 required 0.357 bits per pixel to encode this sequence. As expected, PDE performance decreases as the codeword length limit decreases. The figure also includes the average number of entries (codewords) of the 31 codebooks. For example, the codebooks have an average number of 117 entries when limiting the codeword length to 13. Joint PDE still outperforms MPEG-2 coding when limiting the codeword length to 8 which has an average of 21 entries. However, when the codeword length is limited to 7, MPEG-2 outperforms the joint PDE scheme with less memory requirements (2 codebooks with 114 entries each (*228 total entries*) compared to 31 codebooks with 19 entries(*600 total entries*)).

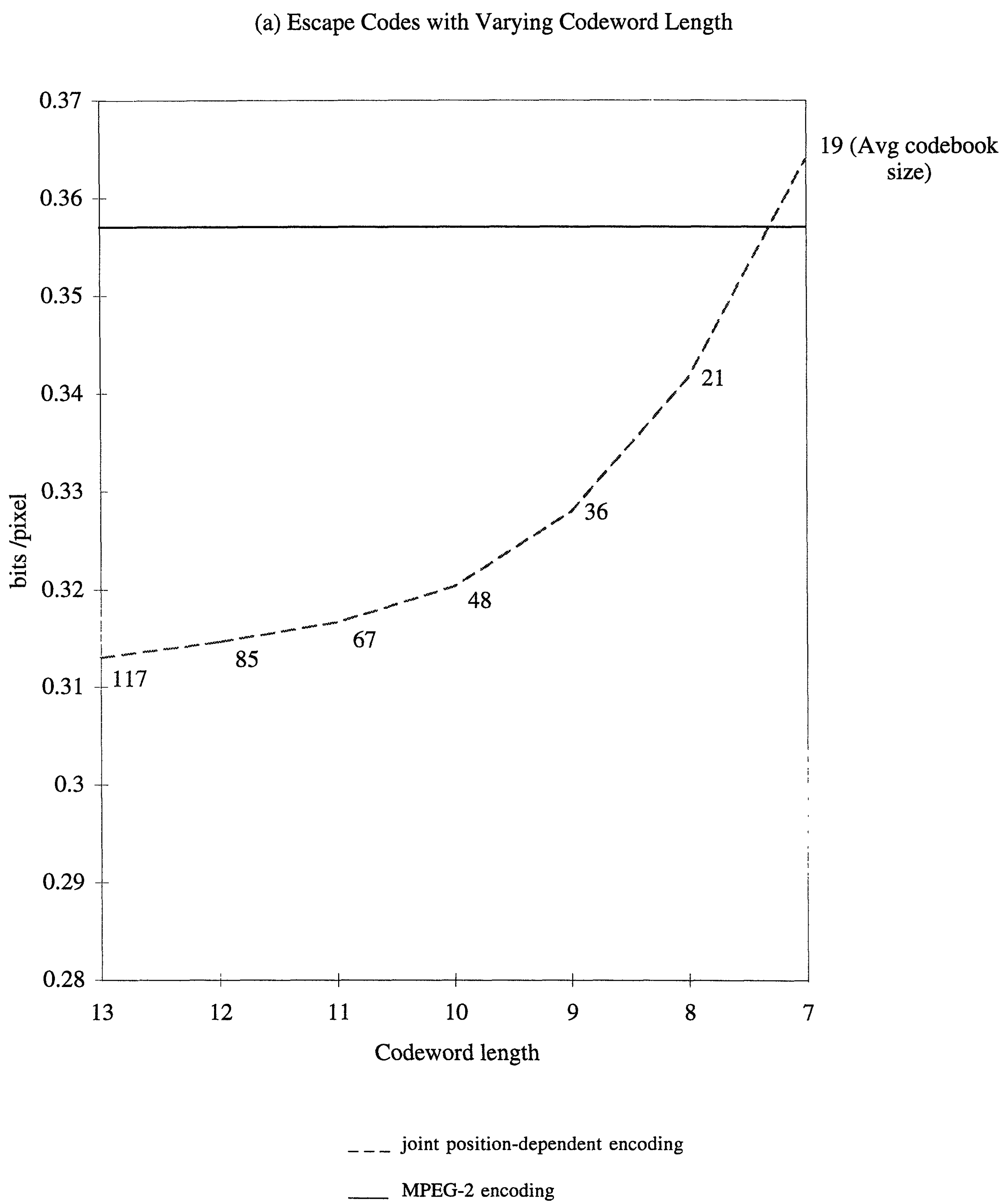


Figure 5-5 Varying codeword length for Escape Codes (Test Sequence 7): (a) Overall Performance.

CHAPTER 6

Concluding Remarks

Comments on Performance

Results indicate that joint PDE performs best for inter blocks. Since results were collected from one GOP plus an additional I-frame, we would expect performance to increase when introducing more P- and B-frames. If results were collected from 2 full GOP's, only P- and B-frames would be introduced into the calculations which would increase performance. Therefore, in a typical video sequence consisting of thousands of frames, it is fair to assume that performance would increase on average. Also, it was evident that the MPEG-2 encoder had trouble compressing many sequences down to 0.34 bits per pixel. In applications requiring this much compression, the MPEG-2 frame structure that was used for testing may not be the most appropriate. Instead, I-frames may be inserted every 15 frames (N=15) which would slightly increase performance.

From the figures presented in Chapter 5, it is evident that the variance of the joint PDE performance for intra blocks is very small. This implies that the intra statistics used for training match very closely the intra coefficients for all the test sequences. This observation demonstrates that most video has predominately low frequency content. On the other hand, it is clear that the variance of the joint PDE performance for inter blocks is very large. The test sequences with the highest overall performance involve sequences where the joint PDE performed above average for inter blocks. The cases with the worse overall performance involve sequences where the joint PDE performed below average for inter blocks. This implies that the collection of inter statistics used for training provided a good representation for some of the sequences but not for others. Therefore, when collecting statistics, it is important to match statistics as closely as possible.

The large variance in performance illustrates that the sequences used for training give significant differences in prediction error statistics compared to the test sequences in many cases. This could be due to the fact that some sequences contain more noise than others adding high frequency content thus resulting in different error statistics. For example, test sequences 5 and 6 involve zoom and pan motion which were synthetically generated at MIT. These sequences contain less noise compared to some of the real video sequences. In addition, the 60 Hz source sequences may have different error statistics compared to the 24 Hz source sequences which may contribute to the differences in performance. Noise variations may be one possible reason for a large discrepancy in joint PDE performance of the inter blocks. In a real implementation, it may be advantageous to collect statistics from the same source.

Applications of Joint Position-Dependent Encoding

Results show that it is possible to reduce the bit rate by more than 7 % with 31 total codebooks. This much reduction in bit rate could be beneficial for many potential applications. In many communication scenarios, it is desirable to keep the bit rate constant and utilize the entire bandwidth available for maximum video quality. Joint PDE could be beneficial for very low bit rate applications such as video conferencing as well as high bit rate applications such as digital television. For example, the extra savings in bit rate could be used to send additional bits to increase video quality. In a digital television application operating at 20 Mbps, a savings of over 7 % may not increase video quality substantially. However, in a video conferencing application operating at 128 Kbits per second, a 7 % savings in bit rate would provide a substantial improvement in video quality. The extra bits could also be used to transmit other services such as news and stock price updates.

There are also many applications where it is desirable to reduce the bandwidth requirements. These scenarios involve communication as well as storage applications. For example, when storing video, it is advantageous to reduce the bits as much as possible to reduce the storage requirements. A scheme that reduces the bit rate by more than 1 Mbps would obviously be beneficial for storage. For example, to store a one hour video sequence, joint PDE would reduce the bits required for storage by more than 3.6 Gbits. In the future, an HDTV format with a resolution of 1080 x 1920 and progressively scanned at a frame rate of 60 Hz will be introduced. In order to send this large amount of information, enhancement data will need to be sent within the 20 MHz channel. Therefore, it is advantageous to reduce the extra bits for enhancement as much as possible. Joint PDE could possibly play a role in coding the enhancement data. In addition, the results obtained using joint PDE are immediately extendible to image compression. Therefore, the joint PDE scheme could be also be used in still frame compression standards such as JPEG.

Future Work

In a real time implementation of joint PDE, statistics will be collected from thousands of frames from many representative test sequences. The results show that it is possible to obtain a decrease in bit rate by more than 16 % if the inter statistics used for training match closely the inter statistics of the playing sequence. Therefore, it would be worthwhile to develop a real time implementation of the joint PDE scheme to see the actual improvements. In this case, it may be advantageous to separate the statistics of the P- and B-frames.

It would also be advantageous to test the joint PDE performance with the video conferencing standards that have been developed. It would be interesting to see the results using a video compression system for video conferencing which is slightly different from MPEG-2. Since the major problem with video conferencing is delay, B-frames are not used in these standards. In addition, many available video conferencing systems are interlaced scanned where all experiments performed in this thesis involve a progressive scan. If the bit rate is reduced by more than 7 %, this could significantly increase the video quality of the video conferencing system.

Appendix A

Distribution of Codebooks

This appendix contains the exact distribution of codebooks used in sections 5.3, 5.4 and 5.5.

k2

8	8	8	8	8	8	8	8
7	8	8	8	8	8	8	8
7	7	8	8	8	8	8	8
5	7	7	8	8	8	8	8
5	6	6	7	8	8	8	8
3	5	6	6	7	8	8	8
2	4	5	6	6	7	8	8
	1	5	5	6	6	7	7

k1

(a) Intra Y

k2

11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11
10	11	11	11	11	11	11	11
	9	11	11	11	11	11	11

k1

(b) Intra UV

Figure A-1 Distribution of Joint PDE codebooks (a) Intra Y (b) Intra UV.

k2

23	23	24	24	24	24	24	24
21	22	23	24	24	24	24	24
20	21	22	23	24	24	24	24
18	20	21	22	23	24	24	24
18	19	20	21	22	23	23	24
15	18	19	20	21	22	23	23
14	16	17	19	20	21	22	23
12	13	17	17	19	20	22	22

k1

(c) Inter Y

k2

31	31	31	31	31	31	31	31
31	31	31	31	31	31	31	31
31	31	31	31	31	31	31	31
30	31	31	31	31	31	31	31
29	30	31	31	31	31	31	31
28	29	30	31	31	31	31	31
27	28	29	30	31	31	31	31
25	26	28	29	30	31	31	31

k1

(d) Inter UV

Figure A-1 Distribution of Joint PDE codebooks (a) Inter Y (b) Inter UV.

Appendix B

Sample Codebook

This appendix contains a sample codebook for the joint PDE scheme. The codebook shown in Table B-1 corresponds to codebook 1 (intra Y) defined in Figure A-1. This codebook corresponds to the joint PDE with escape codes introduced to limit the codeword length to 13. Escape codes follow the MPEG-2 format discussed in section 4.2.

Table B-1 Codebook # 1 with escape codes limiting codeword length to 13.

Variable Length Code (NOTE)	run	level
1110	End of Block	
10 s	0	1
011 s	0	2
1100 s	0	3
0010 s	0	4
0101 1 s	0	5
0011 0 s	0	6
0101 01 s	0	7
0011 11 s	0	8
1101 100 s	0	9
0100 111 s	0	10
1111 1110 s	0	11
1101 1011 s	0	12
1101 0001 s	0	13
0100 1000 s	0	14
1111 1101 1 s	0	15
1101 0100 1 s	0	16
1101 0000 0 s	0	17
0011 1011 0 s	0	18
1111 1100 10 s	0	19
1101 1010 10 s	0	20
0101 0000 10 s	0	21
0100 1011 00 s	0	22
0100 1010 00 s	0	23
1111 1111 010 s	0	24
1101 1010 111 s	0	25
1101 1010 010 s	0	26
1101 1010 110 s	0	27
1101 0000 100 s	0	28
0100 1011 101 s	0	29
1111 1111 1000 s	0	30
1111 1100 0011 s	0	31
1111 1101 0010 s	0	32
1111 1100 0010 s	0	33
1101 1010 0011 s	0	34
1101 0000 1011 s	0	35
0100 1010 0111 s	0	36
0011 1011 1100 s	0	37
1111 1101 0001 0 s	0	38
1101 0101 0101 1 s	0	39
0100 1011 1100 0 s	0	40
0011 1011 1111 0 s	0	41

Variable Length Code (NOTE)	run	level
0100 1011 1100 1 s	0	43
000 s	1	1
1111 0 s	1	2
1111 10 s	1	3
0100 01 s	1	4
1101 011 s	1	5
0011 100 s	1	6
1101 0011 s	1	7
0011 1010 s	1	8
1101 0101 1 s	1	9
0100 1010 1 s	1	10
0101 0000 0 s	1	11
1101 0101 00 s	1	12
1101 0100 01 s	1	13
0100 1011 01 s	1	14
1111 1100 000 s	1	15
1101 0000 110 s	1	16
0100 1010 010 s	1	17
0100 1011 100 s	1	18
1101 1010 0001 s	1	19
1111 1111 1001 1 s	1	20
0011 1011 1101 s	1	21
1111 1111 1001 0 s	1	22
1101 1010 0110 0 s	1	23
1101 1010 0110 1 s	1	24
1101 0101 0100 0 s	1	25
1101 0010 0000 0 s	1	26
0100 1011 1110 0 s	1	27
1101 0010 0001 0 s	1	28
0100 1011 1101 1 s	1	29
1101 11 s	2	1
0101 0001 s	2	2
1111 1100 01 s	2	3
1101 0101 011 s	2	4
1101 0000 1010 s	2	5
1101 0010 0001 1 s	2	6
0100 00 s	3	1
0101 0000 11 s	3	2
0011 1011 1111 0 s	3	3
0101 001 s	4	1
1101 0000 111 s	4	2
1101 0101 0101 0 s	4	3
0100 1001 s	5	1
1111 1100 11 s	6	1
1111 1101 01 s	7	1

Variable Length Code (NOTE)	run	level
1101 0010 1 s	8	1
1101 0010 01 s	9	1
0011 1011 10 s	10	1
1101 1010 0000 s	11	1
1111 1111 011 s	12	1
1111 1111 00 s	13	1
1111 1111 11 s	14	1
1101 0010 001 s	15	1
0100 1010 0110 s	16	1
1111 1101 0011 1 s	17	1
1101 0010 0000 1 s	18	1
1101 1010 0010 s	19	1
0100 1011 1111 s	20	1
0100 1011 1110 1 s	21	1
1111 1101 0001 1 s	23	1
1101 1010 0111 s	24	1
1111 1101 0000 s	25	1
1111 1111 101 s	26	1
1101 0100 00 s	27	1
1101 0101 0100 1 s	29	1
1111 1101 0011 0 s	30	1
0011 1011 1111 1 s	34	1
0100 1011 1101 0 s	39	1
0100 110	Escape	
NOTE - The last bit 's' denotes the sign of the level,'0' for positive '1' for negative.		

References

[1] J.S. Lim, *Two-Dimensional Signal and Image Processing*, Englewood Cliffs, New Jersey: Prentice Hall, 1990.

[2] J. G. Apostolopoulos and J.S. Lim, "Video Compression for Digital ATV Systems," in *Motion Analysis and Image Sequence Processing*, edited by M.I. Sezan and R.L. Lagendijk. Norwell, Massachusetts: Kluwer Academic Publishers, 1993.

[3] A. Netravali and B. Haskell, *Digital Pictures, Representation and Compression.* New York: Plenum Press, 1988.

[4] D. J. DeGall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, Vol. 34, pp.46-58, April 1991.

[5] A. Pfajfer, "Position-Dependent Encoding," Masters Thesis, M.I.T., May 1994.

[6] ITU-T Rec. H.262, *Generic Coding of Moving Pictures and Associated Audio*, Standard draft, March 1994.

[7] J. G. Apostolopoulos, A. Pfajfer, H. M. Jung, and J. S. Lim, "Position-Dependent Encoding," *ICASSP Proc.*, April 1994.

[8] Grand Alliance Members, "The Grand Alliance System for US HDTV," *Proceedings of the IEEE*, Vol. 83, No. 2, pp.158-174, February 1995.

[9] J. G. Apostolopoulos, "Coding of the Motion-Compensated Residual for an All-Digital HDTV System," Masters Thesis, M.I.T., August 1991.