Implementing Rate-Distortion Optimization on a Resource-Limited H.264 Encoder

by

Eric Syu

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 and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2005

Author .....

Department of Electrical Engineering and Computer Science

January 21, 2005

Certified by...

.....

Jae S. Lim

MIT Professor

Thesis Supervisor

Certified by

.....

Tao Shen

Senior Staff Engineer/Manager, QUALCOMM Incorporated

Thesis Supervisor

Accepted by .

.....

Arthur C. Smith

Chairman, Department Committee on Graduate Students

Implementing Rate-Distortion Optimization on a Resource-Limited H.264 Encoder

by

Eric Syu

Submitted to the Department of Electrical Engineering and Computer Science
on January 21, 2005, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis models the rate-distortion characteristics of an H.264 video compression encoder to improve its mode decision performance. First, it provides a background to the fundamentals of video compression. Then it describes the problem of estimating rate and distortion of a macroblock given limited computational resources. It derives the macroblock rate and distortion as a function of the residual SAD and H.264 quantization parameter QP. From the resulting equations, this thesis implements and verifies rate-distortion optimization on a resource-limited H.264 encoder. Finally, it explores other avenues of improvement.

Thesis Supervisor: Jae S. Lim
Title: MIT Professor

Thesis Supervisor: Tao Shen
Title: Senior Staff Engineer/Manager, QUALCOMM Incorporated

Acknowledgments

I would like to acknowledge the following people for their contributions.

First, my family, whose love I will always cherish: my father Jr-Jung Syu, my mother Ying Syu, and my brother Jonathan Syu.

Second, my spiritual families, who have been my homes away from home: the Chinese Bible Church of Greater Boston and Harvest San Diego.

Third, my coworkers and technical mentors at QUALCOMM Incorporated, who have helped me immensely: Xuerui Zhang and Yi Liang.

Finally, all glory be to God, the ultimate author and perfector of all things.

Contents

List of Figures

Chapter 1

Introduction and Background

Digital video compression presents a number of challenges to both academia and industry. For academia, digital video represents the ultimate exercise in compression theory. It requires massive amounts of raw data, yet much is redundant or irrelevant to the human visual system. Video compression tries to eliminate such extraneous information. For industry, consumers are demanding digital video everywhere: in television, movies, telephony, and the Internet. Companies are responding as quickly as possible, but as digital video spreads to new domains, it faces increasingly restrictive resource constraints, whether bandwidth, processing speed, memory size, or power consumption. Cellular phones, for example, can offer very little in all four categories.

In light of these limitations, what is the highest quality video achievable given certain resource constraints? This thesis describes a possible approach by optimizing the process of mode decision in an H.264 encoder using rate-distortion theory.

Chapter 1 provides a background primer to video compression and rate-distortion theory. It also examines prior results established in the literature. Chapter 2 describes the specific problem addressed in this thesis. Chapter 3 models the problem from its parameters: rate, distortion, and λ (Lagrange multiplier). Chapter 4 shows the final solution based on the model.

1.1 Literature Review

The general problem of maximizing quality while minimizing cost is known as *rate-distortion theory* [3]. The terminology originates from Shannon's first formulation of the problem [4], in which he demonstrated how much distortion to expect when transmitting discrete symbols over a noisy channel at a given rate. He showed that at rates below the channel capacity, distortion can be minimized to an arbitrarily small value, whereas at rates above the channel capacity, distortion can never be avoided.

Since Shannon developed his theories of communication fifty years ago, technology has witnessed a race between higher rates and higher channel capacities. Today, video represents one of the most demanding applications in communications. Raw, low-resolution video without audio requires more than 6 megabits per second for accurate reproduction [5]. Few storage mediums, much less transmission channels, can practically handle such large quantities of data.

Because of these resource limitations, it is necessary to compress video to reduce its bit rate. Many application domains, such as file transfers over the Internet, require "lossless" compression, such as by using the classical Lempel-Ziv algorithm [6]. These lossless algorithms exploit statistical properties of the data source to reduce the bit rate. For example, they assign few bits to represent symbols with high probabilities of occurring and many bits for symbols with low probabilities of occurring. Unfortunately, lossless compression algorithms cannot compress raw video (or even some still images) enough for standalone use.

Furthermore, lossless compression, where every bit is reproducible, is unnecessary. The human visual system simply cannot perceive some kinds of information. This information is irrelevant and can be eliminated without loss of quality when performing video compression. Also, certain visual properties are more important to the human visual system than others [7]. Properties of greater importance need to be represented accurately, usually by assigning more bits to them, while those of lesser importance require fewer bits. As a simple example, consider an object with sharp edges in a video sequence. When the object is stationary, any blurring of the edges caused by

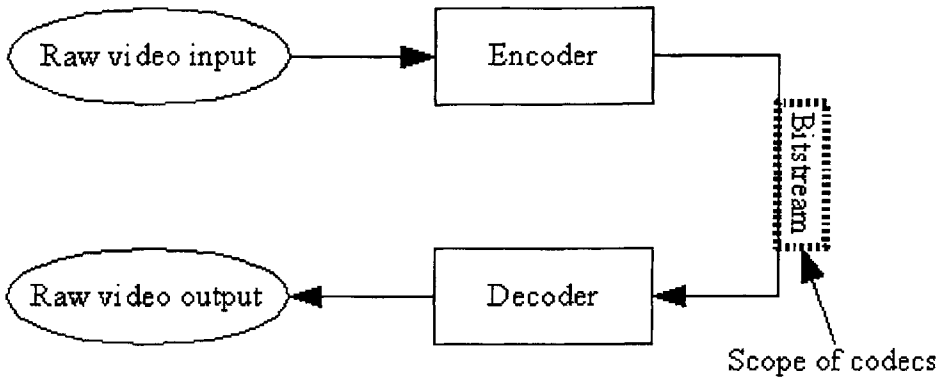

Figure 1-1: Complete video compression system

distortion is noticeable and undesirable. When the object is moving quickly, though, blurred edges are practically expected. A video compression algorithm might take advantage of this phenomenon by assigning less bits to edges during a sequence of rapid motion.

1.2 Codecs

Video evidently contains a significant amount of compressible information. Commercial demand for video applications has led to the development of several standards for compression, known as *codecs*. Two especially important sets of codecs have emerged: the MPEG-x series from the Motion Picture Experts Group (MPEG) and the H.26x series from the International Telecommunications Union (ITU-T). Many of these codecs have entered wide commercial use, such as MPEG-2 for high definition television and H.263 for videoconferencing [1].

Each successive generation of codecs incorporates more advanced compression techniques. The MPEG and ITU-T organizations are working jointly to develop the next codec, known as both H.264 and MPEG-4 version 10. The complete specification for H.264 [8] is extremely intricate, but a good overview can be found in [9].

All codec specifications share a single, narrow goal: to provide a universally decodable bitstream syntax [9]. As shown in Figure 1-1, the bitstream in fact represents only part of a complete video compression system. The specification defines only the

range of possible bitstreams, remaining silent about the actual implementation of the encoder or decoder. An encoder can fail to compress a video sequence at all, yet produce a standard-compliant bitstream.

As a result, engineers have wide latitude when designing a specific encoder. An encoder's design depends heavily upon its purpose. Some applications demand high quality, such as HDTV. Others require real-time encoding, such as teleconferencing. Still others must cope with unreliable channels, such as Internet streaming video. In addition, the content of a given video sequence varies widely with time. An encoder must dynamically decide how best to represent static images, scene changes, and object movement. To do so, it must balance the competing goals of minimum rate and minimum distortion.

In general, though, encoders for nearly any codec share a common structure, as shown in Figure 1-2. First, they divide a video frame into smaller regions known as macroblocks. Encoders may encode a macroblock in two ways: without motion compensation (INTRA-coded) or with motion compensation (INTER-coded). INTRA-coded macroblocks do not depend on any frame other than the current one. They are first converted from their raw format (usually RGB) to YIQ format, where Y is the luma component and I/Q are the chroma components. YIQ is a more suitable basis for compression than RGB because the chroma components can be significantly downsampled without much loss of visual quality.

The YIQ values are further processed with a transform such as the DCT or wavelet transform. These transforms concentrate the macroblock's energy into a small number of large coefficients. They also produce a much larger number of small coefficients. This enables the next step, quantization of the transform coefficients. Quantization is responsible for the "lossy" part of compression. It limits possible coefficient values and eliminates small values. Because many small coefficients usually exist as a consequence of the transform, quantization allows significant reduction of the bit rate. Finally, the quantized coefficients are coded into a bitstream as efficiently as possible and stored in a buffer for eventual output.

INTER-coded macroblocks rely on interframe temporal redundancy to reduce the

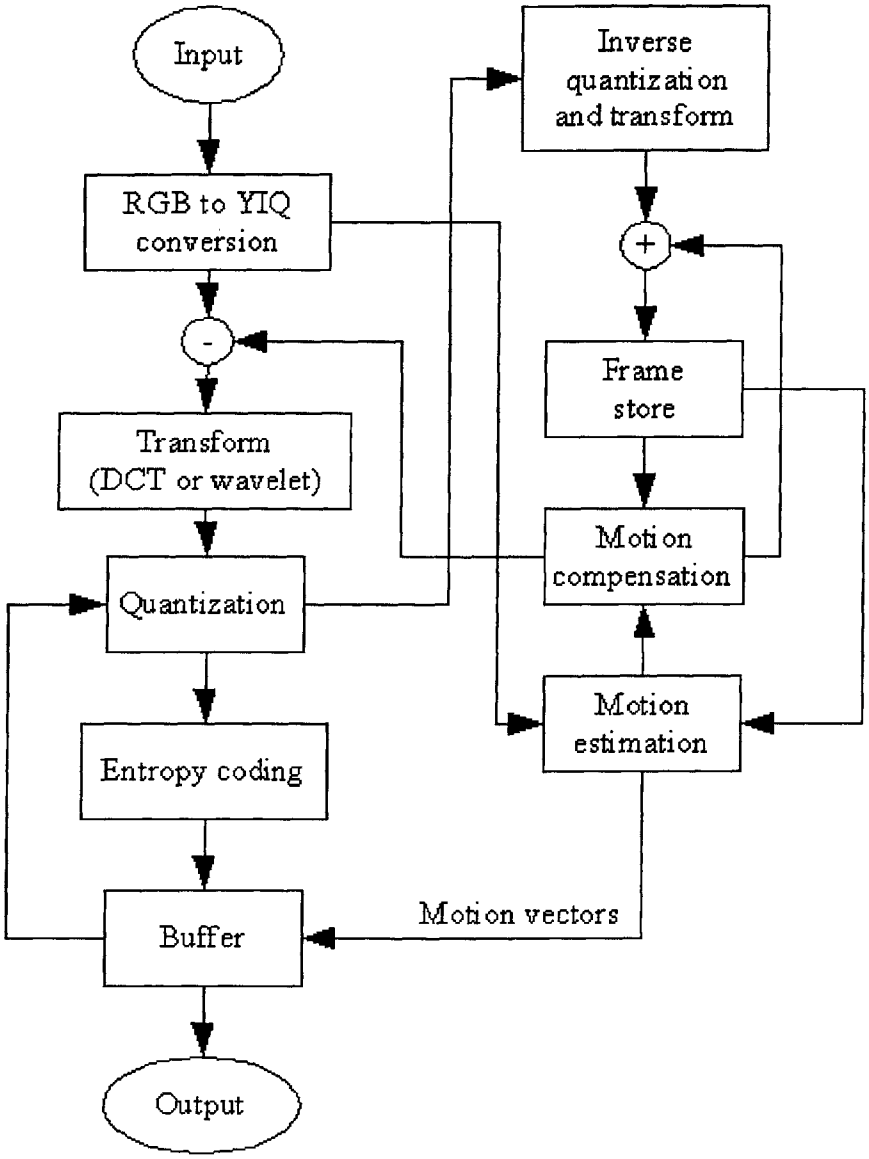

Figure 1-2: Generic interframe video encoder [1, 2]

bit rate. In many video sequences, adjacent frames differ only because of object movement. Motion compensation captures the movement in motion vectors instead of independently coding the final result. This technique produces significant compression because motion vectors require many fewer bits than complete images. Motion compensation typically involves two steps. First, for each region in the macroblock, the encoder determines the optimal motion vector by searching another frame for the best matching reference region, a process called motion estimation. Then the encoder subtracts the current region from the best matching reference region, which yields a residual difference. This residual undergoes the same transform, quantization, and run-length coding as INTRA-coded macroblocks. The residual after transformation usually contains an even greater number of small coefficients than INTRA-coded macroblocks do. As a result, INTER-coded macroblocks can achieve significant bit rate savings over INTRA-coded ones.

1.3 Mode Decision in H.264

Modern codecs such as H.264 have many more modes than simply INTRA or INTER. The process of deciding among them is appropriately named *mode decision.* On the frame level, a frame can be specified as an I-frame (INTRA-coded), a P-frame (predictive-coded), or a B-frame (bipredictive-coded)[1]. I-frames may contain only INTRA-coded macroblocks. P-frames may have INTER-coded macroblocks that use previous frames as reference frames for motion compensation. B-frames can also use future frames as reference frames.

On the macroblock level, the choices multiply significantly. A macroblock in H.264 consists of 16×16 luma samples and 8×8 samples of both chroma components. Two INTRA macroblock modes exist: INTRA-4×4 and INTRA-16×16. INTRA macroblocks in H.264 are independent of other frames. However, they can be spatially predicted from adjoining macroblocks in the same frame. Spatial prediction can reduce the bit rate because adjacent macroblocks are often similar to each other. With

[1]In H.264, slices, not frames, are coded as I, P, or B, but the idea is similar.

INTRA-16 $\times$ 16 mode, the entire macroblock is predicted from another macroblock. With INTRA-4 $\times$ 4 mode, each of the 16 4 $\times$ 4 blocks within a macroblock is predicted from adjoining blocks.

Four INTER macroblock modes exist as well: INTER-16 $\times$ 16, INTER-16 $\times$ 8, INTER-8$\times$16, and INTER-8$\times$8. The INTER-coded macroblocks have motion vectors associated with them. The $m \times n$ notation refers to what size of luma samples each motion vector represents. For example, an INTER-8 $\times$ 8 macroblock has 4 motion vectors. In fact, 8 $\times$ 8 partitions can be even further subdivided into 8 $\times$ 4, 4 $\times$ 8, and 4 $\times$ 4 modes, so a macroblock can potentially have up to 16 motion vectors [9].

For each INTER-coded macroblock, the encoder needs to determine what motion vectors most accurately capture the motion from one frame to another. The motion vectors have quarter-sample granularity, meaning object movement can be represented to an accuracy of one quarter of a luma sample. The more accurate the motion vector is, the smaller the residual becomes. Fast motion estimation techniques such as log search [2] can simplify the computation required to determine accurate motion vectors.

Finally, all modes, whether INTRA or INTER, rely on a quantization parameter QP, which determines how much information is lost during compression. In H.264, QP ranges over 52 values [9]. It corresponds to the quantization step size Q in the following manner [10, 8]:

$$Q = 2^{(QP-4)/6} \tag{1.1}$$

As QP increases, rate decreases and distortion increases.

1.4 Lagrangian Optimization

The multitude of options described in the previous section poses a significant challenge to encoder design. Intuitively, the encoder should make an optimal mode decision by minimizing distortion under a bit rate constraint. The problem can be formulated in the following manner [11]. Consider a vector of source samples $\mathbf{S}$ (for example,

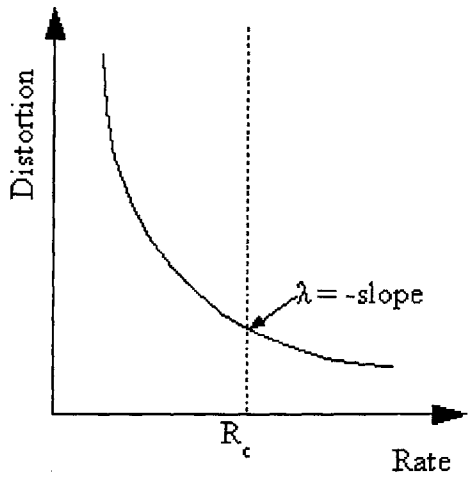

Figure 1-3: Relationship between R_c and λ

macroblocks), and a vector of modes $\mathbf{I}$ such that I_k corresponds to the mode selected for S_k. Let $D(\mathbf{S}, \mathbf{I})$ be the distortion and $R(\mathbf{S}, \mathbf{I})$ be the bit rate. The goal, then, is to find $\mathbf{I}$ such that $D(\mathbf{S}, \mathbf{I})$ is minimized, subject to a constraint $R(\mathbf{S}, \mathbf{I}) \leq R_c$.

Using the preceding formulation, the task of mode decision is reduced to a classic budget constrained allocation problem. The usefulness of Lagrange multipliers from undergraduate calculus is immediately apparent. However, Lagrange multipliers in the context of undergraduate calculus apply only to differentiable functions, which $D(\mathbf{S}, \mathbf{I})$ and $R(\mathbf{S}, \mathbf{I})$ clearly are not. Fortunately, Everett proved that for the purposes of min/max optimization, differentiability or even continuity is not required [12]. As a result, the objective can be described as finding $\mathbf{I}$ such that the Lagrangian cost function $J(\mathbf{S}, \mathbf{I}) = D(\mathbf{S}, \mathbf{I}) + \lambda \cdot R(\mathbf{S}, \mathbf{I})$ is minimized. The Lagrange multiplier λ replaces the rate constraint R_c on the rate-distortion curve as shown in Figure 1-3. In a way, $\lambda = -\frac{dD}{dR}\Big|_{R=R_c}$, except that $D(\mathbf{S}, \mathbf{I})$ is typically not differentiable. However, the intuition holds.

Theoretically, the encoder can try every possible $\mathbf{I}$ to find the minimum $J(\mathbf{S}, \mathbf{I})$. However, such a method would require testing K^N different $\mathbf{I}$s, where K is the number of source samples and N is the number of possible modes. Given that the source samples $\mathbf{S}$ usually consist of macroblocks, this is computationally unacceptable. Consequently, independence among source samples is assumed so that:

$$J(\mathbf{S}, \mathbf{I}) = \sum_{k=1}^{K} J(S_k, I_k)$$

The independence assumption is not wholly realistic. Macroblocks often explicitly depend on other macroblocks [3], either through spatial prediction for INTRA modes or temporal prediction for INTER modes. Techniques that account for dependency do exist [13], but for the purposes of this thesis independence is assumed, as it seems to have little impact on the optimality of the solution [3].

The final formulation of the mode decision problem follows. For each source sample S, choose a mode I such that

$$J(S, I) = D(S, I) + \lambda \cdot R(S, I) \tag{1.2}$$

is minimized, where $D(S, I)$ is the distortion, $R(S, I)$ is the rate, and λ is the Lagrange multiplier that specifies the rate constraint.

1.5 Practical Considerations in H.264

The set of possible modes is still very large, even when macroblocks are assumed to be independent. No encoder can try every single quantization parameter, every candidate motion vector, and every INTRA/INTER prediction option. A less complex procedure, even if suboptimal, would be preferable. The ITU-T H.264 reference software encoder provides one such procedure [14].

The reference encoder performs Lagrangian optimization in an iterative manner. First, it assumes a quantization parameter QP, usually based on the previous frame. Then it finds the optimal motion vector(s) $\mathbf{m}$ for a macroblock S by minimizing the cost function $J_{MOTION}(S, \mathbf{m}) = D_{DFD}(S, \mathbf{m}) + \lambda_{MOTION} R_{MOTION}(S, \mathbf{m})$ for each INTER mode (more on λ_{MOTION} later, DFD stands for displaced frame difference) [5, 11, 15, 16]. Finally, it evaluates the cost function $J_{MODE}(S, I) = D_{REC}(S, I) + \lambda_{MODE} R_{REC}(S, I)$ and determines the optimal mode I among all INTRA and INTER

modes.

At first glance, it appears there are three independent parameters in the algorithm used by the H.264 reference software: QP, λ_{MODE}, and λ_{MOTION}. However, this is not the case, as shown in [5, 15]. The H.264 reference software uses the following experimentally-obtained relationships [11, 14]:

$$\lambda_{MODE} = 0.85 \cdot 2^{(QP-12)/3} \tag{1.3}$$

$$\lambda_{MOTION} = \sqrt{\lambda_{MODE}} \tag{1.4}$$

The existence of these relationships makes intuitive sense because a fixed quantization parameter QP heavily influences which Lagrange parameters λ_{MODE} and λ_{MOTION} are reasonable. For example, consider a low QP, meaning little quantization. Then the rate will likely be high and the distortion low, indicating the objective is to minimize distortion regardless of rate. A low λ_{MODE} would weight the distortion heavily. The square root for λ_{MOTION} is present because the H.264 reference software uses SAD as the distortion measure for motion estimation (D_{DFD}) and SSD for mode decision (D_{REC}). As a result of these relationships, the only independent parameter is QP, which is either experimentally fixed or obtained through a rate control algorithm to approximate the bit rate budget R_c. Section 3.3 explains the derivation of Equation 1.3 in more detail.

Chapter 2

Problem setup

Accurately calculating rate and distortion requires encoding and decoding each source sample for each mode. Unfortunately, some encoders cannot perform the entire sequence of operations in Figure 1-2 for each mode because it requires too many computational resources. Only a subset is possible. As a result, rate and distortion must be estimated from a limited amount of information. This chapter describes what information is available.

2.1 Encoder description

The H.264 encoder in this thesis is limited by the availability of computational resources, which in turn affects its architectural design and the mode decision process. The encoder is part of a chipset used in embedded applications, particularly cellular phones. Like many embedded chipsets, it is more economical to use several specific-purpose chips instead of one general-purpose microprocessor. As a result, the encoder functionality is split among three components: a proprietary digital signal processor (DSP) core, video acceleration hardware, and an ARM macrocell. The DSP chip is the most flexible and easily programmable component of the video encoder. It acts mainly as a control unit by telling the hardware when to run. The hardware performs computationally intensive and repetitive tasks, such as transforms and motion estimation. The ARM macrocell does the final run-length encoding. Such a design

enables the encoder to compress a video sequence at 15 frames per second with a frame size of 288×352 pixels, despite requiring less than 20,000 software instructions to implement.

Because of the encoder's architecture, the mode decision process is concentrated in the hardware. The encoder only supports the following modes: INTRA-4×4, INTRA-16×16, INTER-16×16, and INTER-8×8. For both INTER-16×16 and INTER-8×8 modes, the hardware searches the previous frame for the best matching motion vectors. Once the motion vectors are found, the hardware chooses between INTER-16×16 and INTER-8×8 mode using the following pseudocode:

```
if (SAD(INTER-16x16) < SAD(INTER-8x8) + threshold)
  choose INTER-16x16;
else
  choose INTER-8x8;
```

SAD stands for the *s*um of *a*bsolute *d*ifferences. It is the absolute sum of all values in the residual, which is formed by subtracting the reference macroblock from the current motion-compensated macroblock. The intuition is that smaller SADs are better. A small SAD implies small values in the residual, which raise the chance they will be quantized to 0 and lower the resulting bit rate.

Choosing on the basis of SAD alone, though, leads to incorrect results. The INTER-8×8 SAD should never be greater than the INTER-16×16 SAD. INTER-8×8 mode uses four motion vectors per macroblock as opposed to one motion vector for INTER-16×16 mode. This can produce greater accuracy when estimating motion because each 8×8 block is treated independently. In the worst case, the four motion vectors can yield the exact same SAD as one motion vector by being arranged in a square, as shown in Figure 2-1. However, INTER-8×8 mode comes at a cost not captured by the SAD. It needs to encode four instead of one motion vector, which increases the rate and hence the cost. The threshold estimates the additional cost of selecting INTER-8×8 mode over INTER-16×16 mode.

This thesis models how to choose the correct threshold in order to increase the

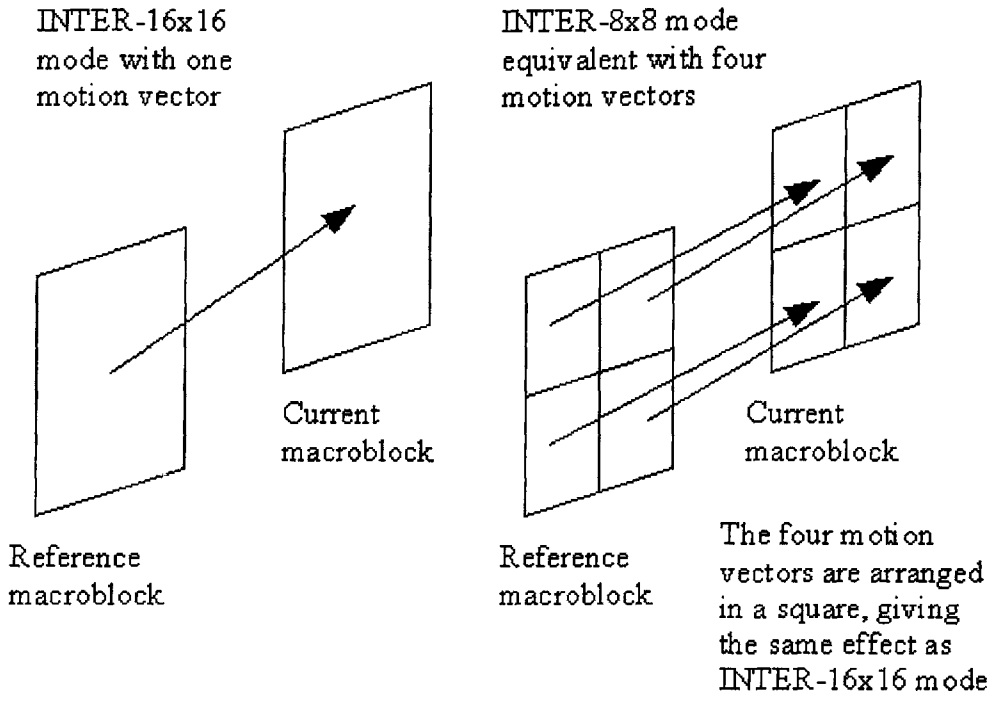

Figure 2-1: Special arrangement of motion vectors in INTER-8 $\times$ 8 mode to mimic INTER-16 $\times$ 16 mode

encoder's rate-distortion performance. Originally, the threshold was a hard-coded number, typically 200. To improve the threshold, it is necessary to describe how the SAD and other available information, especially QP, affect each stage of the encoding process.

2.2 Preliminary experimental results

As an initial experiment, four video sequences were encoded repeatedly. For each of them, the threshold was varied among a range of values and while all other variables remained constant. This process was repeated for different QPs. Using Equation 1.2, the Lagrangian cost for each encoded sequence was calculated. The aggregate size in bits of all the INTER frames was used as the rate. The sum of squared differences (SSD) between every original and compressed INTER frame was used as the distortion. Equation 1.3 was used for λ_{MODE}. Figure 2-2 shows an example plot of threshold versus cost for $QP = 15$ when encoding *Foreman*.

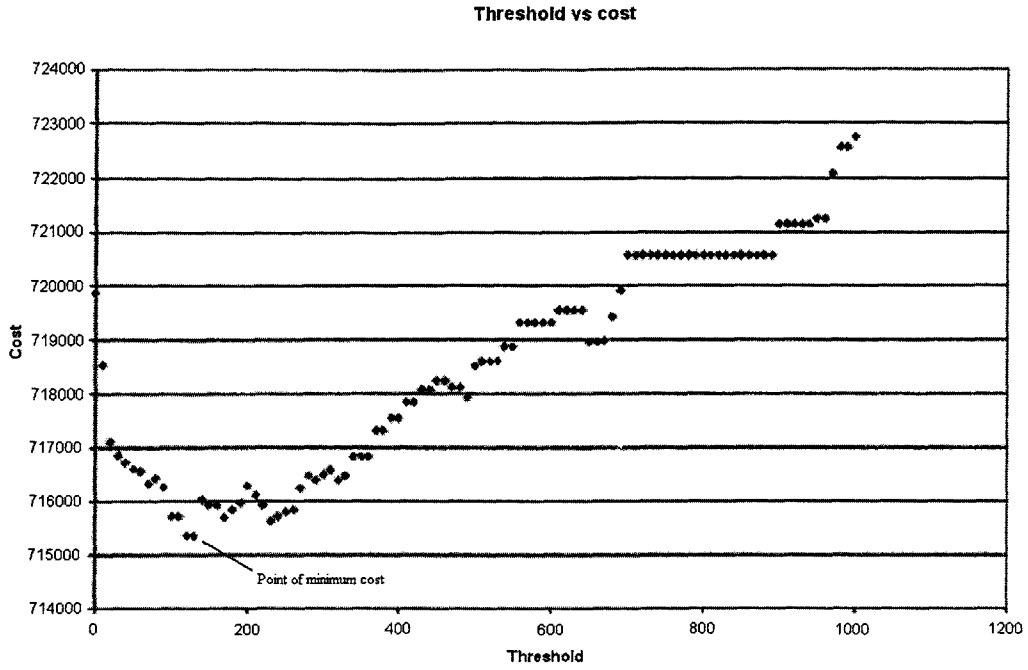

Figure 2-2: Varying INTER-8×8 threshold versus resulting cost (*Foreman*, $QP = 15$)

Other sequences and QPs yield similar plots as Figure 2-2. For low thresholds, the cost is high because only INTER-8 $\times$ 8 modes are being selected, which increases the rate. As the threshold rises, the cost falls but then rises again as INTER-16 $\times$ 16 blocks introduce more distortion.

For each QP, the threshold yielding the lowest Lagrangian cost was determined. The results, depicted in Figure 2-3, were in line with expectations. As QP increased, the optimal threshold increased. The intuition behind it made sense: as the distortion caused by QP increased, the Lagrangian cost formula placed more and more emphasis on the rate. In fact, at the highest QPs that render the video sequence almost unwatchable, the distortion is largely constant, so only rate matters, and in such cases INTER-16 $\times$ 16 mode is the logical choice. At low QPs, rate is always high, and distortion takes greater importance.

Judging from Figure 2-3, the relationship between the optimal threshold and QP is exponential. Figure 2-4 graphs the same data with a logarithmic y-axis. Linear regressions were performed on each sequence. From the graph, it appears that the

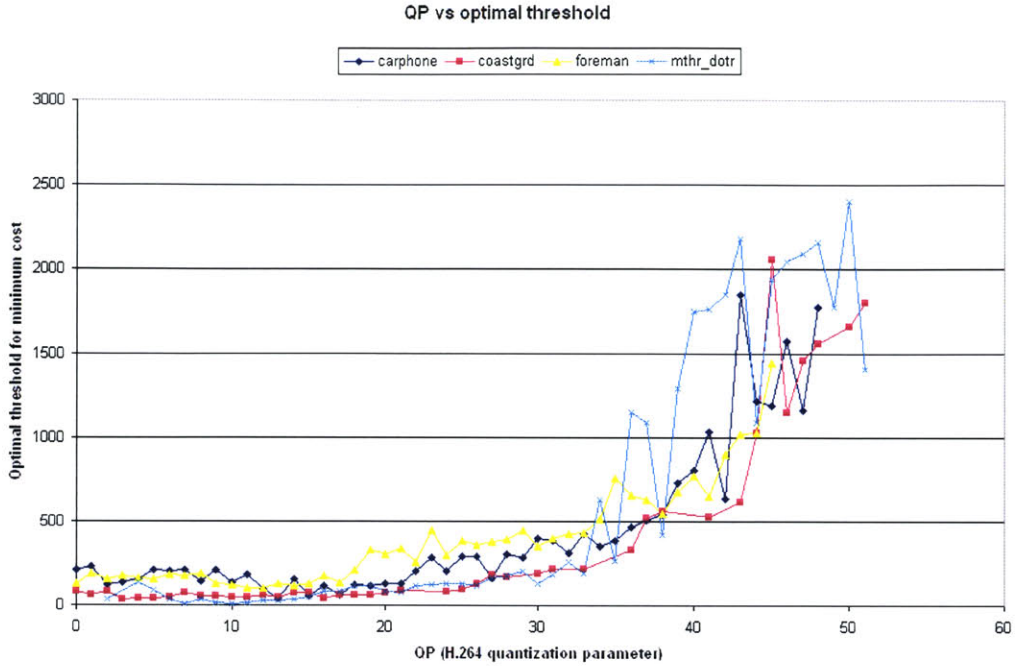

Figure 2-3: QP versus optimal threshold for minimum cost (linear y-axis)

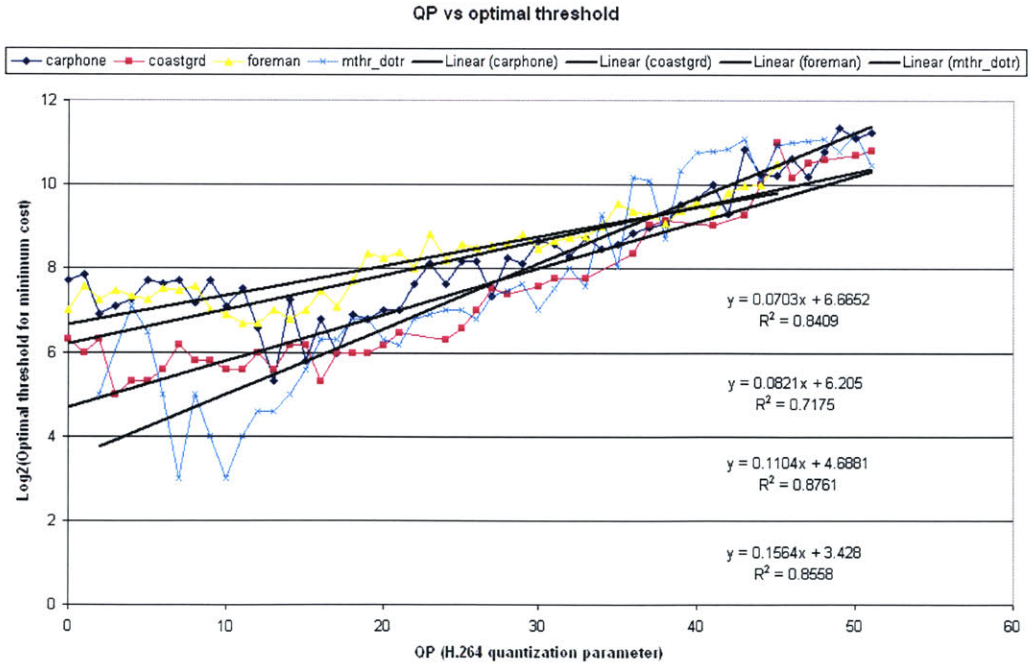

Figure 2-4: QP versus optimal threshold for minimum cost (logarithmic y-axis)

appropriate slope and intercept are highly content-dependent but still consistent with an exponential relationship.

To explain Figure 2-4, this thesis builds a theoretical framework to account for the effects of SAD on the Lagrangian cost function. The SAD affects both the rate and distortion components of the cost function, so they are examined separately and combined later.

Chapter 3

Cost equation parameter models

The cost function in Equation 1.2 involves three components: rate (R), distortion (D), and the Lagrangian multiplier (λ). Each of these components is modeled in this chapter.

3.1 Rate

The relationship between SAD and rate is examined first. Rate is intuitively a function of SAD, because smaller SADs imply that more values in the residual can be eliminated with quantization. The encoder calculates the residual SAD after motion compensation. Then it transforms the residual values, quantizes the transform coefficients using QP, and performs run-length coding on the quantized coefficients. Each step affects the final rate and is modeled here.

First, we explain the relationship between the SAD and the resulting transform coefficients. The H.264 codec uses an integer transform similar to the DCT with energy-preserving properties. In other words, the relationship in Equation 3.1 holds between the space domain and DCT domain.

$$\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}|x(n_1,n_2)|^2 = \sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}|C_x(k_1,k_2)|^2 \tag{3.1}$$

Equation 3.1 says that the energy of the original residual values equals the energy

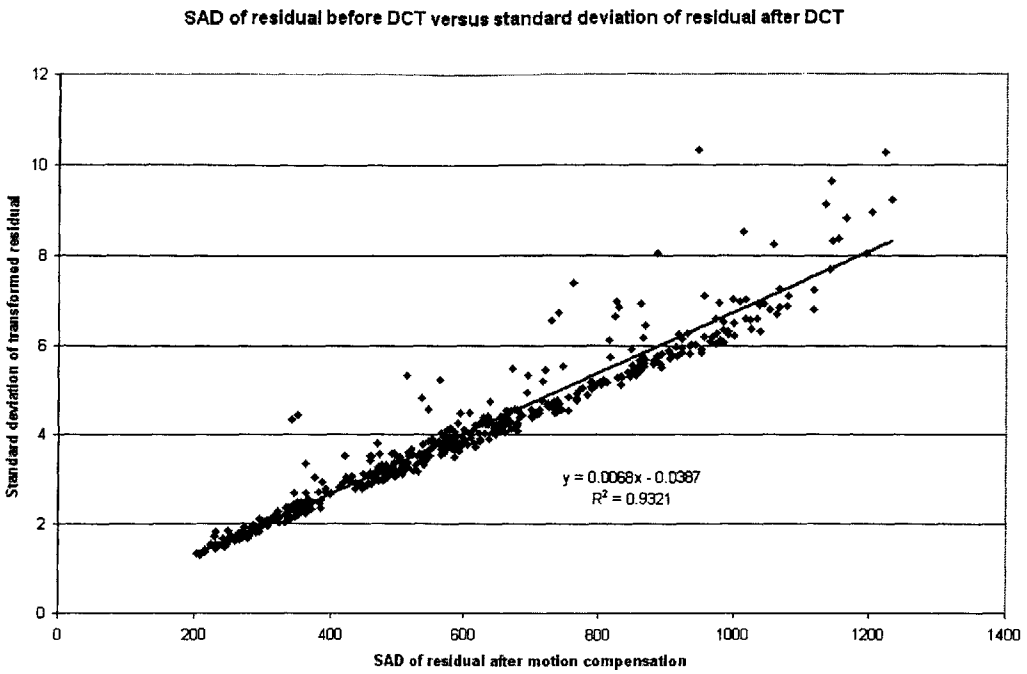

Figure 3-1: SAD of residual error vs standard deviation of DCT coefficients

of their DCT coefficients. The encoder gives only the absolute sum of the residual values, not the energy. However, we can test whether a relationship exists between the SAD and energy. Figure 3-1 graphs the SAD of some macroblock residuals with the standard deviation of their DCT coefficients, calculated after quantization at $QP = 0$. The standard deviation is the square root of the variance, which in turn is the energy divided by the number of macroblock values (256). The graph shows a fairly strong linear relationship between the two metrics.

To explain this relationship, we model the luma coefficients of the residual prior to transformation with a zero-mean Gaussian distribution, whose probability distribution is shown in Equation 3.2.

$$p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)} \tag{3.2}$$

The expected absolute value of a Gaussian random variable is shown in Equation 3.3. The calculation is similar to finding the expected value of a half-normal distribution and may be found in Appendix A.1.

$$E[|X|] = \int_{-\infty}^{+\infty} |x| p_X(x) dx$$
$$= \sigma\sqrt{\frac{2}{\pi}} \tag{3.3}$$

A macroblock consists of 256 luma coefficients. Equation 3.4 expresses the SAD as a sum of random variables.

$$\text{SAD} = \sum_{i=0}^{255} |X_i| \tag{3.4}$$

where the X_is are i.i.d. Gaussian random variables with zero mean and common variance σ^2. The goal is to estimate σ^2 given the SAD. Rearranging Equations 3.3 and 3.4 yields Equation 3.5.

$$E[\sigma] = \frac{\text{SAD}}{256}\sqrt{\frac{\pi}{2}} \tag{3.5}$$

Because of the energy-preserving properties of the DCT in Equation 3.1, the standard deviations of the residual before and after transformation are both equal to σ. However, instead of a Gaussian distribution, the statistics of the transformed coefficients resemble that of a Laplace distribution [17, 18], which has the pdf shown in Equation 3.6. Appendix A.2 shows that b equals $\sigma/\sqrt{2}$.

$$p(x) = \frac{1}{2b} e^{-|x|/b} \tag{3.6}$$

Figure 3-2 graphs the pdf of the Laplace distribution. The integral of $p(x)$ gives the cdf of Equation 3.7, which is easier to use because of the discrete nature of the DCT coefficients.

$$P(x) = \begin{cases} \frac{1}{2} e^{x/b}, & x < 0 \\ 1 - \frac{1}{2} e^{-x/b}, & x \geq 0 \end{cases} \tag{3.7}$$

To verify the accuracy of modeling the DCT coefficients as a Laplace distribution,

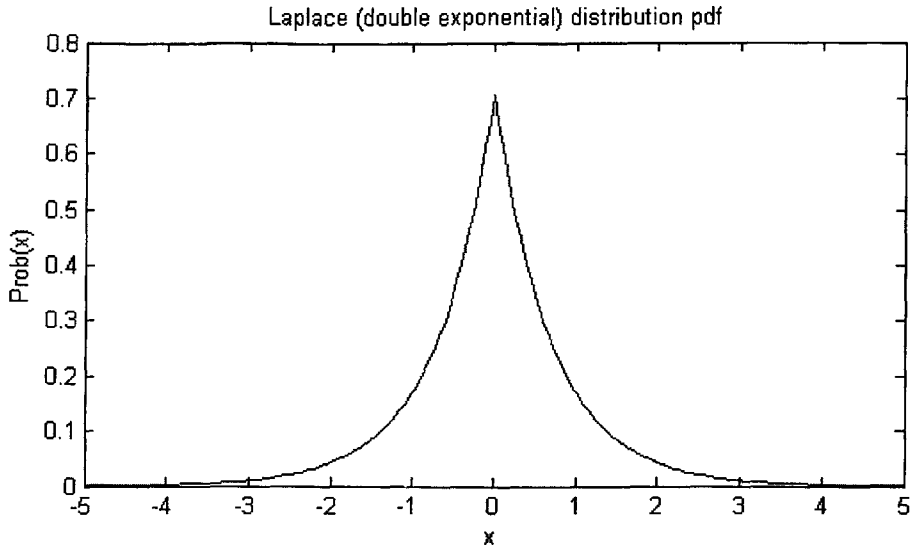

Figure 3-2: Laplace distribution

Figure 3-3 graphs the cdf of an experimental set of such coefficients with Laplace and Gaussian cdfs. Each set of coefficients was normalized to unit variance. It can be seen that the Laplace model is fairly accurate, and it is more accurate than a Gaussian model.

Now the effects of quantization are examined. H.264 uses a scalar quantizer so each quantization interval has the same length, known as the quantization step size Q. Quantization takes all the coefficients within a given step size and represents them with a single value. The quantization operation $Q(\cdot)$ may be represented as a function with the graph in Figure 3-4. Every value between $-Q/2$ and $+Q/2$ is quantized to 0 and so forth. H.264 does not specify Q directly but rather uses a quantization parameter QP, whose relationship to Q is expressed in Equation 1.1.

From [17] and Figure 3-4, we can determine the probability that a quantized coefficient appears in the output. It is the same as the probability that an unquantized coefficient falls in the range of a given quantization interval. Consider an infinite range of discrete intervals iQ with step size Q. The probability of a value being quantized to iQ is shown in Equation 3.8. Its derivation may be found in Appendix A.3.

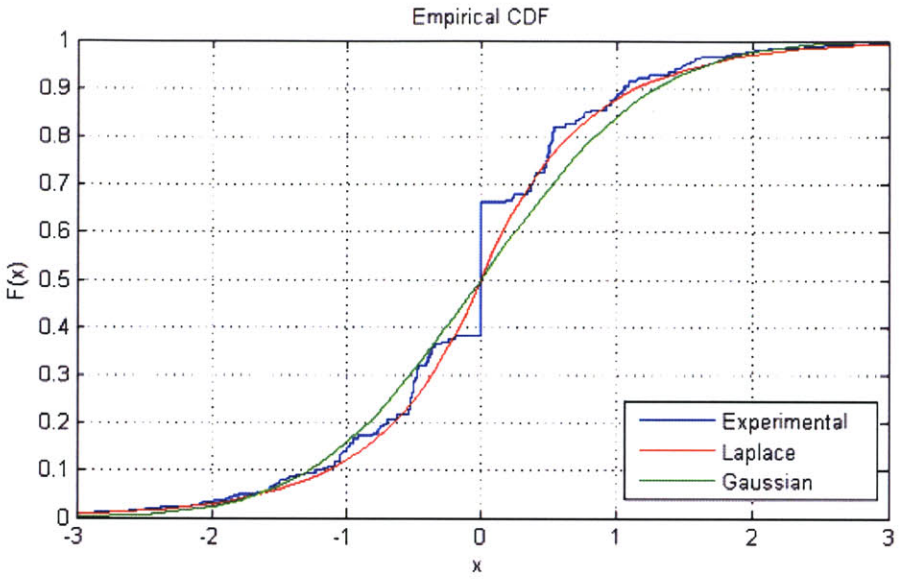

Figure 3-3: Experimental cdf of DCT coefficients, compared to Laplace and Gaussian distributions

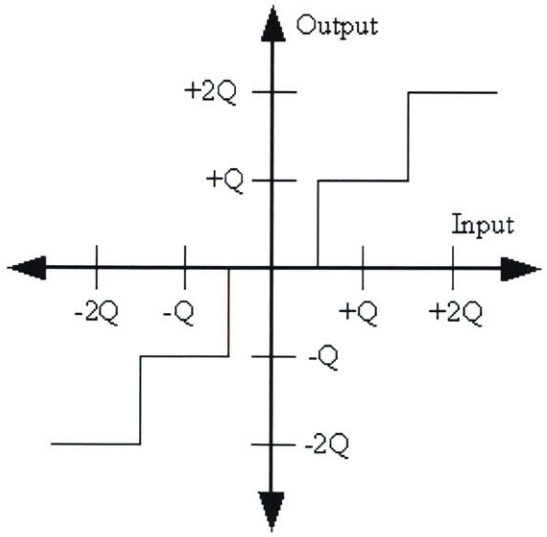

Figure 3-4: Effect of quantization

$$
\begin{aligned}
p(iQ) &= \int_{(i-\frac{1}{2})Q}^{(i+\frac{1}{2})Q} p(x)dx \\
&= \begin{cases} 1 - e^{-r}, & i = 0 \\ e^{-2r|i|} \sinh r, & i \neq 0 \end{cases}
\end{aligned} \tag{3.8}
$$

where $r = Q/(2b)$. Notice that r is the ratio of the quantization step size to the standard deviation of the residual values, multiplied by a constant.

Finally, run-length encoding produces the final bitstream output. A lossless operation, it aims to make the bit rate as close to the entropy of the quantized DCT coefficients as possible. The entropy of a probabilistic distribution is defined as the average amount of information it contains. For example, a degenerate distribution that is a constant value with probability 1 contains no information. A distribution that has a 99% chance of being a certain value contains only a little more information. However, a random variable with a 50-50 chance of being different values contains a great deal of information. Information theory states that no statistical distribution may be losslessly compressed into fewer bits than its entropy. Consequently, we can approximate the bit rate per coefficient using the entropy of the quantized Laplace distribution. The result is Equation 3.9, whose derivation is shown in Appendix A.4. A similar expression may be found in [19].

$$
\begin{aligned}
H &= -\sum_{i=-\infty}^{+\infty} p(iQ) \log_2 p(iQ) \\
&= \frac{1}{\ln 2}\left(-(1 - e^{-r}) \ln(1 - e^{-r}) + \frac{r}{\sinh r} - e^{-r} \ln(\sinh r)\right)
\end{aligned} \tag{3.9}
$$

The actual macroblock bit rate for the luma coefficients may be estimated by multiplying the entropy by 256, the number of DCT coefficients in a macroblock. This results in the final rate expression of Equation 3.10.

$$
R = \frac{256}{\ln 2}\left(-(1 - e^{-r}) \ln(1 - e^{-r}) + \frac{r}{\sinh r} - e^{-r} \ln(\sinh r)\right) \tag{3.10}
$$

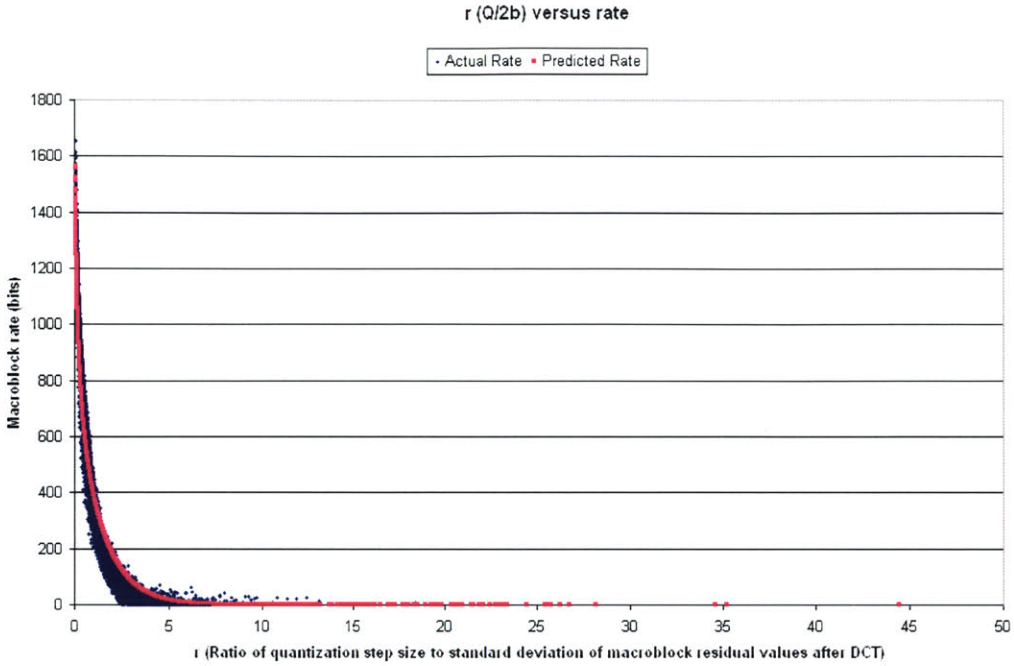

Figure 3-5: Actual and predicted macroblock bit rates

where

$$r = \frac{Q}{2b} = 256 \cdot \frac{2^{(QP-4)/6}}{\text{SAD}\sqrt{\pi}}$$

Figure 3-5 graphs the experimental and predicted rate as a function of r. From this graph, Equation 3.10 does indeed appear to model the rate well given QP and the SAD. Figures 3-6 and 3-7 graph the predicted value of R for different values of QP and SAD. Notice that R appears to be more sensitive to QP than to SAD, which is especially reflected in the change of scale of the y-axes in Figure 3-7.

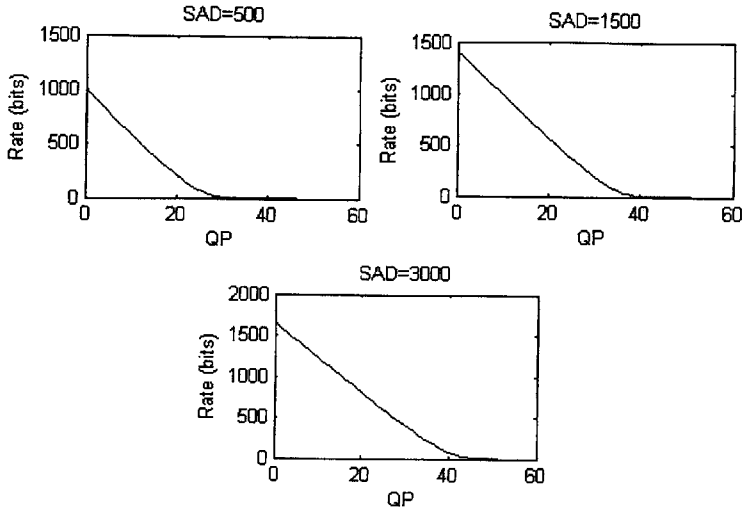

Figure 3-6: Predicted rate as a function of QP for different SADs

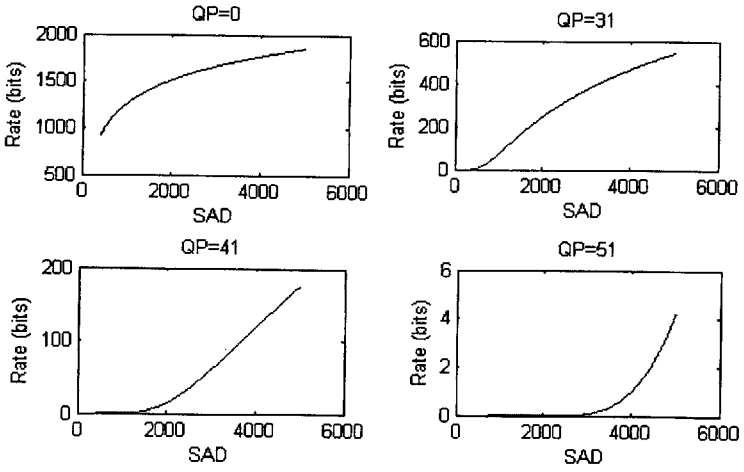

Figure 3-7: Predicted rate as a function of SAD for different QPs

3.2 Distortion

Using the framework developed for rate, a similar mathematical model for distortion can be derived from the SAD and QP. Distortion arises as a result of quantization, which discards some precision to reduce the bit rate. For a given quantization interval iQ, all values in the range from $\left(i - \frac{1}{2}\right) Q$ to $\left(i + \frac{1}{2}\right) Q$ are reduced to iQ. Any value x in that range produces a squared error of $(x - iQ)^2$. Recall the statistical distribution of the unquantized DCT residual coefficients from Equation 3.6.

$$p(x) = \frac{1}{2b} e^{-|x|/b}$$

where

$$b = \frac{\sigma}{\sqrt{2}} = \frac{\text{SAD}\sqrt{\pi}}{512}$$

Distortion, as measured by squared error, can be expressed as Equation 3.11.

$$D = 256 \sum_{i=-\infty}^{+\infty} \int_{(i-\frac{1}{2})Q}^{(i+\frac{1}{2})Q} (x - iQ)^2 p(x) dx \tag{3.11}$$

As shown in Appendix A.5, we find that

$$D = 256 \cdot 2b^2 \left(1 - \frac{r}{\sinh r}\right) \tag{3.12}$$

This formula makes intuitive sense. When $Q = 0$, distortion is zero. When Q goes to ∞, all coefficients are quantized to zero. As a result, distortion becomes $2b^2$, which is actually the variance (σ^2) of the coefficients. Figures 3-8 and 3-9 graph D for different values of QP and SAD.

Unfortunately, Figure 3-10 shows that the SAD is only marginally correlated with distortion. Figure 3-11, meanwhile, shows that the distortion model is more accurate as a function of QP. Section 5.3 speculates on possible improvements to the distortion model.

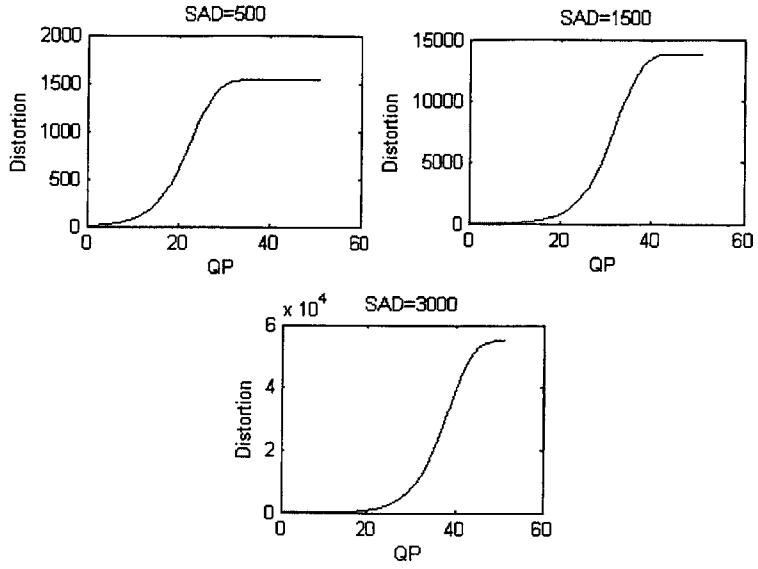

Figure 3-8: Predicted distortion as a function of QP for different SADs

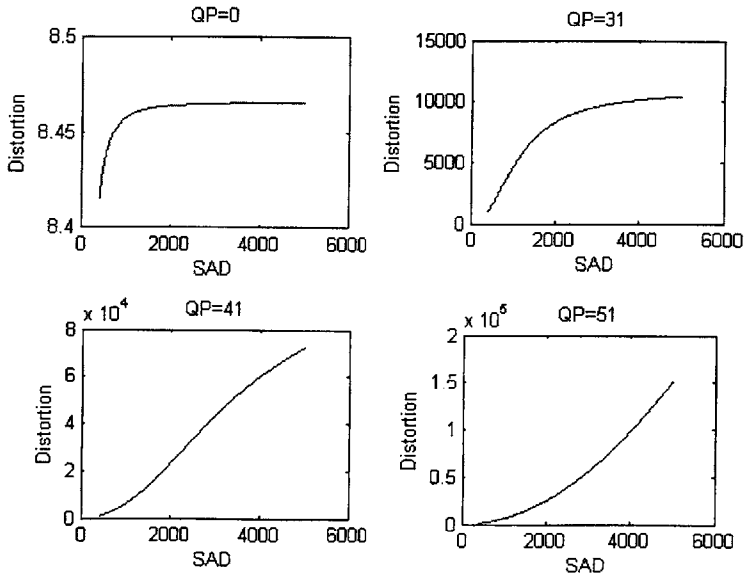

Figure 3-9: Predicted distortion as a function of SAD for different QPs

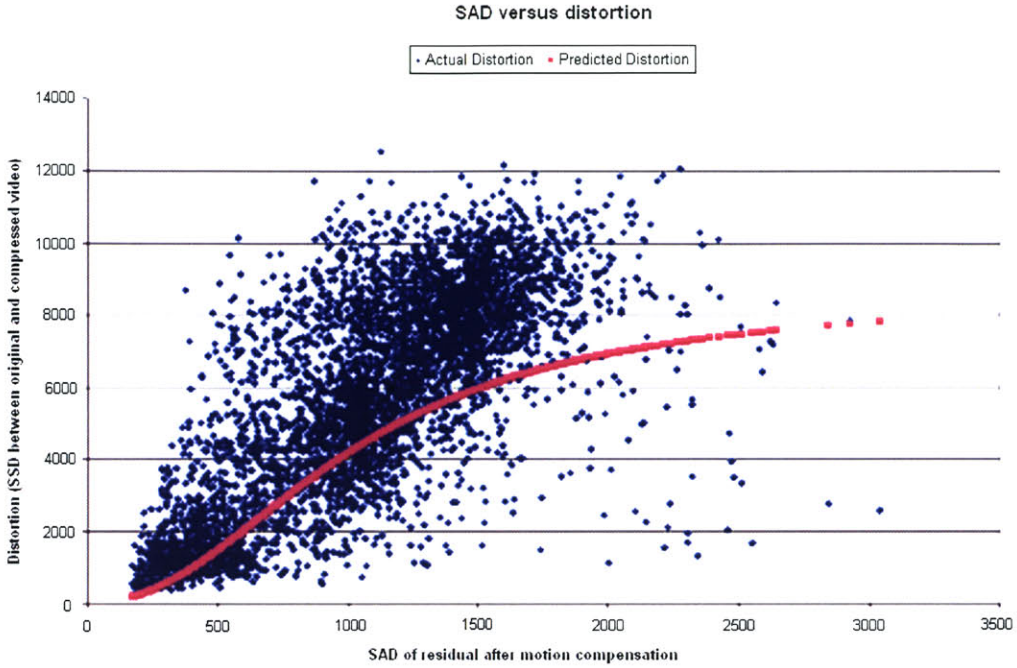

Figure 3-10: SAD versus distortion, actual and predicted

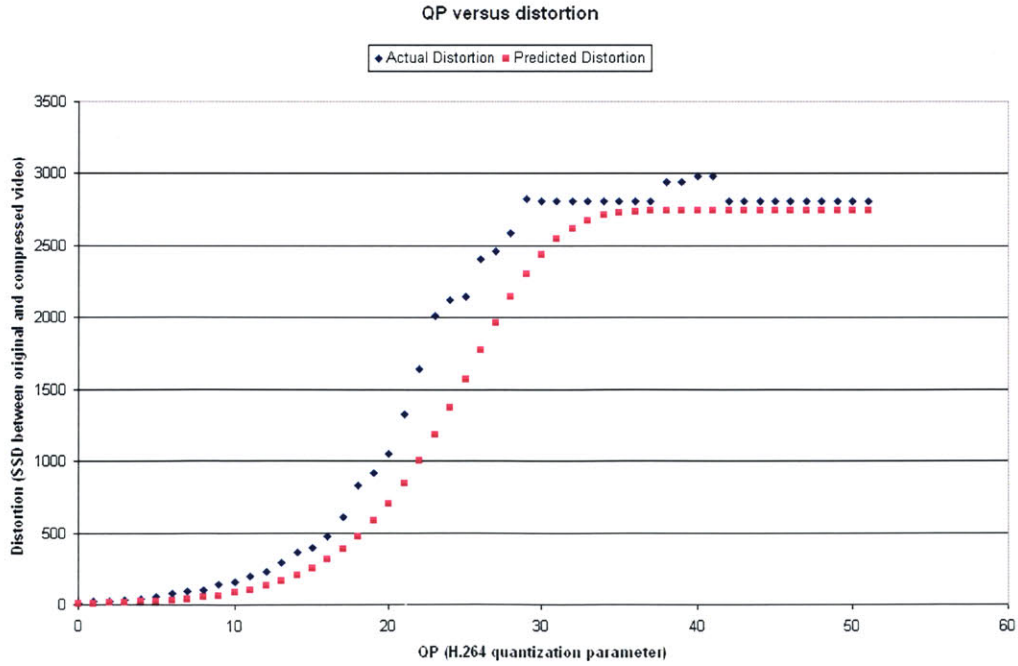

Figure 3-11: QP versus distortion, actual and predicted

3.3 Lambda

Equation 1.3 expresses λ_{MODE} as a function of QP, but it is only an experimental approximation. A theoretical justification is available in [11, 14]. However, the justification only applies to H.263, which uses a different quantization parameter than H.264. Fortunately, it is simple to repeat the analysis here for H.264.

The following analysis is based on [15]. First, we know that

$$\lambda_{MODE} = -\frac{dD}{dR}$$

At high bit rates, we can approximate $R(D)$ as

$$R(D) = a \log_2 \left(\frac{b}{D} \right) \tag{3.13}$$

where a and b are functional parameters. At high rates, distortion can be modeled as a uniform distribution within each quantization interval, meaning

$$D = \frac{Q^2}{12}$$

Substituting Equation 1.1,

$$D = \frac{2^{(QP-4)/3}}{12}$$

The total differentials of rate and distortion are

$$dR = \frac{\partial R}{\partial QP} dQP = -\frac{a}{3} \tag{3.14}$$

$$dD = \frac{\partial D}{\partial QP} dQP = \frac{dD}{dQP} = \frac{\ln 2}{3} 2^{(QP-10)/3} dQP \tag{3.15}$$

As a result, we can see that

$$\lambda_{MODE} = -\frac{dD}{dR} = c \cdot 2^{(QP-12)/3}$$

where c is experimentally determined to be 0.85. The change of 10 to 12 in the exponent probably reflects the fact that $\ln 2 \approx 2^{-2/3}$.

Figure 3-12 shows an empirical graph of frame rate versus distortion for every value of QP. Figure 3-13 graphs the negative slope of Figure 3-12 along with λ as described in Equation 1.3. For small QPs, the two curves are similar, but they begin to diverge later. Section 5.4 discusses some possible reasons for the discrepency.

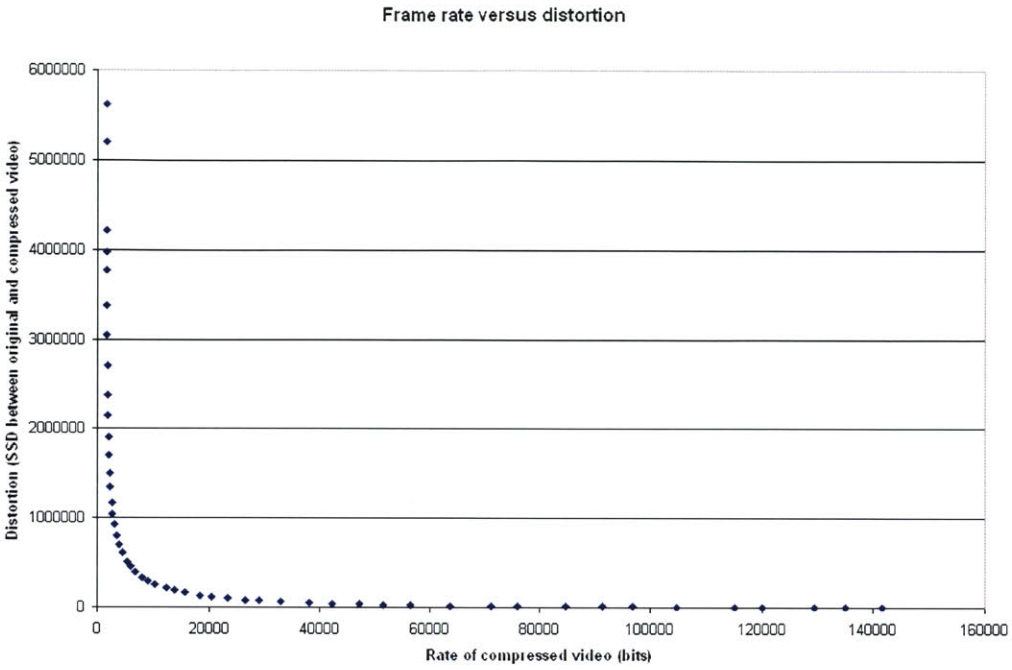

Figure 3-12: Empirical rate versus distortion graph, taken from *Foreman*

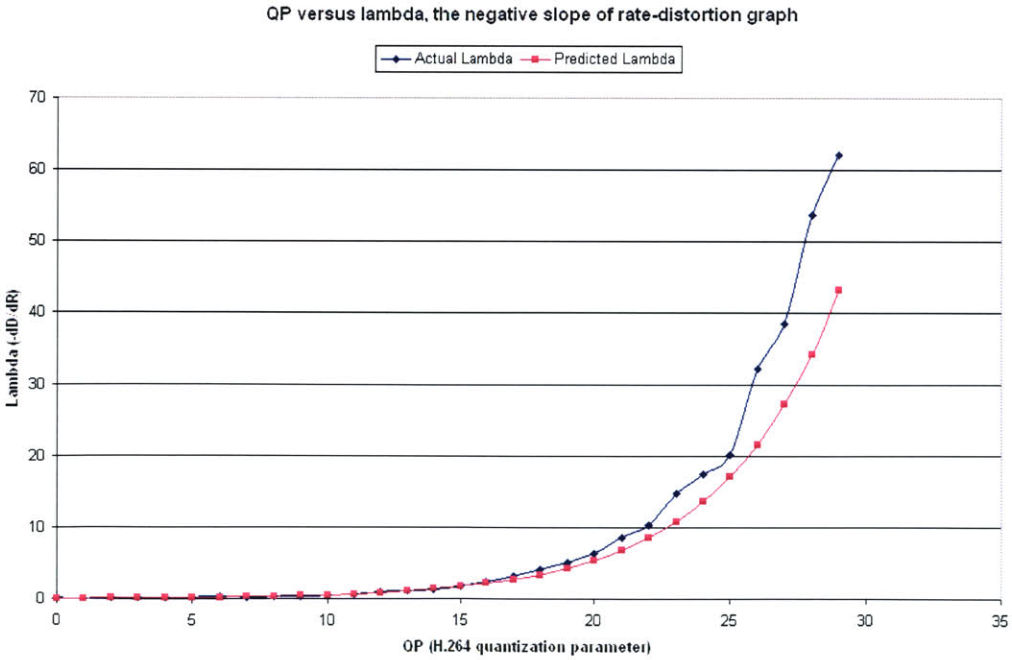

Figure 3-13: QP versus λ, actual and predicted

Chapter 4

Cost model and implementation

Now all three parameters of the Lagrangian cost function have been determined in terms of SAD and QP.

$$
\begin{aligned}
C &= D + \lambda R \\
R &= \frac{256}{\ln 2}\left(-(1-e^{-r})\ln(1-e^{-r}) + \frac{r}{\sinh r} - e^{-r}\ln(\sinh r)\right) \\
&\quad + \text{MV cost} \\
D &= 256 \cdot 2b^2\left(1 - \frac{r}{\sinh r}\right) \\
\lambda &= 0.85 \cdot 2^{(QP-12)/3} \\
b &= \frac{\sigma}{\sqrt{2}} = \frac{\text{SAD}\sqrt{\pi}}{512} \\
Q &= 2^{(QP-4)/6} \\
r &= \frac{Q}{2b} = 256 \cdot \frac{2^{(QP-4)/6}}{\text{SAD}\sqrt{\pi}}
\end{aligned}
$$

We may use these equations to determine the proper threshold when deciding between INTER-16 $\times$ 16 and INTER-8 $\times$ 8 modes. Because of the complexity of the cost model, we will gain intuition into the problem by starting with a simple hypothetical situation.

Let us pretend the cost functions were actually linear with respect to an independent variable x. However, they have different additive constants, as the actual cost

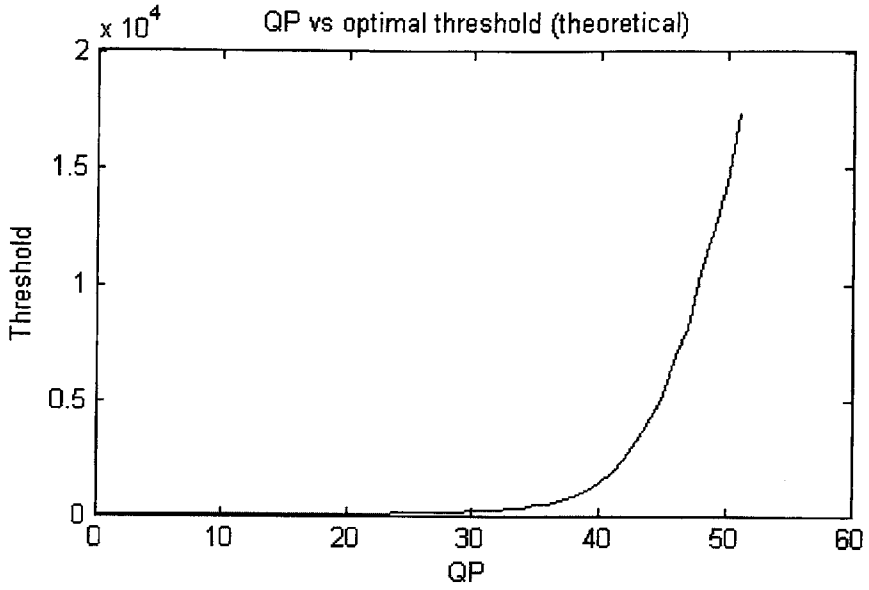

Figure 4-1: QP versus theoretical threshold as derived from cost model

functions do because of their different motion vector rate requirements.

$$\begin{aligned} C_1(x) &= ax + b_1 \\ C_2(x) &= ax + b_2 \end{aligned}$$

Now we are given two different inputs to each function: x_1 for C_1 and x_2 for C_2. We want to find the threshold t such that $x_2 - x_1 + t$ has the same sign as $C_2(x_2) - C_1(x_1)$. Notice that $C_2(x_2) - C_1(x_1) = a(x_2 - x_1) + (b_2 - b_1)$. Assuming a is positive, $t = (b_2 - b_1)/a$ suffices.

Luckily, we can expand the actual cost function using its Taylor series to achieve a similar effect. The input variable is the SAD, and the Taylor series is centered around $\mathrm{SAD} = \mathrm{SAD}_0$.

$$\begin{aligned} C_{t1}(\mathrm{SAD}) &= \left.\frac{dC_1}{d\mathrm{SAD}}\right|_{\mathrm{SAD}=\mathrm{SAD}_0} (\mathrm{SAD} - \mathrm{SAD}_0) + C_1|_{\mathrm{SAD}=\mathrm{SAD}_0} \\ C_{t2}(\mathrm{SAD}) &= \left.\frac{dC_2}{d\mathrm{SAD}}\right|_{\mathrm{SAD}=\mathrm{SAD}_0} (\mathrm{SAD} - \mathrm{SAD}_0) + C_2|_{\mathrm{SAD}=\mathrm{SAD}_0} \end{aligned}$$

The analytical expression for a Taylor series expansion is too complicated to show. Numerically, though, it can be evaluated. Using experimentally determined SAD_0s and motion vector bit rates, Figure 4-1 graphs the theoretically optimal threshold as a function of QP.

Using these thresholds, we can graph the rate-distortion performance of the encoder. Figures 4-2 and 4-3 compare the performance of a fixed threshold at 200 (a typical value) versus a varying threshold for the video sequence *Foreman* at high and low bit rates. Each point represents a different QP. The figures also show the performance of the H.264 reference software, which measures every possible mode and requires many more computational resources. At high rates, a varying threshold has little impact on rate-distortion performance. At low rates, the improvement is significant: a 20 percent bit rate savings without greater distortion. A varying threshold also eliminates the perverse effect where rate and distortion simultaneously increase at very low bit rates.

Implementation is trivial, as it merely involves a table lookup for each QP. Depending on how often QP changes, the threshold may need to be updated at the sequence, frame, or macroblock level.

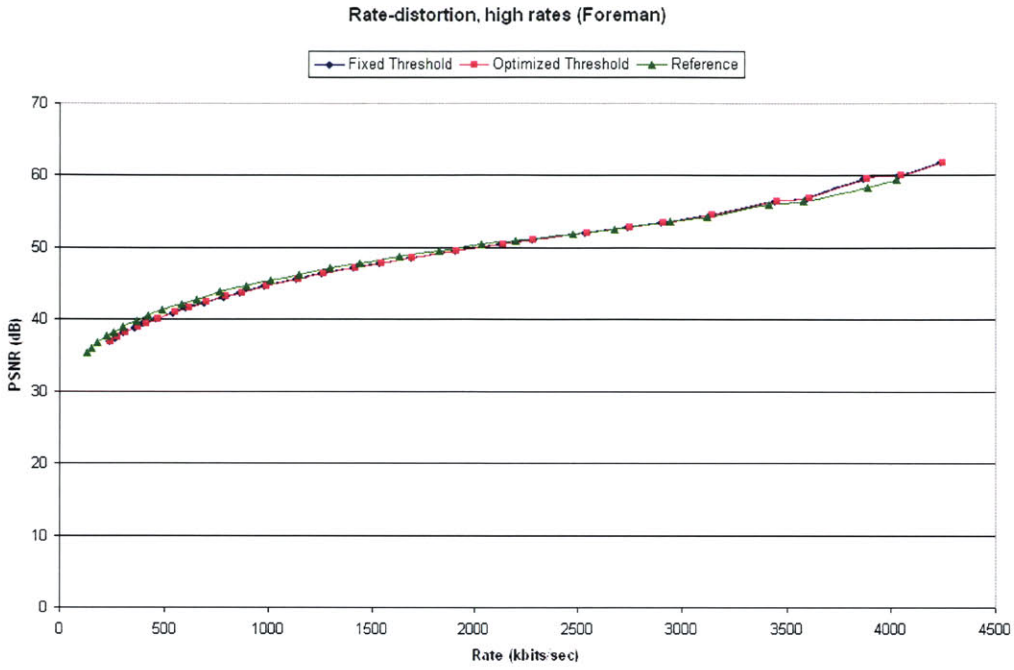

Figure 4-2: Comparison of rate-distortion curves (high bit rates)

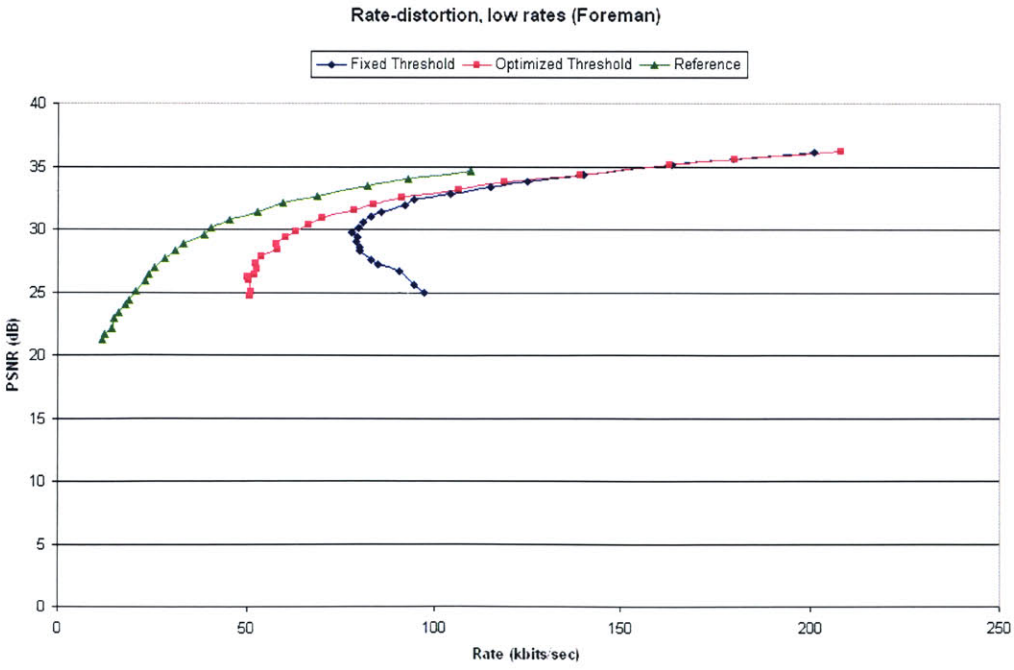

Figure 4-3: Comparison of rate-distortion curves (low bit rates)

Chapter 5

Further work and summary

Figures 4-2 and 4-3 show that the encoder used for this thesis still falls short when compared to the reference encoder. Much of the underperformance is unavoidable because of resource limitations. For example, the motion vector search space is limited compared to the reference encoder. Nevertheless, this chapter outlines some possible further modifications that could improve performance.

5.1 INTRA modes

INTRA modes require some more analysis when modeling their rate and distortion characteristics because they rely on spatial prediction. There are two major intra macroblock modes: INTRA-16 $\times$ 16 and INTRA-4 $\times$ 4, which differ in the size of their predicted blocks. Similar to the INTER modes, a smaller block size usually produces a smaller SAD, but it also requires more bits to code the extra prediction information. INTRA-16 $\times$ 16 mode needs very few extra bits. INTRA-4 $\times$ 4 mode, on the other hand, needs to specify a spatial prediction direction for all of the 16 4 $\times$ 4 blocks in a macroblock.

Unfortunately, the coding of prediction directions in INTRA-4 $\times$ 4 mode is not entirely straightforward. There are a total of 9 possible directions, such as vertical, horizontal, diagonal, and so forth. These directions are not coded directly but rather derived from the directions of neighboring blocks because they are often correlated

[10]. In many cases, a block needs almost no extra bits to signal its prediction direction.

The H.264 reference software [14] employs an algorithm to calculate the final number of bits needed for the prediction directions in each INTER-4×4 macroblock, and the algorithm is used in this thesis's encoder as well. A possible area of improvement is to use the rate and distortion functions of Equations 3.10 and 3.12 to determine the final cost from the SAD.

5.2 Chroma coefficients

This thesis examined only the effects of luma coefficients, not chroma. Chroma coefficients comprise a much smaller component of the encoded bit rate, and chroma distortion is less visible to the human eye than luma distortion. Nevertheless, it may still be useful to include chroma coefficients in a complete rate-distortion model. The encoder used in this thesis only calculates the SAD of luma coefficients, so no information about chroma coefficients is available during mode decision. However, other encoders might use such information to their advantage.

5.3 Improving rate and distortion models

There is evidence that DCT coefficients are better modeled by a Cauchy distribution than a Laplace distribution [18]. A Cauchy distribution has the form

$$p(x) = \frac{1}{\pi} \frac{\mu}{\mu^2 + x^2}$$

where μ is an additional parameter equal to twice the full width at half maximum. While possibly more accurate, a Cauchy distribution is mathematically much more complicated. In particular, there is no analytical symbolic method to derive μ from the SAD. However, a numerical Cauchy model may help somewhat.

A larger question is the lack of experimental correlation between SAD and distor-

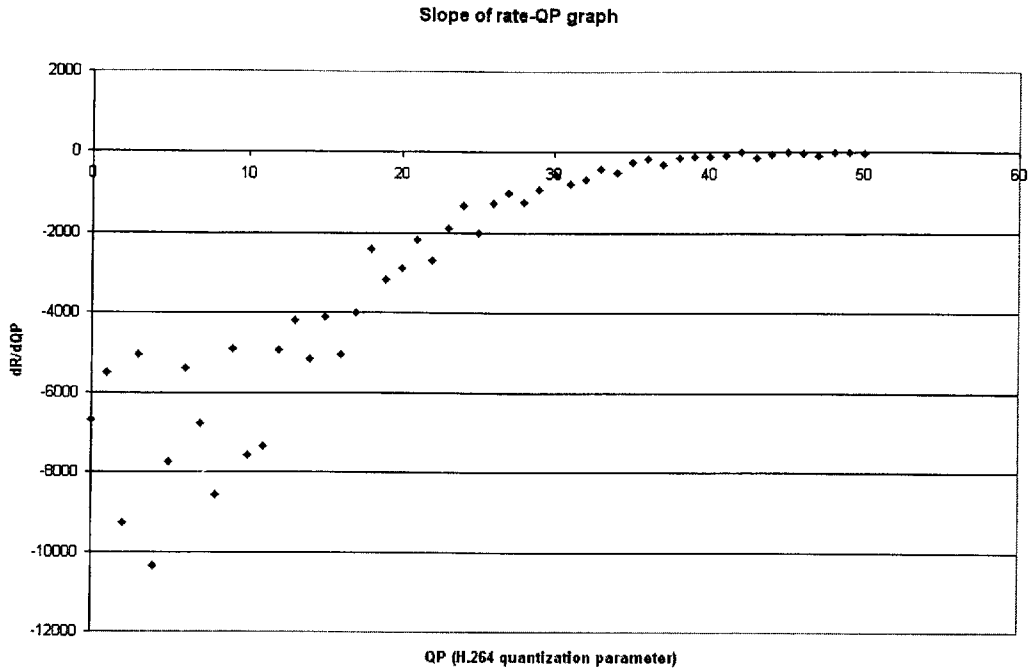

Figure 5-1: Experimental graph of $\frac{dR}{dQP}$

tion, as shown in Section 3.2. A more refined DCT coefficient model, e.g. using the Cauchy distribution, will not help because factors besides SAD and QP is affecting distortion. These other factors and their ability to be quantified remain unknown.

5.4 Improving λ

Section 3.3 demonstrated why Equation 1.3 is reasonable. However, experimental evidence does not seem to support some of assumptions made in [15]. In particular, Equation 3.14 is highly suspect. Figure 5-1 shows an experimental graph of QP versus $\frac{dR}{dQP}$. Clearly it is not constant as claimed in Equation 3.14, particularly at low bit rates.

Because of Equations 3.10 and 3.12, it seems possible to calculate a more accurate theoretical expression for $\lambda = -\frac{dD}{dR}$. We can find the total differentials for R and D:

$$dR = \frac{\partial R}{\partial Q}dQ + \frac{\partial R}{\partial b}db$$
$$dD = \frac{\partial D}{\partial Q}dQ + \frac{\partial D}{\partial b}db$$

Unfortunately, it is difficult to proceed from here. First, R and D contain both variables Q and b, so their total differentials contain dQ and db. There is no way to express both R and D as the function of a single variable. As a result, dD/dR requires some form of mathematical approximation. One possibility is to assume an *a priori* value of b, which is equivalent to setting $db = 0$. It is not clear whether this is a valid approximation. Figures 3-6 and 3-7 show that the rate is significantly more sensitive to changes in QP than SAD, at least in the range of interest. Distortion, on the other hand, is highly dependent on SAD, as illustrated in Figures 3-8 and 3-9. As a result, there is reason to suspect the inadequacy of such an approach. A quick simulation was performed using a fixed b. The results were unsatisfactory because λ was too small at high QPs. Further investigation is necessary.

5.5 Summary

This thesis made the following contributions:

- Provided a primer to video compression techniques and codecs.
- Created a model to estimate rate and distortion characteristics from limited information for mode decision in H.264.
- Implemented rate-distortion optimization and analyzed the results.
- Outlined further areas of improvement for resource-limited H.264 encoders.

Appendix A

Appendix: Mathematical derivations

This appendix shows the detailed mathematical derivations of the equations used in this thesis.

A.1 Expected absolute value of a Gaussian random variable

This section shows the derivation of Equation 3.3. The goal is to calculate

$$E[|X|] = \int_{-\infty}^{+\infty} |x| \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)} dx$$

Because the integrand is an even function of x, we can split it in half and eliminate the absolute value.

$$E[|X|] = \frac{2}{\sigma\sqrt{2\pi}} \int_{0}^{+\infty} x e^{-x^2/(2\sigma^2)} dx$$

Perform the change of variable $y = x^2/(2\sigma^2)$. Then $dy = x/\sigma^2 dx$, or $xdx = \sigma^2 dy$.

$$E[|X|] = \frac{2\sigma^2}{\sigma\sqrt{2\pi}} \int_{0}^{+\infty} e^{-y} dy$$

The integral is equal to 1.

$$E[|X|] = \sigma\sqrt{\frac{2}{\pi}}$$

A.2 Variance of a Laplacian distribution

This section shows how to calculate the variance of a Laplacian distribution. By definition,

$$\sigma^2 = \int_{-\infty}^{+\infty} x^2 p(x) dx$$

From Equation 3.6,

$$p(x) = \frac{1}{2b} e^{-|x|/b}$$

As a result,

$$\sigma^2 = \int_{-\infty}^{+\infty} x^2 \frac{1}{2b} e^{-|x|/b} dx$$

We may eliminate the absolute value operator because the integrand is an even function of x.

$$\sigma^2 = \frac{1}{b} \int_{0}^{+\infty} x^2 e^{-x/b} dx$$

Apply integration by parts, with $u = x^2$, $du = 2xdx$, $v = -be^{-x/b}$, and $dv = e^{-x/b}dx$.

$$\sigma^2 = \frac{1}{b} \left(-bx^2 e^{-x/b}\right)\Big|_{x=0}^{x=+\infty} - \frac{1}{b} \int_{0}^{+\infty} -2xbe^{-x/b} dx$$

Simplify.

$$\sigma^2 = 2\int_0^{+\infty} xe^{-x/b}dx$$

Apply integration by parts again, with $u = x$, $du = dx$, $v = -be^{-x/b}$, and $dv = e^{-x/b}dx$.

$$\sigma^2 = 2\left(-bxe^{-x/b}\right)\Big|_{x=0}^{x=+\infty} - 2\int_0^{+\infty} -be^{-x/b}dx$$

Simplify.

$$\sigma^2 = 2b\int_0^{+\infty} e^{-x/b}dx$$

Perform the final integral.

$$\sigma^2 = 2b\left(-be^{-x/b}\right)\Big|_{x=0}^{x=+\infty}$$

Simplify.

$$\sigma^2 = 2b^2$$

A.3 Probabilistic distribution of quantized Laplacian distribution

This section shows the derivation of Equation 3.8. The objective is to determine

$$p(iQ) = \int_{(i-\frac{1}{2})Q}^{(i+\frac{1}{2})Q} p(x)dx$$

where, from Equation 3.6,

$$p(x) = \frac{1}{2b}e^{-|x|/b}$$

Immediately, we see that

$$p(iQ) = \frac{1}{2b} \int_{(i-\frac{1}{2})Q}^{(i+\frac{1}{2})Q} e^{-|x|/b} dx$$

Two cases are necessary: $i = 0$ and $i \neq 0$. First we consider $i = 0$.

$$p(0Q) = \frac{1}{2b} \int_{-Q/2}^{+Q/2} e^{-|x|/b} dx$$

The integrand is even, so we can eliminate the absolute value operator.

$$p(0Q) = \frac{1}{b} \int_{0}^{+Q/2} e^{-x/b} dx$$

The integral simply becomes

$$p(0Q) = \frac{1}{b} \left(-be^{-x/b}\right)\Big|_{x=0}^{x=+Q/2}$$

Simplify.

$$p(0Q) = 1 - e^{-Q/(2b)}$$

Let $r = Q/(2b)$.

$$p(0Q) = 1 - e^{-r}$$

Now we consider the case $i \neq 0$. First, we see that because $p(x)$ is even, it does not matter whether i is negative or positive. Therefore we may apply the absolute value operator to i and remove it from x.

$$p(iQ) = \frac{1}{2b} \int_{(|i|-\frac{1}{2})Q}^{(|i|+\frac{1}{2})Q} e^{-x/b} dx$$

Perform the integral.

$$p(iQ) = \frac{1}{2b}\left(-be^{-x/b}\right)\Big|_{x=\left(|i|-\frac{1}{2}\right)Q}^{x=\left(|i|+\frac{1}{2}\right)Q}$$

Expand.

$$p(iQ) = \frac{1}{2}\left(-e^{-\left(|i|+\frac{1}{2}\right)Q/b} + e^{-\left(|i|-\frac{1}{2}\right)Q/b}\right)$$

Substitute $r = Q/(2b)$.

$$p(iQ) = \frac{1}{2}\left(e^{-\left(|i|-\frac{1}{2}\right)2r} - e^{-\left(|i|+\frac{1}{2}\right)2r}\right)$$

Expand the exponents.

$$p(iQ) = \frac{1}{2}\left(e^{-2r|i|+r} - e^{-2r|i|-r}\right)$$

Collect common terms.

$$p(iQ) = e^{-2r|i|} \cdot \frac{1}{2}\left(e^{r} - e^{-r}\right)$$

Substitute $\sinh x = \frac{1}{2}(e^x - e^{-x})$.

$$p(iQ) = e^{-2r|i|}\sinh r$$

In summary,

$$p(iQ) = \begin{cases} 1 - e^{-r}, & i = 0 \\ e^{-2r|i|}\sinh r, & i \neq 0 \end{cases}$$

A.4 Entropy of quantized Laplacian distribution

This section shows the derivation of Equation 3.9. The entropy is defined as

$$H = -\sum_{i=-\infty}^{+\infty} p(iQ)\log_2 p(iQ)$$

where, from Equation 3.8,

$$p(iQ) = \begin{cases} 1 - e^{-r}, & i = 0 \\ e^{-2r|i|}\sinh r, & i \neq 0 \end{cases}$$

First, we split the infinite summation of the entropy expression into negative, zero, and positive is.

$$H = -\left(\sum_{i=-\infty}^{-1} p(iQ)\log_2 p(iQ) + p(0Q)\log_2 p(0Q) + \sum_{i=1}^{+\infty} p(iQ)\log_2 p(iQ)\right)$$

It can be seen that $p(iQ)$ is even about i. Therefore, the negative and positive summations are equivalent.

$$H = -\left(p(0Q)\log_2 p(0Q) + 2\sum_{i=1}^{+\infty} p(iQ)\log_2 p(iQ)\right)$$

We now expand $p(iQ)$.

$$H = -(1 - e^{-r})\log_2(1 - e^{-r}) - 2\sum_{i=1}^{+\infty} e^{-2ri}\sinh(r)\log_2(e^{-2ri}\sinh(r))$$

Expand the logarithm and factor out $1/\ln 2$.

$$H = -(1 - e^{-r})\log_2(1 - e^{-r}) - \frac{2\sinh r}{\ln 2}\sum_{i=1}^{+\infty} e^{-2ri}(\ln(e^{-2ri}) + \ln(\sinh r))$$

Simplify the right hand term.

$$H = -(1 - e^{-r})\log_2(1 - e^{-r}) - \frac{2\sinh r}{\ln 2}\sum_{i=1}^{+\infty}(-2rie^{-2ri} + e^{-2ri}\ln(\sinh r))$$

Split the summation.

$$H = -(1-e^{-r})\log_2(1-e^{-r}) + \frac{4r\sinh r}{\ln 2}\sum_{i=1}^{+\infty} ie^{-2ri} - \frac{2\sinh r\ln(\sinh r)}{\ln 2}\sum_{i=1}^{+\infty} e^{-2ri}$$

The rightmost summation is a simple power series.

$$H = -(1-e^{-r})\log_2(1-e^{-r}) + \frac{4r\sinh r}{\ln 2}\sum_{i=1}^{+\infty} ie^{-2ri} - \frac{2\sinh r\ln(\sinh r)}{\ln 2}\frac{e^{-2r}}{1-e^{-2r}}$$

Further simplification is possible.

$$H = -(1-e^{-r})\log_2(1-e^{-r}) + \frac{4r\sinh r}{\ln 2}\sum_{i=1}^{+\infty} ie^{-2ri} - \frac{e^{-r}\ln(\sinh r)}{\ln 2}$$

The middle term presents some difficulty. Let us examine the more general infinite series

$$\sum_{i=1}^{+\infty} ia^i$$

Notice that

$$\sum_{i=1}^{+\infty} ia^i - (i-1)a^i = \sum_{i=1}^{+\infty} a^i$$

The left hand terms can be split, and the right hand term is a simple power series.

$$\sum_{i=1}^{+\infty} ia^i - \sum_{i=1}^{+\infty}(i-1)a^i = \frac{a}{1-a}$$

The index of the right summation may be changed slightly.

$$\sum_{i=1}^{+\infty} ia^i - \sum_{i=0}^{+\infty} ia^{i+1} = \frac{a}{1-a}$$

Pull out a from the right summation, and change the index again.

$$\sum_{i=1}^{+\infty} ia^i - a\sum_{i=1}^{+\infty} ia^i = \frac{a}{1-a}$$

Collect common terms.

$$(1-a)\sum_{i=1}^{+\infty} ia^i = \frac{a}{1-a}$$

Divide to isolate the summation.

$$\sum_{i=1}^{+\infty} ia^i = \frac{a}{(1-a)^2}$$

We substitute this result into the entropy expression, where $a = e^{-2r}$.

$$H = -(1-e^{-r})\log_2(1-e^{-r}) + \frac{4r\sinh r}{\ln 2}\frac{e^{-2r}}{(1-e^{-2r})^2} - \frac{e^{-r}\ln(\sinh r)}{\ln 2}$$

Further simplification yields the final entropy expression.

$$H = \frac{1}{\ln 2}\left(-(1-e^{-r})\ln(1-e^{-r}) + \frac{r}{\sinh r} - e^{-r}\ln(\sinh r)\right)$$

A.5 Distortion

This section shows the derivation of Equation 3.12. We start with Equation 3.11 (ignoring the multiplicative factor of 256).

$$D = \sum_{i=-\infty}^{+\infty} \int_{(i-\frac{1}{2})Q}^{(i+\frac{1}{2})Q} (x-iQ)^2 p(x)dx$$

First, we split the infinite summation into negative, zero, and positive is.

$$D = \sum_{i=-\infty}^{-1} \left(\int_{-Q/2+iQ}^{+Q/2+iQ} (x-iQ)^2 p(x)dx\right) + \int_{-Q/2}^{+Q/2} x^2 p(x)dx$$

$$+\sum_{i=1}^{+\infty}\left(\int_{-Q/2+iQ}^{+Q/2+iQ}(x-iQ)^2p(x)dx\right)$$

Because $p(x)$ is an even function, the negative and positive summations are equivalent. Furthermore, it is worth noticing that the integral in middle term ($i=0$) can be split into two equal halves. As a result, we may rewrite the expression in the following manner.

$$D=2\left(\int_0^{+Q/2}x^2p(x)dx+\sum_{i=1}^{+\infty}\left(\int_{-Q/2+iQ}^{+Q/2+iQ}(x-iQ)^2p(x)dx\right)\right)$$

Now we expand $p(x)$ into its full form. However, in the previous expression, x is always positive in the range of interest. As a result, we can discard the absolute value operator in $p(x)$.

$$D=\frac{1}{b}\left(\int_0^{+Q/2}x^2e^{-x/b}dx+\sum_{i=1}^{+\infty}\left(\int_{-Q/2+iQ}^{+Q/2+iQ}(x-iQ)^2e^{-x/b}dx\right)\right)$$

On the right term, we perform a change of variable $y=x-iQ$.

$$D=\frac{1}{b}\left(\int_0^{+Q/2}x^2e^{-x/b}dx+\sum_{i=1}^{+\infty}\left(\int_{-Q/2}^{+Q/2}y^2e^{-(y+iQ)/b}dy\right)\right)$$

We can factor and pull out a term from the integrand.

$$D=\frac{1}{b}\left(\int_0^{Q/2}x^2e^{-x/b}dx+\left(\sum_{i=1}^{+\infty}e^{-iQ/b}\right)\left(\int_{-Q/2}^{+Q/2}y^2e^{-y/b}dy\right)\right)$$

The infinite series can now be easily calculated.

$$D=\frac{1}{b}\left(\int_0^{+Q/2}x^2e^{-x/b}dx+\left(\frac{e^{-Q/b}}{1-e^{-Q/b}}\right)\left(\int_{-Q/2}^{+Q/2}y^2e^{-y/b}dy\right)\right)$$

Split the right integral in half.

$$D=\frac{1}{b}\left(\int_0^{+Q/2}x^2e^{-x/b}dx+\left(\frac{e^{-Q/b}}{1-e^{-Q/b}}\right)\left(\int_{-Q/2}^{0}y^2e^{-y/b}dy+\int_0^{+Q/2}y^2e^{-y/b}dy\right)\right)$$

Combine common terms. I also change all variables of integration back to x.

$$D = \frac{1}{b}\left(\left(\frac{1}{1-e^{-Q/b}}\right)\left(\int_0^{+Q/2} x^2 e^{-x/b}dx\right) + \left(\frac{e^{-Q/b}}{1-e^{-Q/b}}\right)\left(\int_{-Q/2}^{0} x^2 e^{-x/b}dx\right)\right)$$

Factor out more common terms.

$$D = \frac{1}{b(1-e^{-Q/b})}\left(\int_0^{+Q/2} x^2 e^{-x/b}dx + e^{-Q/b}\int_{-Q/2}^{0} x^2 e^{-x/b}dx\right)$$

Using integration by parts, we find that

$$\int x^2 e^{-x/b}dx = -be^{-x/b}(2b^2 + 2bx + x^2) + C$$

Substitute this expression for the integrals.

$$\begin{aligned} D &= \frac{1}{1-e^{-Q/b}}\left(\left(-e^{-Q/(2b)}\left(2b^2 + Qb + \frac{Q^2}{4}\right) + 2b^2\right)\right. \\ &\quad \left. + e^{-Q/b}\left(-2b^2 + e^{Q/(2b)}\left(2b^2 - Qb + \frac{Q^2}{4}\right)\right)\right) \end{aligned}$$

Multiply out the bottom term.

$$\begin{aligned} D &= \frac{1}{1-e^{-Q/b}}\left(-e^{-Q/(2b)}\left(2b^2 + Qb + \frac{Q^2}{4}\right) + 2b^2\right. \\ &\quad \left. - 2b^2 e^{-Q/b} + e^{-Q/(2b)}\left(2b^2 - Qb + \frac{Q^2}{4}\right)\right) \end{aligned}$$

Cancel out all possible terms.

$$D = \frac{1}{1-e^{-Q/b}}\left(2b^2\left(1-e^{-Q/b}\right) - 2e^{-Q/(2b)}Qb\right)$$

Multiply out.

$$D = 2b^2 - \frac{2e^{-Q/(2b)}Qb}{1 - e^{-Q/b}}$$

In the right hand term, multiply the numerator and denominator by $e^{Q/(2b)}$.

$$D = 2b^2 - \frac{2Qb}{e^{Q/(2b)} - e^{-Q/(2b)}}$$

Substitute $\sinh x = \frac{1}{2}(e^x - e^{-x})$.

$$D = 2b^2 - \frac{Qb}{\sinh \frac{Q}{2b}}$$

Factor out $2b^2$.

$$D = 2b^2 \left(1 - \frac{\frac{Q}{2b}}{\sinh \frac{Q}{2b}}\right)$$

Substitute $r = \frac{Q}{2b}$.

$$D = 2b^2 \left(1 - \frac{r}{\sinh r}\right)$$

Bibliography

[1] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding.* The Institution of Electrical Engineers, 2003.

[2] J. G. Apostolopoulos, "Video compression." Class lecture notes, MIT 6.344, 22 Aug. 2004.

[3] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 23–50, Nov. 1998.

[4] C. E. Shannon, "A mathematical theory of communication," *Bell System Technology Journal*, vol. 27, pp. 379–423, 623–656, July, Oct. 1948.

[5] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, Nov. 1998.

[6] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, pp. 337–343, May 1977.

[7] J. S. Lim, *Two-Dimensional Signal and Image Processing.* Prentice Hall PTR, 1990.

[8] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC), JVT-G050*, Mar. 2003.

[9] 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*, vol. 13, pp. 560–575, July 2003.

[10] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression.* Wiley, 2003.

[11] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 688–703, July 2003.

[12] H. Everett III, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399–417, May–June 1963.

[13] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Transactions on Image Processing*, vol. 3, pp. 533–545, Sept. 1964.

[14] "H.264/AVC reference software, JM 8.4." URL: `http://bs.hhi.de/~suehring/tml/`, 28 July 2004.

[15] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *International Conference on Image Processing*, vol. 3, pp. 542–545, 7–10 Oct. 2001.

[16] S. Ma, W. Gao, P. Gao, and Y. Lu, "Rate control for advance video coding (AVC) standard," in *International Symposium on Circuits and Systems*, vol. 2, pp. 892–895, 25–28 May 2003.

[17] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the dct coefficient distributions for images," *IEEE Transactions on Image Processing*, vol. 9, pp. 1661–1666, Oct. 2000.

[18] Y. Altunbasak and N. Kamaci, "An analysis of the dct coefficient distribution with the h.264 video coder," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 177–180, 17–21 May 2004.

[19] F. Moscheni, F. Dufaux, and H. Nicolas, "Entropy criterion for optimal bit allocation between motion and prediction error information," in *Visual Communications and Image Processing* (B. G. Haskell and H. Hang, eds.), vol. 2094, pp. 235–242, International Society for Optical Engineering, 8–11 Nov. 1993.