A Hybrid System for Video Compression based on H.264 and JPEG2000

by

Zhenya Gu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author ....
Department of Electrical Engineering and Computer Science
May 22, 2009

Certified by.................................May 7, 2009
Jae S. Lim
Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

A Hybrid System for Video Compression based on H.264 and JPEG2000

by

Zhenya Gu

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering

Abstract

A video compression system is created that combines the JPEG2000 and H.264 standards. JPEG2000 is used to encode the I-frames, while H.264 is used to encode the P-frames. The goal of this thesis is to evaluate the performance of this hybrid system. The system is evaluated using a set of eight test video sequences, which cover a range of resolutions (CIF to 1920 × 1080) and picture content. Rate-distortion performance analysis shows the two systems to be comparable. Subjective analysis reveals that the artifacts of JPEG2000 are propagated to the P-frames. This can be useful in reducing blocking artifacts at low bit-rates. However, blurriness and fuzzy edges, which are the artifacts of JPEG2000, replace the blocking artifacts.

Thesis Supervisor: Jae S. Lim
Title: Professor

Acknowledgments

There are several people I would like to thank for their contributions to this thesis work.

First, I would like to thank my thesis supervisor Professor Jae Lim, who introduced me to the field of image processing and guided this thesis work. I would also like to thank Cindy for helping me find my way around the lab and Fatih and Andy for answering my many questions and providing helpful input.

Second, I would like to thank my mother, my father, and my sister Julia for always loving and supporting me.

Lastly, I would like to thank my friends for keeping my spirits high.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

In today's fast-paced world, digital media has become increasingly integrated into our everyday lives. In recent years, technological advances have allowed quality video streaming, vastly improved video conferencing and a proliferation of digital images on the Internet. Digital images and video however require large amounts of storage space and transmission bandwidth. Data compression is necessary to reduce data to a size within current storage and transmission capabilities. Many applications, such as digital television, require compression ratios of 50-70. To address these compression requirements, international committees, such as the International Organization for Standardization (ISO), have developed still image and video compression standards such as the JPEG and MPEG families.

In image and video compression, there is a tradeoff between bit-rate and image quality. New coding standards are judged on their ability to increase image quality at a given bit-rate. This proposal addresses the newest video compression standard, MPEG 4 Part 10, also known as H.264 Advanced Video Coding (AVC), and the newest still image standard, JPEG2000. Both have been compared extensively to previous standards and been shown to have a significant improvement in compression efficiency. Comparisons have also been made between the two standards by using only the intra-prediction mode of H.264, and the coding efficiencies were comparable.

In an encoded video sequence, some frames are compressed independently, the I-frames, while others are predicted based on other frames, the P and B frames. The

I-frames are essentially still images, which allows them to be encoded using a still image standard such as JPEG2000. While JPEG2000 and H.264 have comparable objective performance, they exhibit very different artifacts at low bit-rates. It is easy to distinguish between the two. Because of these differences, JPEG2000 might give better subjective performance. In addition, the P-frames will be affected because JPEG2000 creates different motion-compensated residuals, due to its different artifacts. It is possible that one type of residual might be more compact energy-wise after the H.264 DCT transform. To explore this scenario, we will use JPEG2000 to encode the I-frames and H.264 to encode the P-frames. This thesis will investigate which standard results in greater coding efficiency and higher visual quality.

Chapter 2 provides an introduction of video coding concepts useful for understanding the work presented in this thesis. Chapter 3 presents an overview of H.264 and JPEG2000 and discusses previous work comparing the two standards. Chapter 4 describes our approach for combining the two standards and for comparing the performances of the hybrid system and the original H.264 system. Chapter 5 presents and discusses experimental results.

Chapter 2

Video Coding Concepts

Raw video data consists of a time-ordered sequence of pictures, typically at 30 or 60 frames per second. The pictures come in a range of resolutions, from Quarter Common Intermediate Format (QCIF, 176 pixels × 144 pixels), commonly used for video conferencing, to High-definition (HD, 1920 × 1080) used for high-definition digital television broadcasting and digital film. One goal of video compression is to represent a video sequence with as few bits as possible, and this is achieved in part by exploiting temporal and spatial redundancies. Spatial redundancy refers to the similarity between a pixel and its neighboring pixels. Natural pictures have significant spatial redundancy. For example, there are often large uniform regions. Temporal redundancy refers to the fact that a scene does not change very much from one frame to the next.

Encoded video sequences contain two types of frames: intra-frames and inter-frames. Intra-frames, or I-frames, are compressed independent of all other frames, like a still image. The encoding of I-frames can be treated as still image compression, which will be discussed in Section 2.2. Inter-frames, which can be P or B-frames, are encoded by predicting from other frames. P-frames are predicted from past frames while B-frames are predicted from past and future frames. Inter-prediction will be discussed in Section 2.3.

2.1 Image Quality

Metrics of image quality are needed to evaluate the performance of compression algorithms. Because most algorithms are lossy, to achieve higher compression, there is a tradeoff between image quality and compression ratio. One commonly used qualitative measurement of image quality is peak signal-to-noise ratio (PSNR). PSNR is defined as

$$PSNR = -10log_{10}\frac{MSE}{(2^b-1)^2} \tag{2.1}$$

where b is the bit depth of the original image. The mean square error (MSE) is the average squared difference between the original M × N frame I and the compressed frame Î:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - \hat{I}(i,j)] \tag{2.2}$$

The rate-distortion (R-D) performance of an algorithm describes the tradeoff between image quality and compression ratio. R-D performance is typically represented by graphing PSNR versus bit-rate, which produces a rate-distortion curve, shown in figure 2-1. As figure 2-1 shows, a system has better rate-distortion performance than another system if its rate-distortion curve is higher and more to the left. This means that for a given PSNR, it has a lower bit-rate than the other system.

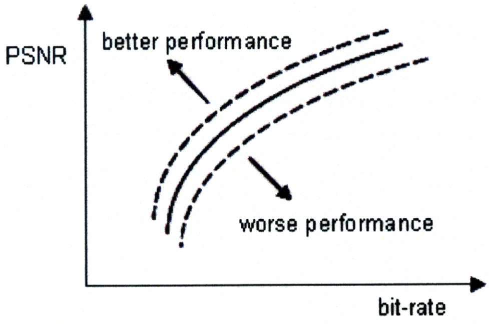


Figure 2-1: Example of a rate-distortion curve.

These objective measurements, however, do not always reflect the best visual quality. Some compressed images will have high PSNR, but will contain artifacts that are not pleasing to the human visual system (HVS). Subjective assessment of an algorithm is also necessary.

2.2 Image Compression

Image encoders exploit spatial redundancy with transform coding. A block diagram of a typical image encoder is shown in figure 2-2. Before transform coding, an image usually undergoes preprocessing. An image is typically converted into the luminance-chrominance (YCbCr) colorspace. This colorspace exploits the HVS's greater sensitivity to luminance versus chrominance. Because the HVS is less sensitive to chrominance, the chrominance, or chroma, channels are downsampled. A common format is 4:2:0, which means that the chroma channels are downsampled in both the horizontal and vertical directions. There is one sample in each chroma channel for every four samples in the luminance, or luma, channel.

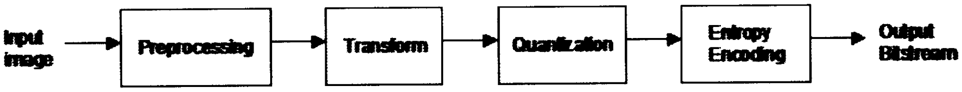


Figure 2-2: A typical image encoder block diagram

After preprocessing, the image is transformed into another domain that decorrelates and compacts the data. One commonly used transform is the block-based Discrete Cosine Transform (DCT). An image is divided into small N × N blocks, and each block goes through the DCT. Because most images do not have very high frequency content, energy is concentrated in the lower frequency coefficients. This compacts the image data into fewer coefficients.

Another transform is the Discrete Wavelet Transform (DWT). In a wavelet transform, signals are represented by wavelets, which are small waves with its energy concentrated in space. The wavelets are generated from a single basis function, called the prototype or mother wavelet, by scalings and time-shifts. The DWT is the discrete version of the wavelet transform. It can be computed by decomposing a signal into subbands and downsampling. The DWT operates on either the entire image or a large section of the image referred to as a 'tile.'

The next step after transform coding is quantization. Quantization is a lossy operation, because it cannot be reversed. During quantization, all input values within a certain interval are mapped to the same output value. The quantization step size (QP) refers to how large the interval is. The input coefficient is divided by QP and rounded to the nearest whole number. Larger QPs result in a coarser quantization and lower image quality. This step is applied after transform coding, and it replaces insignificantly small transform coefficients with zeros. Usually, high spatial frequency values become zero, since natural pictures have little high frequency content. Some algorithms use a quantization table with large QPs for high spatial frequencies and small QPs for low spatial frequencies because the HVS is less sensitive to very high frequencies.

The next step is reordering of the transform coefficients to optimize entropy encoding. A commonly used method is zig-zag scanning. The matrix of quantized transform coefficients is first scanned in a zig-zag pattern, starting from the lowest spatial frequency to the highest spatial frequency. This takes advantage of the fact that most high frequency coefficients are zero. Nonzero coefficients tend to be grouped together in the beginning of the sequence, followed by long strings of zeros. This allows greater compression in the entropy encoding step.

In entropy encoding, the sequence of coefficients is first represented using run-length encoding. Each nonzero coefficient is represented as a (run, level) pair, where run denotes the number of zeros before the current coefficient and level denotes the magnitude of the coefficient. An end symbol signals the last nonzero coefficient, and the decoder will know that all following coefficients are zero. The run-length encoded sequence is then entropy encoded using variable-length coding (VLC). In VLC, each input value is mapped to a codeword. Codewords have varying length, depending on how frequently the codeword is used. Examples of commonly used VLCs are Huffman coding and arithmetic coding. Huffman coding maps each input value to a binary codeword. A sequence of data values is represented by a sequence of codewords. Arithmetic coding maps each input value to an interval less than one. The size of the interval depends on the probability of the input value. A single fractional number is transmitted for each sequence of data values. Binary arithmetic coding is a type of arithmetic coding commonly used in video standards.

2.3 Video Compression

Temporal redundancy in video sequences can be exploited by encoding only the changes from one frame to the next. Only the frame difference, or residual, is transmitted from the encoder to the decoder. An improvement upon this method is motion-compensated prediction. A video sequence might have a still background and a single car moving across the frame. If motion vectors are used to move the pixels representing the car to the new location before computing the frame difference, than the difference would be much smaller. Encoding a smaller residual typically requires fewer bits.

First, the motion from a previous frame to the current frame is estimated. This process is known as motion estimation. One method of motion estimation is block or region matching. The encoder searches in previous frames for the best match to a block in the current frame. The displacement of the block is the motion vector. These motion vectors are used to generate a motion-compensated prediction of the current frame.

An image is typically divided into macroblocks, a 16×16 pixel region. A motion vector is estimated for each macroblock. If smaller macroblock sizes are used, a more refined prediction results, leading to even smaller residuals. However, more motion vectors must be estimated and transmitted. This increases the number of bits required to transmit the motion vectors. In addition, the complexity of motion estimation increases. More searches are required to find these extra motion vectors, and encoding time increases significantly.

Once the motion-compensated prediction is calculated, it is subtracted from the current frame, leaving the motion-compensated residual. The residual is then treated as a still image and undergoes transform coding, quantization and entropy encoding.

In a typical video compression system, shown in figure 2-3, the encoder first generates a motion-compensated prediction for the current frame from past frames stored in the picture buffer. The difference between the original frame and the prediction is then transformed and quantized. Here the path splits in two. One copy undergoes inverse quantization and inverse transform and is stored in the picture buffer to be used for motion estimation in later frames. The other copy is entropy encoded and written to an output bitstream.

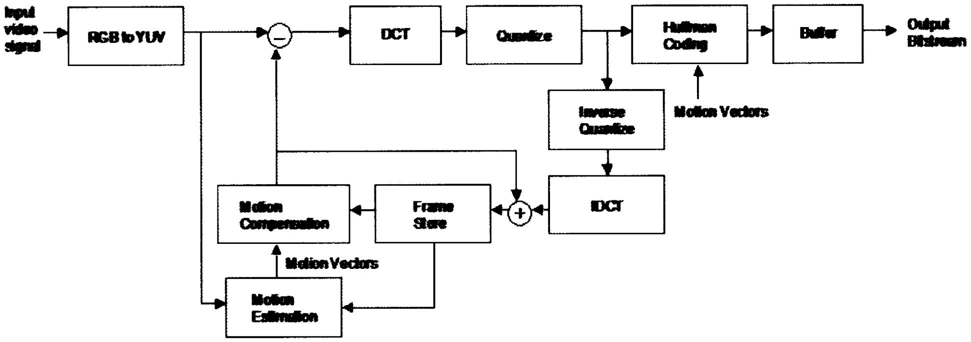


Figure 2-3: A typical video encoder block diagram

Chapter 3

Background

H.264 and JPEG2000 are two recently developed compression standards. They have been shown to have significant compression gains over their predecessors. This chapter presents an overview of the two standards and of previous work comparing the performance of the two.

3.1 H.264/MPEG4 Part 10 System

H.264/MPEG-4 [2] is based on the core MPEG system, but adds many extensions and functionalities. The MPEG standard organizes frames into group of pictures (GOP) that are repeated for the entire video sequence. GOPs can consist of three types of frames: I frames, P frames, and B frames. A common GOP is I0, B1, B2, P3, B4, B5, P6, B7, B8.

Intra-coding in previous MPEG standards uses a DCT to transform samples from the spatial to the transform domain. This transform can be enhanced by applying some kind of prediction between neighboring samples before the transform [8]. One improvement of H.264 is directional spatial prediction of intra-coded macroblocks prior to transform coding. Samples of the block are predicted using neighboring samples. Spatial prediction of the luma component has two modes: Intra4x4 and Intra16x16. In Intra4x4 mode, the encoder predicts each 4×4 block independently. This is useful for pictures with high spatial detail. The Intra16x16 mode is used to predict a whole 16 × 16 macroblock. This is applicable for pictures with large smoothly varying regions. The chroma component can only use 8 × 8 blocks. Prediction occurs along a certain spatial direction. There are nine different prediction modes for 4 × 4 blocks, four for 16 × 16 blocks, and four for chroma components. Figure 3-1 shows five of the nine 4 × 4 modes. The encoder generally chooses the mode that minimizes the prediction error.

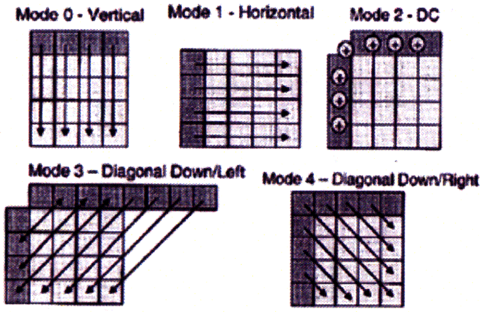


Figure 3-1: Five of the nine 4 × 4 luma intra prediction modes [16].

The transform in H.264 is carried out using a multiplication-free, separable integer transform with a 4 × 4 block size. The integer nature of the transform solves mismatch problems between the encoder and decoder inverse transforms. In Intra16x16 mode, a similar 4 × 4 transform is applied to the 4 × 4 array of luma DC coefficients. A 2 × 2 transform is applied to the chroma DC coefficients.

Entropy encoding in H.264 uses either context-adaptive variable length coding (CAVLC), which refers to a method similar to Huffman coding, or context-adaptive binary arithmetic coding (CABAC). In context-adaptive coding, the coder switches between lookup tables depending on previously encoded elements. Studies have shown CABAC to have significantly improved coding efficiency over CAVLC [9].

Inter-prediction allows variable block-size motion compensation with a minimum luma prediction block size as small as 4 × 4. Other macroblock partitions are 16 × 16, 8 × 16, 16 × 8 and 8 × 8. Sub-macroblock partitions are 8 × 8, 4 × 8, 8 × 4 and 4 × 4. As mentioned earlier, smaller block sizes result in better predictions. However, the search time and bit-rate cost of finding and encoding more motion vectors might not be worth the improvement. The tradeoff depends on the picture characteristics. H.264 allows the encoder to adaptively switch block sizes depending on the video content. Sequences with highly detailed movements can be encoded using the 4 × 4 block size. Sequences with smoother content can use the larger block sizes. H.264 also allows quarter-pixel motion vector accuracy. Subpixel motion compensation interpolates between pixels and searches these sub-samples for the best match to a macroblock. Generally, a finer interpolation leads to better prediction results, at the expense of increased complexity.

H.264 stores a limited number of previously encoded frames in the reference picture buffer. The frame stored is the reconstructed frame, not the original frame. This prevents an increasing "drift" between the encoder and decoder. The encoder and decoder maintain a list of the frames stored in the buffer. P-frames use one reference picture list while B-frames use two lists. As newer frames are stored in the buffer, older frames are removed. Frames can be marked as long term reference pictures, in which case they will not be discarded.

H.264 also implements an in-loop deblocking filter that greatly reduces blocking artifacts. Blocking artifacts are sharp edges that appear at macroblock boundaries, a result of using block-based coding algorithms. The deblocking filter smoothes edges at macroblock boundaries. The filter intelligently chooses whether or not to smooth the edge based on the strength of the edge and the coding modes of adjacent blocks. If the edge gradient is above a certain threshold, the filter does not smooth the edge. The threshold value depends on the quantization parameter. If the QP is small, only very small gradients will be due to blocking effects. If a gradient is very strong, it is likely due to actual image features.

Given the variety of intra and inter prediction modes discussed above, the encoder needs a method for deciding which mode to use. A process called rate distortion optimization looks through all the modes and selects the one that minimizes the amount of bits needed to encode the block and the square of absolute difference (SAD) between the encoded block and the original. A mode is selected independently for each macroblock. If all of the inter-prediction modes are too costly, the block will be intra-coded.

The features described above are encompassed in the H.264 Main Profile. An amendment called the Fidelity Range Extensions (FRExt) was added in 2004, a year after the first version was completed. A new profile, the High Profile, includes these extensions as well as the original Main Profile features. The FRExt amendment extends the 4 × 4 transform and Intra4x4 mode to an 8 × 8 block size. The encoder can select between 16 × 16, 8 × 8 or 4 × 4 for the intra-coding block size and between 4 × 4 and 8 × 8 for transform coding. The High Profile can achieve significant bit-rate savings depending on the sequence content [10].

3.2 JPEG2000

JPEG 2000 [13, 1] is a new still image compression standard developed jointly by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). Its predecessor, JPEG, has been widely used for more than a decade. Since JPEG became an international standard in 1992, many new techniques have been developed for image compression. JPEG2000 utilizes these new techniques, and it not only outperforms JPEG in terms of rate distortion, but it also has many new features for the end user. A quick overview of the JPEG2000 system is presented below.

The compression system can be divided into three phases: image preprocessing, compression, and compressed bitstream formation. The first step of preprocessing is tiling the image into non-overlapping blocks. The tile is the basic unit of the compression system. Larger tiles create fewer tiling artifacts. However, the memory requirements are higher, since each tile is processed as a single unit. Common tile sizes are 256 × 256 or 512 × 512. This is much larger than the 16 × 16 pixel macroblocks used in H.264. This is because JPEG2000 uses the DWT, which is a frame-based transform, instead of a block-based transform like the DCT used in H.264. After tiling, the samples are dc level-shifted and color transformed using either reversible color transform (for lossless compression) or irreversible color transform (for lossy compression).

In the compression stage, the discrete wavelet transform (DWT) is implemented with the Daubechies 9-tap/7-tap filter [5] for irreversible transformation or the Le Gall 5-tap/3-tap filter [7] for reversible transformation. The DWT is useful because it allows multiresolution representation of signals. The DWT decomposes a signal into different subbands. At each decomposition level, one subband represents a coarse, downsampled version of the image, while the other subband represents the details lost by the downsampled version. Successive levels are increasingly finer approximations of the original image. Signal decomposition using the DWT can be implemented using FIR filters, such as the two filters mentioned above. After the DWT, each subband is divided into blocks (called code-blocks) and quantized using uniform scalar quantization with a dead-zone about the origin. JPEG2000 supports a different quantization step-size for each subband.

The next stage is entropy encoding of the quantized wavelet coefficients and bitstream formation. This stage is divided into two steps: Tier-1 and Tier-2 coding. In Tier-1 coding, each code-block is decomposed into bit-planes and each bit-plane is encoded using the embedded block coding with optimized truncation (EBCOT) algorithm by David S. Taubman [14]. EBCOT encodes each bit-plane in three coding passes and generates a context and a binary decision value for each bit position. JPEG2000 uses context-adaptive binary arithmetic coding (CABAC) to encode the binary decision values. The binary arithmetic coder uses the context information generated by EBCOT to select the optimal coding conditions. In Tier-2 coding, the encoder selects which bit-plane coding passes to include in the final code stream and the order of appearance of these passes.

Rate control is left up to the individual developer in the JPEG2000 standard. One method of rate control is to select only a subset of the coding passes to include in the final code stream. A rate-distortion optimization algorithm can decide which passes to include. Another choice is to adjust the quantization step size. However, every time the step size changes, the entropy-encoding step needs to be redone. Because this step is very computationally intensive, such a rate control scheme is not very feasible for real-time applications.

3.3 Previous Work

Since JPEG2000 and H.264 became international standards, many papers have been written comparing their performance. Since one is a still image standard and the other is a video standard, only the intra-prediction performance of H.264 can be compared to JPEG2000. The two have been compared for still images and for intra-only video sequences at a wide range of spatial resolutions. In this section, we present a review of the literature. Section 3.3.1 discusses the software written for JPEG2000 and H.264. Section 3.3.2 summarizes experimental results.

3.3.1 Software

Many software implementations exist for JPEG2000 and H.264. Because the standards only specify the bitstream that the decoder needs to decode into a image or video sequence, there is a lot of flexibility in designing the encoder. What algorithms the encoder uses for motion estimation and mode decision are decisions left up to the individual developer. This leads to variation in rate distortion performance from encoder to encoder.

The Joint Video Team (JVT) has released a reference software implementation for H.264 - Joint Model (JM). JM is the software version used in almost all the previous work. There is almost a reference implementation for JPEG2000, Verification Model (VM), but many researchers prefer the commercial Kakadu implementation. Another implementation by Michael Adams, JasPer, is also commonly used [3, 4]. A study by Dimitriy Vatolin et. al. compared the rate distortion performance for several JPEG2000 implementations [15]. Generally JasPer had average rate distortion performance. A visual comparison also placed JasPer in the middle. Kakadu tended to have worse rate distortion performance that JasPer, but it usually gave better visual results.

For comparisons of JPEG2000 and intra-only H.264 for video sequences, an extension of JPEG2000 called Motion-JPEG2000 is used. Motion-JPEG2000, specified in Part 3 of the JPEG2000 standard, is based on the core system of JPEG2000 Part

1. Motion-JPEG2000 is capable of handling interlaced video. One implementation used is Verification Model of Motion-JPEG2000, a software package built on top of the JPEG2000 VM mentioned above.

3.3.2 Performance Results

Previous work has shown JPEG2000 and H.264 to be comparable in rate-distortion performance with differences depending on the picture content and spatial resolution. JPEG2000 tends to outperform H.264 Main Profile for very high-resolution video, possibly for cinema applications. However, Main Profile outperforms JPEG2000 for low and medium resolution videos. High Profile is comparable with JPEG2000 at high resolutions, and significantly better at low resolutions. Additionally, JPEG2000 tends to perform better with smooth images, which have a lot of spatial correlation. The better decorrelating properties of the wavelet transform give JPEG2000 an advantage with these types of images [8].

Marpe and Wiegand compared H.264 High Profile and JPEG2000 for monochrome still images with resolutions ranging from 512×512 to 2048×2560 [10]. They used H.264 reference software JM v9.4 and JPEG2000 Kakadu v2.2. R-D optimization was enabled for both standards. R-D optimization selects the coding mode that minimizes the difference between the original and encoded image and the number of bits. The FRext 8×8 transform mode was enabled for H.264. For the famous Barbara and Lena images, the overall objective performances of the two standards were nearly identical. The subjective quality was also comparable at all bit-rates. However, for other images, H.264 had a significant gain in PSNR. For one image in particular, an image containing a mixture of natural elements and text, H.264 had a R-D gain of 3-6 dB when the loop filter was disabled. The loop filter created noticeable artifacts. There was an image of a fingerprint for which JPEG2000 had better R-D performance. The performance comparison of the two standards has a significant dependence on image content.

Marpe and Wiegand also compared H.264 and Motion-JPEG2000 for a series of video sequences covering a wide range of resolutions [8]. The software implementations used were H.264 JM v7.1 and Motion-JPEG2000 VM v8.6. One disclaimer is that this study compared H.264 Main Profile to JPEG2000. As has been shown in previous work, H.264 High Profile has better R-D performance than Main Profile [10]. However, the results are presented here to show the general trend of H.264 vs. JPEG2000 performance. H.264 has superior R-D performance for lower resolution video such as CIF (352 × 288). For medium to high resolutions, JPEG2000 and H.264 have comparable performance. At very high resolutions, such as 1920 × 1080, Motion-JPEG significantly outperforms H.264.

A study by Topiwala compared H.264 High Profile and JPEG2000 for high-resolution video sequences [11]. JM 9.2 and Kakadu v2.2 were the software implementation used. Overall H.264 High Profile has an average Y-PSNR gain of 0.5 dB over JPEG2000. For 1280 × 720, H.264/AVC was consistently and significantly better than JPEG2000 while for 1920 × 1080, H.264 was slightly better at higher bit rates and JPEG2000 was slightly better at lower bit rates. This is consistent with other results showing JPEG2000 to have improved R-D performance at high spatial resolutions.

Ouaret, Dufaux and Ebrahimi compared both Main Profile and High Profile Intra-only H.264 to JPEG2000 for low and medium resolution video sequences [12]. For sequences with resolutions of 704 × 576, H.264 High Profile has a ~0.2 dB PSNR gain over JPEG2000 while H.264 Main Profile is ~1 dB worse than JPEG2000. For lower resolutions (QCIF and CIF), Main and High Profile have similar R-D performance and both have a gain of 1 ~ 2 dB over JPEG2000.

While these results vary with picture content, a few trends can be inferred from previous work. In general, JPEG2000 has better performance at higher spatial resolutions while H.264 has better R-D performance at lower spatial resolutions. H.264 High Profile is also significantly better than Main Profile, especially at high resolutions.

Chapter 4

Approach

Previous research has compared JPEG2000 and H.264 for intra coding, and results have shown that the two standards generate very different artifacts at low bit-rates. One might be able to exploit these differences to improve the compression ratio or subjective quality of video sequences. One test is to use JPEG2000 to encode the I-frames and H.264 to encode the P-frames and compare this hybrid system to the original H.264 system. By using JPEG2000 to encode the I-frames, the following P-frames will be predicted using JPEG2000-encoded frames. This could give very different results from the original H.264 system, especially at low bit-rates. The hybrid system is implemented by this thesis work. Since JPEG2000 is a still image standard and H.264 is a video standard, creating the hybrid system requires combining the two encoding systems. This chapter describes the overall system and the software modifications required to create the JPEG2000/H.264 hybrid encoder.

4.1 Overview of the Implemented System

The hybrid encoder will use JPEG2000-encoded I-frames as reference frames for motion estimation and residual encoding. The system does not integrate the two at the bitstream level. For each video sequence, the I-frames are first extracted and encoded using JPEG2000. The encoded frame is than decoded to the raw format and recombined with the rest of the video sequence. A difference will be seen only if the frame undergoes lossy compression. The modified video sequence is then encoded using H.264. The H.264 encoder is configured to encode all I-frames using I_PCM mode. In I_PCM mode, image pixels are directly encoded using entropy encoding, without any transformation or quantization. Figure 4-1 shows the system block diagram.

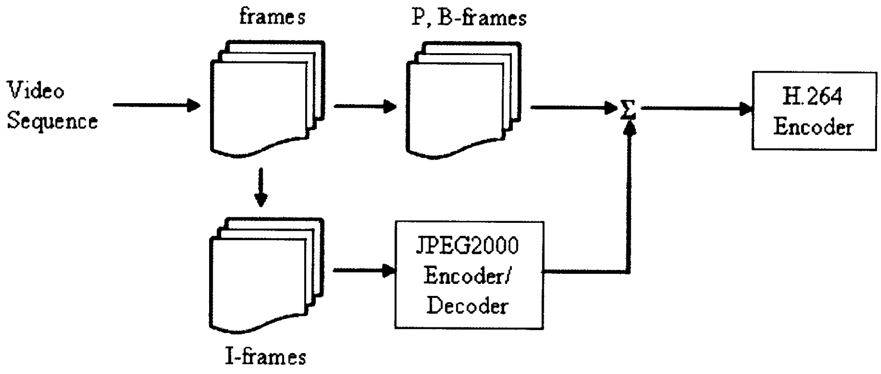


Figure 4-1: System block diagram

4.2 Software

For encoding the video sequence, publicly available software was used. JasPer v1.900.1 was used for JPEG2000 compression and JM v14.2 for H.264 compression. For the hybrid system, the encoder requires a new setting in which all the I-frames are encoded using the I_PCM intra-mode. Normally, this intra-mode is used only when fewer bits are required to directly encode a macroblock than to transform and quantize it. This happens very rarely. However, in the hybrid system, this mode is desired because it does not modify the already JPEG2000-encoded I-frames. In the following sections, we will discuss the two software packages and describe the modifications.

4.2.1 H.264 Encoder

The H.264 JM reference encoder has a configuration file in which the user can specify a wide range of settings. Various features of the H.264 standard, such as the 8 × 8 transform and the deblocking filter, can be enabled or disabled. Many features, such as rate-distortion optimization, have several modes to select from.

To implement the hybrid encoder, we added a new parameter called JpegHybrid. This parameter allows the user to switch between the hybrid encoder and the regular H.264 encoder. When the parameter is enabled (i.e. the hybrid encoder is selected), the mode decision selects I_PCM for all I-frames. Normally, the mode decision function decides which mode to encode the macroblock based on a rate-distortion optimization algorithm. The modification bypasses this step and directly encodes the macroblock using I_PCM mode. This is done for all I-frame macroblocks.

The following H.264 features were enabled:

- High Profile
- 8×8 Transform (adaptive choice between $4 \times 4/8 \times 8$ transform and prediction modes)
- in-loop deblocking filter
- search range ± 32
- CABAC
- R-D optimization
- No B-frames

4.2.2 JPEG2000 Encoder

JasPer is a C implementation of the JPEG2000 standard. It can handle several input file types, such as BMP, PNM, and Sun Rasterfile. There are several encoding options, such as tile size and target rate. The target rate option is used to match the PSNR of JPEG2000-encoded I-frames with the H.264-encoded I-frames. Rate control in JasPer is achieved by using a rate-distortion optimization algorithm to select a subset of the coding passes generated in Tier-1 coding to include in the final code stream. The quantization step sizes are fixed. The JasPer software was not modified for this thesis.

I-frames are input into the encoder as BMP files. The compressed JP2 file is then decompressed back into BMP and recombined with the original video sequence. The encoder options selected were one tile per picture (no tiling) and lossy compression using the Daubechies 9/7-tap filter. Because compression is lossy, the decompressed BMP will be different from the original BMP file.

Chapter 5

Experimental Results

The hybrid system described in Chapter 4 was compared with H.264 for eight video sequences. The video sequences cover a range of resolutions and content type. This range allows a broad evaluation of the two systems.

Section 5.1 describes the video sequence content. Section 5.2 presents an objective analysis of the two systems in the form of rate-distortion data. Section 5.3 presents a subjective analysis and examines artifacts in the decoded video sequences.

5.1 Video Sequences

Eight video sequences were used for comparison. The video sequences cover a range of content types and resolutions. Flower consists of a slow camera pan of a mostly stationary scene. Basketball and DucksTakeOff have multiple objects moving very quickly in a non-uniform way. In the case of the Basketball sequence, multiple players run in different directions on a basketball court. Some sequences show many objects moving at a uniform speed in a constant direction. For example, in Beertruck, cars are moving at a constant speed on a highway. Table 5.1 below shows the resolution and picture content of each sequence. A and B denote different sections of the same scene. All sequences are monochrome and ten frames in length.

Table 5.1: Test sequences.

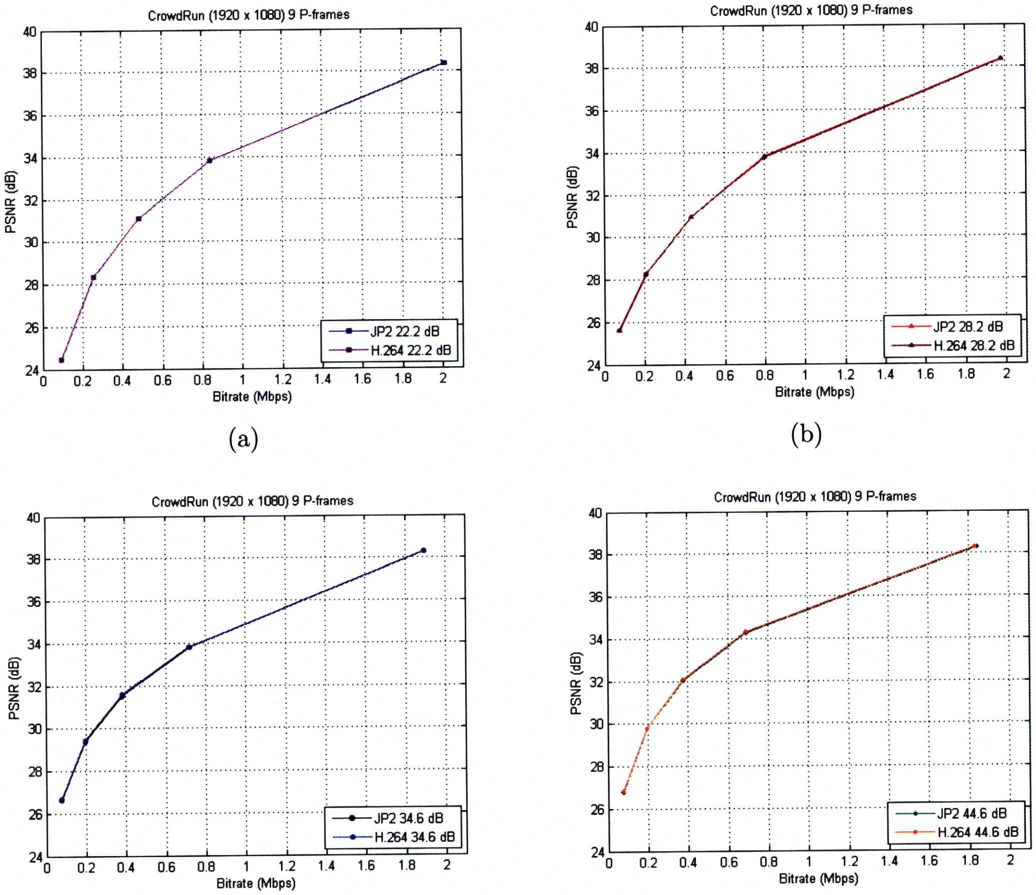
Name	Resolution	Sequence Content
BasketballA	CIF	Camera pan, fast action
BasketballB	CIF	Fast action
Container	CIF	Single object moving slowly
Flower	CIF	High spatial detail, camera pan
Car	720 × 480	Low spatial detail, two objects moving
Mall	1200 × 880	High spatial detail, moving objects
Beertruck	1280 × 720	Multiple objects moving at constant speed
CrowdRun	1920 × 1080	High spatial detail, many objects moving
DucksTakeOffA	1920 × 1080	Low spatial detail, little movement
DucksTakeOffB	1920 × 1080	Low spatial detail, fast action

5.2 Rate-distortion Performance Analysis

In our experiments, we evaluate the rate-distortion performance for only the P-frames of a video sequence. The assumption is that JPEG2000 and H.264 perform comparably for I-frames. We are isolating the problem to how the I-frames affect P-frame R-D performance, since the I-frames are used as reference frames for inter-prediction. To obtain a fair comparison, the PSNR of the JPEG2000-encoded I-frame is matched to the PSNR of the H.264-encoded I-frame.

For each video sequence, ten frames are encoded with the following GOP: I, 9 P-frames. To plot a rate-distortion curve, the P-frame quantization parameter is varied while the I-frame PSNR remains fixed. The average P-frame bitrate is measured and plotted versus average P-frame PSNR. All nine P-frames are encoded at the same QP. Figure 5-1a shows this curve for the CrowdRun sequence for an I-frame PSNR of 22.2 dB. One curve is for the hybrid system, while the other is for the original H.264 system.

Rate-distortion curves were generated for three other I-frame PSNR values, shown in figures 5-1b–d. Figure 5-2a shows all eight curves plotted on the same graph. The sequences with higher I-frame PSNR have a PSNR gain in the P-frames over the lower I-frame PSNR sequences. This is because the higher I-frame quality generates an improved motion-compensated prediction. The residual is therefore much smaller and fewer bits are necessary to encode it.



(a)

(b)

(c)

(d)

Figure 5-1: Rate-distortion curves for CrowdRun (1920 × 1080): (a) I-frame PSNR = 22.2 dB; (b) I-frame PSNR = 28.2 dB; (c) I-frame PSNR = 34.6 dB; and, (d) I-frame PSNR = 44.6 dB.

Figures 5-1a–d show H.264 and the hybrid system to have comparable rate-distortion performance. Figures 5-2b–d show the R-D curves for the Mall, BasketballA and Beertruck sequences, which represent a range of resolutions. These graphs also show the two systems to have comparable rate-distortion performance. The remaining sequences generated similar results. While the R-D plots appear to be almost the same, one can quantize the PSNR gain of one plot over another using the Bjøntegaard Delta bit-rate algorithm [6]. This algorithm finds the average PSNR difference between the two plots over the measured PSNR range. The PSNR gains of the curves plotted in figure 5-2a are shown in Table 5-2. A negative gain means the H.264 rate-distortion curve has a PSNR gain over the hybrid curve. The maximum PSNR gain is very small (~0.05 dB). This is representative of the gain in the other video sequences. The maximum difference in the other video sequences is a gain of ~0.1 dB for H.264 over the hybrid encoder.

Table 5.2: Gain in average P-frame PSNR versus I-frame PSNR for CrowdRun sequence.

I-Frame PSNR (dB)	Gain of Hybrid over H.264
22.2	-0.005
28.2	0.029
34.6	-0.045
44.6	-0.037

More can be learned from analysis of the first P-frame. Because only the I-frames are encoded differently, while the P-frames are encoded with the H.264 standard in both the hybrid and H.264 system, sequential P-frames become more and more similar. Therefore, the most difference between the two systems should be seen in the first P-frame. Figure 5-3 graphs PSNR versus Frame Number for a bitrate of 1 Mbps for the CrowdRun sequence. The difference can be seen more clearly in figure 5-4, which shows the absolute difference in PSNR between the hybrid system and the H.264 system. The answer to which system has higher PSNR for a given bitrate varies with sequence. Frame 0 represents the I-frame, and Frame 1 is the first P-frame, where there is a clear spike. This is representative of the other video sequences as well.

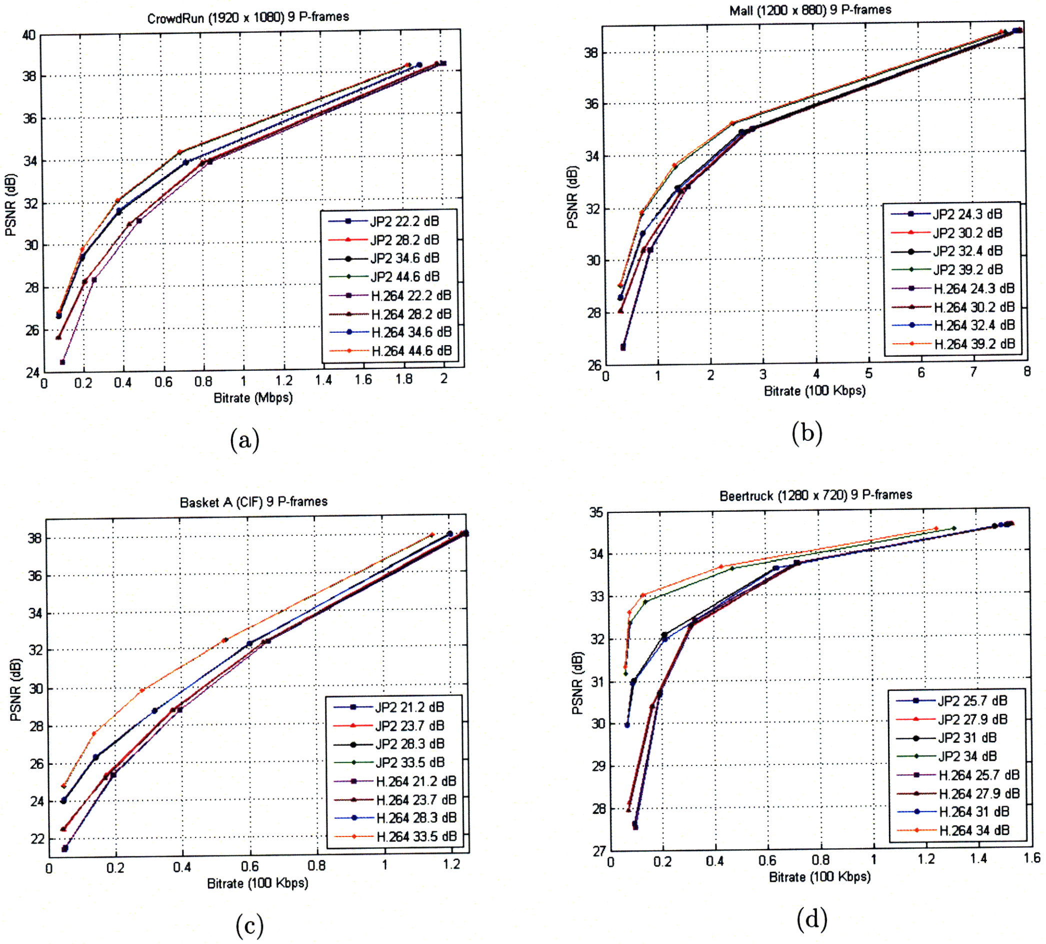


Figure 5-2: Rate-distortion curves averaged over 9 P-frames, shown for 4 different I-frame PSNR values: (a) CrowdRun (1920 × 1080); (b) Mall (1200 × 880); (c) Basketball A (CIF); and, (d) Beertruck (1280 × 720).

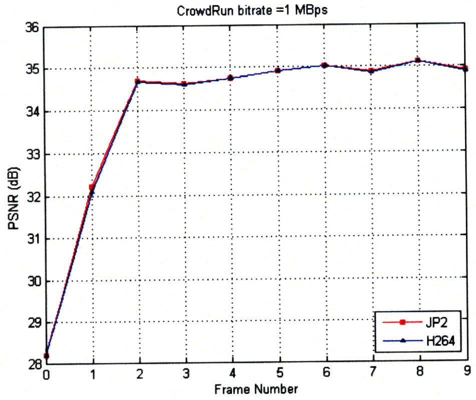


Figure 5-3: PSNR of each frame (CrowdRun, bitrate = 1 Mbps).

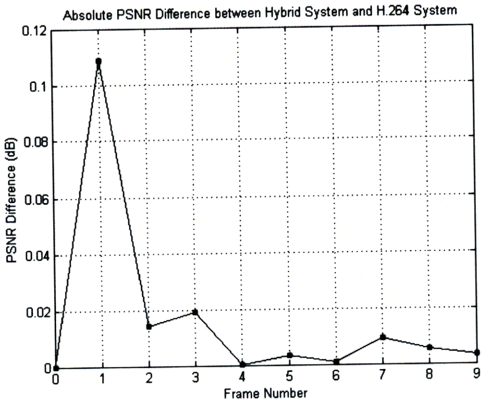


Figure 5-4: Absolute PSNR Difference between Hybrid System and H.264 System (CrowdRun, bitrate = 1 Mbps).

To plot PSNR versus Frame Number for a given bitrate as in figure 5-3, the five points plotted for each R-D curve were linearly interpolated, as illustrated in figure 5-5. The red line indicates the PSNR value of the first P-frame at 1 Mbps. This process was repeated for all 9 P-frames.

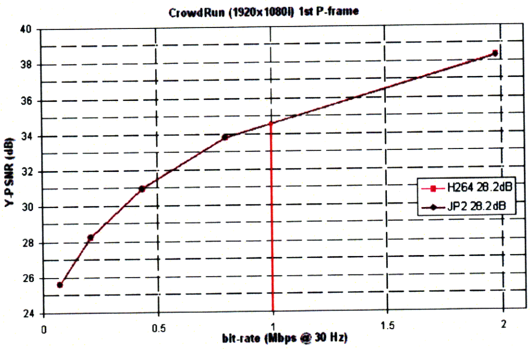
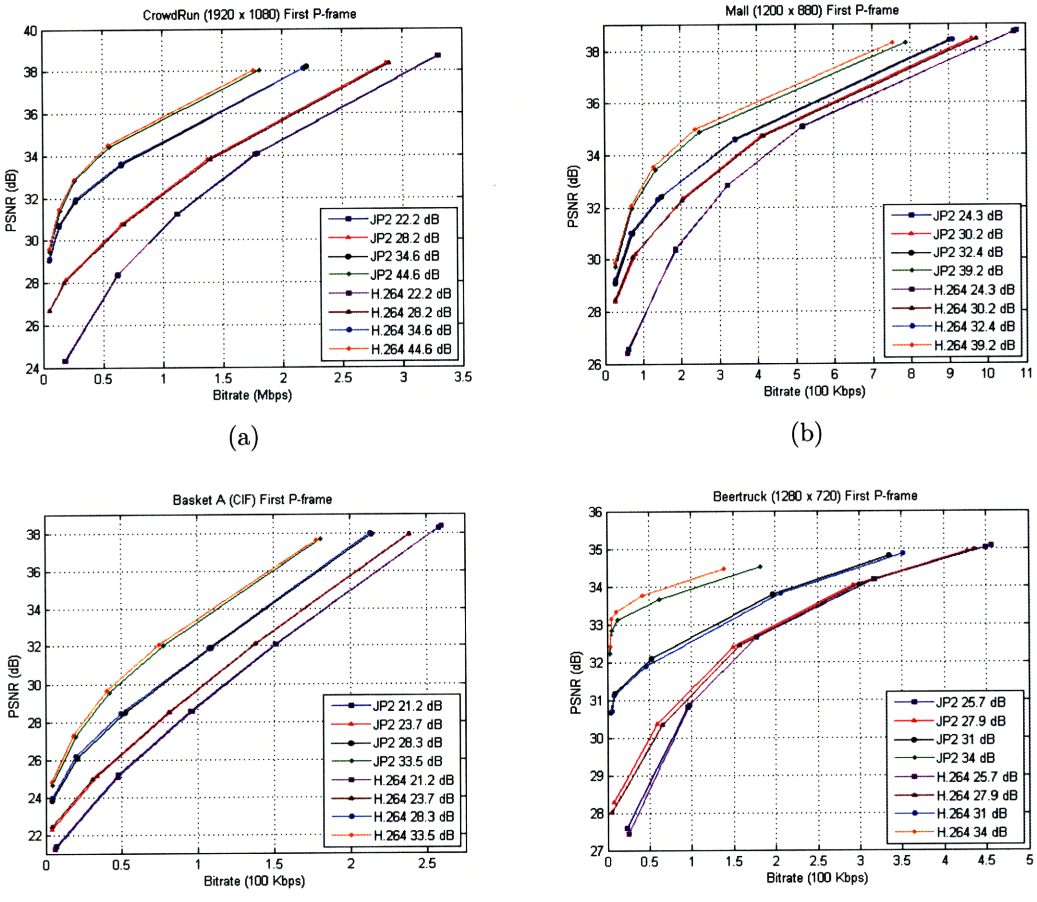


Figure 5-5: Linear interpolation to find PSNR at a certain bit-rate.

Figures 5-6a–d shows the R-D curves for just the first P-frame for the same sequences shown in figures 5-2a–d. Note how the R-D curves for different I-frame PSNR are more separated for the first P-frame than for the average. The first P-frame PSNR has a greater correlation with the I-frame PSNR than the average P-frame PSNR. Despite the greater correlation, the PSNR gain is still very small. The Bjøntegaard Delta bit-rate savings in the first P-frame for different I-frame PSNR is shown in Table 5.3. The maximum observed gain in all the video sequences is ~0.2 dB.



(a)

(b)

(c)

(d)

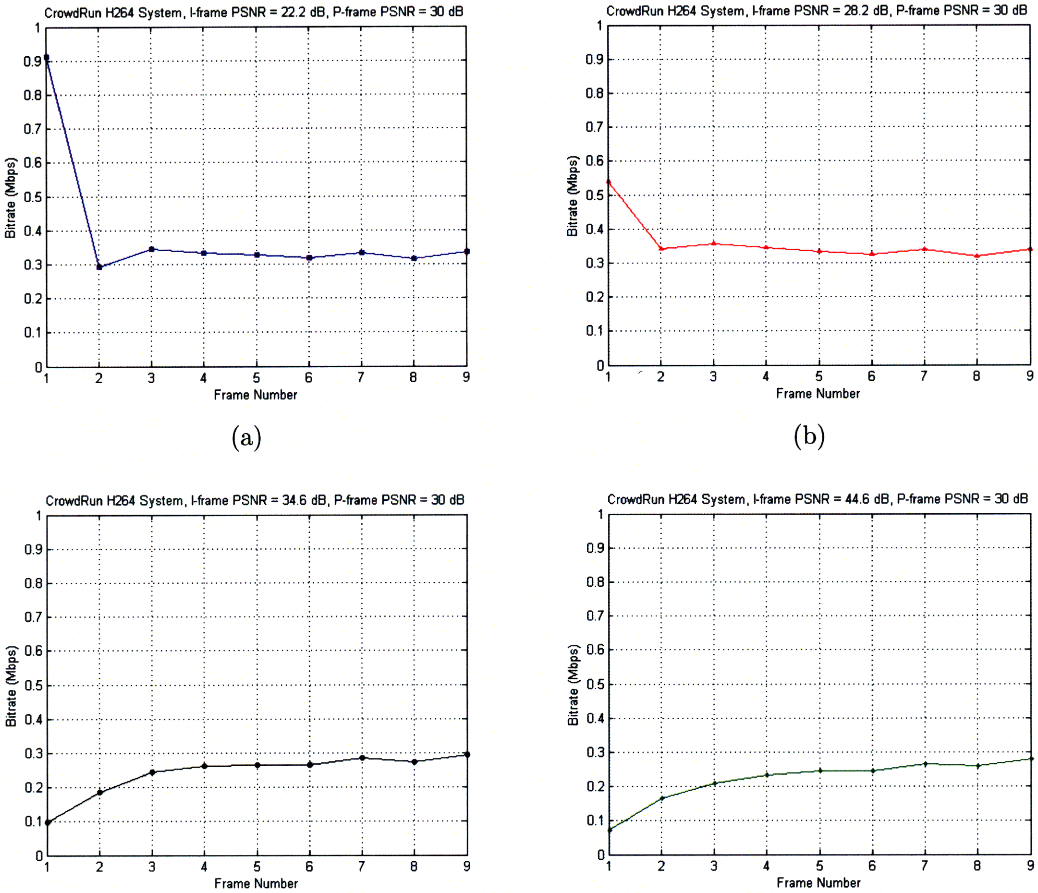
Figure 5-6: Rate distortion curves for first P-frame: (a) CrowdRun (1920 × 1080); (b) Mall (1200 × 880); (c) Basketball A (CIF); and, (d) Beertruck (1280 × 720).

Table 5.3: Gain in first P-frame PSNR versus I-frame PSNR for CrowdRun sequence.

I-Frame Y-PSNR (dB)	Gain of Hybrid over H.264
22.2	0.022
28.2	0.099
34.6	-0.083
44.6	-0.12

In addition, the quality of the I-frame is reflected in how the P-frames settle to a steady state. If the P-frames are encoded at a much higher quality (smaller quantization parameter) than the I-frame, the first P-frame will require a greater number of bits than the average P-frame. This case is illustrated in figure 5-7a for the CrowdRun sequence. The I-frame has a PSNR of 22.2 dB while the P-frames have a PSNR of 30 dB. The plot shows the bitrate in Mbps versus P-frame number. The first P-frame requires substantially more bits than the rest of the P-frames. It needs to "recover" from the low-quality I-frame, so the residual is very large. The plot was obtained using the same linear interpolation method mentioned previously for the PSNR versus frame number plots, but with the PSNR fixed instead of the bitrate.

Conversely, if the P-frames are encoded at a much lower quality than the I-frame, the first P-frame will require fewer bits than the average. This case is illustrated in figures 5-7c–d. While the bitrate settles to around 0.3 Mbps, the first couple of P-frames require significantly fewer bits. The higher quality I-frame forms a better motion-compensated prediction than the ensuing lower quality P-frames, which reduces the number of bits required to encode the residual. All four cases are shown in figure 5-8. One may notice that the graphs do not converge completely as one would expect, since all the P-frames are encoded at the same PSNR. This is because the rate of convergence is very slow for sequences where the I-frame is encoded at a higher PSNR value than the P-frames. Figure 5-9 plots the curves referenced to the 22.2 dB I-frame plot. The first P-frame is left out of the plot, because the difference is so great that it makes it difficult to see the rest of the points. There is a general trend towards zero for all three curves, although convergence is very gradual for the 34.6 dB and 44.6 dB I-frame sequences. Especially for scenes with little movement from one frame to the next, high-PSNR I-frames generate very high quality motion-compensated predictions. Since one can only add bits and not subtract bits, it is much easier to improve the quality of a low-PSNR I-frame than to diminish the quality of a high-PSNR I-frame. Figure 5-9 shows that the 28.2 dB I-frame sequence converges much faster to zero than the other two sequences.



(a)

(b)

(c)

(d)

Figure 5-7: Bitrate of each frame for 4 different I-frame PSNR values and P-frame PSNR = 30 dB: (a) I-frame PSNR = 22.2 dB; (b) I-frame PSNR = 28.2 dB; (c) I-frame PSNR = 34.6 dB; and, (d) I-frame PSNR = 44.6 dB.

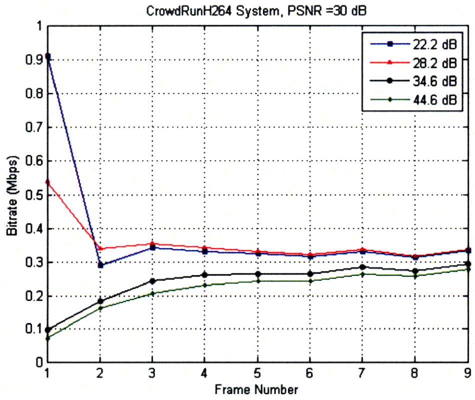


Figure 5-8: Comparison of bitrate of each frame for four different I-frame PSNR values (CrowdRun, P-frame PSNR = 30 dB).

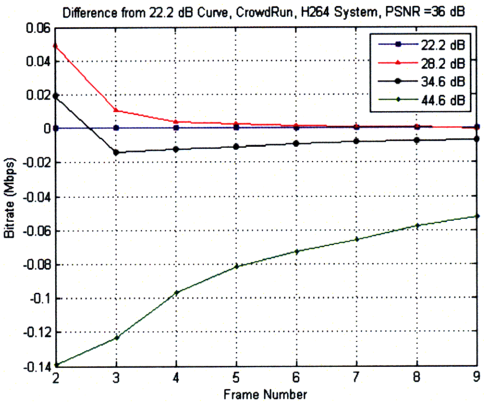


Figure 5-9: Bitrate versus Frame Number Plots referenced to the 22.2 dB I-frame Sequence (CrowdRun, P-frame PSNR = 30 dB).

5.3 Subjective Analysis

While the objective results are very similar, rate-distortion analysis is not always a good indicator of picture quality. Artifacts, such as blocking artifacts, might not greatly affect PSNR, but will be very noticeable and annoying to the viewer. This section discusses the visual artifacts of JPEG2000 and H.264, which appear in the I-frames, and how these artifacts affect the P-frames.

5.3.1 Artifacts

Because JPEG2000 is a wavelet-based transform and H.264 is a block-based transform, the two standards exhibit very different artifacts. H.264 has blocking artifacts because it uses a block-based DCT transform. Because each block is treated independently, artificial edges occur at the horizontal and vertical boundaries at low bit-rates. The in-loop deblocking filter in H.264 does a good job of removing blocking artifacts. Figure 5-10 shows an image with the in-loop deblocking filter disabled, while figure 5-12b shows the same image with the filter enabled. All perceivable traces of blocking artifacts are gone. Figure 5-11 shows the original CrowdRun I-frame as a reference. However, when the compression ratio is very high, the filter cannot remove all blocking artifacts. This is the case in the DucksTakeOff image shown in figure 5-14b. Although the filter does smooth the blocking edges, they are still visible. Figure 5-

13 shows the original DucksTakeOff I-frame as a reference. All the figures in this subsection are I-frames, because JPEG2000 is only used to encode the I-frames.

Another artifact of H.264 is the appearance of fake contours and the smoothing of textures. Quantization step sizes are very large at low bit rates, which causes the fake contours. The smoothing of textures may be due to the spatial prediction step during intra-prediction of a macroblock. If the residual is quantized very heavily, much of the detail will be lost, and only the spatial prediction will be decoded and displayed. This artifact is especially noticeable on the faces in the CrowdRun image. JPEG2000, on the other hand, tends to blur an image, such as the duck shown in figure 5-14a. Edges also become fuzzy, such as the tshirt edges of the runners in figure 5-12a. The lines remain sharp in the H.264-encoded image. On the other hand, JPEG2000 is much better at preserving texture. The grass in the CrowdRun image has more features than the smoothed out version in the H.264-encoded image. However, JPEG2000 creates bright spots that are not smooth with the rest of the image, a result of the wavelet transform. These spots are especially noticeable in the water ripples of the DucksTakeOff image.

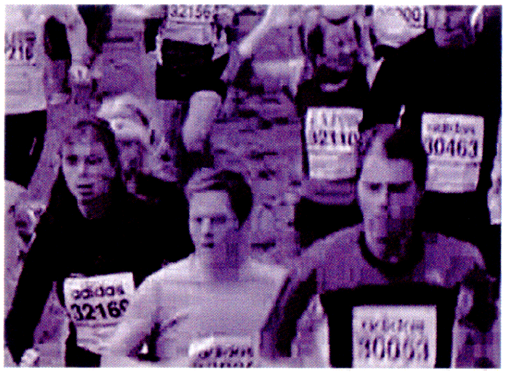


Figure 5-10: CrowdRun (cropped, 321×231) with in-loop deblocking filter disabled.

5.3.2 Artifact Propagation from I-frames to P-frames

If a P-frame has very few intra-coded blocks, many of the artifacts in the I-frame will be propagated to the P-frames. The motion-compensated prediction of the P-frame will retain the artifacts. If a P-frame has many intra-coded blocks, than the

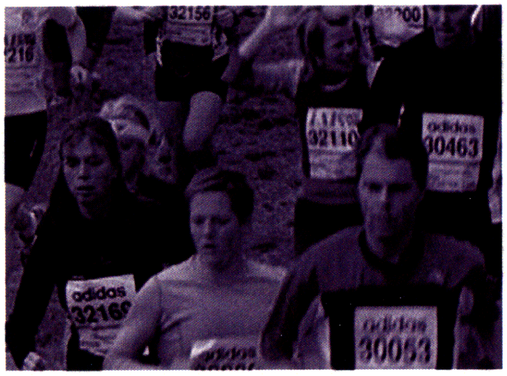
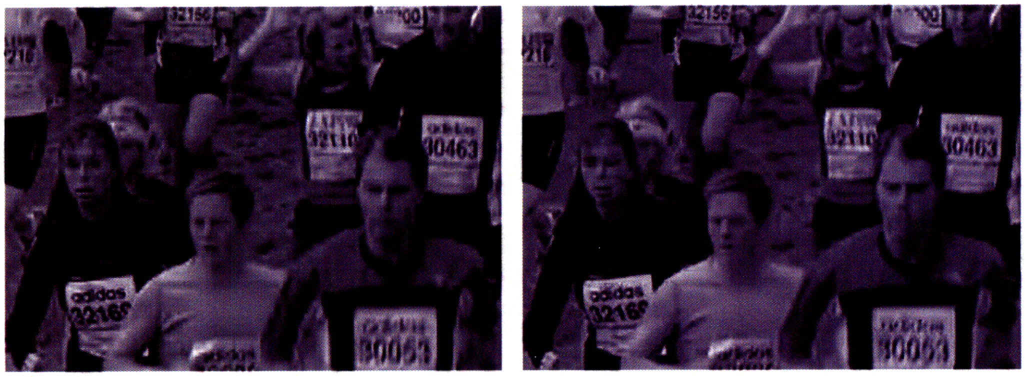


Figure 5-11: CrowdRun (cropped, 321×231) - Original


(a)

(b)

Figure 5-12: CrowdRun (cropped, 321×231): (a) JPEG2000 (PSNR = 28.1 dB); and, (b) H.264 (PSNR = 28.1 dB).

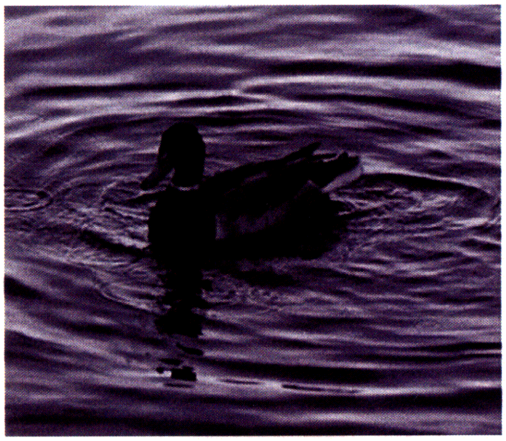


Figure 5-13: DucksTakeOff (cropped, 351×301) - Original

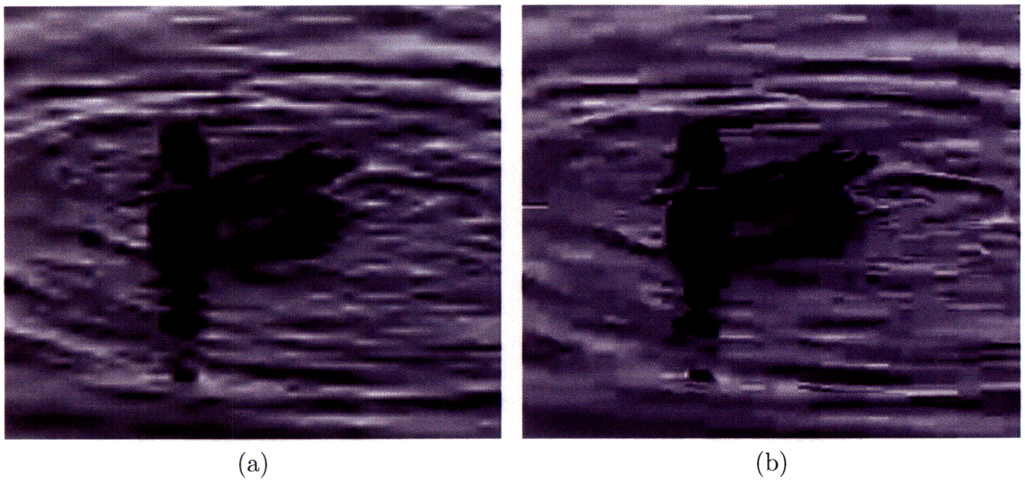


Figure 5-14: DucksTakeOff (cropped, 351×301): (a) JPEG2000 (PSNR = 23.5 dB); and, (b) H.264 (PSNR = 23.5 dB).

H.264 P-frame and the hybrid P-frame would look fairly similar, since both P-frames are encoded with the H.264 algorithm. P-frames tend to have fewer intra-coded blocks when the I-frame PSNR is similar to or greater than the P-frame PSNR and when there is no radical change in the scene content. Figure 5-15 shows a plot of the percentage of intra-coded blocks in the first P-frame versus the PSNR difference between the I-frame and the first P-frame. The figure shows the results for one I-frame PSNR value and five P-frame QP values. Positive values mean that the I-frame PSNR is greater than the P-frame PSNR. JP2 denotes the hybrid system. The general trend is a decrease in intra-coded blocks as the PSNR difference increases. More sequences are not shown in the plot because there is not one trendline for all sequences. The actual percentage of intra-coded blocks also varies with scene content and the absolute PSNR value, not just the PSNR difference. However, the fact that there are fewer intra-coded blocks when the I-frame PSNR is greater than or similar to the P-frame PSNR holds for all sequences.

We will discuss some of the sequences that fit the above criteria and examine the subjective quality of the P-frames. The sequences shown are encoded at low bit-rates so that the encoder artifacts are more prominent. Because the two systems have different artifacts, the visual quality depends heavily on the sequence content. H.264 tends to preserve edges and lines while JPEG2000 preserves textures.

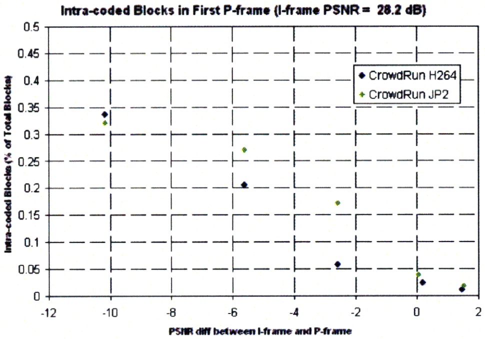


Figure 5-15: Percentage of intra-coded blocks in first-P-frame versus PSNR difference between I-frame and first P-frame. Positive differences denote I-frame PSNR values that are greater than the P-frame PSNR values.

Container

In the container sequence, the scene is dominated by a blocky object, the ship. The original I-frame and first P-frame are shown in figures 5-16a–b. The only motion is the slow movement of the ship across the scene. Figures 5-17a–b shows the JPEG2000 and H.264-encoded I-frames. H.264 preserves the sharp lines of the ship, while JPEG2000 blurs these lines. H.264 does, however, create fake contours on the water and strange artifacts on the trees in the background. Because the P-frame PSNR is approximately the same as the I-frame PSNR, these artifacts are propagated to the P-frames. The first P-frame is shown in figures 5-18a–b. It looks almost identical to the I-frame because there was very little movement. The rest of the P-frames are similar.

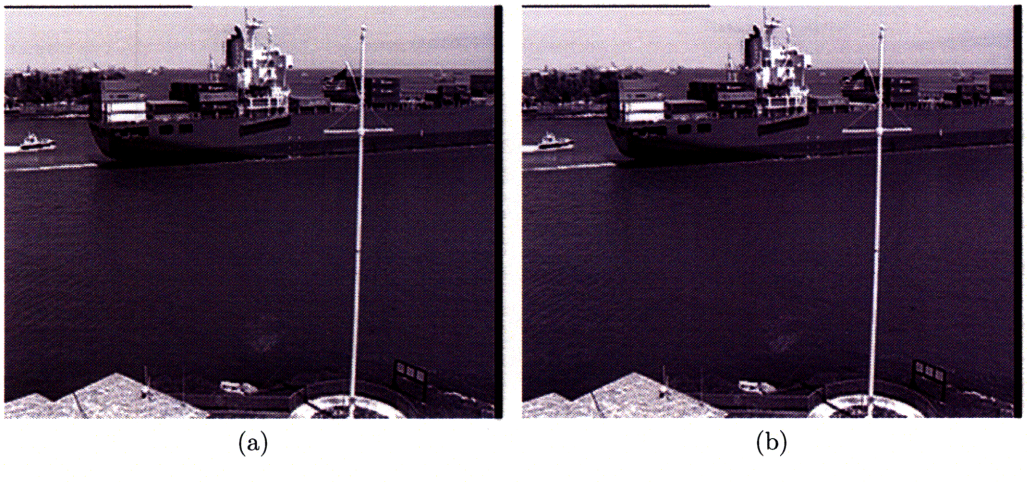
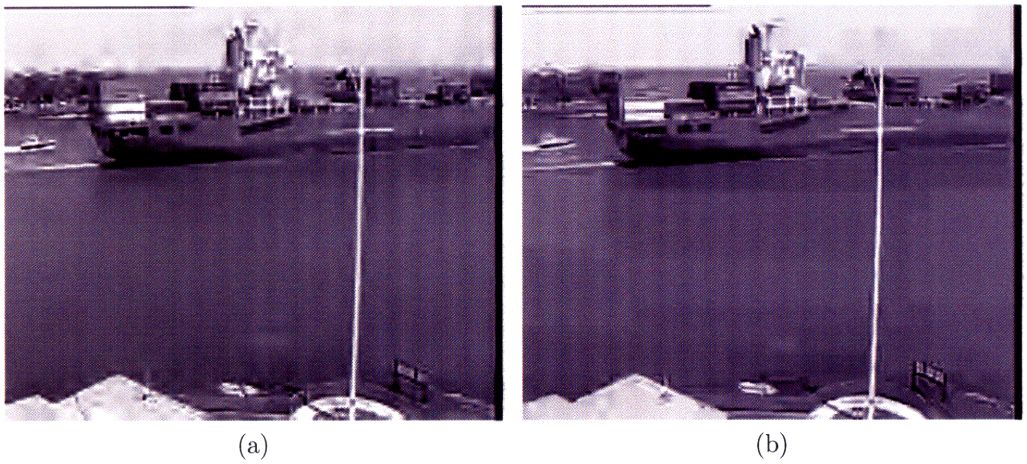
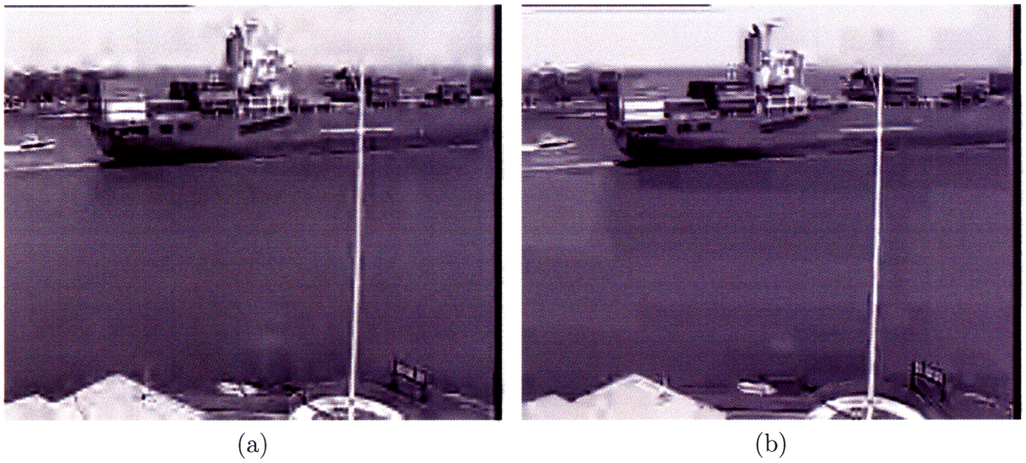


Figure 5-16: Container (CIF): (a) Original I-frame; and, (b) Original first P-frame.



(a) (b)

Figure 5-17: Container (CIF) I-frame: (a) JPEG2000 (PSNR = 25.7 dB); and, (b) H.264 (PSNR = 25.7 dB).



(a) (b)

Figure 5-18: Container (CIF) first P-frame: (a) JPEG2000 (QPPSlice = 44, PSNR = ∼26 dB); and, (b) H.264 ((QPPSlice = 44, PSNR = ∼26 dB).

Flower

In the flower sequence, there are sharp edges on the house, windmill and lamppost and texture in the field of flowers, shown in figures 5-19a–b. This sequence is a good indication of which type of scene content each system is better at encoding. The JPEG2000 and H.264-encoded I-frames are shown in figures 5-20a–b. In the JPEG2000-encoded I-frame, the trees and windmill are very blurry. In the H.264 image, the lines are a bit sharper and there is no halo effect around the windmill blades and the lamppost. In the H.264 image, some parts of the flower field are replaced by solid gray blocks. In the JPEG2000 image, there is a smoother transition between areas that have less spatial detail and the surrounding flowers. Most of these artifacts are carried over to the first P-frames, shown in figures 5-21a–b.

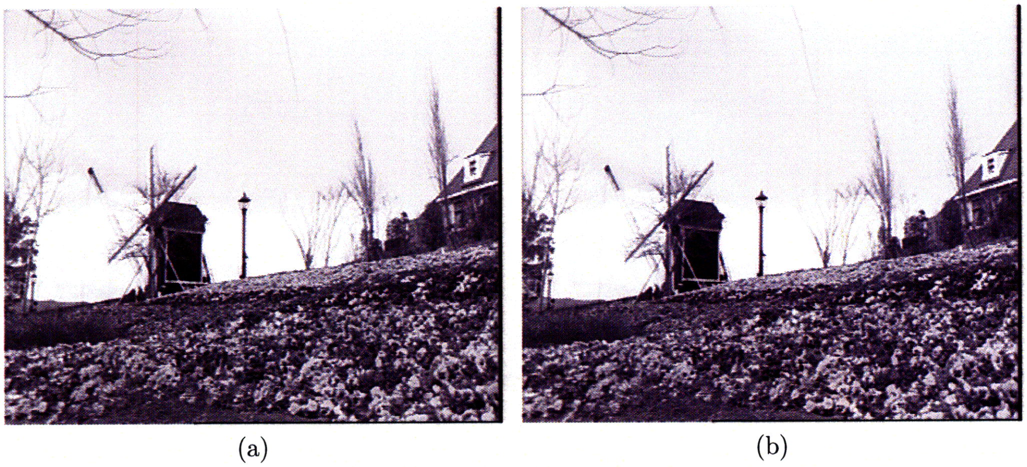
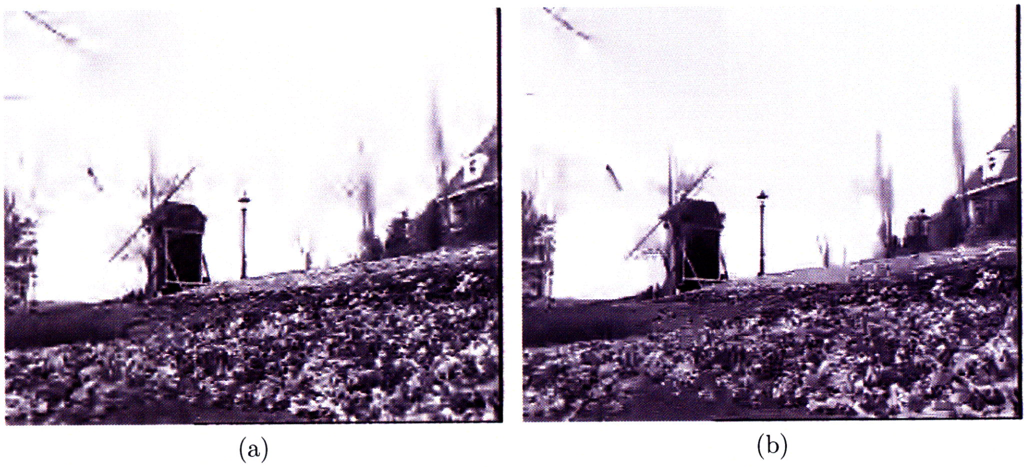


Figure 5-19: Flower (CIF): (a) Original I-frame; and, (b) Original first P-frame.

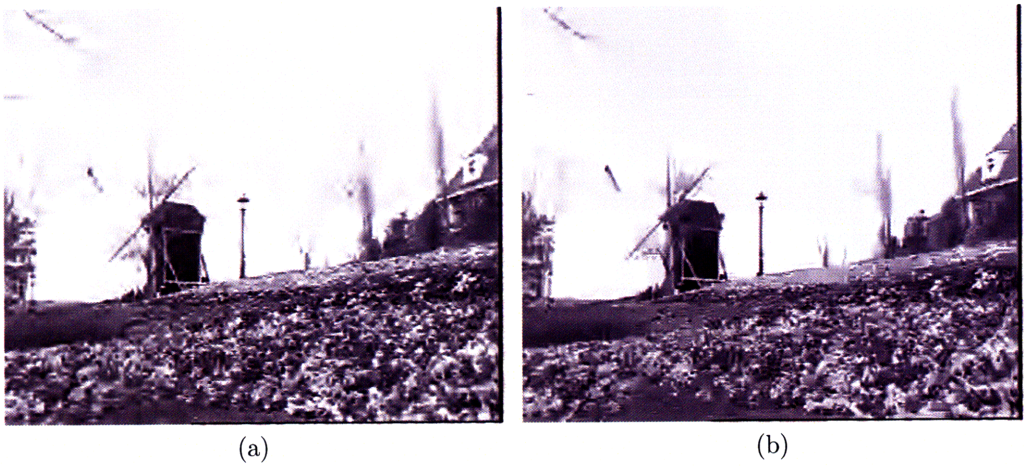
Beertruck

Figure 5-22 shows cropped versions of the I-frame and first P-frame of the Beertruck sequence, in which cars and trucks are moving along a highway. This video sequence has many straight lines that run diagonally across the scene. The H.264 I-frame has noticeable blocking artifacts, shown in figure 5-23b. In addition, one of the cars behind the truck is almost completely erased due to quantization noise. This car is preserved in the JPEG2000 I-frame. The JPEG2000 I-frame also lacks blocking



(a) (b)

Figure 5-20: Flower (CIF) I-frame: (a) JPEG2000 (PSNR = 22.8 dB); and, (b) H.264 (PSNR = 22.8 dB).



(a) (b)

Figure 5-21: Flower (CIF) first P-frame: (a) JPEG2000 (QPPSlice = 44, PSNR = 22.9 dB); and, (b) H.264 (QPPSlice = 44, PSNR = 22.9 dB).

artifacts. Instead, the image is a bit blurry and the sharp edges of the highway are fuzzy. The first P-frame, shown in figure 5-24, has the same artifacts as the I-frame. The car preserved in the JPEG2000 I-frame still appears in the P-frame, while in the H.264 P-frame, it is still erased.

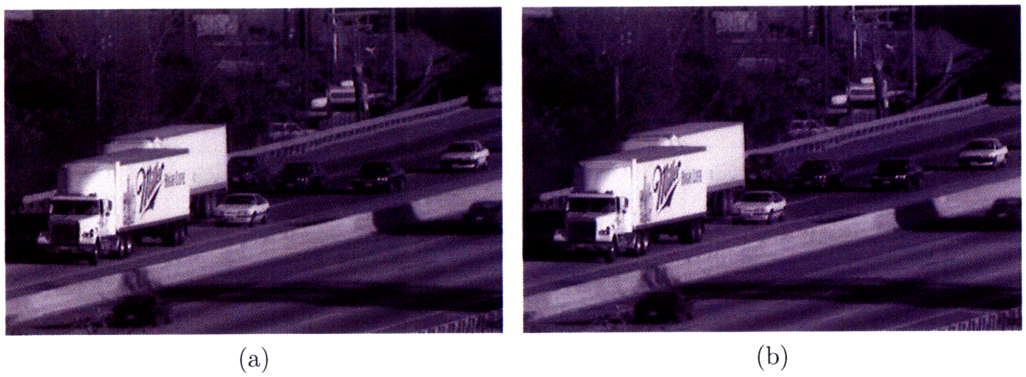

(a)
(b)

Figure 5-22: Beertruck (cropped 531×341): (a) Original I-frame; and, (b) Original first P-frame.

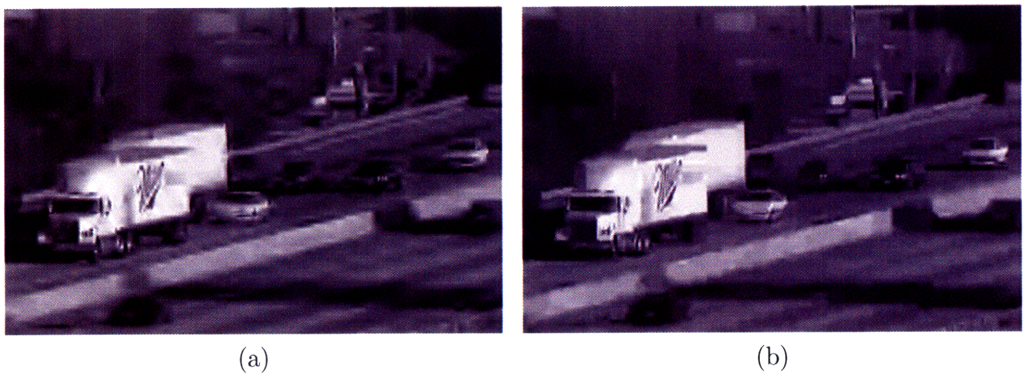
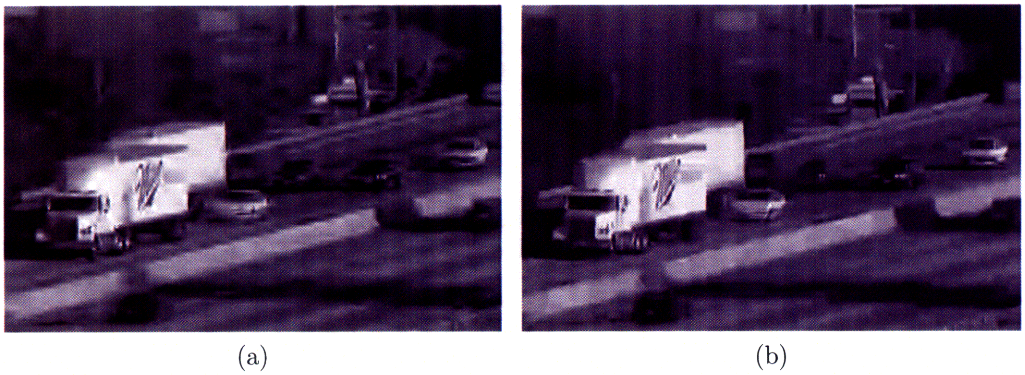

(a)
(b)

Figure 5-23: Beertruck (cropped 531×341) I-frame: (a) JPEG2000 (PSNR = 27.9 dB); and, (b) H.264 (PSNR = 27.9 dB).



(a)

(b)

Figure 5-24: Beertruck (cropped 531×341) first P-frame: (a) JPEG2000 (QPPSlice = 44, PSNR = ∼28 dB); and, (b) H.264 ((QPPSlice = 44, PSNR = ∼28 dB).

Chapter 6

Conclusion

The trend today towards increasing video resolution and quality demands higher compression ratios in order to meet the available storage space and transmission bandwidth. Compression algorithms are evaluated based on objective rate-distortion analysis and subjective quality. In this thesis, we combined JPEG2000, a still image standard, with H.264, a video standard. JPEG2000 was used to encode the I-frames of a video sequence while H.264 was used to encode the P-frames. Because JPEG2000 and H.264 generate very different images at low bit-rates, the expectation was that perhaps using JPEG2000 would lead to improved coding efficiency in the P-frames.

The hybrid system was created by modifying the H.264 reference software to compress the video sequence without compressing the I-frames. For each video sequence, the I-frames would first be compressed independently using JPEG2000. This modified sequence is then input to the H.264 encoder. The JPEG2000-encoded I-frames are used for predicting the P-frames.

The two systems were compared for eight video sequences, which ranged in resolution and picture content. The rate-distortion results, averaged over all 9 P-frames, showed JPEG2000 and H.264 to be comparable. There was more separation between the two systems when only the first P-frame was studied, but overall, the results were still comparable. The maximum first P-frame PSNR difference was a ~0.2 dB gain for H.264.

Subjective results showed JPEG2000 and H.264 to have very different artifacts at low bit-rates. JPEG2000 blurs images and makes edges fuzzy. H.264 smoothes out textures and creates fake contours, but it usually maintains the sharpness of edges. JPEG2000 maintains the appearance of textures, but has very noticeable bright spots. When the number of intra blocks in the P-frames is low, these artifacts are propagated from the I-frame to the P-frame. For sequences that have strong blocking artifacts when encoded with H.264, using the hybrid system eliminates almost all trace of blocking artifacts. For sequences with many strong edges, such as Container, H.264 tends to do a better job of preserving these lines, while the hybrid system blurs them. The subjective performance of the hybrid system depends on the video sequence content.

Finally, this work has shown the hybrid and H.264 systems to have comparable objective performance, but significantly different subjective quality. The image quality of each system is dependent on the video sequence content, so no one system works better in all cases. However, JPEG2000 has many advantages beyond simply image quality, such as scalability and region-of-interest coding. These features have powerful applications and future research can look into exploiting these features for video applications. Scalability allows users with varying internet access capabilities to view different quality versions of the same file. Region-of-interest coding allowing a certain area to be encoded at a much higher quality than the rest. This is potentially useful for video conferencing, where typically only the person's face is of interest. While these features only exist for still images, i.e. I-frames, I-frames require the most bits to transmit. Any savings in the I-frames would be significant for the entire video sequence.

Bibliography

[1] ITU-T recommendation T.800 and ISO/IEC 15444-1 JPEG2000 image coding system: Core coding system (JPEG2000 Part 1). Technical report, 2000.

[2] ITU-T recommendation H.264 and ISO/IEC 14496-10 MPEG-4 part 10, advanced video coding (AVC). Technical report, 2003.

[3] M. Adams and F. Kossentini. JasPer: A software-based JPEG-2000 codec implementation. In *Proc. IEEE ICIP*, 2000.

[4] M. Adams and F. Kossentini. JasPer: A portable flexible open-source software tool kit for image coding/processing. In *Proc. IEEE ICASSP*, 2004.

[5] M. Antonini, M. Barlaud, and I. Daubechies. Image coding using the wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, April 1992.

[6] G. Bjontegaard. Calculation of average PSNR differences between RD-curves. In *VCEG-M33*, 2001.

[7] D. L. Gall and A. Tabatabai. Subband coding of digital images using symmetric short kernel filters and arthmetic coding techniques. In *Proc. IEEE International Conference ASSP*, pages 761–765, New York, 1988.

[8] D. Marpe, V. George, H. L. Cycon, and K. U. Barthel. Performance evaluation of Motion-JPEG2000 in comparison with H.264/AVC operated in intra coding mode. In *Proc. SPIE*, pages 129–137, February 2004.

[9] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):620–636, July 2003.

[10] D. Marpe, T. Wiegand, and S. Gordon. H.264/MPEG4-AVC fidelity range extensions: Tools, profiles, performance and application areas. In *IEEE International Conference on IMage Processing*, pages 593–596, Genoa, Italy, September 2005.

[11] D. Marpe, T. Wiegand, and S. Gordon. Comparative study of JPEG2000 and H.264/AVC FRExt I-frame coding on high-definition video sequences. In *Proc. SPIE Applications of Digital Image Processing*, 2006.

[12] M. Ouaret, F. Dufaux, and T. Ebrahimi. On comparing JPEG2000 and intraframe AVC. In *Proc. SPIE Applications of Digital Image Processing*, August 2006.

[13] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, September 2001.

[14] D. S. Taubman. High-performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, July 2000.

[15] D. Vatolin, A. Moskvin, O. Petrov, and A. Titarenko. JPEG 2000 image codecs comparison. Technical report.

[16] T. Wiegand, G. J. 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):560–576, July 2003.