Joint MPEG-2/H.264 Scalable Coding and Enhancements to H.264 Intra-Mode Prediction

by

Duane Thomas Storey

B.A.Sc., University of British Columbia, 2000

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

The University of British Columbia

June 2006

Abstract

In this thesis, several enhancements and optimizations related to H.264, the latest and most advanced video standard, are explored. The first part of this work involves the investigation of a new spatial scalability scheme involving MPEG-2 as a base-layer and H.264 as a spatial enhancement-layer. This scheme does not involve any bitstream modifications, and can be configured using off-the-shelf components. While simplistic, this scheme results in bitrate reductions for the tested CIF sequences of approximately 50% (depending on the quality of the base-layer) and bitrate reductions on the order of 20% for high-definition 720p imagery.

An alternate scheme is also proposed that extends the H.264 bitstream syntax by adding a new 4x4 Intra-block prediction mode. For Intra-frames, bitrate reductions on the order of 25% are possible depending on the quality of the base-layer. Since this scheme can also choose to ignore distorted base-layer blocks, it is not subject to the large bit-penalty associated with low-bitrate base-layer sequences demonstrated using the first proposed scheme.

Finally, a modified Intra-mode prediction scheme is proposed for H.264 that involves forming a real-time estimate of the joint probability distribution functions for the Intra prediction modes. Using these distributions, it is possible to obtain an increase in prediction accuracy of approximately 50% over the default H.264 prediction method.

Contents

List of Tables

List of Figures

Acknowledgements

First, I would like to thank Kemal Ugur and Panos Nasiopoulos for guiding my research over the last three years.

I would like to thank Desiree Ord, Tinkerbell, for being understanding with the late nights and constant frustration my research has caused me. Without her support and friendship, I may have strayed from this path a long time ago.

Thanks to Dustin Harrison, for understanding that the path of least resistance often takes one to the front door of the nearest bar. May we solve many more problems together over a few cups of coffee.

And last but not least, I would like to thank my family for their support, inspiration, and words of encouragement. I would like to thank my mom, for her love, her support, and for letting me dismantle the family computer so I could figure out how it worked. To my dad, for all his advice, and for keeping the rum cabinet always stocked. I would like to thank my step-mom, Cathy, for her support, her insight, and for being able to produce a cooked turkey on demand. And lastly, to my sister and brother-in-law, for always letting me know that you care.

DUANE THOMAS STOREY

The University of British Columbia

June 2006

Chapter 1

Introduction

We live in a world that is driven by media. From our telephone, to our television, the set-top boxes in our living rooms down to the digital projectors in the local movie theater, the enjoyment and distribution of media is tightly coupled with everything that we do. The digital versatile disc (DVD), combined with the MPEG-2 video codec, revolutionized the world of digital video. In contrast to the existing VHS tape format it originally competed with, the DVD also had the following advantages:

- superior image quality
- movie-specific special features
- menuing systems
- variable rate fast-foward and fast-rewind
- 100 year archival life

These attributes quickly catapulted MPEG-2 and the DVD to the forefront of the digital video world, and put an advanced video decoder into most homes.

The evolution of video compression technology has been a slow, steady journey, with each generation of video codecs pushing the envelope of current technology. The general goal for each new video codec is to reduce the amount of bits necessary for a given quality picture. For example, H.263 video offered a substantial improvement over H.261 video, while H.264 encoded video humbles even the best H.263 encoder. Ten years ago, it was not possible to stream SQCIF (88 pixels wide by 72 high) imagery over a local area network (LAN) in real time. Now, with the advances in video codec technology and the increase in computation power, it's possible to stream CIF or VGA imagery in real-time over a typical broadband internet connection at home. Apple's iChat product (a real-time audio/video conferencing application) can even stream multiple H.264 streams and form a real-time video conference on the Internet.

H.264 achieves compression ratios unheard of several years ago. Figure 1.1 shows the relative compression ratios for H.263+, MPEG4 and H.264. As can be seen, at the corresponding picture quality (measured by the peak signal to noise ratio, or PSNR) of 30dB, H.264 uses approximately 40% less bits than H.263+ [1].

While video compression technology has slowly improved over time, it is clear that audio technology has far outpaced video technology. The transition from eight-track tapes to cassette tapes brought about a large improvement in

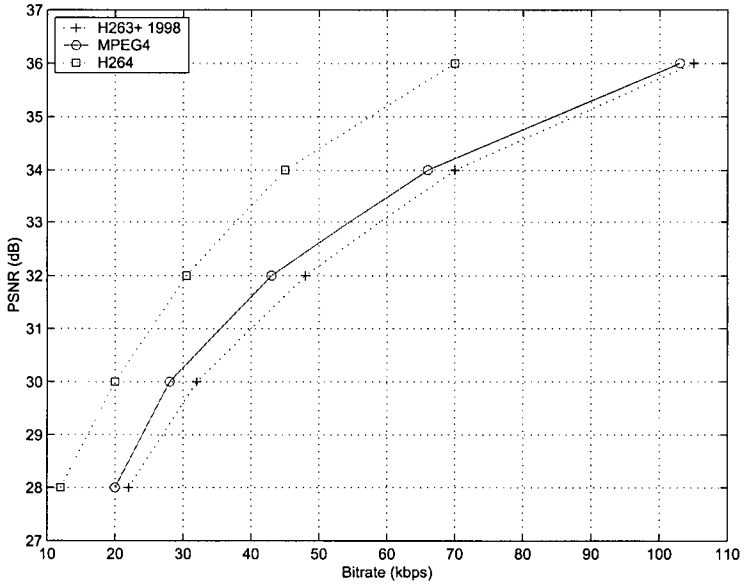


Figure 1.1: The rate distortion curves for three popular video codecs. Note how H.264 uses roughly half the bits for the same quality as H.263+.

listening quality, as did the progression from cassettes to compact disc (CD) audio. Sony now has a new audio format called Super Audio CD (SACD) that offers 96 kHz audio recordings which many individuals claim are indistinguishable from the gold studio master tapes that are made in the recording studio. Since most humans can only hear frequencies between 20Hz and 20kHz, a 96kHz audio recording contains more information than most people can actually hear [2]. In terms of audio, we have reached our own biological auditory limits using the available technology, and it is hard to foresee any necessary changes in the near future.

The situation is somewhat different for video however. The holy grail of digital imagery, high definition, is just slowly beginning to take hold in North

America. Unfortunately, the costs associated with adopting high definition are currently too high for most individuals to justify obtaining this technology [3]. In addition, there are still several technological challenges to overcome before the average consumer throws his or her 720x480 DVD discs away, and embraces high-definition whole-heartedly. One problem to overcome is that a new digital medium capable of storing a high-definition movie must be brought to market. Currently there are two competing technologies – Blu-Ray[4] (a 50 GB disc that uses a 405nm blue laser) and HD-DVD[5] (a 30 GB disc that uses similar technology to the DVD). The second challenge is to obtain a video codec that is able to provide a high-definition movie using less bits than MPEG-2. After years of research, H.264 is ready, and is well suited to replace the aging MPEG-2 standard.

1.1 Research Objectives

Since MPEG-2 will continue to exist in the marketplace for quite some time, the original goal of this thesis was to attempt to combine elements of MPEG-2 with H.264 so that H.264 implementations could benefit from MPEG-2 streams that already exist. Ultimately, a digital broadcasting system was envisioned that could transmit a MPEG-2 low-definition sequence on one channel (which is currently compatible with most digital broadcasting systems such as digital TV), and also transmit supplemental high-definition H.264 data on an alternate channel.

The main thesis goal was to investigate the bitrate reduction possible when combining a H.264 high-resolution stream with a lower resolution MPEG-2 stream. Some literature exists that explores the combination of non-scalable MPEG2 codecs for a spatial scalability scheme, but no work has been done to explore the combination of MPEG2 with the new H.264 standard. The results from this investigation led us to explore modifications to the H.264 standard with respect to Intra-prediction modes for Intra-frames. The purpose of this investigatation was to enhance the efficiency of the proposed MPEG2/H.264 scalability scheme by modifying the H.264 standard to improve the bitrate reduction possible when a spatial scalability scheme based on an alternate codec is used.

The proposed changes to the H.264 standard also led to the realization that the Intra-mode prediction mechanism currently used in H.264 could be enhanced by several small changes. The proposed changes lead to a decrease

in bits neccessary for the proposed MPEG2/H.264 scalability schemes, and also improves the prediction mechanism for normal H.264 bitsreams.

The remainder of this thesis is structured as follows:

In Chapter 2, an overview of video compression technology is presented. This chapter details video compression fundamentals, as well as background material needed to understand subsequent chapters.

In Chapter 3, a spatial scalability scheme that combines aspects of MPEG-2 and H.264 (to allow for a high resolution H.264 enhancement-layer to be based on a low-resolution MPEG-2 base layer) is explored. This scheme does not involve any changes to the H.264 standard, and can be created using off-the-shelf components.

In Chapter 4, extensions are made to the proposed scalability scheme that involves changes to the H.264 standard in the area of 4x4 Intra-block prediction, and results in a further bitrate reduction for the scheme presented in Chapter 3. These changes led to the realization that improvements could be made in the H.264 4x4 Intra-mode prediction scheme as well.

In Chapter 5, enhancements are explored for the H.264 4x4 Intra-mode prediction mechanism. These extensions allow for a reduction in bits in a given Intra frame due to improvements in the overall accuracy of the prediction scheme, and further enhance the two schemes proposed in Chapters 3 and 4.

Finally, in Chapter 6, we provide a summary along with possible future work.

Chapter 2

Background

2.1 Video Compression Basics

A normal uncompressed video frame typically consists of a 24-bit RGB triplet indicating the amount of red (R), green (G) and blue (B) components for each pixel in the frame. This combination is well suited for display on home televisions and computer monitors, since these devices typically use red, green and blue phosphors to create a colour gamut. While RGB images are widely used, research has shown that the human visual system (HVS) is extremely sensitive to brightness (luminance), but not very sensitive to deviations in colour (chrominance) [6]. This perceptual knowledge makes it possible to reduce the size of a frame with little deterioration in quality simply by converting to a different colourspace [7], and discarding some of the colour data.

Most video codecs convert the RGB colour space into a I420 colour space, yielding an immediate decrease in data size. The I420 colourspace

contains the same amount of luminance information as a RGB-24 image of the same size, but contains only half of the color information.

A typical resolution for a DVD movie is 720x480 pixels, which yields a raw frame size of 4.04 megabits (Mb) in I420. At thirty frames per second, a raw, uncompressed DVD movie sequence would require 121 megabits (Mb) for each second of video. The goal of video compression is to remove as much redundancy as possible from a video sequence, and reduce the size requirements of a video sequence.

Video compression employs many methods to convert raw image data, which is very large, into a lossy, reduced bitrate representation of the original video sequence. This section presents a quick overview of the general procedures for compressing video.

2.1.1 Intra Frames

There are several frame types that are used in modern video compression. Intra frames, or I-frames, are completely independent, and represent a resynchronization point for a decoder when frames are lost. Intra frames can also form the basis for other types of frames.

Each I-frame is divided into a series of blocks, usually on the order of 8x8 or 16x16 pixels. Figure 2.1 shows a sample image subdivided into 16x16 macroblocks.

For most block-based video encoders the encoder loops over each block in the I-frame, and computes the two-dimensional discrete cosine transform

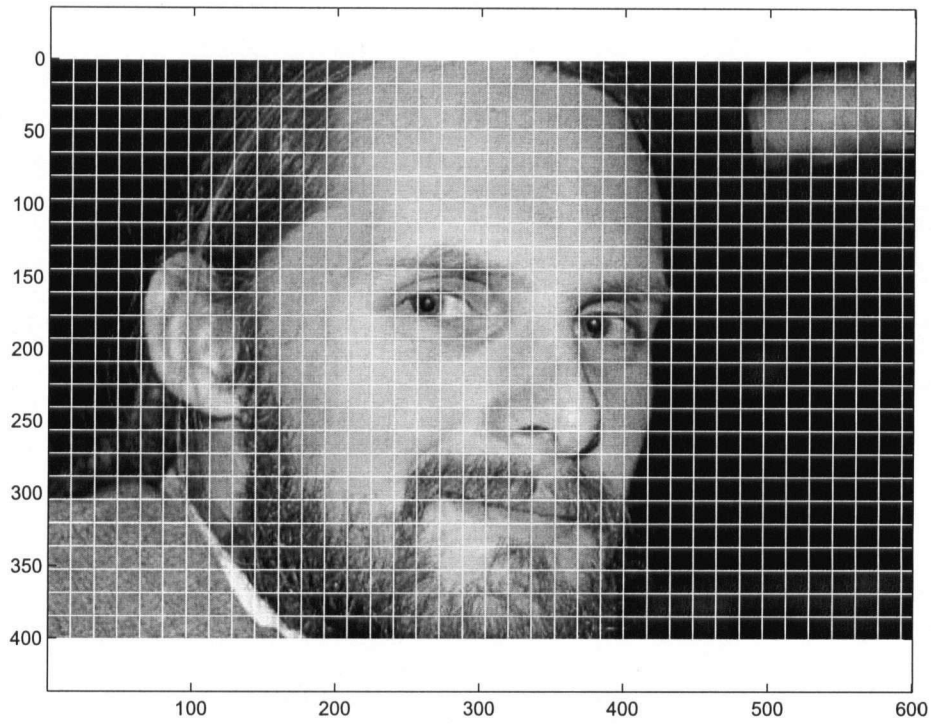


Figure 2.1: This image has been divided into 8x8 macroblocks, as indicated by the white lines. Notice how the pixel variation within most blocks is small.

(DCT) for the block. Since each block, and video in general, is predominantly composed of low-frequency information, the DCT usually results in significant energy compaction. Figure 2.2 shows an example of energy compaction using the DCT.

Next, the values of the DCT are written out to the bitstream in a particular order (often called the zig-zag order in literature) that places low-frequency information first in the stream, followed by the high frequency information. This arrangement maximizes the number of consecutive zeros (from the lack of high-frequency energy in the block), and results in efficient run-

length-encoding (RLE) during the entropy encoding phase (see section 2.1.4).

Modern day decoders also make use of I-frames to provide video previews during fast-forward or fast-rewind methods on DVD or set-top boxes.

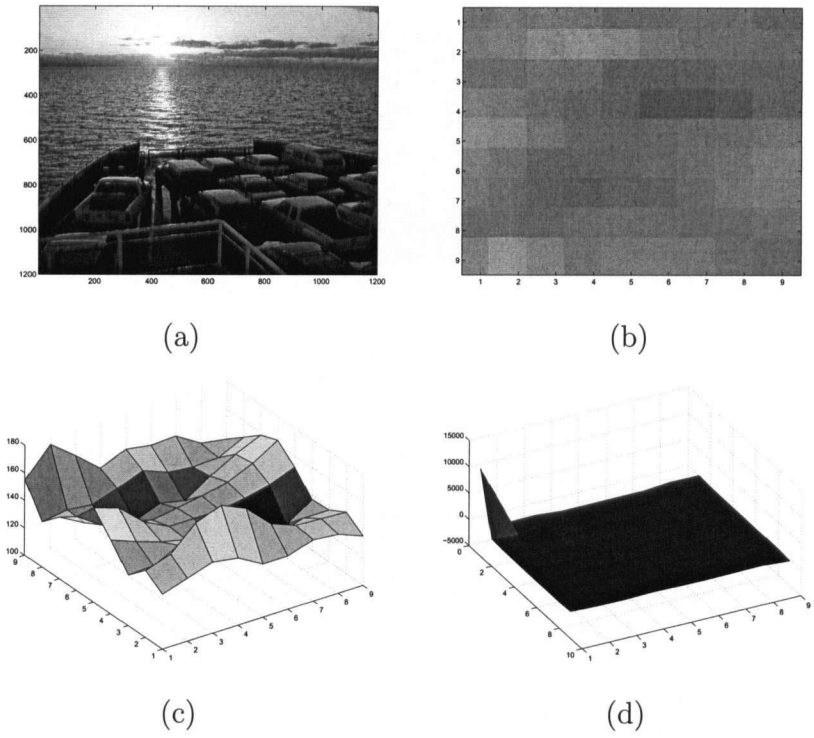


Figure 2.2: These figures show how the DCT results in energy compaction. (a) the original sample image (b) a 8x8 region located in the water of the image. (c) a view of the actual pixel values for the 8x8 region. (d) 8x8 DCT of the water region. Notice how almost all of the energy is located at the DC component.

2.1.2 Predictive Frames

Predictive frames, or P-frames, represent a large portion of the size reduction in video compression. Since each frame in a video sequence usually represents

a fraction of a second, there is large amount of spatial redundancy between frames. P-frame compression exploits this redundancy by utilizing information in previous I or P frames to achieve a very high compression ratio.

For each block in a P-frame, the encoder attempts to find the closest matching block from the previous I-frame or P-frame (or some other reference image, depending on the complexity of the encoder). Most encoders allow at least an 8x8 pixel deviation from the current macroblock position to the macroblock in the reference frame. For each macroblock in the search region, the encoder usually computes the sum of absolute differences (SAD):

$$M_{SAD} = \sum_{j=0}^{s-1} \sum_{i=0}^{s-1} |I_{(i,j)} - \hat{I}_{(i,j)}| \tag{2.1}$$

where M_{SAD} represents the SAD result, i is the x coordinate, j is the y coordinate, $I_{(i,j)}$ is the current macroblock pixel value for position (i, j), s is the number of positions in each direction that can be used for motion estimation, and $\hat{I}_{(i,j)}$ is the reference macroblock value at position (i, j). For H.264, which can use 8x8 search regions and quarter-pel accuracy, the value of s is 32 in each direction. Of all the computationally complex operations involved in video compression, determining the best block in a reference frame (which is called *motion estimation*) is the most intense. For H.264, nearly 1024 4x4 values can be estimated for each block when an exhaustive search is being done. A great deal of research goes into fast motion estimation algorithms that minimize the amount of physical calculations needed[8, 9, 10]. Advanced codecs usually allow motion estimation to occur for non-integer pixel values [11]. For example, H.263 allows for half-pel accuracy of motion estimation,

while H.264 allows for quarter-pel accuracy[1].

The macroblock that results in the lowest amount of bits written to the stream is chosen as the basis for the currently encoded macroblock, and is signalled to the decoder by a two-dimensional motion vector (MV). Most encoders also allow each macroblock in a P-frame to be encoded as if it were an Intra-macroblock. This is useful if no suitable match in the previous reference frame can be found for the current macroblock in the P-frame. In this case, basing the prediction on previous data would result in a greater amount of bits than encoding the macroblock directly. P-frames can only be predicted from I-frames or previously decoded P-frames.

Once the motion-vector is specified, the encoder subtracts the reference macroblock from the current macroblock. The two-dimensional spatial difference block is then converted into the frequency domain via the DCT, quantized, and then written to the bitstream.

The loss of a P frame is problematic since future P frames were most likely based upon the lost P frame. The visual quality of the sequence will therefore continue to deteriorate until the reception of the next full Intra frame.

2.1.3 Bi-predictive Frames

The last major frame type used in video is called a bi-predictive frame, or B-frame. B-frames are similar to P-frames, but they are based on predictions from past and future frames. They generally achieve better compression ratios

[1]eighth-pel accuracy is currently being experimented with

than P-frames, but are more computationally complex. In addition, they also introduce a one-frame latency into the bitstream (since the previous and future frames must first be decoded before the B-frame can be decoded). The addition of latency makes B-frames unpopular for most real-time video systems such as video over IP.

Since no other frames depend on decoded B-frames, they are completely discardable, and will not result in the further deterioration of video quality should a B-frame be lost from the stream.

2.1.4 Entropy Encoding

The final area where video achieves compression is by entropy encoding. Common entropy reducing methods usually revolve around run-length encoding, such as in a bitmap, or around variable length codes (such as Huffman or Exp-Golomb). These codes are based on the probability distribution of the symbols in the source stream.

H.264 makes use of Exp-Golomb codes, which are optimal codes for exponential or geometric distributions [12]. Table 2.1 shows several codewords for the Exp-Golomb codes used in H.264. The cost of a zero using this scheme is only one-bit. Since H.264 encodes only the difference between the actual block and the predicted block, many of these elements are usually zero, and only cost one-bit to write each element to the bitstream.

Table 2.1: H.264 makes use of Exp-Golumb codes. These codes are optimized for exponential distributions. The first nine of these codes are shown here. Notice that the cost of encoding a zero is simply one bit.

Value	Codeword
0	0
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001

H.264 also has context adaptive variable length codes. These codes

change their meaning depending on the content of surrounding marcoblocks, or encoding modes. Using context adaptive codes further reduces the bit requirements of the H.264 stream.

2.1.5 Peak Signal-to-Noise Ratio

To most common metric that is used amongst researchers to gauge video quality is the peak signal-to-noise ratio, or PSNR. The PSNR for an image is defined by the following equations:

$$MSE = \frac{1}{MN} \sum_{i}^{M} \sum_{j}^{N} |I(i,j) - K(i,j)|^2 \tag{2.2}$$

$$PSNR = 20 \log \frac{255}{\sqrt{MSE}} \tag{2.3}$$

where M and N represent the width and height of the image, I represents the source image, K represents the deteriorated image, and (i, j) are the coordinates within the image. Typical values for the PSNR are between 20dB and 40dB, although it varies based on the source material. As a general rule of thumb, 20dB is heavily distorted video, 30dB is acceptable for most people, and 40dB is nearly indistinguishable from the original.

While the PSNR is useful for gauging relative quality within a sequence, it does not provide any absolute measurement ability. For the purpose of this thesis, PSNR measurements are generated using a fixed set of input sequences, and compared. This allows a fair comparison of quality within each sequence.

2.2 Overview of H.264

Several years ago, an effort was initiated to replace the aging MPEG-2 standard with a new video compression design, one that would take advantage of the improvements in computational power since the time MPEG-2 was released. In 1997, the ITU-T Video Codec Experts Group started work on a new codec which was named H.26L[13]. The goal of this group was to create a new codec that would outperform H.263++ (the year 2000 version of ITU-T recommendation H.263), and MPEG4 simple profile.

In 2003, the JVT in combination with the ITU-T standardized the H.26L draft and officially renamed it to H.264. Figure 2.3 shows the progression of all major video codecs over time.

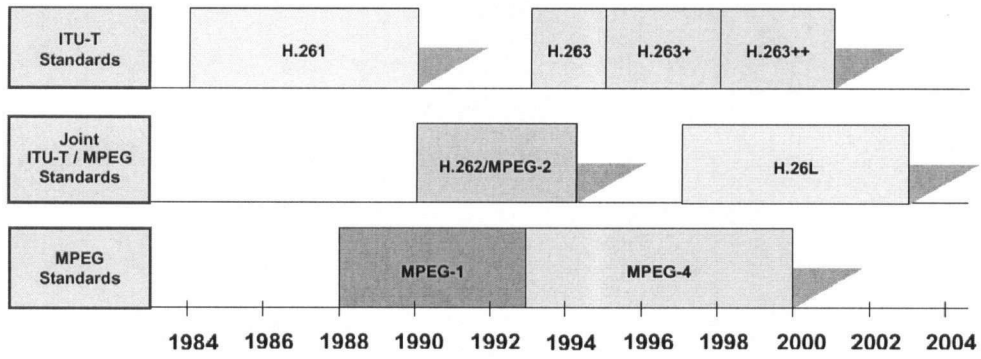


Figure 2.3: This figure shows the progression of the various video codecs developed in the last twenty years (Source: UB Video)

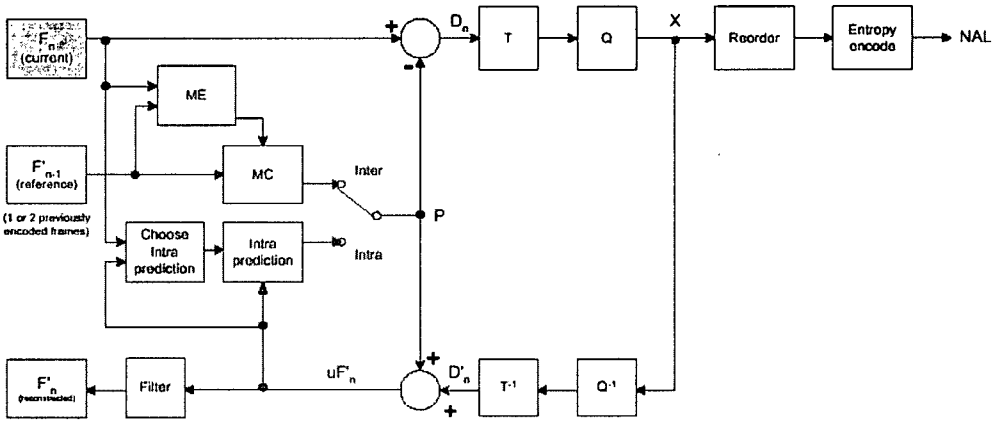


Figure 2.4: A typical H.264 encoder is shown here. Compared to H.263, H.264 also contains an in-the-loop deblocking filter, advanced quarter-pel prediction accuracy, Intra-mode prediction and adaptive entropy coding schemes.

2.2.1 H.264 Encoder Details

The bitrate savings in H.264 are the result of advanced Intra-prediction modes, strong motion isolation due to quarter-pel accuracy, multiple reference frames for predictions, Intra-frame prediction, weighted bi-predictive frames, and by context-adaptive block coding (CABAC) [14]. These added features make H.264 extremely computationally complex [15], [16].

Figure 2.4 represents a H.264 encoder. Ever since H.263+, block-based coders have benefited from an in-the-loop deblocking filter [17]. This filter greatly enhances the perceived image quality by reducing the blocking artifacts caused by the DCT at low bitrates [18]. H.264 includes a deblocking filter that changes its filter taps depending on the amount of blocking present. While a deblocking filter usually improves the subjective quality, the PSNR

after deblocking usually decreases slightly since the image has technically been altered.

2.2.2 H.264 Intra Prediction

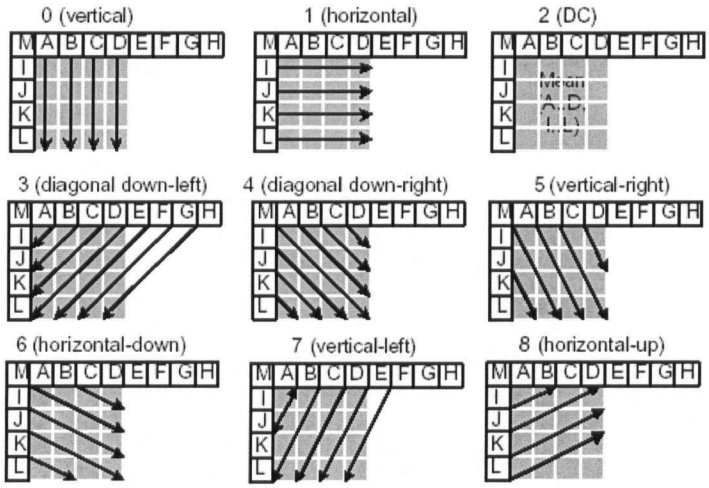


Figure 2.5: H.264 allows prediction to occur within each Intra frame. Shown here are the nine modes used by H.264 for the Intra 4x4 predictor. The letters represent pixels in the adjoining macroblocks, while the arrows represent the direction of the prediction.

In H.264, it is possible to predict the value of a intra 4x4 or 16x16 block based from another adjacent 4x4 or 16x16 block. This functionality is called Intra-prediction, and it is not included in other major codes [19]. Intra prediction is usually limited to data within a slice, so that if surrounding slices are lost, the Intra prediction values for the current slice will remain the same.

There are nine distinct prediction modes that are currently used within

4x4 Intra blocks. These modes are shown in figure 2.5. Pixels A through F represent pixels in the blocks directly above the current block, while I through L represent pixels in the previous horizontal block. For modes 0 and 1, each value in the current block is determined by duplicating the values of the pixels at the base of each arrow. For example, in mode 0, all pixels under pixel A will assume the same value as the pixel at A. For mode 2, a DC value is computed based on the pixel values at M, A, B, C, D, I, J, K and L. Modes 3 through 8 are determined by fitting planes between the indicated pixels.

To reduce bits, H.264 does not directly encode the predicted intra 4x4 mode used for each block. Instead, the encoder estimates the most probable prediction mode for the current block based on the modes chosen for the adjacent MBs. If the encoder makes an accurate prediction of the Intra mode, a single bit is written to the bitstream; the cost of an accurate prediction is therefore only one bit.

If the estimate was not correct, then the encoder must signal the correct mode to the decoder. Since there are nine prediction modes, and the most probable mode is not correct, three bits are needed to indicate the proper mode. The cost of a prediction miss is four bits. Intra-mode predictions are discussed in more detail in chapter 5.

H.264 also includes a network abstraction layer (NAL). This layer deals with the packetization of data so that H.264 can easily be transmitted over a network such as the Internet [20].

Encoding video using H.264 is extremely computationally complex, and

much research has gone into reducing the complexity for use in real-time environments ([21], [22]). Some implementations have even optimized enough to work on digital cameras [23] or other hand-held devices[24].

2.2.3 H.264 Decoder Details

Figure 2.6 represents the minimal functionality for a H.264 decoder. The first operation the decoder performs is to parse the bitstream. Once the bitstream is parsed, the decoder converts the entropy-coded data into macroblock data. This data may have been compressed with either CAVLC or CABAC. Inverse quantization is then performed and the IDCT is computed for each encoded block. For Intra frames, the result of the IDCT is the spatial block. For Predictive and Bi-predictive frames, the spatial data is combined with the motion compensated block computed by using the past and future reference frames (in the case of B frames) in conjunction with the signalled motion vectors. Deblocking is performed on the image to reduce compression artifacts. The decoded frames are then added to the stored picture buffer for use in the future decoding of other P or B frames.

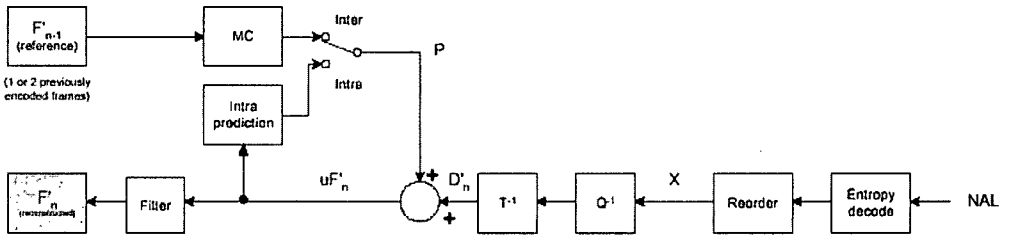


Figure 2.6: Shown here is a block diagram representing a typical H.264 decoder.

Most decoders also contain additional logic for dealing with missing blocks or corrupted data. Typically these elements will interpolate missing blocks within a frame, or use elements from previous reference frames to fill in erroneous information.

In comparison to encoding, decoding video is far less computationally complex. However, some of the research done lately has been to reduce the complexity of H.264 encoding and decoding so that more individuals can be able to decode larger frame sizes (such as high-definition) [25]. Today, one needs a fast computer (roughly a 3GHz machine) to decode high-definition H.264 in real time.

2.3 Video Scalability

One of the most active research areas in video compression is scalability. The primary concept of scalability is that a basic form of the video is sent to the receiver in a base-layer. This layer consists of a reduced quality version (for SNR scalability), a frame reduced version (temporal scalability), or a spatially reduced version of the video (for spatial scalability). The base-layer is typically encoded using a minimum number of bits to ensure that there is always enough bandwidth for its transmission.

An enhancement-layer is created that contains additional information to add to the base-layer. This enhancement-layer is meant to augment the base-layer, improving either the quality, the temporal resolution, or the spatial resolution of the final sequence. When bandwidth permits, the enhancement-layer, or a portion of it, is sent to the receiver. If the bandwidth decreases suddenly during transmission, the streaming server can omit either part or all of the enhancement-layer. Since the receiver generally receives the base-layer in its entirety (this is usually a requirement for scalability schemes, and can be accomplished using forward error correction techniques as in [26]), a lower quality version of the video can still be shown when the enhancement-layer is partially or totally missing.

2.3.1 Temporal Scalability

Temporal scalability is one of the simplest forms of video scalability, as it involves changing the frame rate of the video sequence in real time. One way

to accomplish this is through the use of bi-predictive (B) frames. Since no frames (I or P) are predicted from B frames, B frames can be dropped at any time without impacting the future PSNR quality of the video. This can be utilized to drop frames if the decoder is having difficulties maintaining the frame rate due to computational complexity, or if other hardware limitations exist.

Most modern video compression schemes (H.263, H.264, MPEG4) employ bi-predictive frames for temporal scalability.

2.3.2 PSNR Scalability

PSNR scalability refers to a video sequence that can change its video quality under certain conditions. There are several different methods to achieve PSNR scalability. One method is to encode a certain video sequence multiple times using different bitrate options for each stream. If the streams are being broadcast over a bandwidth limited medium, the low bitrate stream can be sent by default, and then increased to a higher bitrate stream when conditions allow. After the switch is complete, the video sequence at the decoder would have improved visual quality.

Another popular method of spatial scalability involves a scheme called Fine Granular Scalability, or FGS. Typical PSNR scalability usually involves only a few streams at different bitrates. Each switch is therefore very coarse in terms of the bandwidth and quality level acheived during the switch. FGS allows very fine resolution switches, since data can change at the bit level

[27, 28]. The original ideas related to FGS were first added to the MPEG-4, and later proposed for H.264 in [29], [30], [31] and [32]. Improvements to the FGS scheme based on context-adaptive binary arithmetic coding were also proposed in [33] and [34].

It should be noted that when scalability is used, it is usually impossible to recover elements in an enhancement-layer if a portion of the base-layer is missing. This phenomenon is known as *drift* , and it indicates that errors will propagate in the enhancement-layer when errors occur in the base-layer [35]. It is one of the major problems with scalability schemes. Several means of correcting for drift are discussed in [36].

2.3.3 Spatial Scalability

Spatial scalability aims to add an enhancement-layer with additional spatial resolution. An example of this would be a moderate resolution base-layer (720x480) with a high-definition (1920x1080) enhancement-layer. Switching from the base-layer to the enhancement layer would yield an increase in resolution and greater resolving power for the features within the video sequence. Several examples of spatial scalability schemes can be found in [37].

2.4 Previous Research

Most of the exciting research in terms of video compression over the last decade has been in the area of scalability. Scalability allows the quality of the video sequence to be adjusted when the bandwidth of the channel fluctuates. When excess bandwidth is available, the full version of the video is transmitted, either in a full encoded sequence, or via a base-layer and one or more enhancement-layers.

One popular method of spatial scalability, often used when only one codec is available, is to encode the same sequence at different bitrates using an encoder such as MPEG-2, and then send different versions based on the current bandwidth conditions [38]. This method, while easy to implement, forces extra computational requirements onto the encoder, since multiple versions must be encoded offline. Another constraint is that the bitstream can usually only switch on an I-frame, otherwise drift errors will occur in the stream.

As part of the MPEG-4 standard, a new scalability scheme called Fine Granular Scalability (FGS) was proposed. The purpose of FGS "can be viewed as optimizing the video quality over a given bit range instead of at a given bit rate." [39]. The enhancement layer for a FGS scheme is devised so that it can be broken down into a series of bit-planes, each with a different significance. If bandwidth allows, the encoder may choose to transmit the five most significant bit-planes, allowing partial reconstruction of the enhancement-layer, and partial improvement in quality. If the bandwidth suddenly drops, the FGS decoder can send a small number of bit-planes, or terminate the enhancement-

layer completely. In the second scenario, only the base-layer would be received and decoded.

In H.264, the ability to switch between different streams at places other than an I-frame was added. These special frame types are denoted SI, for switching I-frame, and SP, for switching P-frame [40]. These frames represent the information needed to go from one layer to another without introducing any errors. Ugur extended this feature, which provides course bitrate changes via stream switching, with FGS, allowing both course and fine granular bitrate changes [29].

While SI and SP frames allow course bitrate changes within H.264, they do not allow for any bitstream changes if another codec were to be used in the base-layer. FGS allows for fine granular changes, but also requires that the encoder and decoder be modified to understand FGS bitstreams. To satisfy the requirements that the base-layer be based on MPEG-2 (see section 1.1), a different type of scalability scheme is necessary.

In [41] the authors investigated the use of two independent encoders to create a spatial scalability scheme. The benefit of the proposed scheme was that it could be constructed using off-the-shelf MPEG-2 encoders. This novel scheme showed a bitrate reduction compared to encoding a high-resolution stream using MPEG-2 alone. While this scheme presents a basis for forming a spatial scalability scheme using non-scalable MPEG-2 codecs, it would advantageous if a similar scheme could be modeled around H.264, since it is emerging as the dominant video codec for the future.

The first part of this thesis extends the ideas presented in [41] in an attempt to validate this scheme when used with non-scalable MPEG-2 and H.264 codecs. It is not immediately obvious that the new scalability scheme based on spatial difference images would result in a reduction of bits for the H.264 stream, since many aspects of the advanced codec are fine-tuned for standard content images. For example, there is no guarantee that the H.264 deblocking filter would correctly handle a primarily high-frequency spatial difference image without distorting some of the frequency content. The scheme presented in [41] forms the basis for the research done in chapter 3.

Chapter 3

Scalability with MPEG-2

3.1 Motivation

While H.264 offers significant bitrate improvements over MPEG-2, the widespread usage of MPEG-2-enabled set-top boxes and DVD players ensures that the MPEG-2 codec will continue to dominate the consumer market for quite some time. Because of this, the transition over to H.264 will take time, and the two codecs will undoubtedly co-exist in the market place over the next few years.

As most digital broadcasts are currently encoded using MPEG-2, it would be advantageous to utilize redundant information in a MPEG-2 stream to enhance a concurrent high-definition (HD) digital broadcast in another codec, such as H.264. This configuration would allow digital broadcasters to offer additional channels to their customers, and increase their revenue streams. A scalability scheme that combines a low-resolution MPEG-2 base-layer with a high-defitinion H.264 enhancement layer is explored in this section.

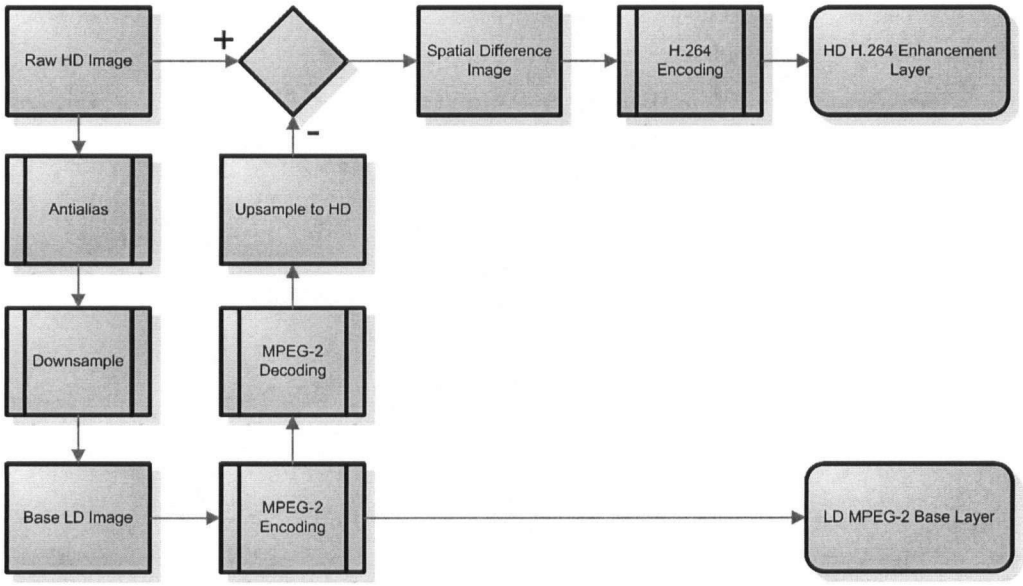


Figure 3.1: This figure shows the encoding details for the proposed scalability scheme. The scheme conbines elements from a low-resolution MPEG-2 stream and a high-resolution H.264 streams.

3.2 Encoding Details

3.2.1 Base-Layer

Figure 3.1 depicts the modified encoding setup for this scalabilty scheme. The first encoding step is to convert the HD source sequence into a sequence suitable for the base-layer encoding. The HD sequence is downsampled:

$$F_{LD} = (\downarrow n)[H * F_{HD}] \tag{3.1}$$

where F_{LD} represents the base-layer source sequence, n is the downsampling factor (which can be fractional), H is a two-dimensional anti-aliasing

filter, and F_{HD} is the high resolution source image. The usage of the anti-aliasing filter is important, since it ultimately affects the bitrate reduction of this scheme (see section 3.2.3).

Since the decoder will only have access to the decoded MPEG-2 image, and not the source image, the HD encoder must utilize the decoded MPEG-2 sequence (and not the original source sequence) to construct the enhancement layer.

3.2.2 Enhancement Layer

The enhancement layer uses H.264, and is composed of spatial difference images representing the changes between the decoded base-layer and the source HD sequence. We can form the enhancement layer source sequences by the following:

$$F_{EL} = C\left(F_{HD} - (\uparrow n)\hat{F_M}\right) \tag{3.2}$$

where F_{EL} is the spatial difference image for the H.264 enhancement-layer, F_{HD} is original high resolution source image, $\hat{F_M}$ is the decoded MPEG-2 base-layer image, and $C(B)$ is an invertible function that maps the 9-bit result of the pixel subtraction operation to the 8-bit value allowed for the pixel . This function is discussed more in section 3.2.4.

Figure 3.2 shows the spatial difference image formed by equation 3.2 for the first frame of the Foreman CIF sequence. Since the downsampling operation results in a reduction of high-frequency components, the primary

Figure 3.2: This image shows an example spatial difference image that is formed using the proposed scalability scheme. Notice how the image is composed of primarily high-frequency information.

component of the spatial different image are those components that were filtered out by equation 3.1.

3.2.3 Resizing Filters

Since the HD enhancement-layer is formed by creating a spatial difference image from an encoded, downsampled version of the original sequence, it is important that a high quality resizing algorithm be used to minimize any artifacts or anomalies (such as ringing).

Most popular image conversion programs offer various options for resizing. ImageMagick was used for these experiments, and it offers the following conversion filters: Point, Box, Triangle, Hermite, Hanning, Hamming, Blackman, Gaussian, Quadratic, Cubic, Catrom, Mitchell, Lanczos, Bessel and Sinc. Based on the documentation, a small subset of these were used to determine which combinations produced the highest PSNR based on the image resampling requirements of the proposed scheme.

One-hundred frames of the Foreman CIF sequence were used for the image resizing tests. These tests amounted to the following:

1. Downsample source sequence from CIF to QCIF using the specified filter. ImageMagick chooses a filter automatically if one is not specified.

2. Upsample source sequence from QCIF to CIF using the specified filter.

The results of the experiments, sorted by descending image quality, are displayed in table 3.1.

Based on these results, the Lanczos filter was chosen for both the upsampling and downsampling portions of the scalability scheme.

Table 3.1: Shown here are different combinations of filters for upsampling and downsampling. The resampling filters chosen can have a significant impact on final image quality. The difference between the highest and lowest values is 1.3 dB, which is quite noticeable. The Lanczos filter yields the highest PSNR value, and was chosen for both upsampling and downsampling in the proposed scalability scheme.

Filter down	Filter up	PSNR (dB)
unspecified	lanczos	31.9
lanczos	lanczos	31.9
catrom	lanczos	31.6
unspecified	catrom	31.5
lanczos	catrom	31.5
catrom	catrom	31.0
mitchell	lanczos	31.0
unspecified	unspecified	30.9
unspecified	mitchell	30.9
lanczos	unspecified	30.9
lanczos	mitchell	30.9
unspecified	bessel	30.8
lanczos	bessel	30.8
bessel	bessel	30.7
catrom	unspecified	30.6
catrom	mitchell	30.6
mitchell	catrom	30.5
catrom	bessel	30.4
bessel	mitchell	30.3
mitchell	unspecified	30.0
mitchell	mitchell	29.0
mitchell	bessel	29.7
bessel	unspecified	29.7
bessel	lanczos	29.7
bessel	catrom	29.6

3.2.4 Quantization Function

In equation 3.2, an invertible function[1] is required that maps a 9-bit value to an 8-bit value. Several clamping functions were investigated for this design.

Linear Clamp

The first is a simple linear clamping function:

$$C(x) = \begin{cases} 255 & \text{if } x > 127, \\ x + 128 & \text{if } -128 < x < 128, \\ 0 & \text{if } x \leq -128. \end{cases} \tag{3.3}$$

The benefit of this scheme is that there is no quantization error for the 256 values centered around zero error. Unfortunately, this benefit is at the expense of poor accuracy for large errors.

Linear Scale

The next quantization scheme investigated involves scaling the 9-bit input signal into an 8-bit output signal. The mapping function is:

$$C(x) = \frac{255 + x}{2} \tag{3.4}$$

Equation 3.4 gives nearly equal quantization errors for all values.

Figure 3.3 utilizes the quantization functions for the two quantization schemes.

[1]since we are dealing with finite precision numbers, there will inherently be quantization errors in this mapping, even though it is by definition invertible

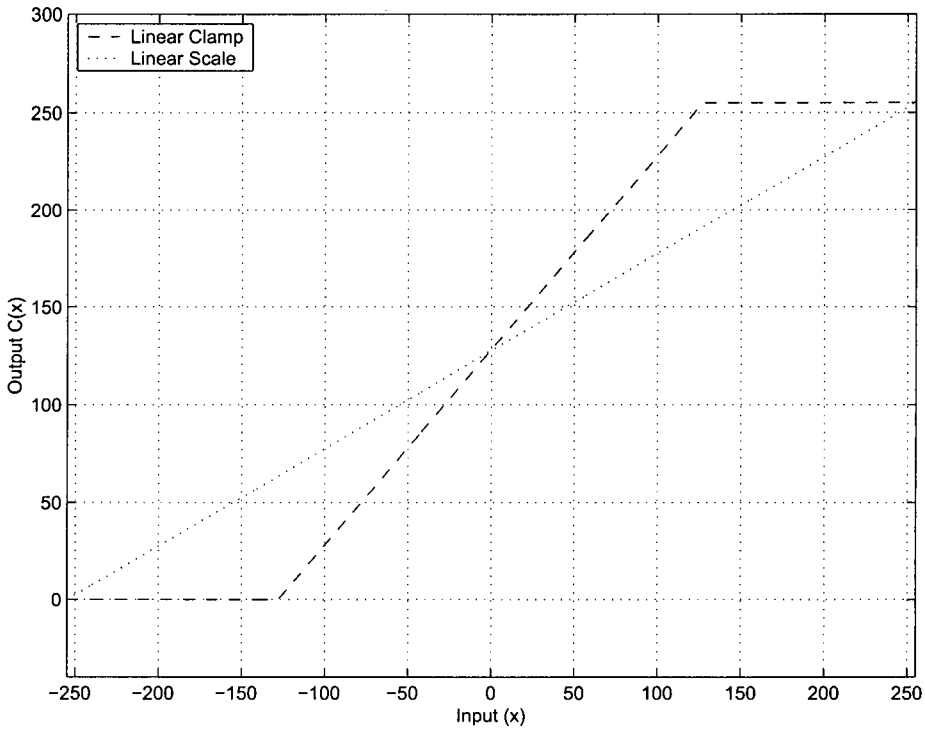


Figure 3.3: This figure demonstrates the two quantization schemes used for this experiment.

Figure 3.4 shows the resultant rate-distortion curves for both quantization schemes. Based on the results, it appears that both are adequate, and provide minimal error for low-values, which will predominate the signal values. The linear-scale method was chosen for all future experiments.

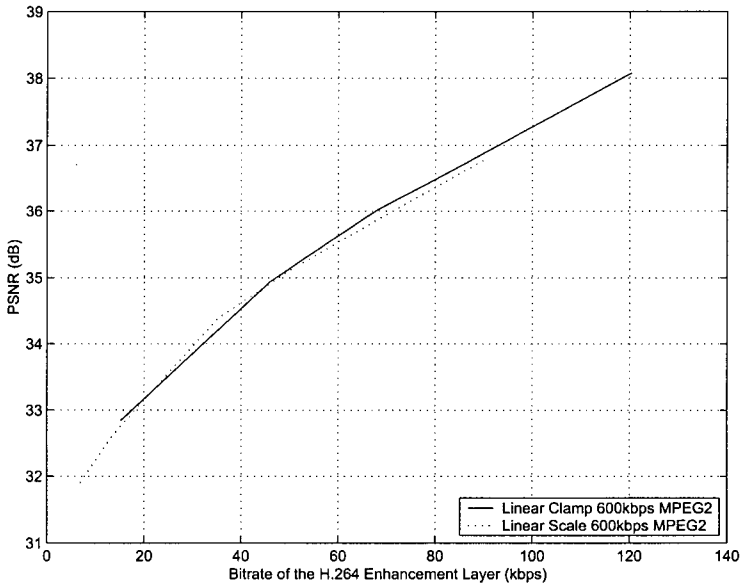


Figure 3.4: This figure shows the results of the quantization scheme test when used in the proposed scalability scheme. Based on the results, the linear-scale method was chosen.

3.3 Decoder Details

Decoding using this scalability scheme involves decoding both the MPEG-2 and H.264 stream. If the MPEG-2 stream is lost or damaged, the quality of the H.264 will suffer significantly. No work was done to limit the damage to the H.264 scheme based on loss in the base layer.

Figure 3.5 shows the steps neccessary to decode the modified H.264 stream. First the MPEG-2 and H.264 streams are decoded on a frame by frame basis. The H.264 decoder will output the spatial difference image for the sequence, and the MPEG-2 stream will output the low resolution frame. The H.264 decoder was modified to read the MPEG-2 decoded imagery, scale the

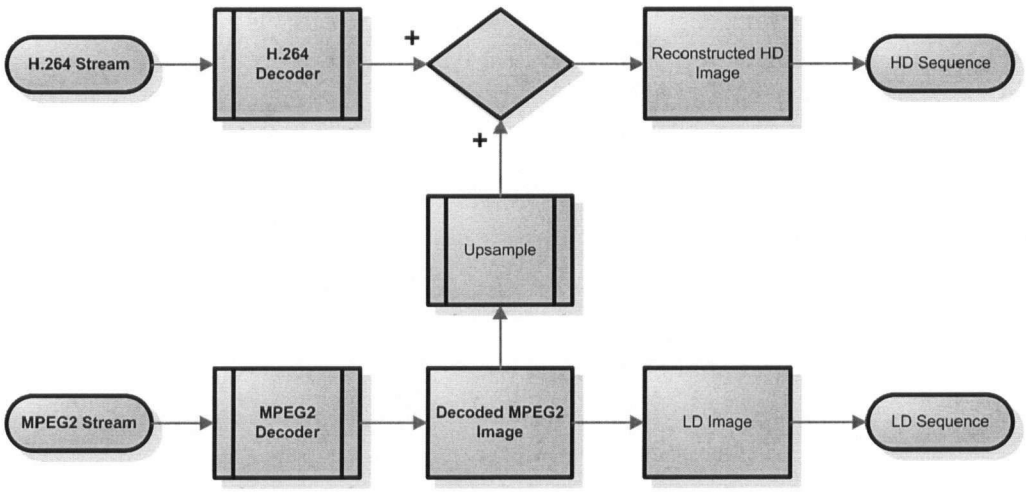


Figure 3.5: Shown here are the decoding steps associated with the proposed scalability scheme. The decoder forms a high-resolution image by combining elements from a low-resolution MPEG-2 layer and a high-resolution H.264 layer.

images up to match the high-resolution layer, and re-create the original H.264 stream by adding the MPEG-2 spectral information to the H.264 difference image.

3.4 Results

In the previous sections the components of a new scalability scheme based on work done in [41], but extended to be used with MPEG-2 and H.264, were described. In section 3.2, the layout of the encoder block was described. Part of the encoding and decoding process involves resampling sequences to different resolutions. Based on the results show in section 3.2.3, the Lanczos resampling filter yields the highest PSNR values. Since the result of doing a spatial subtraction between two eight-bit values is a nine-bit value, two quantization schemes were evaluated. The results showed that both methods performed equally well. For all future experiments, the linear-scale method was chosen. The decoder for the proposed scheme was shown in section 3.5, and represents the final block for the proposed scheme.

The following subsections detail the results for the proposed scalability scheme. All PSNR values refer to the luminance components only. For all sequences discussed, fifty frames, representing progressive sequences of approximately two seconds, were encoded using this scalability scheme. Each point on the results graphs represents approximately 15 minutes of computation time using only fifty frames.

The motivation for each experiment was to examine the bitrate changes in the H.264 enhancement-layer by basing it on various base-layers. The base-layers that were used were 100 - 1000 kb/s MPEG-2 encodings, as well as an uncompressed base-layer that had been downsampled, and then upsampled (i.e., an MPEG-2 layer with no MPEG-2 compression artifacts).

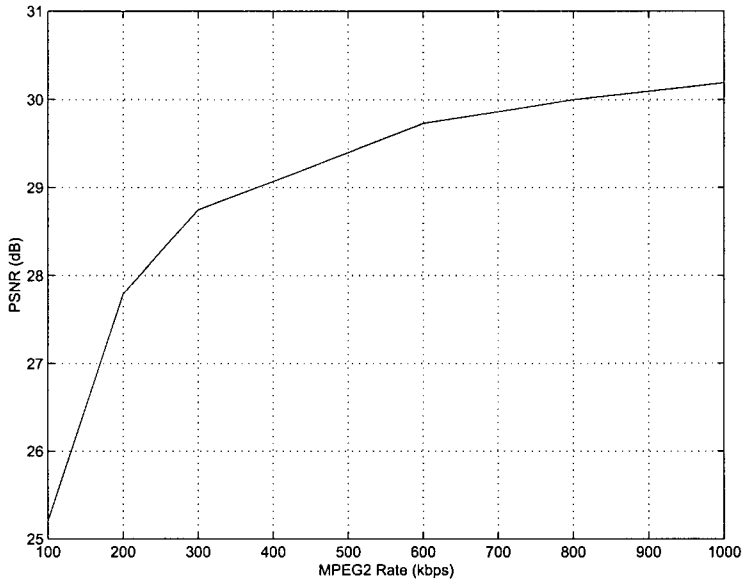


Figure 3.6: This figure demonstrates the rate-distortion curve for the Foreman sequence that has only undergone Lanczos downsampling from CIF to QCIF, followed by Lanczos upsampling from QCIF to CIF. The PSNR values represent the upper-limit on the quality obtainable using this scheme, since the sequence has not undergone any MPEG-2 compression.

3.4.1 Foreman Results

The original Foreman sequence was a I420 CIF sequence. This sequence was downsampled to QCIF resolution and used for the base-layer. The base-layer was encoded using MPEG-2 at rates of between 100 - 1000 kb/s. The decoded MPEG-2 sequences were then upsampled offline to form the reference images for the enhancement layer. Figure 3.6 shows the resultant CIF PSNR for the base-layer sequences.

Figure 3.7 shows the results of using the proposed scalability scheme on the CIF Foreman sequence. The lower solid line in figure 3.7 represents

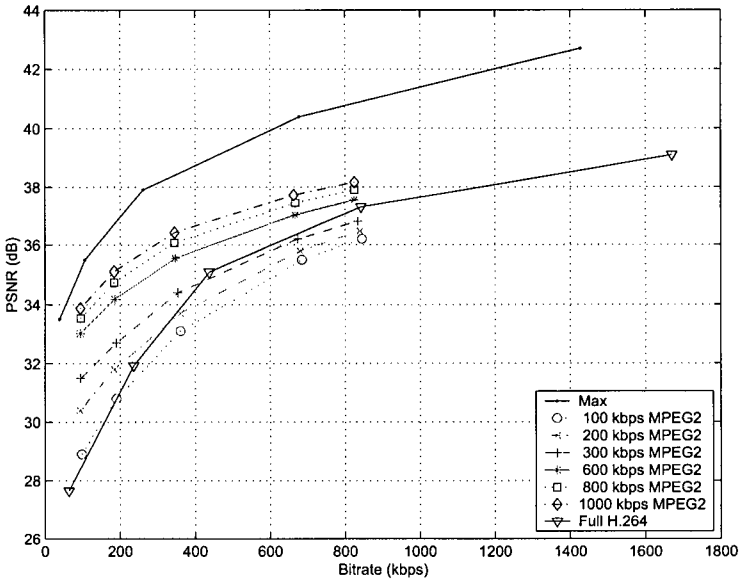


Figure 3.7: Shown here is the results of the proposed scalability scheme using the Foreman sequence. For MPEG-2 rates of at least 300 kb/s, the scheme usuaully results in a bitrate reduction. For lower rates, a bitrate penalty sometimes occurs.

the PSNR curve for H.264 using the CIF Foreman sequence without using the scalability scheme. For that sequence, Foreman was encoded using only H.264. Any curve that is shifted towards the top of the graph with respect to that curve represents a reduction in bits for a given PSNR.

The dotted curves in Figure 3.7 show the resultant H.264 bitrates when our scalability scheme is used. They represent the rate-distortion curves for the H.264 enhancement layers when they are based on the shown MPEG-2 base layer sequences. For most levels of compression, the proposed scalability scheme offers a bitrate reduction in the H.264 enhancement-layer. For example, if the desired enhancement-layer PSNR is 36dB, H.264 normally requires

approximately 650 kb/s to encode the CIF sequence. When the proposed scheme is used, the bitrate in the enhancement layer can be reduced to as low as 250 kb/s, depending on the compression used in the MPEG-2 layer.

It should be noted that some of the MPEG-2 curves fall below the H.264 curves. This indicates that the proposed scalability scheme actually used more bits at this level than would have been used had the original sequence would have been encoded using only H.264. This can be explained by realizing that low-bitrate MPEG-2 sequences have very noticeable compression artifacts (namely blocking) that would require substantial high-frequency information in the H.264 layer to compensate for.

Table 3.2: This table shows the bitrate reduction for the Foreman sequence. The percentage values indicate the size of the bitstream as a fraction of the original H.264 bitstream. For resultant CIF PSNR values of 36dB, bitrate reductions of approximately 50% are achievable.

Base Layer	Enhancement Layer PSNR		
Bitrate (kb/s)	32dB	34dB	36dB
100			
200	85%		
300	56%	84%	
600	34%	47%	68%
800	26%	31%	58%
1000	21%	26%	55%

Table 3.2 shows the bitrate reductions possible by using the proposed scheme (these data can be obtained directly from figure 3.7). The percentage values in tables 3.2, 3.3, 3.4, and 3.5 indicate the size of the resulting bitstreams (when using our proposed scalability scheme) as a percentage of the original

H.264 bitstream size. As is evident in table 3.2, given a high bitrate MPEG-2 stream (1000 kb/s), it is possible to achieve bitrate savings of up to 80%. Even at modest bitrates of 600 kb/s, the resulting bitrate uses only 34% of the bits in the original bitstream. It should be noted that MPEG rates of 100 kb/s - 200 kb/s are heavily distorted, and have a very poor visual quality.

Sample images from the Foreman sequence can be found in Appendix A.

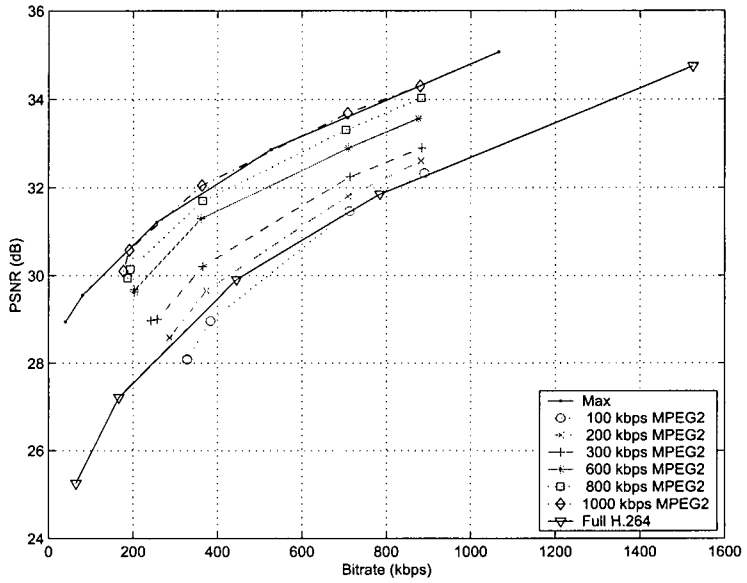


Figure 3.8: Results for Coastguard, CIF Source

3.4.2 Coastguard Results

The results of the Coastguard CIF sequence can be interpreted similarly to those discussed in the Foreman section. Figure 3.8 shows the results of the scalability scheme when using the Coastguard sequence.

Since the Coastguard sequence is less complex than the Foreman sequence in terms of subject movement, the MPEG-2 bitrate needed to obtain a high-quality sequence is reduced. As a result, most curves in Figure 3.8 lie above the H.264 curve, indicating a bitrate savings using this scheme. It should also be noted that the 1000 kb/s MPEG-2 encoding approaches the bitrate savings of the uncompressed CIF sequence (shown as the upper solid line marked "Max"). This indicates that 1000 kb/s in the MPEG-2 layer is

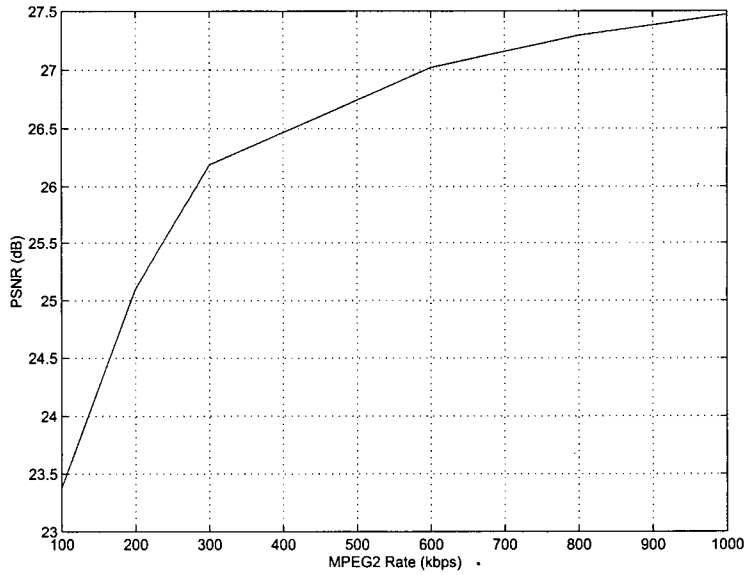


Figure 3.9: This sequence shows the PSNR values for Coastguard sequence after undergoing downsizing to QCIF, followed by upsizing to CIF.

sufficient to represent the data in the original sequence with minimal loss upon decoding.

Figure 3.9 represents the PSNR of the upsampled decoded MPEG-2 sequence at various bitrates.

Table 3.3 shows the bitrate savings using this scheme at various quality levels. For a modest MPEG-2 rate of 600 kb/s, the resultant bitstream uses between 29% and 68% of the bits as the original H.264 stream.

Sample images from the Coastguard sequence can be found in Appendix A.

Table 3.3: Shown here are the bitrate reductions possible using the Coastguard sequence with the proposed scalability scheme. For a high-resolution PSNR value of 36 dB, the resultant bitstream uses only 13% of the bits as the original H.264 stream.

Base Layer	Enhancement Layer PSNR		
Bitrate (kb/s)	32dB	34dB	36dB
100	92%		
200	75%	93%	96%
300	41%	81%	88%
600	29%	44%	68%
800	20%	37%	59%
1000	13%	23%	50%

3.4.3 Silent Results

Silent is a CIF image sequence tested using the proposed scalability scheme. The results from this sequence are shown in table 3.4. In terms of image content, it is fairly complex, having rapid movement in several locations within the frame. For this type of sequence, a lower bitrate reduction should be expected using H.264 directly, since H.264, with its quarter-pel accuracy, is well suited for compressing images with high motion.

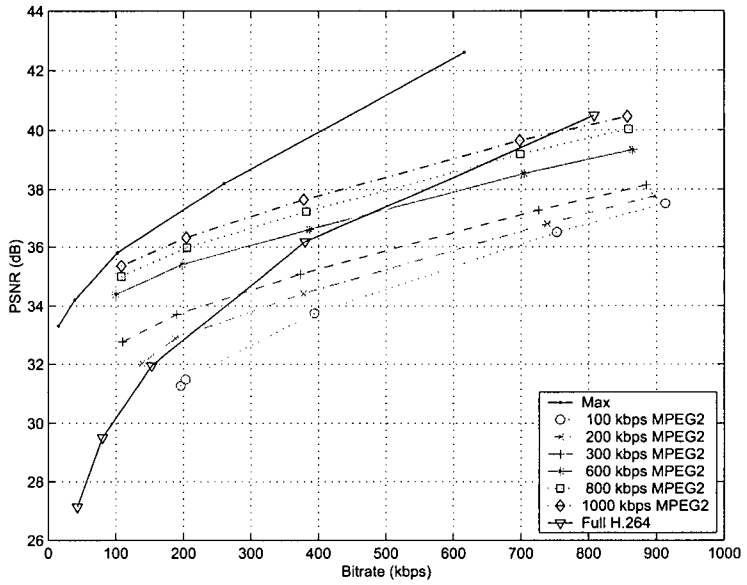


Figure 3.10: This graph shows the results for the proposed scalability scheme with the Silent Sequence

For a low bitrate base-layer, the new scheme uses approximately 93% of the bits as the original H.264 stream as at PSNR of 32dB. For higher bitrate base-layer sequences (1000 kb/s), the proposed scheme uses between 20% and 67% of the bits as the original H.264 stream, depending on the quality of the base-layer used.

Figure 3.10 depicts the bitrate reductions for this sequence graphically. As can be seen, many instances of Silent perform better when encoded using H.264 directly.

Sample images from the Silent sequence can be found in Appendix A.

Table 3.4: Shown here at the bitrate reductions possible using the Silent sequence. For a high-resolution PSNR value of 36 dB, the resultant bitstream uses approximately 67% of the bits as the original H.264 stream.

Base Layer	Enhancement Layer PSNR		
Bitrate (kb/s)	32dB	34dB	36dB
100			
200	93%		
300	46%	81%	
600	33%	46%	86%
800	27%	42%	75%
1000	20%	35%	67%

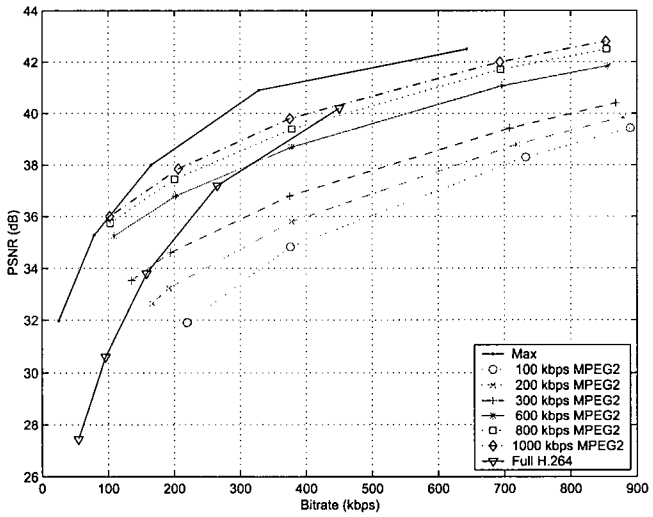


Figure 3.11: This graph shows the results for the proposed scalability scheme with the News sequence.

3.4.4 News Results

News is the final CIF image sequence tested using the proposed scalability scheme. It represents a fairly complex scene with multiple people and large

Table 3.5: Shown here are the bitrate reductions possible using the News sequence. For a high-resolution PSNR value of 36dB, the resultant bitrate uses approximately 72% of the bits as the original H.264 stream.

Base Layer	Enhancement Layer PSNR		
Bitrate (kb/s)	34dB	36dB	38dB
100			
200			
300	91%		
600	67%	74%	
800	62%	66%	76%
1000	59%	64%	72%

movement.

The results from the News sequence are depicted in Figure 3.11. As can been seen, low bitrate MPEG-2 sequences do not always yield a bitrate reduction using the proposed scheme. This is due to the high amount of blocking artifacts in the base-layer due to the high motion within the News sequence.

Table 3.5 shows the bitrate reductions for the News sequence. For a low-bitrate MPEG-2 base layer using 300 kb/s, the resultant bitstream is approximately 91% of the size of the original H.264 stream as a PSNR of 34 dB.

For a higher bitrate MPEG-2 base-layer at 1000 kb/s, the resultant bitstreams are between 59% and 72% of the size of the original H.264 streams.

Sample images from the News sequence can be found in Appendix A.

3.4.5 High Definition Enhancement Layer

A 1920x1080 high definition video sequence was obtained from Digital Film Group, a Vancouver based company. This image was obtained to attempt to validate the proposed scalability scheme when used for high-definition enhancement layers. A sample image from this sequence is shown in Figure 3.12. The sequence was resampled down to 720x480 to simulate a DVD digital broadcast for the base-layer. This layer was then encoded using two bitrates: 8 Mb/s and 10 Mb/s.

Table 3.6 shows the approximate reduction in bitrate for the high definition sequence using this new scalability scheme. These results are depicted graphically in Figure 3.13.

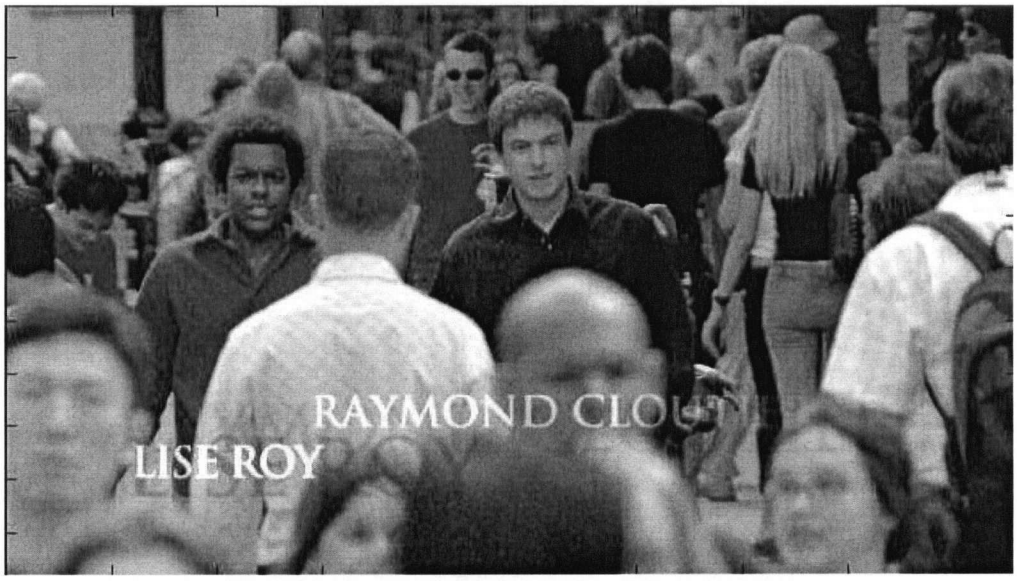


Figure 3.12: High Definition Sample Frame

Table 3.6 shows the enhancement layer bitstream sizes as a percentage

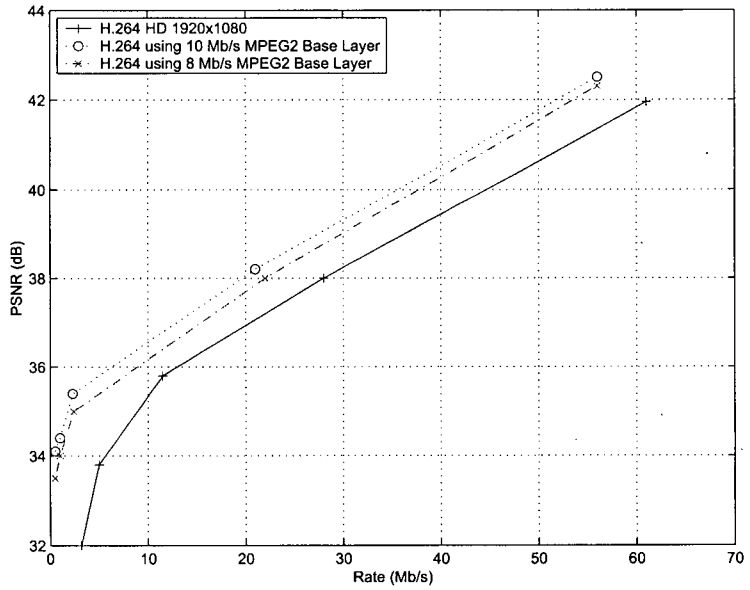


Figure 3.13: Result for High Definition Sequence

	High Resolution PSNR			
MPEG-2 Rate	36dB	38dB	40dB	42dB
8 Mb/s	75%	83%	85%	82%
10 Mb/s	68%	79%	83%	84%

Table 3.6: High Definition Rate Reduction Results

of a high-definition layer encoded using only H.264. As can be seen, for a high definition PSNR of 38 dB, and a base-layer bitrate of 10 Mbps, the resultant stream is approximately 79% of the original H.264 stream (or roughly a 20% reduction in bits).

3.5 Discussion

3.5.1 Frequency Content of Different Layers

In an effort to understand the behaviour of the proposed scalability scheme, a detailed frequency analysis is performed here. Figure 3.14 (a) shows the luminance value for the first frame in the Foreman sequence, while figure 3.14 (b) shows the frequency response for the first frame of the Foreman sequence. The frequency response shows a large amount of frequency content at low frequencies, with a few dominant high frequency bands primarily representing the sharp edges along the bricks in (a).

The first part of the scalability scheme involves an anti-aliasing step followed by a decimation step to form the base-layer image. This operation will cause slight aliasing (since it is impossible to have an ideal anti-aliasing filter in practice), as well as loss of high-frequency information above the Nyquist rate for the base-layer image. In the reverse operation, the decoded base-layer image is upsampled to the same resolution as the enhancement-layer image prior to forming the final combined image. Figure 3.14 (c) shows the image in (a) after is has been filtered, downsampled and then resampled: it represents the best approximation to (a) since no MPEG-2 distortion has been introduced.

Figure 3.14 (e) shows the spatial difference image formed by subtracting (a) from (c) and adding a DC offset to put the average image level at grey. It is evident that this image primarily contains high-frequency information

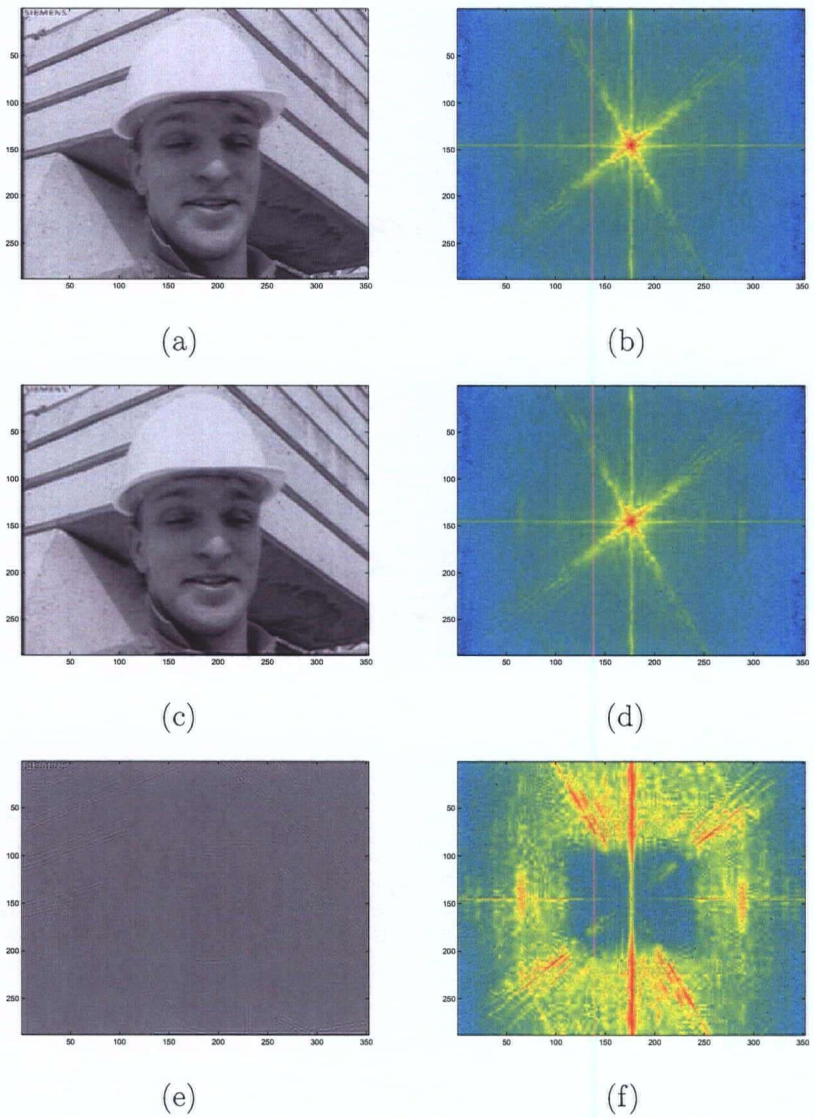


Figure 3.14: Shown here are several images that detail the frequency content of the foreman sequence. a) luminance plot b) Frequency content of luminance data c) Luminance of image that was first downsampled by a factor of two, then upsampled by a factor of two. d) Frequency content of (c) e) Spatial difference image of (a) and (c) f) Frequency content of (e)

which intuitively matches the information filtered out of (a) during the resizing process. Figure 3.14 (f) shows the frequency content of the spatial difference image in (e).

The success or failure of the proposed scheme essentially rests on the ability of the enhancement-layer encoder to be able to encode the frequency content shown in (f) using less bits than it would take to encode the content shown in (b) directy. Not only does (f) contain the high frequency information lost during the resizing operation, but also aliasing distortion not originally present in (a). This places an added burden on the encoder, since it must allocate bits to information that was not originally in the enhancement-layer image.

Figure 3.15 (a) shows the luminance component of the first frame from the Foreman sequence after undergoing MPEG-2 compression at 100kb/s. Some mild blocking artifacts can be noticed in this image. (b) shows the frequency content of (a), and is primarily composed of low-frequency information.

Figure 3.15 (c) shows the difference between figure 3.15 (b) and figure 3.14 (b). In essence, this is the frequency information that the enhancement-layer encoder must encode in order to reconstruct the original image. To gain an understanding of the added difficultly encoding this information, it can be compared against the original frequency content shown in Figure 3.14 (b). This result is shown in Figure 3.15 (d), and represents erroneous image information that must be encoded to properly restore (a) to the original image shown in

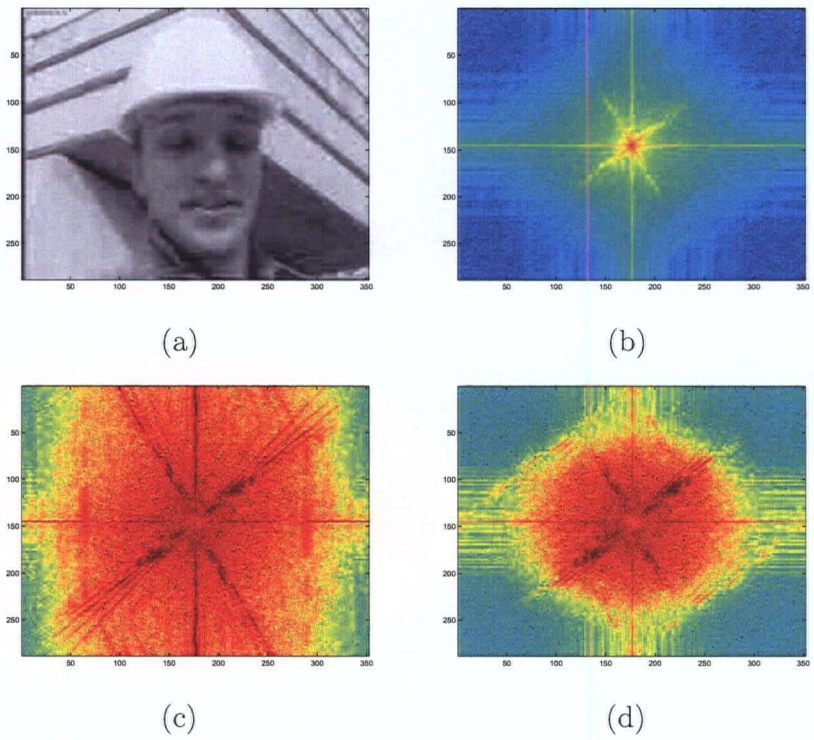


Figure 3.15: Analysis of the first frame from the Foreman Sequence after undergoing MPEG-2 compression. a) Image after undergoing MPEG-2 compression at 100kb/s b) the frequency content of (a). c) the frequency content difference between (b) and Figure 3.14 (a). d) Frequency information from (c) that was not originally contained in Figure 3.14 (b).

3.14 (a). This information is the cumulative result of DCT quantization errors inherent to the MPEG-2 encoding process, and the anti-aliasing filters used in the resizing operation.

3.5.2 Limit on Bitrate Reduction

Figure 3.16 shows the results of the MPEG-2 compression artifacts in the frequency domain. For example, (a) shows the difference in spectrum be-

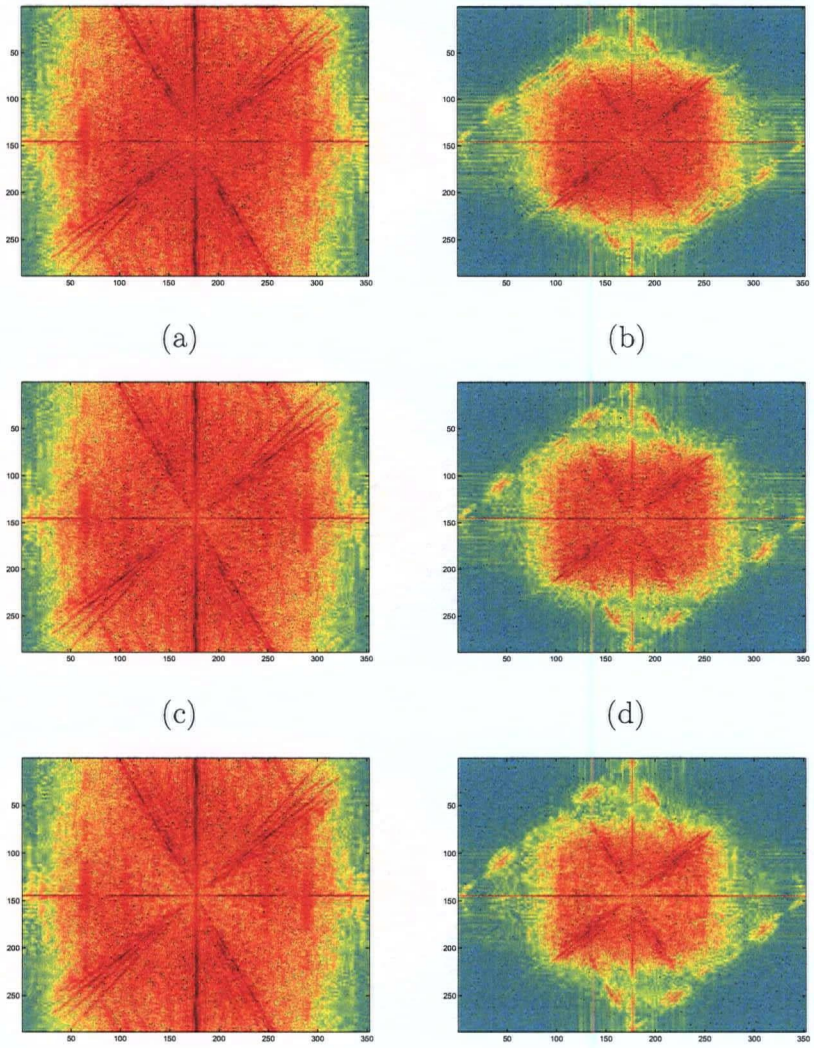


Figure 3.16: The figure shows the MPEG-2 compression artifacts in the frequency domain.

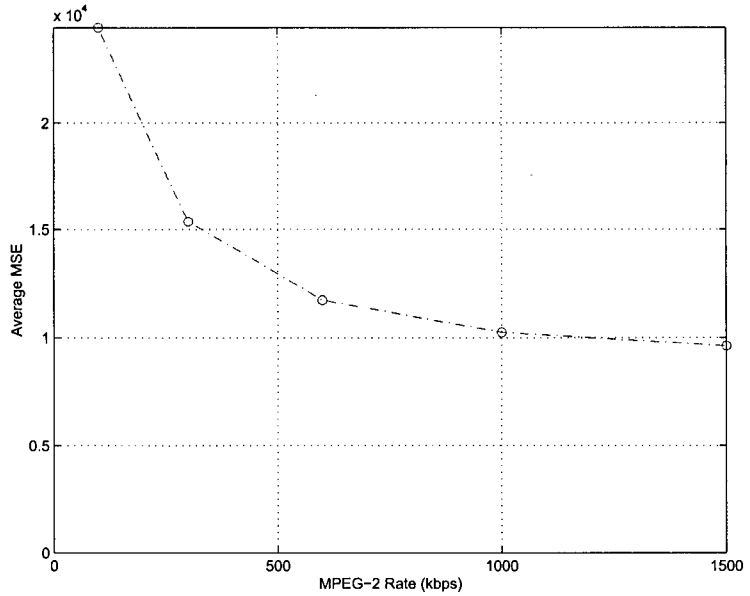


Figure 3.17: This figure represents the erroneous frequency information compared with vary amounts of MPEG-2 compression. The graph approaches a non-zero MSE, indicating a fundamental bitrate limit for this sequence.

tween Figure 3.14 (b) and the first frame of the Foreman sequence encoded at 300kb/s, while (b) shows the erroneous information in (a) that is not present in Figure 3.14 (d).

Comparing Figure 3.16 (f) with (b), it is evident that overall there is less speckle, especially around the center of the images.

Figure 3.15 (b), (d) and (f) show various levels of erroneous frequency information that must be encoded using the enhancement-layer encoder. We can use a mean-squared error metric to determine the amount of erroneous information present:

$$MSE = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |\tilde{F}(i,j) - \hat{F}(i,j)|^2 \tag{3.5}$$

where M is the width of the image, N is the height of the image, $\tilde{F}$ is the fourier transform of the MPEG-2 image, and $\hat{F}$ is the fourier transform of the zero-compression image (the one created by downsampling to the base-layer resolution, and then upsampling to the enhancement-layer resolution), similar to the one shown in Figure 3.14 (c).

Using equation 3.5, it is possible to come up with an error measure based on the MPEG-2 compression rates. The results for the Foreman sequence are shown in Figure 3.17. The non-zero asymptote indicates a lower-limit on the distortion introduced into the Foreman sequence due to the aliasing artifacts and then MPEG-2 compression artifacts due to DCT quantization.

3.6 Summary

In this chapter, we introduced a spatial scalability scheme that combined a MPEG-2 base layer with a H.264 ehancement layer. In general, this scheme resulted in a bitrate reduction for low-bitrate to moderate-bitrate H.264 streams. One benefit of this scheme is that it does not require any changes to either the MPEG-2 or H.264 bitstream formats: it can be created using off-the-shelf components.

In the next chapter, we investigate modifying the H.264 Intra prediction modes to allow prediction from another layer. Using this new ability, we

can increase the likelihood of a bitrate reduction, since each Intra frame can selective choose whether or not to include elements from the base-layer.

Chapter 4

Modified H.264 Scalability Scheme

4.1 Motivation

In chapter 3, a spatial scalability scheme was presented that often decreased the bitrate requirements of the high-definition layer. The main benefit of the presented scheme is that it works with independent codecs (in the case of the proposed scheme, MPEG-2 was combined with H.264), and does not require any bitstream modifications to be made. Unfortunately, by not modifying the bitstream of the enhancement-layer, an upperbound is placed on the quality achievable in the high-definition layer. This bound is ultimately due to aliasing distortion introduced during the resizing operations, and is also influenced by the distortion introduced in the base-layer encoding operation.

In this section, an alternate scalability scheme is presented based on

the methods discussed in chapter 3. This scalabilty involves modifying the bitstream for H.264, and is therefore not compatible with other implementations. These enhnacements could easily be added at a future date as an annex to the H.264 standard [1].

4.2 Method

In section 2.5, the various 4x4 Intra prediction modes are discussed for H.264. These modes are used for prediction within an Intra frame only. For our proposed scheme, an extension is proposed that would provide one-additional prediction mode. This new prediction mode will be called MPEG-2.

In essence, this prediction mode forms a prediction based on the exact 4x4 block in the upsampled base-layer. The benefit of forming a scalability scheme in this manner is that the encoder has the ability to use the base-layer prediction if it would result in the lowest amount of bits, and to use a normal H.264 prediction mode if the prediction formed from the base-layer requires too many bits to encode. In the scalability scheme proposed in chapter 3, the encoder was forced to encode elements of the base-layer in every block, which often resulted in additional bits being used to compensate for the base-layer errors.

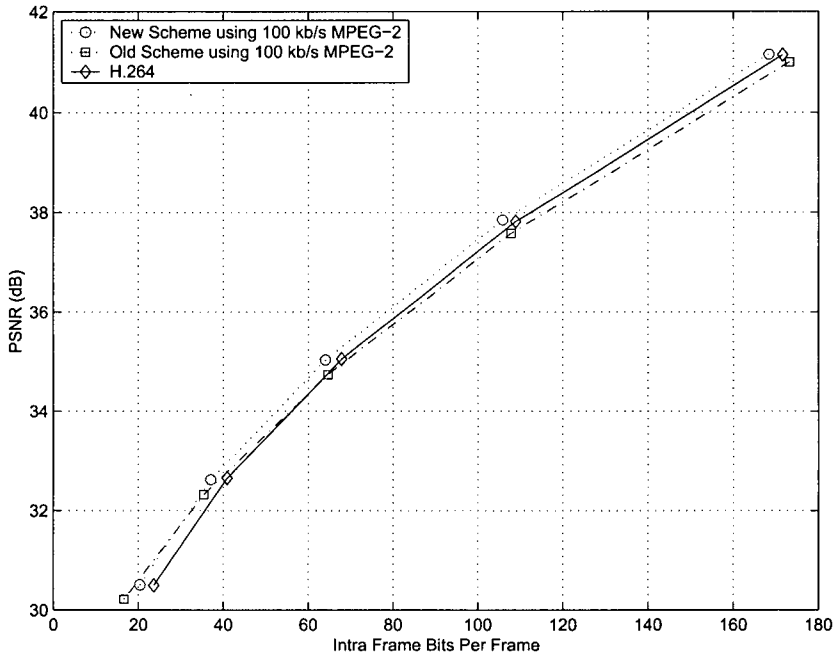


Figure 4.1: This figure shows the bitrate of the Foreman Intra-frames after addition of the MPEG-2 mode

4.3 Results

4.3.1 Intra-Prediction Mode Modifications

As a first step, the new MPEG-2 Intra prediction mode was added to the reference encoder. Figure 4.1 shows the results from the proposed scheme. The first evident feature of figure 4.1 is that while the bitrate of the 100 kb/s MPEG-2 based scalability scheme (based on the old method proposed in the previous chapter) is initially smaller, at higher enhancement-layer bitrates the scheme quickly breaks down, and the bitrate surpasses that of pure H.264. In contrast,

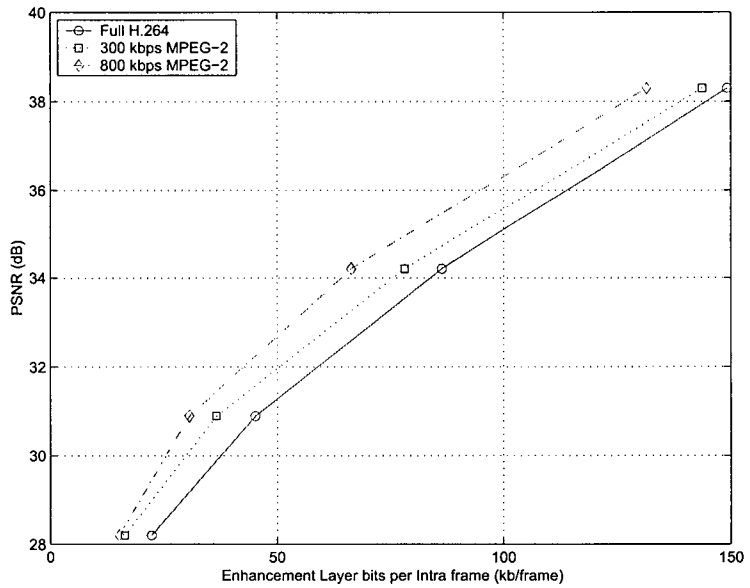


Figure 4.2: This figure shows the average bits per Intra frame for the Coastguard sequence after addition of the MPEG-2 mode

the new modifications show a reduced bitrate at each of the enhacement-layer bitrates shown in figure 4.1.

Figure 4.2 shows the results of the Coastguard sequence after addition of the MPEG-2 mode. Three rates are shown: Full H.264 (i.e. no scalability scheme used), a 300 kb/s MPEG-2 base-layer, and a 800 kb/s MPEG-2 layer. In all cases, the MPEG-2 layer results in a modest bit reduction per Intra frame.

Cost of new MPEG-2 mode

While adding a new Intra mode increases the flexibiliy of the H.264 encoder, it also adds an extra overhead in terms of bits. In H.264, the encoder expects to

have a large amount of redundancy between adjacement Intra modes. When the mode for a particular block is the same as the adjacent blocks, the encoder can signal this to the decoder using only one bit. If, however, the mode is different, the encoder takes a penalty, and must write out the actual mode that was used. The cost of writing out an inaccurate estimate is four bits.

To estimate the number of bits required for each Intra-block, we can use the relation:

$$N_b = P(S_n|S_{n-1}) + 4(1 - P(S_n|S_{n-1})) \tag{4.1}$$

Equation 4.1 shows that the number of bits required in a function of the accuracy of the Intra-mode prediction mechanism. Typical prediction accuracies for the default H.264 scheme are between 30% and 70% (see section 5.3). For an accuracy of 50%, the average number of bits used per Intra-block is 2.5. For a CIF image, there are 6336 blocks. Using the average of 2.5 bits per block, there are approximately 15840 bits in each Intra-frame dedicated to which Intra-prediction mode was used.

By adding an extra mode, the encoder must now use five bits for a prediction miss, instead of four. This yields the new bit cost for Intra frame as:

$$N_b' = P(S_n|S_{n-1}) + 5(1 - P(S_n|S_{n-1})) \tag{4.2}$$

Using the same metric of 50% for the accuracy, the addition of an extra Intra mode yields an average of 3.0 bits per Intra block, for a total of 19008

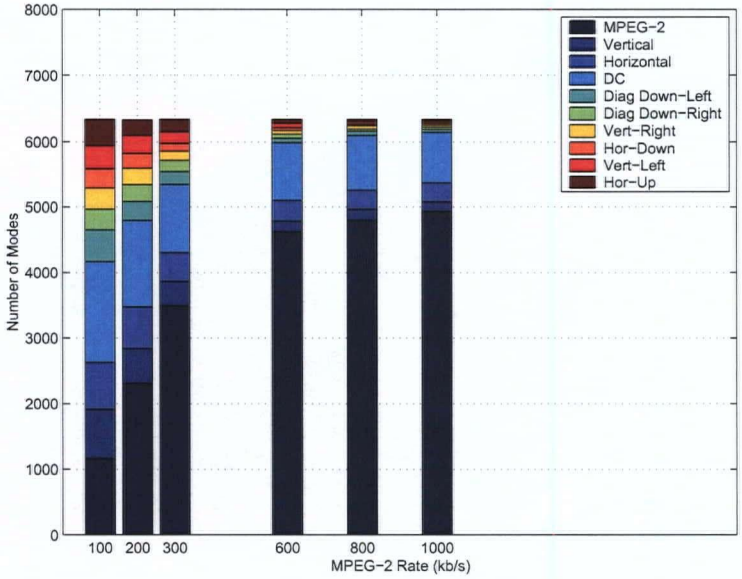


Figure 4.3: This figure shows the selection of Intra prediction modes for a fixed enhancement layer bitrate, but varying degrees of base-layer MPEG-2 bitrates

bits per Intra frame. The cost of adding an additional Intra mode is 3168 bits per frame when the Intra-prediction accuracy is 50%.

Selection of Modes for Fixed Enhancement-Layer Rate

Figure 4.3 shows a stacked graph detailing the internal four by four Intra prediction modes chosen by the encoder for this scheme. For low-bitrate MPEG-2 streams, such as the 100 kb/s stream, the encoder only choses the MPEG-2 mode 18% of the time, indicating that for the other 82% of the time, it was more advantageous to not base a prediction on the base-layer. As the bitrate of the MPEG-2 stream increase, the encoder predominantly chooses the MPEG-2 mode more often. For the highest bitrate MPEG-2 stream measured, 1000

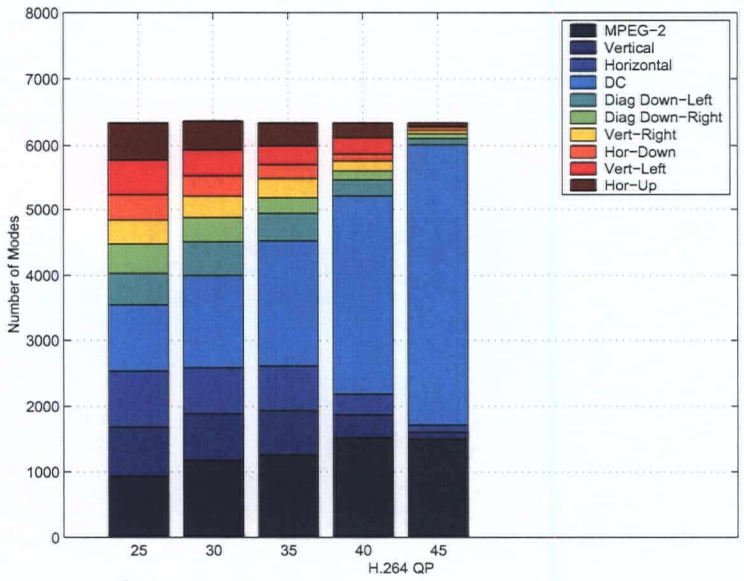


Figure 4.4: This figure shows the selection of Intra prediction modes for a fixed base-layer bitrate, but varying degrees of enhancement-layer bitrates (lower QP indicates higher bitrate)

kb/s, the encoder chose the MPEG-2 mode 78% of the time.

Selection of Modes for Fixed Base-Layer Rate

Figure 4.4 shows a stacked graph detaliing the internal four by four Intra prediction modes chosen by the encoder when a fixed MPEG-2 base-layer rate of 100 kb/s was chosen. This is a very low bitrate, which visually has a large amount of blocking artifacts. It is noteworthy that the amount of blocks that use the MPEG-2 mode is nearly constant across all enhancement-layer bitrates. This implies that the amount of modes chosen for the MPEG-2 mode is ultimately dominated by the amount of compression in the base-layer. At

high bitrates (low QP), the Intra prediction modes are nearly equal, with the MPEG-2 and DC modes occuring the most.

4.4 Summary

In this section, a new 4x4 Intra-mode is proposed. The purpose of this mode is to allow a 4x4 Intra prediction to occur from a 4x4 block in a base-layer. The addition of this mode allows the modified H.264 encoder to selectively utilize a base-layer macroblock, or choose not to use it if encoding the block using H.264 directly would result in fewer bits used.

Using this scheme, an additional bitrate reduction on the order of 20% for the CIF sequences tested is achieved. In terms of the proposed scalability schemes, these changes allow for improved compression since the encoder can selectively choose whether or not to include a block from the base-layer.

For video sequences that contain periodic Intra frames (such as DVD video or real-time video conferencing data), the bitstream size is largely dominated by Intra frames. For this reason, no changes are proposed for P or B frames, since the bitrate reductions would not justify the complexity changes involved by modifying the stored picture buffer of the encoder and decoder.

While investigating these the MPEG-2 mode and 4x4 Intra prediction in general, it became apparent that the 4x4 Intra-mode prediction mechanism in H.264 can be enhanced to give improved accuracy. This improvement in accuracy would further enhance the bitrate reductions possible with the proposed scalability scheme, and is explored in the next chapter.

Chapter 5

Intra Mode Prediction Modifications for the H.264 Standard

In the previous chapter, a new 4x4 Intra mode called MPEG-2 was discussed that would enhance the proposed scalability scheme. While investigating the addition of this mode, it became apparent that the Intra-mode prediction mechanism in H.264 can be improved. This improvement in prediction accuracy leads directly to a reduction in bits for our proposed schemes.

Ever since H.263+, encoders have been able to do predictions within each Intra frame. H.264 is no exception, and also has this ability. Prior to H.264 and H.263+, most video encoders encoded macroblocks within an intra frame completely independent of other macroblocks. While this makes each intra frame more resiliant to error, it does not exploit the spatial redundancy

between each macroblock.

With H.264, each macroblock can be encoded independently, or it can be predicted from an adjacement macroblock. This prediction is determined by a particular mode. For 4x4 macroblocks, nine modes are available for prediction, depending on which adjacement macroblocks are available.

5.1 4x4 Intra Prediction

5.1.1 Choosing the Intra Mode

When using Intra-mode prediction, the encoder first makes an estimate of the most-probable prediction mode for a particular macroblock. This prediction is called the most probable prediction mode, or MPP. Once the MPP mode for a particular MB is determined, the encoder cycles through all predictions to determine which mode results in the lowest distortion. The encoder determines this by computing the sum of absolute differences (SAD), which is given by:

$$V(m) = \sum_{m=0}^{N_M-1} \sum_{x=0}^{R-1} \sum_{y=0}^{R-1} |I(x,y) - \hat{I}(m,x,y)| \tag{5.1}$$

where x and y represent the coordinates within a particular MB, R represents the size of a macroblock, in pixels, $I(x,y)$ represents a pixel value in the source image at location (x,y), $\hat{I}(m,x,y)$ represents the encoding estimate of the MB value at location (x,y) based on prediction mode m, and N_M is the number of intra 4x4 predictions modes.

The mode m that is used is m such that the SAD function $V(m)$is minimized. If the chosen prediction mode is the same as the MPP mode, a bit

savings is achieved since only one bit needs to be written to the stream. If, however, the MPP mode is incorrect, the encoder must write out the actual mode used, which results in a bit penalty. The relationship between accuracy and average bits needed for the MPP is discussed in section 4.3.1.

5.2 New Statistical Methods

The current method of determining the MPP mode is based on a simple comparison of the modes used in the upper and previous MBs (if they exist). While this is computationally trivial, it assumes a stationary statistical distribution of these modes, and does not take into account any spatially-dependant correlation between MB modes.

Figure 5.1 shows the accuracy of the current scheme used by the H.264 standard. As can be seen, for high QP values (which represent high compression), the accuracy of the prediction mechanism is close to 85%. For lower QP values (which represent less compression), the accuracy of the current scheme is approximately 25% for these sequence.

To improve this scheme, our method uses the statistics accumulated in a Intra frame or slice to better estimate this mode. Since the intra predicted mode that will result in the lowest SAD for a particular MB is correlated with the modes in the adjacent MBs, we set out to determine:

$$E[s(i,j)|s_{(i-1,j)}, s_{(i,j+1)}] = \sum_{i=0}^{N_M-1} P(s|s_i, s_j)i \tag{5.2}$$

where $E[s(i,j)|s_{(i-1,j)}, s_{(i,j+1)}]$ represents the expected value of the MPP mode

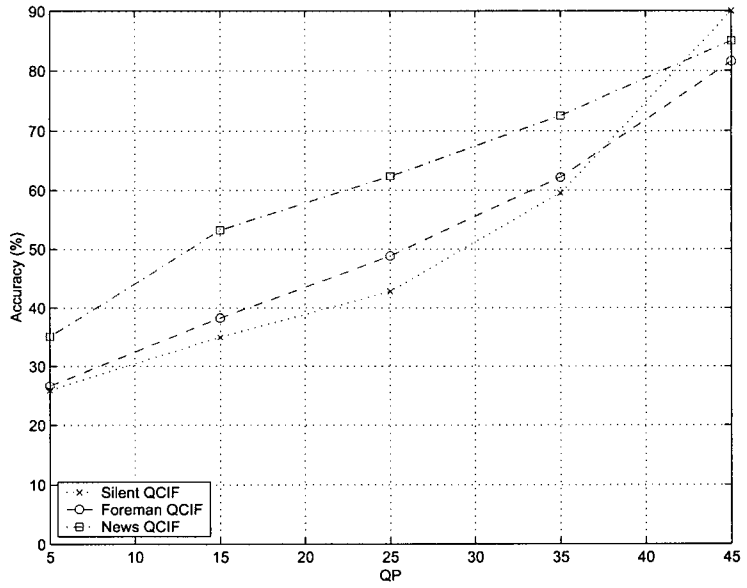


Figure 5.1: The Intra-mode prediction accuracy of H.264 is shown here. For low values of QP, the prediction accuracy is around 80%. For high values of QP, the prediction accuracy approaches 25%.

for the 4x4 Intra-block at position (i, j) when the previous 4x4 Intra-block modes for the adjacent 4x4 Intra blocks $s_{(i-1,j)}$ and $s_{(i,j+1)}$ are known. This can be estimated in real time by computing:

$$P(s|s_l, s_u) = \frac{M_P[s_l][s_u][s]}{\sum_{j=0}^{N_M} M_P[s_l][s_u][j]}; \tag{5.3}$$

where $P(s|s_l, s_u)$ represents the probabilty that a 4x4 Intra block is a certain type when the modes of the previous 4x4 Intra blocks (s_l and s_u) are known, M_P is a matrix with dimensions (N_M, N_M, N_M). Equations 5.2 and 5.3 can be combined to yield the new MPP mode:

$$E[s|s_{(i-1,j)}, s_{(i,j+1)}] = \sum_{i=0}^{N_M-1} i \frac{M_P[s_l][s_u][s]}{\sum_{j=0}^{N_M} M_P[s_l][s_u][j]}; \tag{5.4}$$

This algorithm was coded into the JVT reference software, version 8.4.

5.2.1 Encoder and Decoder States

For this algorithm to work, it is imperative that the encoder and decoder always predict the same value of a MB mode given the same initial conditions. To achieve this, the state matrices must be seeded with initial values that are known to both the encoder and decoder. For this paper, the initial probabilities were determined by using the currently implemented algorithm in H.264, which makes a simple prediction based on the adjacent MBs.

5.2.2 Statistical Memory

The statistical distribution represented by equation 5.4 will be updated over the course of the entire video sequence encoded. To limit the memory of the distribution, the probabilities within the distribution are reset at the start of each intra frame. This allows the encoder and decoder a chance to synchronize if a intra frame is lost.

While refreshing the distribution is obvious at frame boundaries, we also investigated increasing this refresh interval to encompass MB and GOB boundaries. Three types of adjustments were investigated for each refresh interval:

1. Scaling the entire distribution by a factor of $\frac{1}{2}$
2. Scaling the entire distribution by a factor of $\frac{1}{4}$

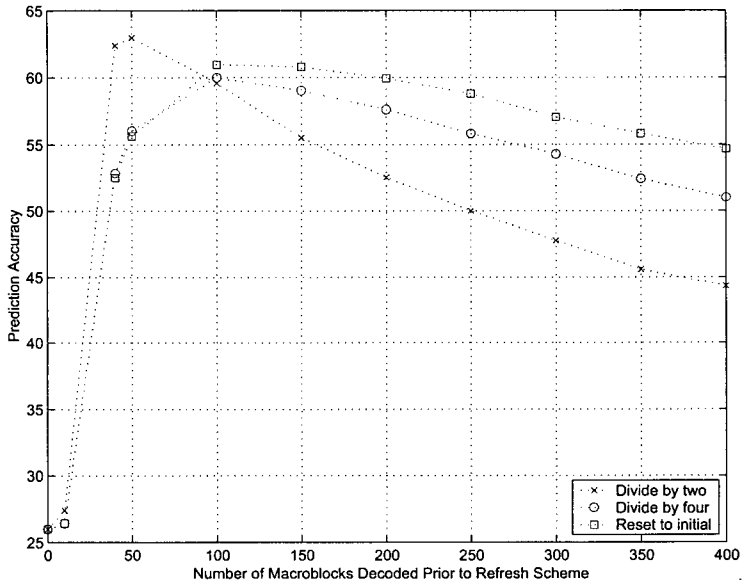


Figure 5.2: In this figure, the refresh scheme for the statistical information is presented. Tbe best performance occurred when all accumulated data was divided by two approximately every fifty macroblocks.

3. Resetting the entire distribution to zero

5.3 Results

Figure 5.2 shows the results of the statistical memory investigation for the Silent QCIF sequence. As is evident, the greatest accuracy was achieved by scaling the entire distribution by two after every 50 MBs. Resetting the distribution approximately every 100 MBs also produced excellent results, and also performed better at higher refresh intervals. This method is also more resilient to errors, since the encoder and decoder can synchronize at multiple

locations within each frame: this scheme was adopted for all experiments with this scheme.

5.3.1 Improvement in Accuracy

A simple method to determine the accuracy of the new statistical method is to compute an accuracy metric M:

$$M = \frac{N_{hits}}{N_{hits} + N_{misses}} \tag{5.5}$$

Figure 5.1 depicts the Intra 4x4 prediction accuracy using the current scheme in H.264. While the accuracy is acceptable for low-bitrates (around 80% for a QP of 45 for all three sequences), the current prediction implementation performs quite poorly at higher bitrates – the accuracy approaches 30% for a QP of 15 for the Silent sequence.

Figure 5.3 shows the prediction accuracy of the proposed scheme. While the gain at low bitrates is notable, there is a substantial increase in prediction accuracy at high bitrates.

Figure 5.4 demonstrates the improvement of the proposed method over the current method in H.264. At high bitrates, the improvement is as much as 100%.

The improvement in prediction accuracy directly leads to a decrease in the bits required for each intra frame. Figure 5.3 shows the resulting average bit savings for each of the three sequences tested. For high bitrate sequences, nearly 2000 bits are saved per Intra frame using the proposed method – for

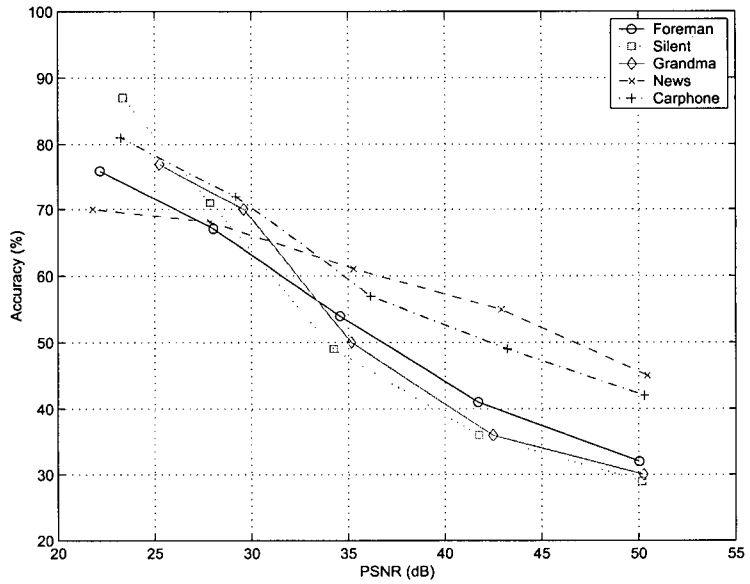


Figure 5.3: This figure shows the prediction accuracy of the new scheme.

this sequence, where the bits per Intra frame varied between 10000 and 40000 bits, the bit savings that results is between 5% and 20%, depending on the level of compression used.

5.4 Computational Complexity

The current implementation of the H.264 encoder makes an intra-prediction, then proceeds to calculate the bitrates needed for each of the other intra predicted modes. Based on the rate-distortion information obtained at this step, the encoder makes a final decision regarding which intra mode to use. If certain modes are statistically determined to have a low probability of occurrence, then these calculations can be skipped by the encoder, resulting in a

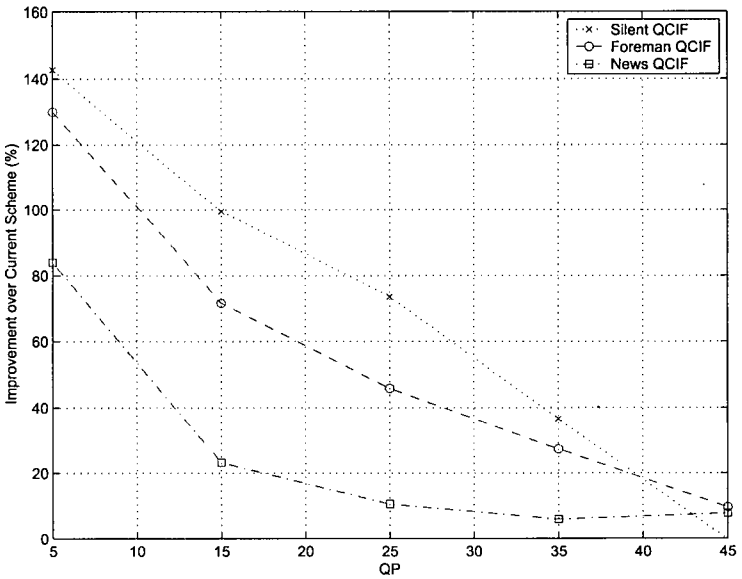


Figure 5.4: This figure shows the prediction accuracy improvement of the new scheme when compared to the old scheme.

speed increase for intra-frame encoding.

5.5 Adjusting for the Dominant Mode

Since the distribution is reset to zero at the start of each intra frame, there are no statistics for which to base future Intra-mode predictions on. In this scenario, the encoder defaults to the original method implemented currently within H.264. This method is

$$MODE(x) = \begin{cases} \text{DC MODE} & \text{if either macroblocks is unavailable} \\ \text{UP MODE} & \text{if UP MODE less than LEFT MODE} \\ \text{LEFT MODE} & \text{otherwise} \end{cases} \tag{5.6}$$

While the method represented by equation 5.6 is simplistic, it does take into account the local macroblock statistics, and forms the basis for the default scheme in H.264.

We can define the dominant mode for each prediction method as

$$M_{DOM} = max(MPP[i][j][k][i = 0..N_M - 1]) \tag{5.7}$$

$$P_{DOM} = \frac{\sum_{i=0}^{N_M-1}\sum_{j=0}^{N_M-1}\sum_{k=0}^{N_M-1} max(MPP[i][j][k])}{\sum_{i=0}^{N_M-1}\sum_{j=0}^{N_M-1}\sum_{k=0}^{N_M-1}\sum_{m=0}^{N_M-1} MPP[i][j][k][m]} \tag{5.8}$$

5.6 Summary

In this section, a novel Intra-mode prediction method was proposed that involved tracking the Intra-frame statistics for Intra-block modes. Using this new scheme, accuracy improvements of up to 100% are achievable over the default H.264 method. This trade off comes at the expense of reduced ability to handle errors within a frame, and a slight computational increase.

Chapter 6

Summary and Future Work

6.1 Summary

With the recent completion of the H.264 standard, and the certain increase in high-definition capable televisions in the future, H.264-based spatial scalability schemes are promising areas of research. In this thesis, our main contribution is a new scalability scheme that combines a MPEG-2 base layer with a H.264 high resolution enhancement layer. To demonstrate the effectiveness of our proposed scheme, several CIF sequences were evaluated and compared against bitstreams compressed directly with H.264. For these sequences, bitrate reductions of around 25% are obtainable for moderate quality MPEG-2 base-layers (600 kb/s). For higher-quality base-layers (of between 800 kb/s and 1000 kb/s), bitrate reductions of approximate 50% are obtainable for these sequences. Our proposed scalability scheme does not involve modifying the H.264 or MPEG-2 bitstreams, and can therefore be used with off-the-shelf

components.

In chapter 4, an alternate scalability scheme is proposed that requires modifications to the H.264 bitstream syntax. In particular, a new Intra-mode prediction method called "MPEG-2" is proposed. This mode allows the encoder to base a particular block in the enhancement-layer on a block in the base-layer, or to selectively ignore it if using the base-layer block would require additional bits. Although this approach causes a modest bit overhead due to signalling an extra Intra mode, it results in significant bitrate reductions for Intra frames of up to 25%, and further enhances our proposed scalability scheme.

In chapter 5, further improvements to our scalability scheme were explored by modifiying the Intra-mode prediction scheme within H.264. Our contribution in this area is a new scheme which involves tracking the local Intra-mode statistics for a particular frame, and basing a prediction on this information. Using our proposed changes, accuracy improvements of between 10% and 140% were demonstrated, depending on the sequence used, and the desired quality of the encoded video. This improvement in prediction accuracy leads directly to a reduction in the bits required for each Intra-frame, since less bits are used to indicate the result of an inaccurate prediction. The reduction in bits further enhances our proposed scalability schemes, reducing the bits required for the enhancement layers.

6.2 Future Work

In chapter 3, a spatial scalability scheme is proposed that demonstrated a bitrate reduction when the base-layer was of a sufficiently high quality. If this scheme is used when the quality of the base-layer is low, it may yield a slight increase in bitrate, indicating that the scheme is inefficient in these scenarios.

The analysis done in chapter 3 shows that the ultimate limit on bitrate reduction is asymptotically dependant on the aliasing and quantization distortion inherent in the base-layer. More work should be done to understand this fundamental limit, and to obtain a quantitative metric for when this scheme should and should not be used.

For the proposed scalability scheme in chapter 4, it was shown that a bitrate reduction can be obtained by adding a new Intra-mode that corresponds to a prediction from a block in the base-layer. This addition adds a modest bit overhead to the stream, but in general, but allows for a modest bit-reduction for Intra-frames when used with a base-layer. The main cause of the bit overhead is because the addition of a new mode causes 4 bits to be used instead of 3 bits for an incorrect prediction. Based on the results in chapter 4, it may be possible to replace one of the less likely Intra-modes with the MPEG-2 Intra-mode when the scheme is used. This would reduce the 1-bit penalty for using the MPEG-2 mode, and increase the attractiveness of the scheme.

In addition, it is clear that work needs to be done to understand how to increase the bit reductions with Predictive and Bi-predictive frames. Since the

majority of bit reduction for video compression comes from these two frames, it is important that they also contribute to a bitrate reduction for a scalability scheme.

For the Intra-prediction mode changes proposed in chapter 5, it may be possible to realize additional bit savings by exploiting the fact that the transition between modes is correlated. In the current scheme, an inaccurate prediction causes one bit to be written (indicating that it was inaccurate), followed by three bits, indicating which mode is the correct one. If it is known which modes are more accurate than others, it may be possible to use entropy-coding for the inaccurate prediction mode. In the case of Exp-Golomb codes, if only two modes are dominant, then the cost of a inaccurate prediction can be written using only two bits instead of four. Depending on the statistics of the Intra-frame, this may or may not result in a bitrate savings, but it is definitely worth exploring.

Bibliography

[1] T. Wiegang, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), July 2003.

[2] B. Rulon, M. Shaw, and K. Donohue. A Comparison of Audio Compression Transforms. *Southeastcon '99. Proceedings. IEEE*, pages 253–257, March 1999.

[3] G. Lilienfield and J. Woods. Scalable High Definition Video. *Image Processing, 1995. Proceedings., International Conference on*, 2:567–570, October 1995.

[4] D. Kelly, W. Van Gestel, T. Hamada, M. Kato, and K. Nakamura. Blu-Ray Disc - A Versatile Format for Recording High Definition Video. 2003.

[5] D. Huang, T. Jeng, G. Wu, H. Yang, Y. Yung, and J. Ju. New Optical Storage Formats Developed In Taiwan. *IEEE Transactions on Magenetics*, 41(2), February 2005.

[6] K. Rao and Y. Huh. JPEG 2000. *Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom*, pages 1–6, June 2002.

[7] L. Chang, C. Wang, and S. Lee. Designing JPEG Quantization Tables Based on Human Visual System. *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, 2:24–28, October 1999.

[8] S. Deshpande and J. Hwang. A New Fast Motion Estimation Method Based on Total Least Squares for Video Encoding. *Acoustics, Speech, and Signal Processing, 1998. ICASSP '98. Proceedings of the 1998 IEEE International Conference on*, 5:2797–2800, May 1998.

[9] Y. Liang, I. Ahmad, J. Luo, and Y. Sun. Fast Motion Estimation Using Hierarchical Motion Intensity Structure. *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, pages 699–702, June 2004.

[10] D. Yu, S. Jang, and J. Ra. A Fast Motion Estimation Algorithm for MPEG-4 Shape Coding. *Image Processing, 2000. Proceedings. 2000 International Conference on*, 1:876–879, September 2000.

[11] S. Brofferio and F. Rocca. Interframe Redundancy Reduction of Video Signals Generated by Translating Objects. *IEEE Transactions on Communications*, 25:448–455, Apr 1977.

[12] W. Di, W. Gao, H. Mingzeng, and J. Zhenzhou. An Exp-Golomb Encoder and Decoder Architecture for JVT/AVS. *ASIC, 2003. Proceedings. 5th International Conference on*, 2:910–913, October 2003.

[13] A. Jerbi. Emerging H.26L Standard. 2002.

[14] R. Osorio and J. Bruguera. Arithmetic Coding Architecture for H.264/AVC CABAC Compression System. *Digital System Design, 2004. DSD 2004. Euromicro Symposium on*, pages 62–69, September 2003.

[15] A. Ahmad, N. Khan, D. Masud, and M. Maud. Performance Evaluation of Advanced Features of H.26L Video Coding Standard. *Multi Topic Conference, 2003. INMIC 2003. 7th International*, pages 141 – 145, December 2003.

[16] N. Kamaci and Y. Altunbasak. Performance comparison of the emerging H.264 video coding standard with the existing standards. *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, 1:345–358, July 2003.

[17] M. Hong. An Efficient Loop/Post Filter to Reduce Annoying Artifacts of H.26L Video Coder. *Consumer Electronics, 2000. ICCE. 2000 Digest of Technical Papers. International Conference on*, pages 240–241, June 2000.

[18] S. Tai, Y. Chen, and S. Sheu. Deblocking filter for low bit rate mpeg-4 video. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(6), June 2005.

[19] Y. Cheng, Z. Wang, J. Guo, and K. Dai. Research on Intra Modes for Inter-Frame Coding in H.264. *Computer Supported Cooperative Work in Design, 2005. Proceedings of the Ninth International Conference on*, 2:740–744, May 2005.

[20] T. Stockhammer, M. Hannuksela, and S. Wenger. H.26L/JVT Coding Network Abstraction Layer and IP-based Transport. *Image Processing. 2002. Proceedings. 2002 International Conference on*, 3:485–488, September 2002.

[21] Q. Peng and Y. Zhao. Study on Parallel Approach in H.26L Video Encoder. *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pages 834–837, August 2003.

[22] V. Lappalainen, A. Hallapure, and T. Hamalainen. Optimization of emerging H.26L video encoder. *Signal Processing Systems, 2001 IEEE Workshop on*, pages 406–415, September 2001.

[23] T. Wang, P. Tseng, and L. Chen. H.26L Intra Mode Encoder Architecture for Digital Camera Application. *Consumer Electronics, 2001. ICCE. International Conference on*, pages 132–133, June 2001.

[24] Performance Analysis of Hardware Oriented Algorithm Modifications in H.264. *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, 2:493–496, April 2003.

[25] V. Lappalainen, A. Hallapuro, and T. D. Hamalainen. Complexity of Optimized H.26L Video Decoder Implementation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13, July 2003.

[26] Y. Charfi and R. Hamzaoui. Packet Loss Protection of Scalable Video Bitstreams Using Forward Error Correction and Feedback. *Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on*, 1:370–374, September 2003.

[27] M. van der Schaar and H. Radha. A Hybrid Temporal-SNR Fine Granular Scalability for Internet Video. *IEEE Transactions on Circuits and Systems for video technology*, 11(3), March 2001.

[28] W. Peng and Y. Chen. Mode-Adaptive Fine Granularity Scalability. *Image Processing, 2001. Proceedings. 2001 International Conference on*, pages 993–996, October 2001.

[29] K. Ugur and P. Nasiopoulos. Combining Bitstream Switching and FGS for H.264 Scalable video Transmission over Varying Bandwidth Networks. *Communications, Computers and signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on*, 2:972–975, August 2003.

[30] Y. He, Li S., Y. Zhong, and S. Yang. H.26L-based Fine Granularity Scalable Video Coding. *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, 4:548–551, May 2002.

[31] Fine granular scalability for H.26L-based video streaming. *Consumer Electronics, 2002. ICCE. 2002 Digest of Technical Papers. International Conference on*, pages 346–347, June 2002.

[32] Y. He, F. Wu, S. Li, Y. Zhong, and S. Yang. H.26L-Based Fine Granular Scalable Video Coding. *Proceedings of IEEE International Symposium on Circuits and systems (ISCAS)*, 4:548–551, May 2002.

[33] B. Wang, X. Gu, and H. Zhang. An Improvement to Fine Granularity Scalability Based on H.26L. 3:833–836, May 2004.

[34] W. Peng, T. Chiang, and H. Hang. Context-based Binary Arithmetic Coding For Fine Granuality Scalability. *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, 1:105–108, July 2003.

[35] C. Zhu, Y. Gau, and L. Chau. Reducing Drift for FGS Coding Based on Multiframe Motion Compensation. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, 3:253–256, May 2004.

[36] O. Harmanci and A. M. Tekalp. A Zero Error Propagation Extension to H264 for Low Delay Video Communications Over Lossy Channels.

Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, 2:185–188, March 2005.

[37] N. Laurent, A. Buisson, and S. Brangoula. A Hybrid Mesh-H264 Video Coder. *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, 3:865–8, September 2003.

[38] Q. Hu and S. Panchanathan. A comparative evaluation of spatial scalability techniques in the compressed domain. *1996 Canadian Conference on Electrical and Computer Engineering*, 1:474 – 477, May 1996.

[39] Q. Li, H. Cui, and K. Tang. Fine granularity scalability video coding algorithm. *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*, 1:5–9, June 2002.

[40] R. Kurceren and M. Karczewicz. Synchronization-Predictive Coding for Video Compression: The SP Frames Design for JVT/H.26L. *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2:497–500, September 2002.

[41] R. Dugad and N. Ahuja. A Scheme for Spatial Scalability Using Non-scalable Encoders. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(10):923–99, October 2003.

Appendix A

Reference Images

This appendix includes reference images for the video sequences used in this thesis.

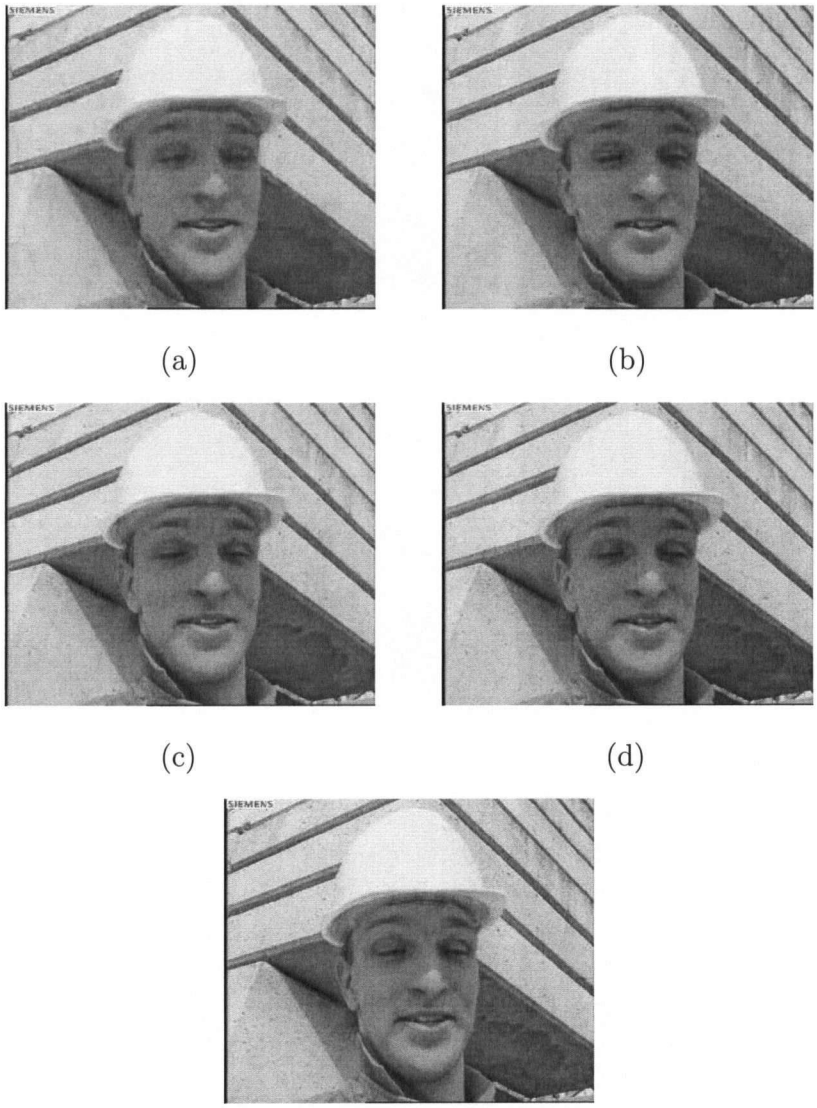


Figure A.1: Shown here are several Foreman reference images. a) PSNR of 32dB b) PSNR of 34dB c) PSNR of 36dB d) PSNR of 38dB e) Original Image

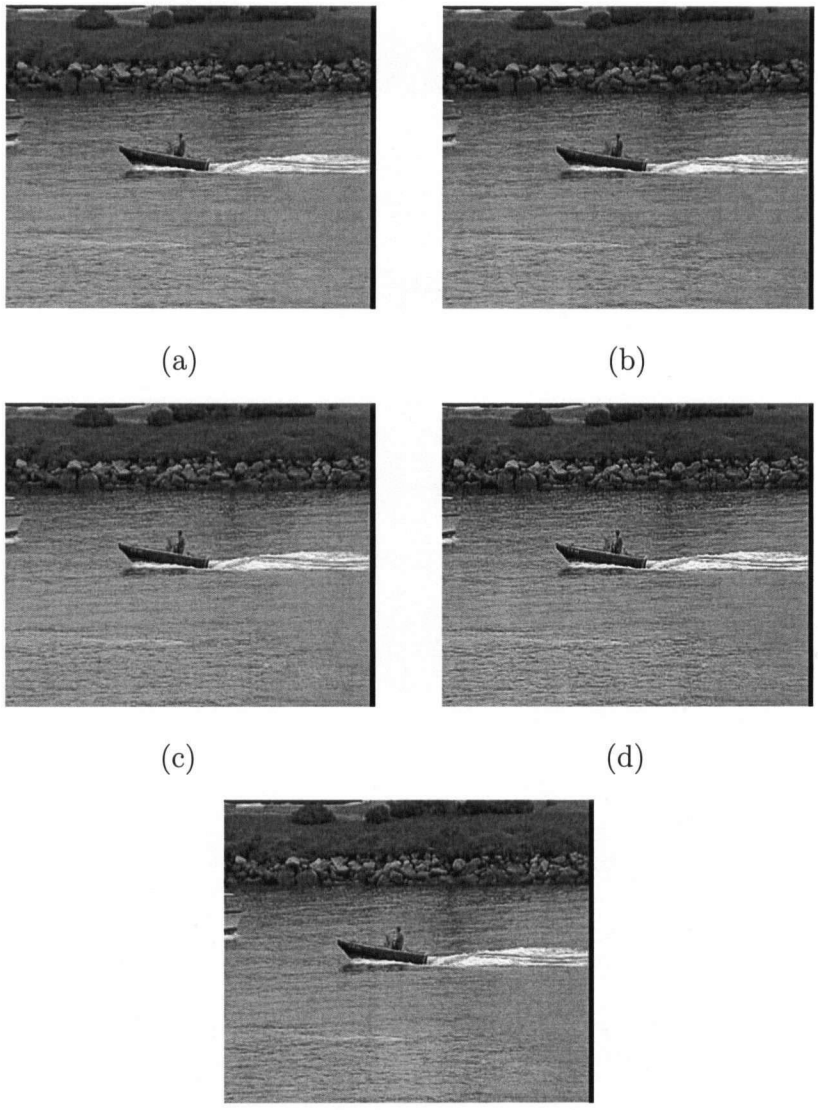
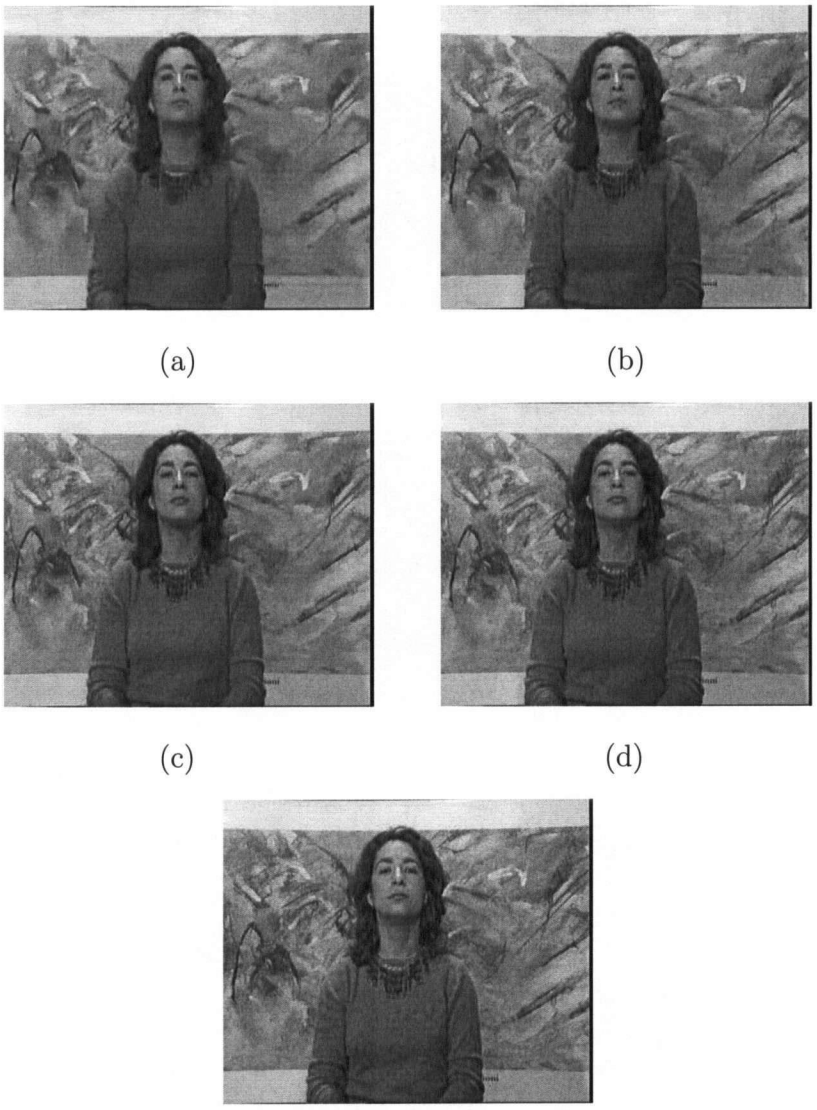


Figure A.2: Shown here are several Coastguard reference images. a) PSNR of 30dB b) PSNR of 32dB c) PSNR of 34dB d) PSNR of 34dB plus post-processing sharpening e) Original Image



(a) (b)

(c) (d)

(e)

Figure A.3: Shown here are several Silent reference images. a) PSNR of 30dB b) PSNR of 32dB c) PSNR of 34dB d) PSNR of 36dB e) Original Image

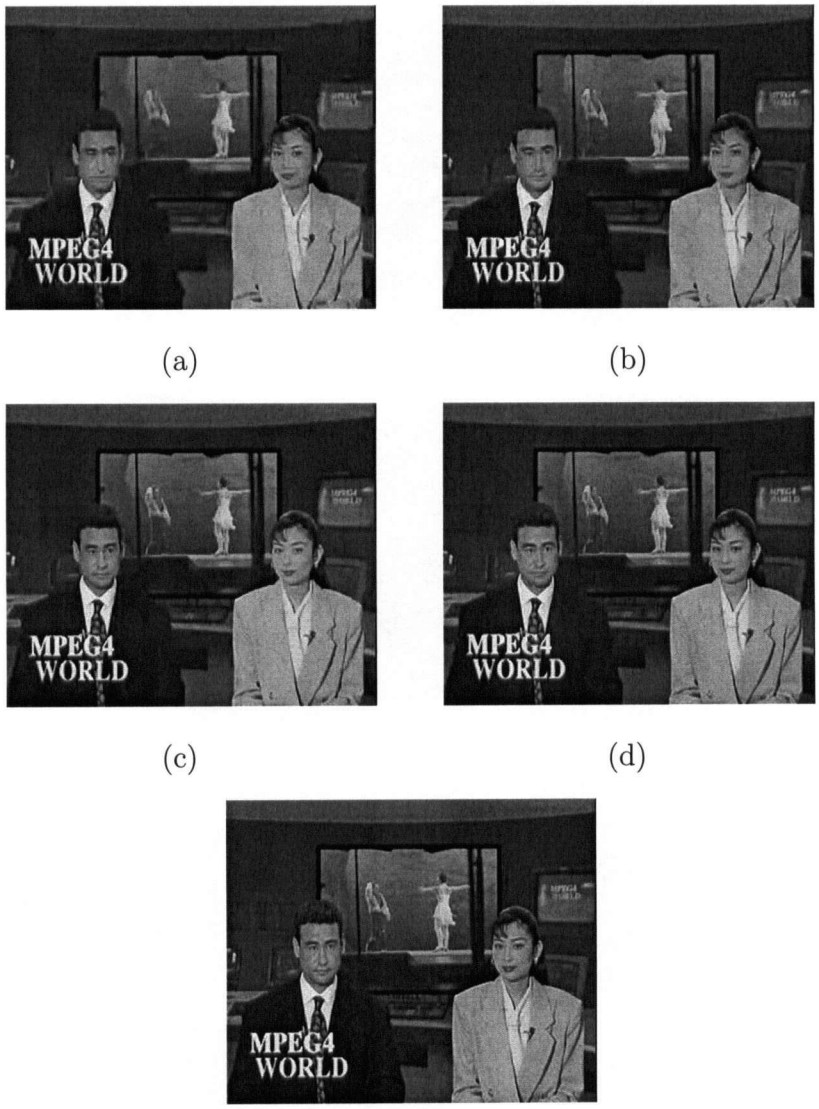


Figure A.4: Shown here are several Silent reference images. a) PSNR of 30dB b) PSNR of 32dB c) PSNR of 34dB d) PSNR of 36dB e) Original Image