# Optimization of Multi-resolution Source Codes

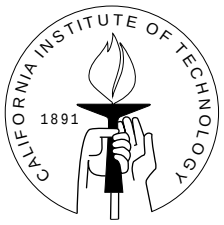
Thesis by

Diego G. Dugatkin

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy




California Institute of Technology

Pasadena, California

2004

(Defended October 27, 2003)

*This thesis is dedicated to my wife, Tania,*
*for her inconditional love,*
*continuous support,*
*and everlasting faith.*

# Abstract

What is an optimal multi-resolution source code? This thesis studies the optimization of multi-resolution source codes. A multi-resolution source code is a data compression algorithm that generates a bit-stream that can be truncated at any point to reconstruct low-resolution representations of the original data. By progressively refining the description, these codes allow the receiver to get representations of progressively increasing quality from a single file.

The optimization methods presented here are based on the minimization of a Lagrangian performance measure, which is a weighted sum of rates and distortions at the different resolutions of the multi-resolution code. The Lagrangian coefficients are the weights that parameterize the priorities assigned to the resolutions. The relative value of these parameters can be set according to the user's preferences regarding which rates are more important, the probability of decoding the file at each possible rate, or any other prioritization rationale. We present a method for converting design constraints into the corresponding Lagrangian parameters.

We also use a Lagrangian analysis to investigate optimality properties of multi-resolution codes. Specifically, we explore the characterization of the theoretically optimal output density functions of a two-resolution source code for any arbitrary set of priorities over the resolutions.

Once the priority function has been identified, the goal is to design the multi-resolution code that yields the best rate-distortion trade-off for those priorities. The minimization of the multi-resolution Lagrangian is somewhat specific to the framework and type of multi-resolution code. We pursue this goal in several coding frameworks.

The first framework is the multi-resolution vector quantizer (MRVQ) framework. Prior work on the topic described optimal MRVQ design for both fixed- and variable-rate systems but implemented only fixed-rate codes. The earliest portion of this

thesis began with the implementation of the earlier described algorithm for variable-rate MRVQ for use as a testbed for understanding the important question of how to choose the Lagrangian parameters for multi-resolution codes to meet a collection of desired constraints.

Armed with a new understanding of parameter choice in the MRVQ framework, we moved next to the more sophisticated coding framework of wavelet-based embedded bit-plane coders. New results in this framework include improvements on the Set Partitioning in Hierarchical Trees (SPIHT) and the Group Testing for Wavelets (GTW) algorithms that apply the lessons learned from MRVQ theory in these more sophisticated wavelet coding frameworks. Experimental results demonstrate the performance benefits associated with this approach.

# Contents

# List of Figures

# List of Tables

# Chapter 1 Introduction

In this thesis, we are interested in the theory and practice of multi-resolution coding. Multi-resolution source codes, also called progressive transmission or embedded codes, are data compression algorithms that generate a bit-stream that can be truncated at any point to reconstruct low-resolution representations of the original data. The higher the number of decoded bits, the better the reconstruction.

Current computer networks and wireless communication systems often present situations where each user wants to access the same file at a different connection speed or with different requirements for the minimum acceptable quality of the reconstructed data. Multi-resolution codes give a scalable solution that allows each receiver to get representations of progressively increasing quality from a single file. Thus users able and willing to wait or pay for the best (in some cases lossless) reconstruction of the original image will decode the whole bit-stream of compressed data. Users that can afford a representation of the data with more distortion can decode only a fraction of the bit-stream by stopping at the rate of interest and discarding all following bits. This feature is useful to the file-owner as well, since only one file is needed regardless of the number of different users with differing requirements.

Beginning with a theoretical investigation of multi-resolution codes helps to provide solid ground on which to base practical techniques. It also allows us to better understand the possible limitations of real multi-resolution systems. While prior results in multi-resolution coding theory include the solution to the optimal rate-distortion trade-off, evaluating that solution for specific sources is difficult.

The first major portion of this thesis is an investigation of the rate-distortion region. In particular, we seek an understanding of the reproduction alphabet size needed to achieve the optimal rate-distortion trade-off for general sources.

The second portion of this thesis treats parameter choice in practical code design. The theoretical development relies heavily on the use of a Lagrangian characterization

of code performance. This characterization is critical because it allows us to combine a large collection of rates and distortions into a single performance criterion. Critical to successful application of this approach in practice is an understanding of how to convert functional design constraints (e.g., bounds on rates and distortions) into their corresponding Lagrangian parameters. We therefore seek a new algorithm for Lagrangian parameter choice.

The third main contribution of this thesis is a pair of methods for applying Lagrangian optimization to wavelet-based multi-resolution algorithms. We present these techniques as well as experimental results.

The thesis organization follows. (Most chapters can be read separately.) Chapter 2 defines notation and describes general background. Chapter 3 presents the Lagrangian analysis to investigate reproduction alphabet size for optimal multi-resolution source codes. In particular, we explore the characterization of the theoretically optimal output density functions of a two-resolution source code for an arbitrary set of priorities over the resolutions. Chapter 4 treats the Multi-Resolution Vector Quantization (MRVQ) algorithm originally introduced in [19]. The focus of this chapter is the development of new methods for choosing Lagrangian parameters. This chapter also describes a new linear complexity MRVQ–TSVQ hybrid algorithm and presents experimental results for those algorithms with comparisons to other methods. The material in this chapter also appears in [16, 21].

Chapter 5 introduces several new algorithms using a Lagrangian minimization. Efficient wavelet-based bit-plane coders, SPIHT and GTW, are optimized in a rate-distortion sense to achieve performance improvements of up to 1 dB. Work done here on the optimization of zero-tree codes led to a collaboration with Prof. Hao-Min Zhou (now at Georgia Tech) and Prof. Tony F. Chan (UCLA) to yield our method for the joint optimization of the adaptive Essentially Non-Oscillatory (ENO) Wavelet transform and the GTW algorithm, also described in Chapter 5. The material in this chapter also appears in part in [17, 18].

Finally, in Chapter 6 we summarize our contributions.

# Chapter 2 Background

Figure 2.1 shows a sequence of four images reconstructed from a multi-resolution code at different rates and distortions. The rate $R$ is measured in bits per symbol, and it describes the average number of bits used to describe each source symbol. We measure distortion $D$ as the mean squared error (MSE) between original source symbols $x_i$, $i = 1, \ldots, n$, and their corresponding reproduction symbols $\widehat{x}_i$. We calculate MSE as $D = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{x}_i)^2$.* In the figure, the first reproduction is at a low rate $R_1$ and has a high average distortion $D_1$ with respect to the original source image. By decoding an additional $R_2 - R_1$ bits, the total decoded rate increases to $R_2$, and the reconstructed image improves to yield a lower distortion $D_2 \leq D_1$. As we continue to increase the rate ($R_1 \leq R_2 \leq \cdots \leq R_{max}$), the distortion decreases ($D_1 \geq D_2 \geq \cdots \geq 0$).

In general, multi-resolution source codes cannot simultaneously achieve the best possible performance at all resolutions.† Figure 2.2 illustrates a rate-distortion function (dashed line) for a non-successively refinable source, and two possible scenarios (solid lines) for two-resolution coding of the same source. In one of those scenarios the encoder sets the coarse description to be identical to the optimal single-resolution code, forcing the high-resolution performance to deviate from the rate-distortion bound. In the other case, the high-resolution description is set to be optimal, and the low-resolution description fails to achieve the rate distortion bound.

Because there exist sources that are not successively refinable [23, 14], the optimization of multi-resolution source codes relies on the use of priority functions. A priority function is a weighting function describing the relative importance of the different resolutions to the system designer.

*For images, for example, the average bit-rate is measured in bits per pixel (*bpp*), and the distortion is usually expressed in decibels (*dB*) by the peak signal-to-noise ratio PSNR $= 10\log_{10}(255^2/\text{MSE})$ dB. Notice that PSNR increases with increasing image quality, and it assumes that the original picture is represented with 8 bits per pixel, where pixel values go from 0 to 255.

†The degradation in performance associated with multi-resolution coding is generally small, as exemplified in [20] and bounded for general sources in [35, 36, 26, 25].

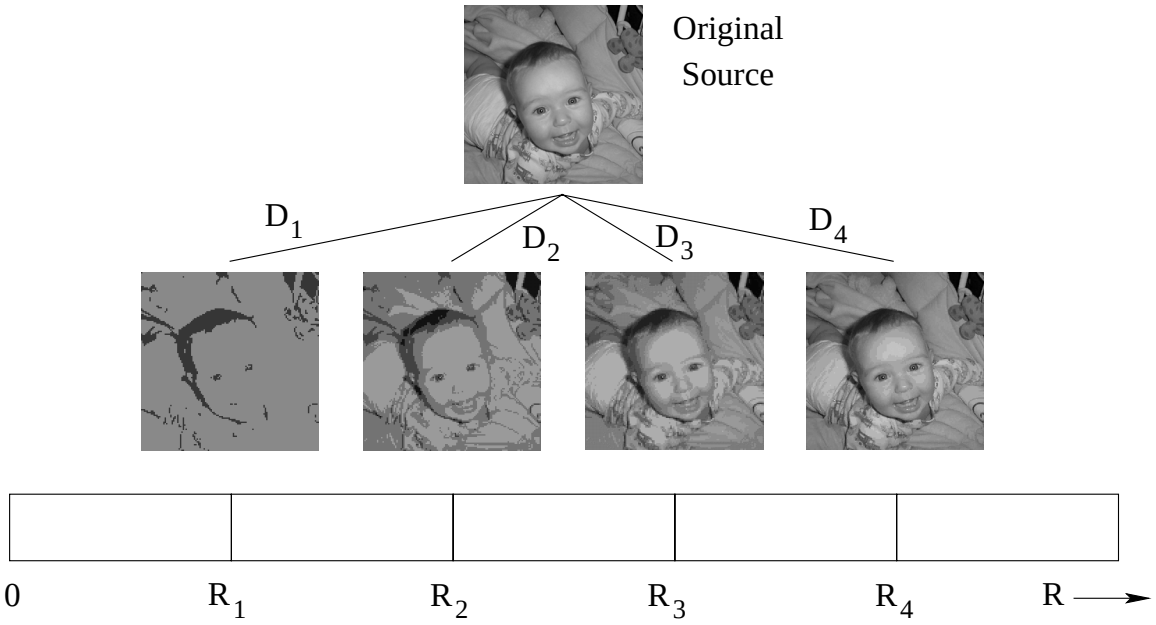



Figure 2.1: Multi-resolution source codes. Example of a 4-resolution reproduction sequence, comparing the original source image with the four quantized images with decreasing distortions $D_1$, $D_2$, $D_3$, and $D_4$, and their associated increasing bit-rates $R_1$, $R_2$, $R_3$, and $R_4$.

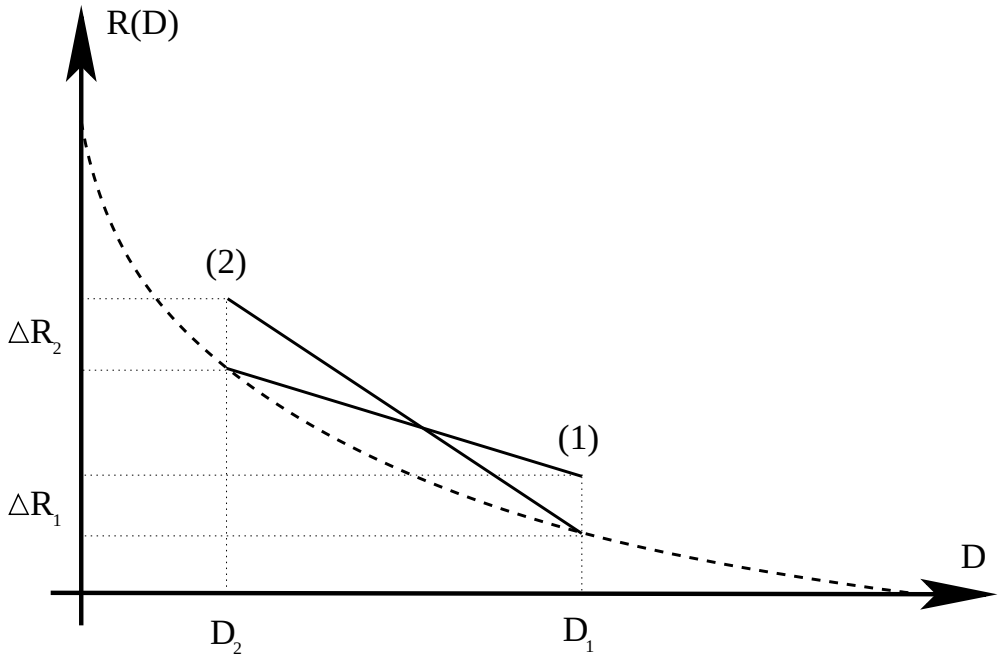



Figure 2.2: A single-resolution rate-distortion function $R(D)$ (dashed line) for a non-successively refinable source, and two possible scenarios (solid lines) for two-resolution coding of the same source. In (1), $R_2 = R(D_2)$, forcing $R_1 > R(D_1)$. In (2), $R_1 = R(D_1)$, forcing $R_2 > R(D_2)$.

In [19], Effros introduces a multi-resolution optimization criterion based on a weighted sum of rates and distortions. For any multi-resolution code with performance on the lower convex hull of achievable rate-distortion vectors, there exist non-negative constants $(\alpha_\ell, \lambda_\ell)_{\ell=1}^{L}$, later denoted by $(\alpha^L, \lambda^L)$, such that the given code's performance minimizes the multi-resolution Lagrangian performance measure

$$J = \sum_{\ell=1}^{L} \alpha_\ell (D_\ell + \lambda_\ell R_\ell) \tag{2.1}$$

over all multi-resolution codes. Here $D_\ell$ and $R_\ell$ denote the code's expected distortion and total rate at resolution $\ell$. We use $J$ as our performance criterion in multi-resolution code design throughout this thesis. Intuitively, $\alpha_\ell$ describes the priority on the $\ell$-th resolution and $D_\ell + \lambda_\ell R_\ell$ is the familiar rate-distortion Lagrangian for resolution $\ell$. Here the vector $(\alpha^L, \lambda^L)$ characterizes the direction of a hyper-plane tangential to the lower convex hull of the achievable rate-distortion region at a single point [20, 16].

The relative values of the $(\alpha^L, \lambda^L)$ parameters can be set according to the user's relative preference for some resolutions over the others, the probability of utilization of each resolution, the corresponding cost/benefit trade-off, or any other prioritization rationale. Once the priority function has been identified, we design the multi-resolution code that yields the best rate-distortion trade-off for those priorities.

# Chapter 3 Reproduction Alphabet Size

## 3.1 Introduction

The rate distortion function for a source random variable $X$ is denoted by $R(D)$. It is a non-increasing continuous convex $\cup$ function of $D$, and it represents the minimum rate necessary to achieve an average distortion not exceeding $D$. For the squared error distortion measure, the alphabet for the reproduction that achieves $R(D)$ is purely discrete if the Shannon lower bound (SLB) and $R(D)$ function do not coincide [47]. The multi-resolution rate-distortion bound for memoryless sources appears in [44]. We are interested in characterizing the alphabet size for the reproduction distribution that achieves the multi-resolution rate-distortion bound. While [50] generalizes some of the results of [47] from single- to multi-resolution codes, the techniques used in [47] to solve the alphabet size problem do not easily generalize to allow solution of the corresponding alphabet size problem for multi-resolution codes. We use the multi-resolution Lagrangian to pursue that goal. The investigation begins in Section 3.2 with a sketch of a new derivation of the single-resolution result of [47]. The hope is that the given technique can be generalized to characterize the optimal reproduction alphabet for multi-resolution codes. Section 3.3 sketches the proposed method of attack, giving full details for the parts of that argument that have been solved to date.

## 3.2 Single-resolution Case

Let random variable $X$ have probability density function (pdf) $p(x)$ on continuous alphabet $\mathcal{X}$. Use $Y$ to denote the reproduction random variable with output density $q(y) = \int_{\mathcal{X}} p(x)\, q(y|x)\, dx$ on output alphabet $\mathcal{Y}$. The goal here is to determine whether

the output alphabet $\mathcal{Y}$ that achieves the rate-distortion function

$$\begin{aligned} R(D) &= \min_{q(y|x):E_{p(x)\,q(y|x)}\rho(x,y)\,\le\,D} I(X;Y) \\ &= \min_{q(y|x):\int_{\mathcal{X}}\int_{\mathcal{Y}} p(x)\,q(y|x)\,\rho(x,y)\,dy\,dx\,\le\,D} \int_{\mathcal{X}}\int_{\mathcal{Y}} p(x)\,q(y|x)\,\ln\left[\frac{q(y|x)}{q(y)}\right] dy\,dx \end{aligned}$$

is discrete or continuous.

We begin by formulating the above constrained minimization as a Lagrangian.

In particular, the above constrained minimization is equivalent to the unconstrained minimization

$$J^{\star}_{\alpha,\beta} = \min_{q(y|x)} J_{\alpha,\beta}(q)$$

for the appropriate choice of Lagrangian parameters $\alpha$ and $\beta$,where

$$\begin{aligned} J_{\alpha,\beta}\left[q(y|x)\right] &= \alpha\, E_{p(x)\,q(y|x)}\rho(X,Y) + \beta\, I(X,Y) \\ &= \alpha \int_{\mathcal{X}}\int_{\mathcal{Y}} p(x)\,q(y|x)\,\rho(x,y)\,dy\,dx + \beta \int_{\mathcal{X}}\int_{\mathcal{Y}} p(x)\,q(y|x)\,\ln\left[\frac{q(y|x)}{q(y)}\right] dy\,dx \\ &= \beta \int_{\mathcal{X}}\int_{\mathcal{Y}} p(x)\,q(y|x)\,\ln\left[\frac{q(y|x)}{q(y)\,e^{-\frac{\alpha}{\beta}\rho(x,y)}}\right] dy\,dx. \end{aligned} \tag{3.1}$$

To identify the optimal conditional distribution $q(y|x)$, we employ calculus of variations. First we apply a perturbation of the distribution $q(y|x)$. That is, we replace $q(y|x)$ with

$$q^{\epsilon}(y|x) \triangleq (1-\epsilon)\,q(y|x) + \epsilon\,\xi(y|x)\,, \tag{3.2}$$

where $\epsilon > 0$, and $\xi(y|x)$ is an admissible perturbation, here any conditional density on $Y$ given $X$ for which the integrals exist and such that it is valid to change the order of integration and differentiation where required. By further replacing $q(y) = \int'_{\mathcal{X}} p(x')\,q(y|x')\,dx'$, Equation (3.1) becomes

$$\begin{aligned} J_{\alpha,\beta}\left[q^{\epsilon}(y|x)\right] = \beta \int_{\mathcal{X}}\int_{\mathcal{Y}} & p(x)\left\{q(y|x) + \epsilon\left[\xi(y|x) - q(y|x)\right]\right\} \\ & \cdot \ln\left[\frac{\left\{q(y|x) + \epsilon\left[\xi(y|x) - q(y|x)\right]\right\}}{\int_{\mathcal{X}'} p(x')\,\left\{q(y|x') + \epsilon\left[\xi(y|x') - q(y|x')\right]\right\}\,dx'\,e^{-\frac{\alpha}{\beta}(x-y)^2}}\right] dy\,dx\,. \end{aligned}$$

Setting $\frac{\partial}{\partial \epsilon} J_{\alpha,\beta}\left[q^{\epsilon}(y|x)\right] = 0^{*}$ allows us to characterize the optimal $q(y|x)$. For a fixed $p(x)$, a necessary condition for the optimality of a given $q(y|x)$ is that **all** admissible perturbations of $q(y|x)$ increase the value of the functional $J_{\alpha,\beta}\left[q^{\epsilon}(y|x)\right]$. The details follow.

$$\begin{aligned} 0 &= \left.\frac{\partial J_{\alpha,\beta}\left[q^{\epsilon}(y|x)\right]}{\partial \epsilon}\right|_{\varepsilon=0} \\ &= \int_{\mathcal{X}}\int_{\mathcal{Y}} p(x)\left[\xi(y|x) - q(y|x)\right] \cdot \ln\left[\frac{q(y|x)}{\int_{\mathcal{X}'} p(x')\, q(y|x')\, dx' \cdot e^{-\frac{\alpha}{\beta}(x-y)^2}}\right] dy\, dx\,. \end{aligned} \tag{3.3}$$

Define $g(x,y) = \frac{\alpha}{\beta}(x-y)^2 + \ln\left[\frac{q(y|x)}{q(y)}\right]$. Given this definition, we can rewrite (3.3) as

$$E_{p(x)q(y|x)}\, g(x,y) = E_{p(x)\xi(y|x)}\, g(x,y) \tag{3.4}$$

for all admissible conditional distribution perturbations $\xi(y|x)$ and a given $p(x)$. Intuitively, any function $f(x,y)$ that has the same expected value for all $x$ and $y$ must be everywhere constant. Likewise, we expect that any function $g(x,y)$ that has the same expected value for a fixed density $p(x)$ and all conditional densities $\xi(y|x)$ can vary with $y$ only on sets of measure zero with respect to the fixed distribution on $X$. Using this intuition, Equation (3.4) suggests that $g(x,y)$ is almost surely a function only of $x$. We note that formalizing this intuition into a result would require a proof that we have not completed to date. The remainder of the argument suggests how to proceed if the given intuition is correct.

If the conjecture is verified, then there exists a function $\psi(x)$ such that

$$e^{g(x,y)} = \frac{q(y|x)}{q(y)\, e^{-\frac{\alpha}{\beta}(x-y)^2}} = \psi(x)\,,$$

almost surely $p(x)$, implying

$$q(y|x) = \psi(x)\, q(y)\, e^{-\frac{\alpha}{\beta}(x-y)^2}\,, \tag{3.5}$$

*Notice that by applying calculus of variations, the first order partial derivative of the Lagrangian (with respect to the scalar $\epsilon > 0$ of the $\epsilon$-perturbation) should be sufficient to determine the minimum of the Lagrangian functional, given that the Lagrangian functional itself is a convex function of $q$.

which is the Gibbs distribution. Here

$$\psi(x) = \frac{1}{\int_{\mathcal{Y}} q(y)\, e^{-\frac{\alpha}{\beta}(x-y)^2} dy} \tag{3.6}$$

is the normalization factor needed to give a legitimate conditional density. Based on (3.5),

$$q(x,y) = p(x)\, \psi(x)\, q(y)\, e^{-\frac{\alpha}{\beta}(x-y)^2}\,.$$

If we assume that there is an open interval $I_0$ in the support of the output random variable $Y$, then

$$q(x|y) = p(x)\, \psi(x)\, e^{-\frac{\alpha}{\beta}(x-y)^2} \tag{3.7}$$

for all $y \in I_0$. In particular we require $q(y) > 0$ as a condition for finding $q(x|y)$. Integrating (3.7) with respect to $x$ and taking the first partial derivative with respect to the reproduction $y$, we have

$$\begin{aligned} 0 &= \int_{\mathcal{X}} p(x)\, \psi(x) \frac{\partial}{\partial y} e^{-\frac{\alpha}{\beta}(x-y)^2}\, dx \\ 0 &= \int_{\mathcal{X}} p(x)\, \psi(x)\, (x-y)\, e^{-\frac{\alpha}{\beta}(x-y)^2}\, dx\,. \end{aligned} \tag{3.8}$$

Notice that (3.8) is identical to the Euler-Lagrange equation from [47, p. 1942]. The argument that follows is therefore identical to the argument in [47]. We reproduce it here for the sake of clarity. The right-hand side of (3.8) is zero everywhere in the interval $I_0$, which ensures all derivatives at a given point $y_0 \in I_0$ must be zero. Thus

$$\begin{aligned} 0 &= \frac{\partial^n}{\partial y^n} \int_{\mathcal{X}} p(x)\ \psi(x)\ e^{-\frac{\alpha}{\beta}(x-y)^2} dx \Bigg|_{y=y_0} \\ &= \int_{\mathcal{X}} p(x)\ \psi(x)\ \frac{\partial^n}{\partial y^n}\ e^{-\frac{\alpha}{\beta}(x-y)^2} \Bigg|_{y=y_0} dx \end{aligned} \tag{3.9}$$

for all $n \geq 1$, where the justification for the exchange in differentiation and integration order can be found, for example, in [47, p. 1950]. Notice also the possible represen-

tation of the partial derivatives of the exponential in (3.9) as Hermite polynomials[†], which can be written by Rodrigues' Formula as

$$H_n(x-y_0) = e^{\frac{\alpha}{\beta}(x-y_0)^2} \frac{d^n}{dy_0^n} e^{-\frac{\alpha}{\beta}(x-y_0)^2}. \tag{3.10}$$

Then, as in [47], equation (3.9) can be written as

$$\int_{\mathcal{X}} p(x)\,\psi(x)\,H_n(x-y_0)\,e^{-\frac{\alpha}{\beta}(x-y_0)^2}\,dx = 0 \tag{3.11}$$

for all $n \geq 1$, which implies that $p(x)\,\psi(x)$ must be a constant, since it is orthogonal to all Hermite polynomials for $n \geq 1$. That is,

$$p(x)\,\psi(x) = \frac{p(x)}{\int_{\mathcal{Y}} q(y)\,e^{-\frac{\alpha}{\beta}(x-y)^2}\,dy} = \text{constant}. \tag{3.12}$$

The constant can be determined by integrating (3.12) with respect to $x$ to obtain the relationship between the input and output density functions

$$p(x) = \sqrt{\frac{\frac{\alpha}{\beta}}{\pi}} \int_{\mathcal{Y}} q(y)\,e^{-\frac{\alpha}{\beta}(x-y)^2}\,dy\,. \tag{3.13}$$

Recall the assumption that support of the output random variable $Y$ contains an interval $I_0$. Any $q(y)$ not satisfying equation (3.13) would contradict the assumption of existence of an interval $I_0$ in the support of $Y$. This contradiction would imply that the reproduction alphabet is discrete.

For the single resolution case, $p(x)$ and $q(y)$ satisfy (3.13) if and only if the Shannon lower bound is tight (see, e.g., [5]).

As a summary for this section, let us emphasize two points. First, note that our potential alternative proof for [47] relies on an unproven conjecture. Second, notice that while the single-resolution result is not new, the above technique may provide useful insight for deriving a multi-resolution result. We extend the proposed argument

[†]Hermite polynomials are an orthogonal set in $(-\infty, \infty)$ with respect to the weight function, here $e^{-\frac{\alpha}{\beta}[x-y]^2}$.

to the two-resolution case, again relying on the conjecture, in the following section.

## 3.3 Two-resolution Case

Let us consider a two-resolution source code. Denote $X$ as the source random variable with a probability density function $p(x)$ on a continuous alphabet $\mathcal{X}$, and $Y_1$ and $Y_2$ as the reproduction random variables with output alphabets $\mathcal{Y}_1$ and $\mathcal{Y}_2$ respectively, and output probability density functions $q(y_1) = \int_{\mathcal{X}} p(x)\, q(y_1|x)\, dx$ and $q(y_1, y_2) = \int_{\mathcal{X}} p(x)\, q(y_1, y_2|x)\, dx$.

The achievable rate-distortion region is the set of vectors $(R_1, R_2, D_1, D_2)$ such that

$$\begin{cases} R_1 \geq I(X; Y_1) \\ R_2 \geq I(X; Y_1, Y_2) \\ D_1 \geq E\rho(X, Y_1) \\ D_2 \geq E\rho(X, Y_2) \end{cases}$$

for some conditional density $q(y_1, y_2|x)$. The mutual informations are

$$I(X; Y_1) = \int_{\mathcal{X}} \int_{\mathcal{Y}_1} p(x)\, q(y_1|x)\, \ln \left[ \frac{q(y_1|x)}{q(y_1)} \right] dy_1\, dx \tag{3.14a}$$

$$I(X; Y_1, Y_2) = \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x)\, q(y_1, y_2|x) \ln \left[ \frac{q(y_1, y_2|x)}{q(y_1, y_2)} \right] dy_2\, dy_1\, dx \tag{3.14b}$$

and the expected distortions are

$$E\rho(X, Y_1) = \int_{\mathcal{X}} \int_{\mathcal{Y}_1} p(x)\, q(y_1|x)\, \rho(x, y_1)\, dy_1\, dx \tag{3.15a}$$

$$E\rho(X, Y_2) = \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x)\, q(y_1, y_2|x)\, \rho(x, y_2)\, dy_2\, dy_1\, dx. \tag{3.15b}$$

As in Section 3.2, we find the lower convex hull of this region using a Lagrangian functional

$$J^{\star}_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \min_{q(y_1, y_2|x)} J_{\boldsymbol{\alpha},\boldsymbol{\beta}}(q),$$

where the two-resolution Lagrangian is

$$J_{\boldsymbol{\alpha},\boldsymbol{\beta}}(q) = \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x)\, q(y_1, y_2|x) \cdot \left[ \alpha_1\, \rho_1(x, y_1) + \beta_1 \ln \left[ \frac{q(y_1|x)}{q(y_1)} \right] + \alpha_2\, \rho_2(x, y_2) + \beta_2 \ln \left[ \frac{q(y_1, y_2|x)}{q(y_1, y_2)} \right] \right] dy_2\, dy_1\, dx, \tag{3.16}$$

where

$$\begin{aligned} q(y_1) &= \int_{\mathcal{X}} \int_{\mathcal{Y}_2} p(x)\, q(y_1, y_2|x)\, dy_2\, dx \\ q(y_1|x) &= \int_{\mathcal{Y}_2} q(y_1, y_2|x)\, dy_2 \\ q(y_1, y_2) &= \int_{\mathcal{X}} p(x)\, q(y_1, y_2|x)\, dx\,. \end{aligned}$$

(We here use notation that is slightly different from, but functionally equivalent to, the notation of Chapter 2. In particular, here $\beta_\ell = \alpha_\ell\, \lambda_\ell$.)

We again apply calculus of variations to find optimal density $q(y_1, y_2|x)$. Thus, we replace $q(y_1, y_2|x)$ with

$$q^\epsilon(y_1, y_2|x) \triangleq (1 - \epsilon)\, q(y_1, y_2|x) + \epsilon\, \xi(y_1, y_2|x)\,, \tag{3.17}$$

where $\epsilon > 0$ and $\xi(y_1, y_2|x)$ is an admissible perturbation. We then take the first order partial derivative of the Lagrangian

$$\begin{aligned} J_{\boldsymbol{\alpha},\boldsymbol{\beta}}\left[q^\epsilon(y_1, y_2|x)\right] = \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} & p(x)\, q^\epsilon(y_1, y_2|x) \\ \cdot \quad & \left[ \beta_1 \ln \left[ \frac{\int q^\epsilon(y_1, y_2'|x)\, dy_2'}{e^{-\frac{\alpha_1}{\beta_1}(x - y_1)^2} \iint p(x') q^\epsilon(y_1, y_2'|x')\, dy_2'\, dx'} \right] \right. \\ & \left. + \ \beta_2 \ln \left[ \frac{q^\epsilon(y_1, y_2|x)}{e^{-\frac{\alpha_2}{\beta_2}(x - y_2)^2} \int p(x')\, q^\epsilon(y_1, y_2|x')\, dx'} \right] \right] dy_2\, dy_1\, dx \end{aligned}$$

with respect to $\epsilon$. Setting $\frac{\partial}{\partial \epsilon} J_{\boldsymbol{\alpha},\boldsymbol{\beta}}\left[q^\epsilon(y_1, y_2|x)\right] = 0$ allows us to find properties of the optimal $q(y_1, y_2|x)$. For a fixed $p(x)$, a necessary condition for the optimality of a

given $q(y_1, y_2|x)$ is that all admissible perturbations of $q(y_1, y_2|x)$ increase the value of the functional $J_{\boldsymbol{\alpha},\boldsymbol{\beta}}\left[q^{\epsilon}(y_1, y_2|x)\right]$. Again, an admissible perturbation $\xi(y_1, y_2|x)$ is a conditional distribution for which the integrals exist and such that it is valid to change the order of integration and differentiation where required. The details follow.

$$
\begin{aligned}
0 = & \left. \frac{\partial J_{\boldsymbol{\alpha},\boldsymbol{\beta}}\left[q^{\epsilon}(y_1, y_2|x)\right]}{\partial \epsilon} \right|_{\varepsilon=0} \\
= & \ \beta_1 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x) \left[\xi(y_1, y_2|x) \ - \ q(y_1, y_2|x)\right] \ \ln \left[\int q(y_1, y_2'|x)\, dy_2'\right] \, dy_2\, dy_1\, dx \\
- & \ \beta_1 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x) \left[\xi(y_1, y_2|x) \ - \ q(y_1, y_2|x)\right] \\
& \qquad \cdot \ln \left[e^{-\frac{\alpha_1}{\beta_1}(x-y_1)^2} \iint p(x')\, q(y_1, y_2'|x')\, dy_2'\, dx'\right] \, dy_2\, dy_1\, dx \\
+ & \ \beta_1 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x)\, q(y_1, y_2|x) \\
& \qquad \cdot \left[\frac{\int \left[\xi(y_1, y_2|x) \ - \ q(y_1, y_2|x)\right] \, dy_2'}{\int q(y_1, y_2'|x)\, dy_2'}\right] \, dy_2\, dy_1\, dx \\
- & \ \beta_1 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x)\, q(y_1, y_2|x) \\
& \qquad \cdot \left[\frac{e^{-\frac{\alpha_1}{\beta_1}(x-y_1)^2} \iint p(x') \left[\xi(y_1, y_2'|x') - q(y_1, y_2'|x')\right] dy_2'\, dx'}{e^{-\frac{\alpha_1}{\beta_1}(x-y_1)^2} \iint p(x')\, q(y_1, y_2'|x')\, dy_2'\, dx'}\right] dy_2\, dy_1\, dx \\
+ & \ \beta_2 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x) \left[\xi(y_1, y_2|x) \ - \ q(y_1, y_2|x)\right] \ \ln \left[q(y_1, y_2|x)\right] \, dy_2\, dy_1\, dx \\
- & \ \beta_2 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x) \left[\xi(y_1, y_2|x) \ - \ q(y_1, y_2|x)\right] \\
& \qquad \cdot \ln \left[e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2} \int p(x')\, q(y_1, y_2|x')\, dx'\right] \, dy_2\, dy_1\, dx \\
- & \ \beta_2 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x) \ q(y_1, y_2|x) \left[\frac{\xi(y_1, y_2|x) \ - \ q(y_1, y_2|x)}{q(y_1, y_2|x)}\right] \, dy_2\, dy_1\, dx \\
+ & \ \beta_2 \int_{\mathcal{X}} \int_{\mathcal{Y}_1} \int_{\mathcal{Y}_2} p(x)\, q(y_1, y_2|x) \\
& \qquad \cdot \left[\frac{e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2} \int p(x') \left[\xi(y_1, y_2|x') - q(y_1, y_2|x')\right] dx'}{e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2} \int p(x')\, q(y_1, y_2|x')\, dx'}\right] dy_2\, dy_1\, dx. \quad (3.18)
\end{aligned}
$$

Equivalently,

$$\int_{\mathcal{X}}\int_{\mathcal{Y}_1}\int_{\mathcal{Y}_2} p(x)\,\left[\xi(y_1,y_2|x)\;-\;q(y_1,y_2|x)\right] \cdot\left\{\beta_1\ln\left[\frac{\int q(y_1,y_2'|x)\,dy_2'}{e^{-\frac{\alpha_1}{\beta_1}(x-y_1)^2}\iint p(x')\,q(y_1,y_2'|x')\,dy_2'\,dx'}\right] + \beta_2\ln\left[\frac{q(y_1,y_2|x)}{e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\int p(x')\,q(y_1,y_2|x')\,dx'}\right]\right\} dy_2\,dy_1\,dx\;=0\,. \quad (3.19)$$

Defining

$$\begin{aligned} g(x,y_1,y_2) &= \frac{\alpha_1}{\beta_1}(x-y_1)^2+\beta_1\ln\left[\frac{q(y_1|x)}{q(y_1)}\right]+\frac{\alpha_2}{\beta_2}(x-y_2)^2+\beta_2\ln\left[\frac{q(y_1,y_2|x)}{q(y_1,y_2)}\right] \\ &= \beta_1\ln\left[\frac{\int q(y_1,y_2'|x)\,dy_2'}{e^{-\frac{\alpha_1}{\beta_1}(x-y_1)^2}\iint p(x')q(y_1,y_2'|x')dy_2'dx'}\right] \\ &\quad+\beta_2\ln\left[\frac{q(y_1,y_2|x)}{e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\int p(x')q(y_1,y_2|x')dx'}\right] \end{aligned} \quad (3.20)$$

allows us to rewrite (3.19) as

$$E_{p(x)q(y_1,y_2|x)}\,g(x,y_1,y_2) \;=\; E_{p(x)\xi(y_1,y_2|x)}\,g(x,y_1,y_2) \quad (3.21)$$

for all admissible reproduction conditional distribution perturbations $\xi(y_1,y_2|x)$ and a given $p(x)$. Based on the same intuition described for the single-resolution case, Equation (3.21) suggests that $g(x,y_1,y_2)$ only depends on $x$, almost surely–$p(x)$. As in the single-resolution case, we rely on but do not prove this conjecture. The remainder of the argument suggests how to proceed if the given intuition is correct. If this conjecture is verified, then

$$g(x,\hat{y_1},\hat{y_2}) = g(x,\hat{\hat{y}_1},\hat{\hat{y}_2})\,;\quad \forall\,(x,\hat{y_1},\hat{y_2}),(x,\hat{\hat{y}_1},\hat{\hat{y}_2})\in\mathcal{X}\times\mathcal{Y}_1\times\mathcal{Y}_2. \quad (3.22)$$

That is,

$$\frac{\alpha_1}{\beta_1}(x-\hat{y_1})^2+\beta_1\ln\left[\frac{q(\hat{y_1}|x)}{q(\hat{y_1})}\right]+\frac{\alpha_2}{\beta_2}(x-\hat{y_2})^2+\beta_2\ln\left[\frac{q(\hat{y_1},\hat{y_2}|x)}{q(\hat{y_1},\hat{y_2})}\right]$$
$$=\frac{\alpha_1}{\beta_1}(x-\hat{\hat{y_1}})^2+\beta_1\ln\left[\frac{q(\hat{\hat{y_1}}|x)}{q(\hat{\hat{y_1}})}\right]+\frac{\alpha_2}{\beta_2}(x-\hat{\hat{y_2}})^2+\beta_2\ln\left[\frac{q(\hat{\hat{y_1}},\hat{\hat{y_2}}|x)}{q(\hat{\hat{y_1}},\hat{\hat{y_2}})}\right],$$

which yields

$$q(y_1,y_2|x)=\phi(x)\;q(y_1,y_2)\left[\frac{q(y_1|x)}{q(y_1)}\right]^{-\frac{\beta_1}{\beta_2}}e^{-\frac{\alpha_1}{\beta_2}(x-y_1)^2}e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}, \tag{3.23}$$

where

$$\phi(x)=\frac{1}{\int_{\mathcal{Y}_1}\int_{\mathcal{Y}_2}q(y_1,y_2)\left[\frac{q(y_1|x)}{q(y_1)}\right]^{-\frac{\beta_1}{\beta_2}}e^{-\frac{\alpha_1}{\beta_2}(x-y_1)^2}e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}dy_2\,dy_1}. \tag{3.24}$$

Consider an arbitrary $\hat{y_1}\in\mathcal{Y}_1$ for which $q(\hat{y_1})>0$. Let us assume there is an open interval $I_{2_0}$ such that $q(y_2|\hat{y_1})>0$ for all $y_2\in I_{2_0}$. Our goal is to find an expression that relates $p(x)$ and the optimal $q(y_1,y_2)$ as $p(x)=f(q(y_1,y_2))$. In order to accomplish that, we will use the fact that $q(x|y_1,y_2)$ must integrate to one to get an equation that relates $p(x)$ to $q(y_1|x)$. And to further solve, we will apply the fact that $q(y_1|x)=\int_{\mathcal{Y}_2}q(y_1,y_2|x)\,dy_2$.

If we multiply both sides of (3.23) by $p(x)$ and divide by $q(y_1,y_2)$, we get

$$q(x|y_1,y_2)=p(x)\;\phi(x)\left[\frac{q(y_1|x)}{q(y_1)\,e^{-\frac{\alpha_1}{\beta_1}(x-y_1)^2}}\right]^{-\frac{\beta_1}{\beta_2}}e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}. \tag{3.25}$$

Integrating with respect to the input $X$ and taking the first partial derivative with respect to the second-resolution reproduction $Y_2$ gives

$$0=\int_{\mathcal{X}}p(x)\,\phi(x)\left[\frac{q(y_1|x)}{q(y_1)\,e^{-\frac{\alpha_1}{\beta_1}(x-y_1)^2}}\right]^{-\frac{\beta_1}{\beta_2}}(x-y_2)\,e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}dx. \tag{3.26}$$

The right-hand side of (3.26) is zero for all $y_2$ in the interval $I_{2_0}$, which ensures

all derivatives at a given point $\hat{y_2} \in I_{2_0}$ must be zero. Exchanging the order of differentiation and integration, with equivalent arguments as in (3.9) and [47, p. 1950], we get

$$\begin{aligned} 0 &= \frac{\partial^n}{\partial y_2^n}\left|_{y_2=\hat{y_2}} \int_{\mathcal{X}} p(x)\,\phi(x) \left[\frac{q(\hat{y_1}|x)}{q(\hat{y_1})\,e^{-\frac{\alpha_1}{\beta_1}(x-\hat{y_1})^2}}\right]^{-\frac{\beta_1}{\beta_2}} e^{-\frac{\alpha_2}{\beta_2}(x-\hat{y_2})^2} dx \right. \\ &= \int_{\mathcal{X}} p(x)\,\phi(x) \left[\frac{q(\hat{y_1}|x)}{q(\hat{y_1})\,e^{-\frac{\alpha_1}{\beta_1}(x-\hat{y_1})^2}}\right]^{-\frac{\beta_1}{\beta_2}} \frac{\partial^n}{\partial y_2^n}\left|_{y_2=\hat{y_2}} e^{-\frac{\alpha_2}{\beta_2}(x-\hat{y_2})^2} dx, \right. \end{aligned} \tag{3.27}$$

for all $n \geq 1$. As in (3.11), we can write an equivalent expression to (3.27) using the Hermite polynomials with respect to the function $e^{-\frac{\alpha}{\beta}(x-\hat{y_2})^2}$, as

$$\int_{\mathcal{X}} p(x)\,\phi(x) \left[\frac{q(\hat{y_1}|x)}{q(\hat{y_1})\,e^{-\frac{\alpha_1}{\beta_1}(x-\hat{y_1})^2}}\right]^{-\frac{\beta_1}{\beta_2}} H_n(x-\hat{y_2})\,e^{-\frac{\alpha}{\beta}(x-\hat{y_2})^2} dx = 0 \tag{3.28}$$

for all $n \geq 1$. Then

$$p(x)\,\phi(x) \left[\frac{q(\hat{y_1}|x)}{q(\hat{y_1})\,e^{-\frac{\alpha_1}{\beta_1}(x-\hat{y_1})^2}}\right]^{-\frac{\beta_1}{\beta_2}} = K(\hat{y_1})\,, \tag{3.29}$$

given that the L.H.S. of (3.29) is orthogonal to all Hermite polynomials $H_n(x-\hat{y_2})$ of $n \geq 1$. Notice also that it doesn't vary with $\hat{y_2}$, as the same result applies for all $\hat{y_2}$ in the interval $I_{2_0}$.

In order to further analyze the above result from (3.29), we revisit (3.23) to solve for $q(y_1|x)$. Integrating both sides of (3.23) with respect to $y_2$ gives

$$\begin{aligned} q(y_1|x) &= \int_{\mathcal{Y}_2} q(y_1, y_2|x)\,dy_2 \\ &= \int_{\mathcal{Y}_2} \phi(x)\; q(y_1, y_2) \left[\frac{q(y_1|x)}{q(y_1)}\right]^{-\frac{\beta_1}{\beta_2}} e^{-\frac{\alpha_1}{\beta_2}(x-y_1)^2} e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\,dy_2 \\ &= q(y_1)\,\phi(x) \left[\frac{q(y_1|x)}{q(y_1)}\right]^{-\frac{\beta_1}{\beta_2}} e^{-\frac{\alpha_1}{\beta_2}(x-y_1)^2} \int_{\mathcal{Y}_2} q(y_2|y_1)\,e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\,dy_2, \end{aligned}$$

which yields

$$\left[\frac{q(y_1|x)}{q(y_1)}\right]^{\frac{\beta_1+\beta 2}{\beta_2}} = \phi(x)\, e^{-\frac{\alpha_1}{\beta_2}(x-y_1)^2} \int_{\mathcal{Y}_2} q(y_2|y_1)\, e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\, dy_2\,. \tag{3.30}$$

Rewriting (3.29) for clarity as

$$p(x)\,\phi(x)\,\frac{q(y_1|x)}{q(y_1)}\left[\frac{q(y_1|x)}{q(y_1)}\right]^{-\left(\frac{\beta_1+\beta_2}{\beta_2}\right)} e^{-\frac{\alpha_1}{\beta_2}(x-y_1)^2} = K(y_1)\,,$$

and applying (3.30) to it, we get

$$p(x)\,\frac{q(y_1|x)}{q(y_1)}\left[\int_{\mathcal{Y}_2} q(y_2|y_1)\, e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\, dy_2\right]^{-1} = K(y_1)\,, \tag{3.31}$$

and the equivalent

$$p(x|y_1) = K(y_1) \int_{\mathcal{Y}_2} q(y_2|y_1)\, e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\, dy_2\,, \tag{3.32}$$

where integrating with respect to $x$ gives

$$K(y_1) = \frac{1}{\int_{\mathcal{Y}_2} q(y_2|y_1) \int_{\mathcal{X}} e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\, dx\, dy_2} = \sqrt{\frac{\frac{\alpha_2}{\beta_2}}{\pi}}\,. \tag{3.33}$$

This means that for each and every $y_1$ compliant with the assumptions made earlier, the function $K(y_1)$ is actually a constant. Notice that the above discussion again relies on the unproven result described in Section 3.2. A proof of that result, when combined with the analysis given here, would give a two-resolution parallel to Rose's theorem about the optimal reproduction alphabet size for one-resolution source codes. In particular, that missing piece would allow us to relate input and output reproduction densities as

$$p(x) = \sqrt{\frac{\frac{\alpha_2}{\beta_2}}{\pi}}\;\; \frac{\int_{\mathcal{Y}_2} q(y_1, y_2)\, e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\, dy_2}{q(y_1|x)}\,. \tag{3.34}$$

The conjecture is that when this relationship is not satisfied, then the output alphabet at the second resolution is discrete. Further, the conjecture suggests that the second

resolution reproduction is continuous only if the source and its optimal reproductions are related through an additive Gaussian random variable.

## 3.4 Conclusions

The conjecture for the two-resolution case reflects a parallel with Rose's one-resolution result. For single-resolution source coding, the optimal reproduction alphabet is continuous if and only if $p(x) = \sqrt{\frac{\alpha}{\pi\,\beta}}\ \int_{\mathcal{Y}} q(y)\, e^{-\frac{\alpha}{\beta}(x-y)^2} dy$. The conjecture for the two-resolution case is that the optimal second resolution reproduction alphabet is likewise continuous if and only if $q(y_1|x)\ p(x) = \sqrt{\frac{\alpha_2}{\pi\,\beta_2}}\ \int_{\mathcal{Y}_2} q(y_1, y_2)\, e^{-\frac{\alpha_2}{\beta_2}(x-y_2)^2}\, dy_2$. For the single-resolution case, the given condition is met if and only if the SLB on $R(D)$ is tight at the given rate and distortion, which occurs if and only if the source and its optimal reproduction are related through an additive Gaussian random variable. The form of the two-resolution conjecture suggests a similar relationship.

# Chapter 4 Practical MRVQ Design

## 4.1 Introduction

A multiple stage vector quantizer (MSVQ) [34], also called a residual quantizer [2, 3, 27, 4] is a multi-resolution code in which the first stage uses a small codebook to output a crude quantization of the input, the second stage quantizes the error between the original and the output of the first stage, and so on. A tree-structured vector quantizer (TSVQ) [8, 45] gives an alternative approach to multi-resolution coding where the encoder describes a source vector by mapping it in a top-down greedy fashion to a single path through a tree-structured codebook. The progressive vector quantizer (progressive VQ) of [46] also uses a tree-structured codebook but relies on an encoder that considers only the highest-level reproduction in choosing its path through the tree. The pruned TSVQ (PTSVQ) [13] combines the top-down greedy tree-design algorithm and encoder of TSVQ with a pruning algorithm.

More recently, [29, 6, 7] use a design algorithm that minimizes an expected distortion $\sum_\ell p_\ell D_\ell$ with respect to a given distribution $\{p_\ell\}$ over the available rates $\{R_\ell\}$ in a fixed-rate code. Here $D_\ell$ and $R_\ell$ are the expected distortion and total rate, respectively, in the $\ell^{th}$ reproduction of a given source. In [33, 7], the problem of scalar quantizer design given a distribution $\{p_\ell\}$ over the resolutions is considered. For these scalar quantizers, optimality is defined in terms of minimization of the expected distortion $\sum_\ell p_\ell D_\ell$ with respect to a constraint on the expected rate $\sum_\ell p_\ell R_\ell$, giving a final optimization criterion equal to $\sum_{\ell=1}^{L} p_\ell D_\ell + \lambda \sum_{\ell=1}^{L} p_\ell R_\ell$.

Also tree-structured, the multi-resolution vector quantizer (MRVQ)* introduced in [19] uses rate distortion theory to develop optimality criterion $\sum_{\ell=1}^{L} \alpha_\ell (D_\ell + \lambda_\ell R_\ell)$ for both code design and encoder path choice. This criterion agrees with [33, 7] for cases with fixed slopes $\lambda_\ell = \lambda$, for all $\ell$, but differs otherwise. The optimality criterion

*The same code was called a multi-resolution TSVQ on its introduction in [19].

of [19] has $L-1$ more free parameters than the criterion used in [33, 7]. These $L-1$ degrees of freedom are crucial for obtaining arbitrary points on the outer convex hull of the achievable rate-distortion region.

The MRVQ work described in this thesis began with an implementation of variable-rate MRVQ and the goal of understanding how to choose the Lagrangian parameters for the optimality criterion introduced in [19]. The remainder of this chapter provides answers to those questions, discusses the complexity of the MRVQ algorithm and proposes a new lower complexity algorithm, and demonstrates the rate-distortion performance of practical implementations of the algorithm and some new variations on the MRVQ.

Section 4.2 briefly describes the optimal multi-resolution vector quantizer design algorithm introduced in [19]. Section 4.3 presents a method for converting design constraints into the corresponding Lagrangian parameters for general multi-resolution source codes. An analysis of MRVQ encoding complexity appears in Section 4.4, followed by a new linear complexity algorithm. Section 4.5 includes a series of experimental results comparing the performance of optimal multi-resolution vector quantizers to the theoretical bounds and to other single- and multi-resolution source codes. A summary and conclusions follow in Section 4.6.

## 4.2 Background: MRVQ Algorithm

MRVQ code design uses an iterative descent technique analogous to the generalized Lloyd algorithm for designing optimal fixed-rate vector quantizers [38] and the entropy constrained vector quantization (ECVQ) design algorithm for optimal variable-rate code design [12]. The goal of code design is to minimize the Lagrangian $\sum_{\ell=1}^{L} \alpha_\ell (D_\ell + \lambda_\ell R_\ell)$. Without loss of generality, the optimality criterion applied in [19, 20, 21] and throughout this chapter uses incremental rates $\{r_\ell\}$ instead of total rates $\{R_\ell\}$. Here $R_\ell = \sum_{j=1}^{L} r_j$ for all $j \in \{1, \ldots, L\}$. The resulting Lagrangian functional is again written $\sum_{\ell=1}^{L} [\alpha_\ell \, D_\ell + \beta_\ell \, r_\ell]$ although here $\beta_\ell = \sum_{i=\ell}^{L} \alpha_i \, \lambda_i$ for all $\ell$. For any given Lagrangian parameters $\alpha^L, \beta^L \geq 0$ our design objective is to minimize the Lagrangian

functional over all possible fixed-rate multi-resolution source codes or over all possible variable-rate multi-resolution source codes.

The algorithm iteratively optimizes quantizer encoder $A$, quantizer decoder $B$, and lossless encoder $\Gamma$ (and its lossless decoder). It is initialized with an arbitrary $L$-resolution tree-structured codebook, an arbitrary quantizer decoder $B$, and arbitrary lossless code $\Gamma$. Each iteration requires three steps that sequentially optimize the quantizer encoder $A$ for the given quantizer decoder $B$ and lossless code $\Gamma$, the quantizer decoder $B$ for the given quantizer encoder $A$ and lossless code $\Gamma$, and the lossless code $\Gamma$ for the given quantizer encoder $A$ and quantizer decoder $B$. The algorithm is run to convergence. The three steps required in each iteration are

1. nearest neighbor encoding

2. decoding to the centroid

3. optimizing the prefix code.

(See [19, 21] for a detailed description.)

The above MRVQ design algorithm jointly optimizes all resolutions of a multi-resolution code rather than designing the code one resolution at a time. At each step in each iteration of the above algorithm, the Lagrangian functional $\sum_{\ell=1}^{L} \left[\alpha_\ell \, D_\ell + \beta_\ell \, r_\ell\right]$. cannot increase. Since the functional is bounded below by 0, the algorithm is guaranteed to converge. Since each step in the algorithm produces a global minimum of the Lagrangian functional relative to the fixed source coding components, the algorithm is guaranteed to converge to a local optimum.

## 4.3 Choosing Lagrangian Parameters $(\alpha^L, \beta^L)$

The MRVQ design algorithm must be run separately for each $(\alpha^L, \beta^L)$ value of interest. Since each $(\alpha^L, \beta^L)$ corresponds to a single point on the desired convex hull, tracing out the entire convex hull requires a repetition of the above procedure at a variety of values of $(\alpha^L, \beta^L)$.

The search for the $(\alpha^L, \beta^L)$ vectors of interest is, in some sense, analogous to the search for the appropriate value for $\lambda$ (the "slope" of the $R(D)$ function [16]) in entropy constrained vector quantizer (ECVQ) design [12]. For ECVQ, instead of designing the code for all possible slopes, [12] proposes a bisection approach to allow code design for a particular desired rate (or entropy). The ECVQ algorithm designs a vector quantizer for a specific slope $\lambda$ at the middle of a range $[\lambda_{min}, \lambda_{max}]$. The design process then shrinks the range to the lower or higher half in the direction that decreases the gap between the observed and desired rate. The process continues, at each step cutting the range of slopes in half until reaching the target rate.

Similarly, for a specific set of target rates in a multi-resolution code, it would be impractical to design a code for each possible $(\alpha^L, \beta^L)$ vector until hitting the desired target rates. Searching this space is difficult since these parameters are interdependent, which makes the challenge of finding the target vector $(\alpha^L, \beta^L)$ an interesting problem.

Next, we propose a method based on an interval partitioning approach, for converting a collection of functional design constraints into corresponding Lagrangian parameters $(\alpha^L, \beta^L)$; while the approach is here demonstrated in the context of MRVQ design, it applies for general multi-resolution codes. An independently derived method to determine Lagrangian coefficients for navigation of the rate-distortion surface (in particular to determine the average distortion achievable for a given pair of rates in a two-resolution code) appears in [50]; that work was published after the work presented here was completed.

### 4.3.1 New Method

Let us suppose that a system designer wishes to design the variable-rate code with (incremental) rate vector $(r^L)^\star = (r_1^\star, \cdots, r_L^\star)$ that minimizes $\sum_{\ell=1}^{L} p_\ell D_\ell$ for some given priority vector $p^L$ with $p_\ell \geq 0$ for all $\ell \in \{1, \ldots, L\}$ and $\sum_{\ell=1}^{L} p_\ell = 1$. Alternative problem formulations (e.g., matching distortion constraints given priorities over the rates) can be handled similarly.

To incorporate the given priorities in our Lagrangian parameters and maintain the symmetry between rate and distortion, we set $\alpha_\ell = (1-c)\,p_\ell$ and $\beta_\ell = c\,q_\ell$ for some $c \in [0,1]$ and $q^L \geq 0$ with $\sum_{\ell=1}^L q_\ell = 1$. There is no loss of generality in this choice since only the relative values of these parameters is meaningful [20, Lemma 4].

The next argument allows us to further restrict the space of $q^L$ vectors over which we must search, by first noting that $\beta_1 \geq \cdots \geq \beta_L \geq 0$, as follows.

Lagrangian coefficients must be non-negative [20], which applies both for coefficients associated with incremental rates as well as total rates. Thus $\alpha_\ell \geq 0$ and $\lambda_\ell \geq 0$ for all $\ell \in \{1,\ldots,L\}$, where the corresponding Lagrangian for total rates is

$$\begin{aligned} J &= \sum_{\ell=1}^{L} \left[ \alpha_\ell \left( D_\ell + \lambda_\ell\, R_\ell \right) \right] \\ &= \sum_{\ell=1}^{L} \left[ \alpha_\ell \left( D_\ell + \lambda_\ell \sum_{i=1}^{\ell} r_i \right) \right] \\ &= \sum_{\ell=1}^{L} \left[ \alpha_\ell\, D_\ell + \left( \sum_{i=\ell}^{L} \alpha_i\, \lambda_i \right) r_\ell \right]. \end{aligned} \tag{4.1}$$

Thus $\beta_\ell = c\,q_\ell = \sum_{i=\ell}^L \alpha_i\,\lambda_i$, where $\alpha_i \geq 0$ and $\lambda_i \geq 0$ for all $i$, imply that we need only consider $q^L \in \mathcal{Q}_L$, where $\mathcal{Q}_L = \{q^L : \sum_{\ell=1}^L q_\ell = 1 \ \wedge \ q_1 \geq \cdots \geq q_L \geq 0\}$.

Since the priority vector $p^L$ is given, it remains only to choose the value of $(c, q^L) \in [0,1] \times \mathcal{Q}_L$. A wide variety of techniques can be applied to search for an optimal $(c^\star, (q^L)^\star) \in [0,1] \times \mathcal{Q}_L$. The simple method that follows takes a bisection-style approach. We set $(\underline{c}_0, \overline{c}_0) = (0,1)$ and choose an initial value $(c_0, q_0^L)$ in some central location in the allowed region $[\underline{c}_0, \overline{c}_0] \times \mathcal{Q}_L$. For example, when $L = 3$, $\mathcal{Q}_L = \{q^3 : \sum_{\ell=1}^3 q_\ell = 1 \ \wedge \ q_1 \geq q_2 \geq q_3 \geq 0\}$, and we choose $(c_0, q_0^3) = (1/2, (11/18, 5/18, 2/18))$. At each time $t \geq 0$, we design a multi-resolution code for Lagrangian parameters $(\alpha^L, \beta^L) = ((1-c_t)p^L, c_t q_t^L)$, calculate the resulting performance $(r^L, D^L)$, and find $(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q_{t+1}^L)$ according to the following rules.

At time $t$, define $\mathcal{U} = \{\ell : r_\ell < r_\ell^\star - \varepsilon\}$ and $\mathcal{V} = \{\ell : r_\ell > r_\ell^\star + \varepsilon\}$, where $\varepsilon > 0$ describes a target margin of error (i.e., we are aiming for $r_\ell \in [r_\ell^\star - \varepsilon, r_\ell^\star + \varepsilon]$).†

†Asymmetrical error margins ($r_\ell \in [r_\ell^\star - \varepsilon, r_\ell^\star]$) and multiplicative error margins ($r_\ell \in [r_\ell^\star(1 -$

Codes for which both $\mathcal{U}$ and $\mathcal{V}$ are empty use the full available rate and thus do not have the potential to achieve a lower value of $\sum_{\ell=1}^{L} \alpha_\ell D_\ell$. In theory it will not always be possible to find a code with both $\mathcal{U}$ and $\mathcal{V}$ empty since we are restricting our attention to codes whose performance lies on the lower convex hull of achievable $(r^L, D^L)$ vectors. Experimentally, the set of points on the lower convex hull seems to be extremely rich for the sources considered here, and thus this problem has not been observed in practice. We therefore run the following iterative search procedure until at least $\mathcal{V}$ is empty.

- If $\mathcal{U}$ and $\mathcal{V}$ are both empty, then the procedure stops.
- Otherwise, if $\mathcal{V}$ is empty, then $(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q^L_{t+1}) = (\underline{c}_t, c_t, (\underline{c}_t + c_t)/2, q^L_t)$.
- Otherwise, if $\mathcal{U}$ is empty, then $(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q^L_{t+1}) = (c_t, \overline{c}_t, (c_t + \overline{c}_t)/2, q^L_t)$.
- Otherwise, we leave $\underline{c}$, $\overline{c}$, and $c$ unchanged ($(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}) = (\underline{c}_t, \overline{c}_t, c_t)$) and search the space $\mathcal{Q}_L$ of allowed $q^L$ vectors using the iterative approach described below. This procedure outputs a modified vector $q^L_{t+1} \neq q^L_t$ such that at least one of $\mathcal{U}$ and $\mathcal{V}$ is empty for parameters $(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q^L_{t+1})$.

Given a fixed $c$ and some initial $q^L_t \in \mathcal{Q}^L$ for which sets $\mathcal{U}$ and $\mathcal{V}$ are both nonempty, the procedure for searching the space $\mathcal{Q}_L$ of allowed $q^L$ vectors likewise uses an iterative approach. Since both $\mathcal{U}$ and $\mathcal{V}$ are nonempty, we rule out the subspace $\{q^L \in \mathcal{Q}_L : [q_i > q_{t,i} \forall i \in \mathcal{U}] \wedge [q_j < q_{t,j} \forall j \in \mathcal{V}]\}$, choose a central point in the region that remains, test the resulting rates, and continue the iterative procedure until achieving a point for which at least one of $\mathcal{U}$ and $\mathcal{V}$ is empty. In the procedure used for the experimental results section, the choice of a tentative value for $q^L_{t+1}$ given $q^L_t$ maintained the ratios $q_{t+1,i}/q_{t+1,j} = q_{t,i}/q_{t,j}$ for all $(i,j) \in (\mathcal{U} \times \mathcal{U}) \cup (\mathcal{V} \times \mathcal{V})$, giving

$$q_{t+1,\ell} = \begin{cases} a q_{t,\ell} & \text{for all } \ell \in \mathcal{U} \\ \left(1 + \frac{(1-a) q(\mathcal{U})}{q(\mathcal{V})}\right) q_{t,\ell} & \text{for all } \ell \in \mathcal{V} \\ q_{t,\ell} & \text{for all } \ell \in \mathcal{U}^c \cap \mathcal{V}^c, \end{cases}$$

$\varepsilon), r^\star_\ell(1+\varepsilon)])$ can be handled similarly.

where $q(\mathcal{U}) = \sum_{i \in \mathcal{U}} q_{t,i}$, $q(\mathcal{V}) = \sum_{j \in \mathcal{V}} q_{t,j}$, and $a < 1$ is the mid-point of the segment of values for which the resulting $q_{t+1,\ell}$ falls in the unsearched remaining region of $q^L$. By shrinking the sub-space of values that must be searched at each step, the algorithm narrows its way to a solution.

## 4.4 MRVQ Complexity

The MRVQ's nearest neighbor encoder chooses a complete path through the multi-resolution source coding tree in one operation rather than making a sequence of choices on a resolution-by-resolution basis. This has serious consequences with respect to the code complexity, since the code complexity of the nearest neighbor encoder for a multi-resolution source code is roughly equivalent to the complexity of a nearest neighbor single-resolution encoder with a codebook size equal to the number of nodes in the multi-resolution tree. For example, for a binary tree, the nearest neighbor encoder has complexity roughly twice that of a nearest neighbor encoder for a vector quantizer with the same (maximal) rate since the number of nodes in a binary tree is roughly twice the number of leaves in the tree. The complexity of nearest neighbor encoders on fixed-rate tree-structured codebooks grows exponentially in the dimension $n$ for any particular rate vector $R^L$, while the complexity of the TSVQ algorithm grows only linearly in $n$. Inspired by this discrepancy, we next present a new linear complexity variation of the MRVQ algorithm. (Another way to decrease the MRVQ's complexity while maintaining its high performance would be to apply the hierarchical VQ techniques of [51, 9] to perform an approximation to the optimal MRVQ encoder entirely through table look-up.) Experimental results for both the linear complexity code and the MRVQ algorithm appear in Section 4.5.

The linear complexity MRVQ combines a TSVQ encoder with an MRVQ decoder in an attempt to get performance as close as possible to that of the MRVQ at lower computational expense. An early version of this technique was first introduced in [21]. The aim of the approach is to bridge the performance gap between TSVQ and MRVQ, with a trade-off in complexity. The proposed method uses a multi-path search

instead of searching for the single best "greedy" path, as in TSVQ. The multi-path search is also "greedy" in the same sense as TSVQ, but allows searching for more than one path in a multi-resolution tree-structured codebook designed using the MRVQ design algorithm. For example, given a binary tree and a 2-path search, no decision is necessary at the first layer of the tree since there are only two possible paths. To choose the best paths at the next layer of the tree – with four nodes – we compare the weighted rate-distortion performance of the four possible paths, picking the two with lowest Lagrangian performance measure. At the following layer we choose the best two paths out of the four paths descending from the paths chosen in the previous layer, and so on. Finally, upon reaching the leaves at the last layer of the tree, the encoder chooses the best single path among the paths that remain at that point. The two extreme cases of multi-path search are one-path search, i.e., TSVQ, and all-path search, which is the MRVQ algorithm described above. In the multi-path scenario we also need to explicitly set the relative priorities at each step between resolutions, using the Lagrangian parameters $(\alpha^L, \beta^L)$.

## 4.5 Experimental Results

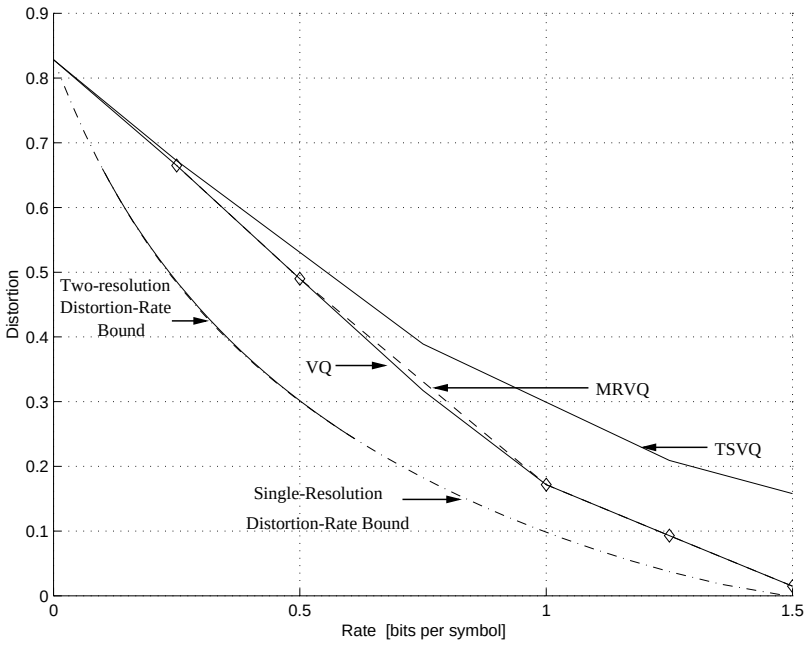
In this section we examine the empirical performance of fixed- and variable-rate multi-resolution codes designed using the MRVQ design algorithm. We compare the performance of these codes on synthetic data to both the theoretically optimal performance achievable on that source and the performance of alternative single- and multi-resolution vector quantizers. We also examine the convergence properties of the given codes as a function of the vector dimension $n$. On natural image sources (for which the optimal performance cannot be calculated) we compare the fixed- and variable-rate MRVQ performance to the performance of alternative single- and multi-resolution codes. In both cases, comparisons are restricted to vector quantizers of the same dimension $n$.

### 4.5.1 Synthetic Data

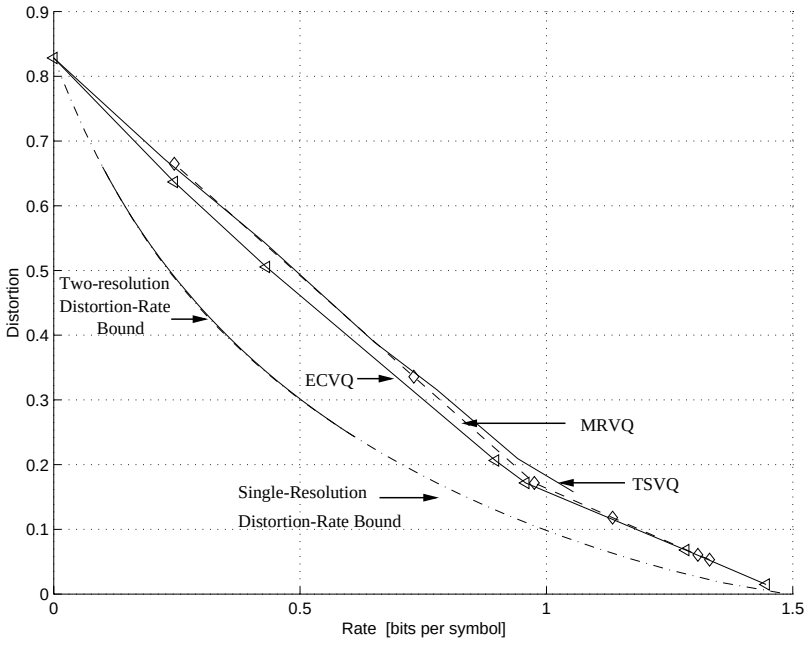
The synthetic data set consists of i.i.d. samples drawn according to the distribution $\mu = \{(1-p)/2, p, (1-p)/2\}$ on alphabet $\{1, 2, 3\}$ with $p = 0.171$. This Gerrish distribution is treated in [22, 23, 20]. We use half of the data samples for training and report results on the remaining half. The distortion measure considered is the absolute difference distortion measure, and all codes are optimized relative to this criterion.

While the theoretical results of [20] demonstrate that the penalty associated with using a multi-resolution code on the given three-symbol source is very small, those results treat only the asymptotic case, where the coding dimension $n$ is allowed to be arbitrarily large. The results of Figure 4.1 give empirical evidence suggesting that similar statements hold on this source even at very small coding dimensions. Figure 4.1 shows the performance of (a) fixed-rate and (b) variable-rate MRVQ of dimension $n = 4$. In this figure, each MRVQ curve shows the performance of a *collection* of MRVQ codes. Each MRVQ in the collection achieves first-resolution performance that is identical to the performance of the best available single-resolution code of the same rate. The codes differ from each other only in their second-resolution rates and distortions, and thus only the second-resolution (total) rates and distortions of these codes appear in Figure 4.1. In both the fixed- and the variable-rate examples, the MRVQ achieves second-resolution performance very near to the performance of the best single-resolution code of the same dimension. In both cases, the MRVQ gives better performance than a TSVQ of the same dimension.

The dimension-4 coding performance shown in Figure 4.1, is quite far from the optimal performance. Figure 4.2 demonstrates the performance improvements associated with higher coding dimensions. In both fixed- and variable-rate codes, increasing the coding dimension from 4 to 16 gives significant performance improvement. Further improvement could be obtained by increasing the dimension even more. The penalty for this improvement is an increase in computational complexity, as discussed in Section 4.4.

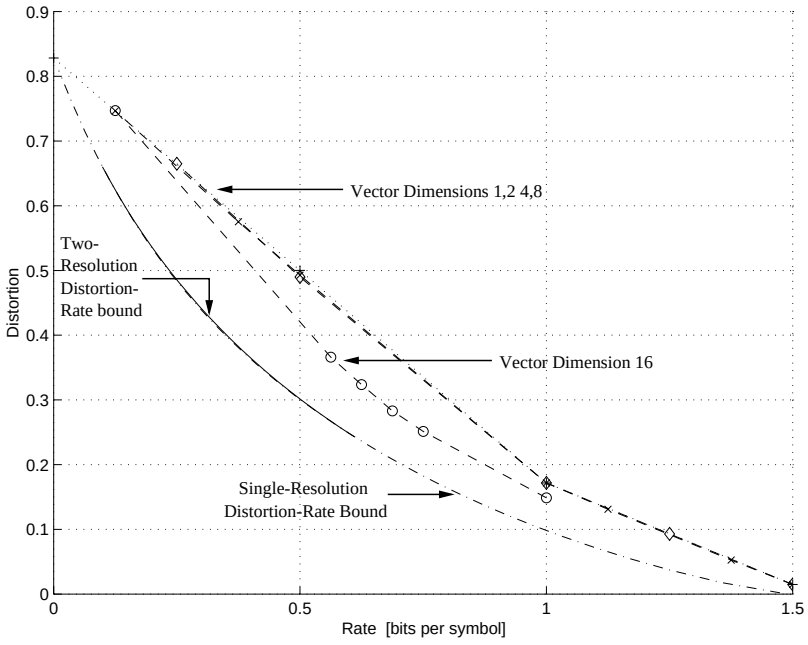



(a)

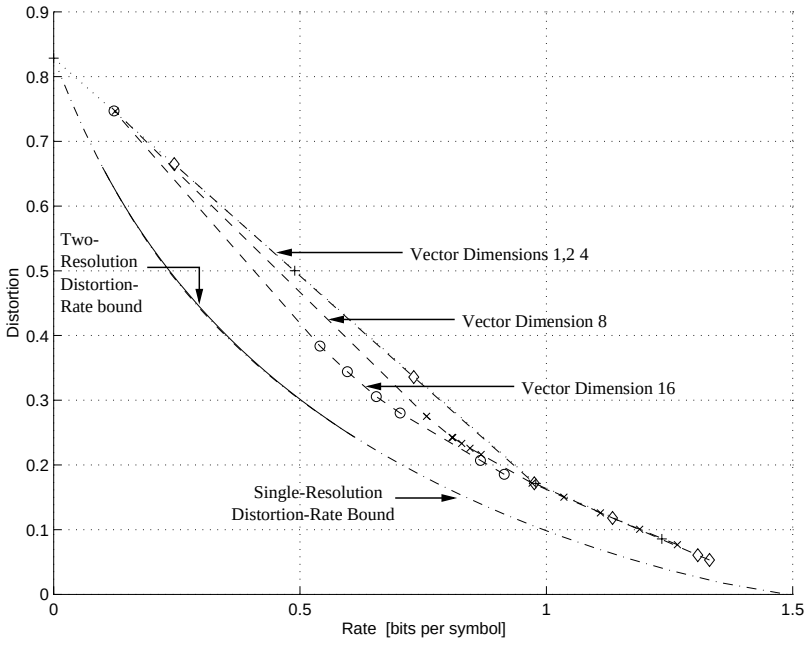



(b)

Figure 4.1: Comparison of (a) fixed-rate and (b) variable-rate MRVQ performance to the performance of prior multi-resolution codes (fixed- and variable-rate TSVQ), the best available single-resolution codes (fixed-rate VQ and ECVQ), and the theoretically optimal performance. All codes have vector dimension 4. The source is the synthetic data set. In each case, the MRVQ curve gives the second-resolution performance of a collection of MRVQs, each with the same first-resolution performance. The first-resolution MRVQ performance is identical to that of the best rate-0.25 VQ in the fixed-rate case and the best rate-0.246 ECVQ in the variable-rate case.




(a)




(b)

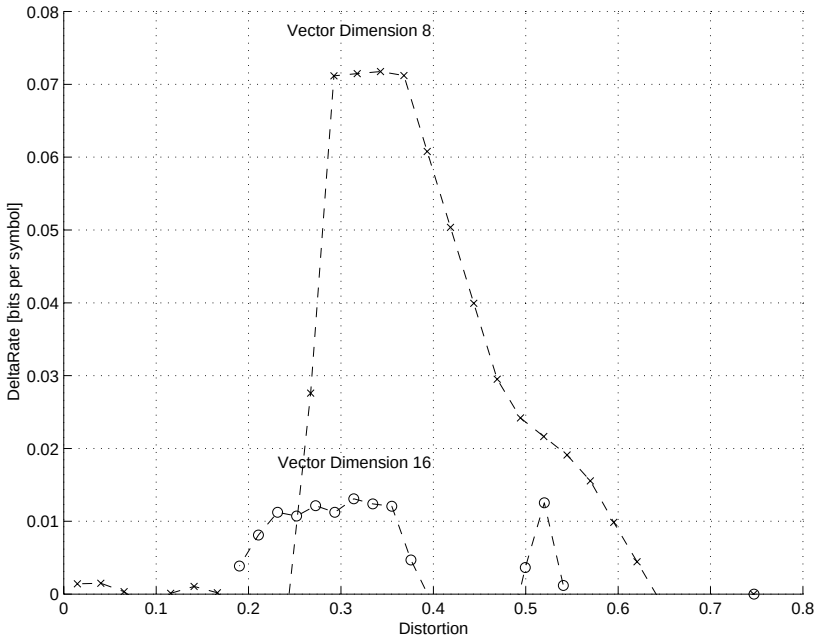
Figure 4.2: Second-resolution (a) fixed-rate MRVQ and (b) variable-rate MRVQ performances for vector dimensions 1, 2, 4, 8, and 16 on the synthetic data set.

Figure 4.3 characterizes, for several dimensions, the second-resolution rate penalty associated with multi-resolution coding. In this case, we constrain the first-resolution performance to be identical to that of the best corresponding single-resolution code and then measure the (total) rate needed at the second resolution to get a range of possible second-resolution distortion values. Each (total) second-resolution rate is then compared to the rate that would be required to get the same distortion with a single-resolution code of the same dimension. The difference is plotted in Figure 4.3. These experimental results are analogous to theoretical results given in [20]. The rate penalty varies both as a function of the first-resolution rate and as a function of the coding dimension.

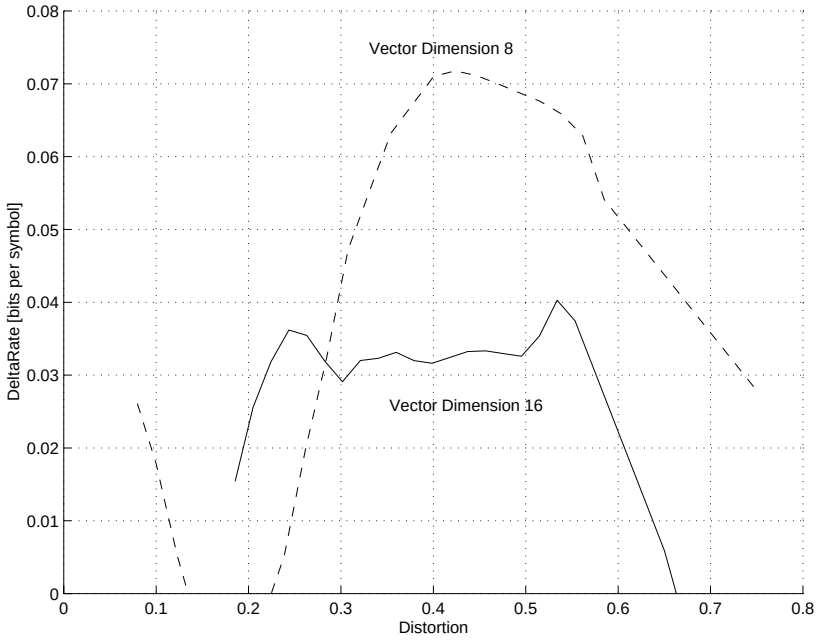
### 4.5.2 Natural Data

The natural data set consists of twenty $256 \times 256$ medical brain-scan images used for training, and five $256 \times 256$ brain-scan images used for testing. The training and test sets do not overlap. All experiments on the natural data set use the squared error distortion measure and coding dimension $n = 4$.

Figure 4.4 shows (a) fixed-rate and (b) variable-rate performance results of the MRVQ on the natural data set. Here performance is shown by plotting signal to quantization noise ratio (SQNR) as a function of rate. Each MRVQ curve represents the performance, at a sequence of resolutions, of a *single* multi-resolution code. The performances achieved with a variety of values for $(\alpha^L, \beta^L)$ are included to show a range of performances achievable with the given algorithm. These results were achieved using the method for choosing $(\alpha^L, \beta^L)$ described in Section 4.3. The MRVQ performance is compared both to the performance of a collection of single-resolution codes (VQs and ECVQs for the fixed- and variable-rate cases, respectively) and to the performance of the TSVQ algorithm. In both cases, the performance of the best single-resolution code of a given rate can be obtained exactly using a multi-resolution code if the priority at the appropriate resolution is made sufficiently high. The potential expense of this choice, however, is a degradation of the performance

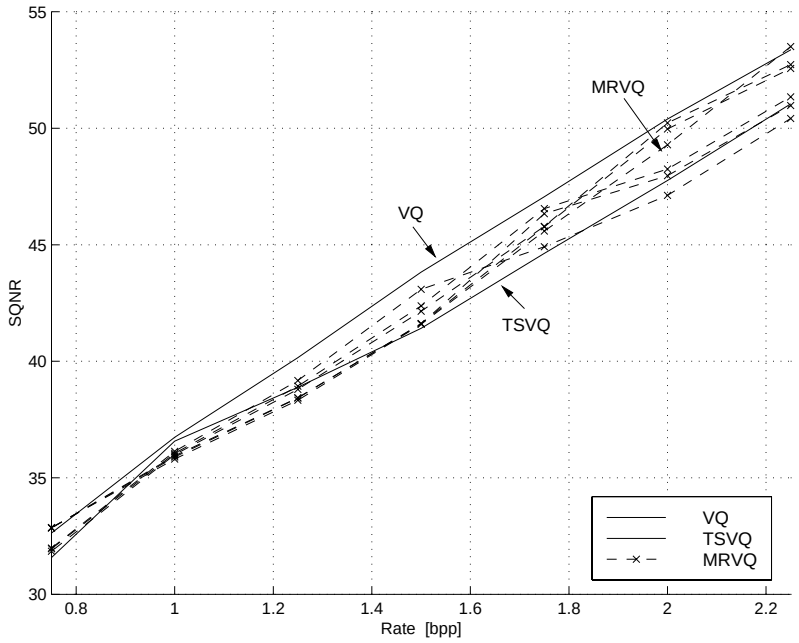



(a)

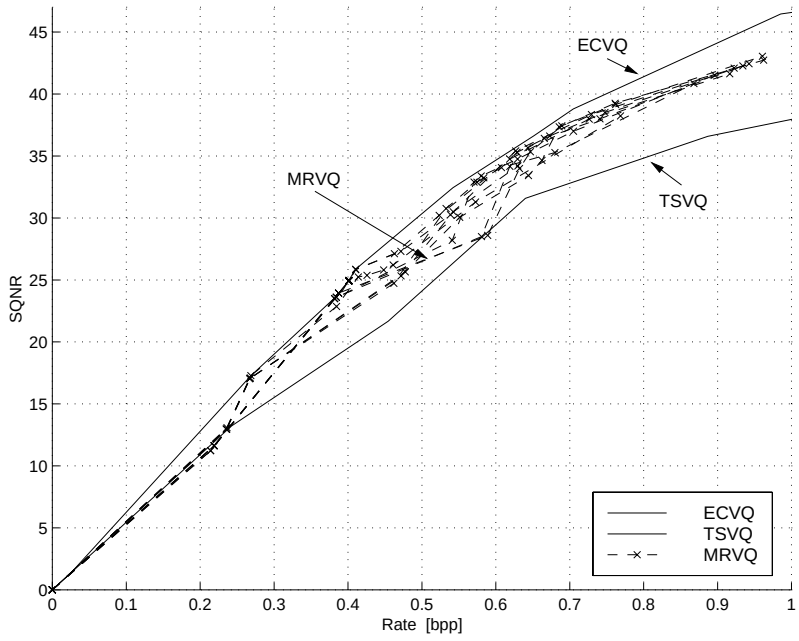



(b)

Figure 4.3: The second-resolution rate penalty for MRVQ on the synthetic data set, for vector dimensions 8 and 16. For each distortion value, the graphs show the difference between the rate required to achieve that distortion in the second resolution of an MRVQ and the rate required to achieve that distortion with a single-resolution code. In each case, the first-resolution of the MRVQ is constrained to achieve performance identical to that of the best corresponding single-resolution code at rate $\simeq 0.125$ bps. Graph (a) compares fixed-rate MRVQ to fixed-rate VQ. Graph (b) compares variable-rate MRVQ to ECVQ.




(a)




(b)

Figure 4.4: SQNR vs. rate results for (a) fixed-rate MRVQ, a collection of independent (non-embedded) fixed-rate VQ, and TSVQ and (b) variable-rate MRVQ, a collection of independent (non-embedded) ECVQs, and TSVQ. All results show performance on the medical image data set.

of the code at another resolution. Figure 4.4 includes examples both of cases where the MRVQ performance is set equal to the corresponding VQ or ECVQ performance at a given resolution – giving the best possible performance at the given resolution but causing performance degradation at other resolutions – and examples where the MRVQ performance is everywhere near but no where equal to the performance of the best single-resolution code. As expected, the MRVQ exceeds the performance of the TSVQ except, occasionally, at the lowest rates where the TSVQ's "greedy" strategy can give good performance. Figure 4.5 shows examples of compressed images from single- and multi-resolution codes.

### 4.5.3 Linear Complexity MRVQ

Figure 4.6 shows the peak signal-to-noise ratio (PSNR) as a function of rate, where PSNR $= 10 \log_{10}(255^2/\text{MSE})$ dB, for a collection of multi-path search MRVQ codes, with increasing complexity and improving performance, on the natural data set. The performances of all codes shown in the figure, correspond to fixed rate codes of vector dimension 4, and all use the same depth-9 initial tree-structured codebook. While 1-path MRVQ is equivalent to TSVQ in encoding complexity, the results given here are better since the code was not designed using the greedy TSVQ algorithm. In Figure 4.6, the results for MRVQ and all multi-path MRVQs use identical priorities.

## 4.6 Summary and Conclusions

This chapter presents a method for converting a collection of functional design constraints into Lagrangian parameters. The approach is a bisection search technique that approximates the values of the Lagrangian parameters $(\alpha^L, \beta^L)$ for given target rates and relative priorities over the resolutions. We also examine the empirical performance of vector quantizers for multi-resolution source coding and present an algorithm for $m$-path optimization, spanning from encoder complexity comparable to TSVQ, when $m = 1$, to MRVQ, when $m$ is equal to the number of leaves. The approach finds the best $m$ paths within a tree-structured multi-resolution codebook,

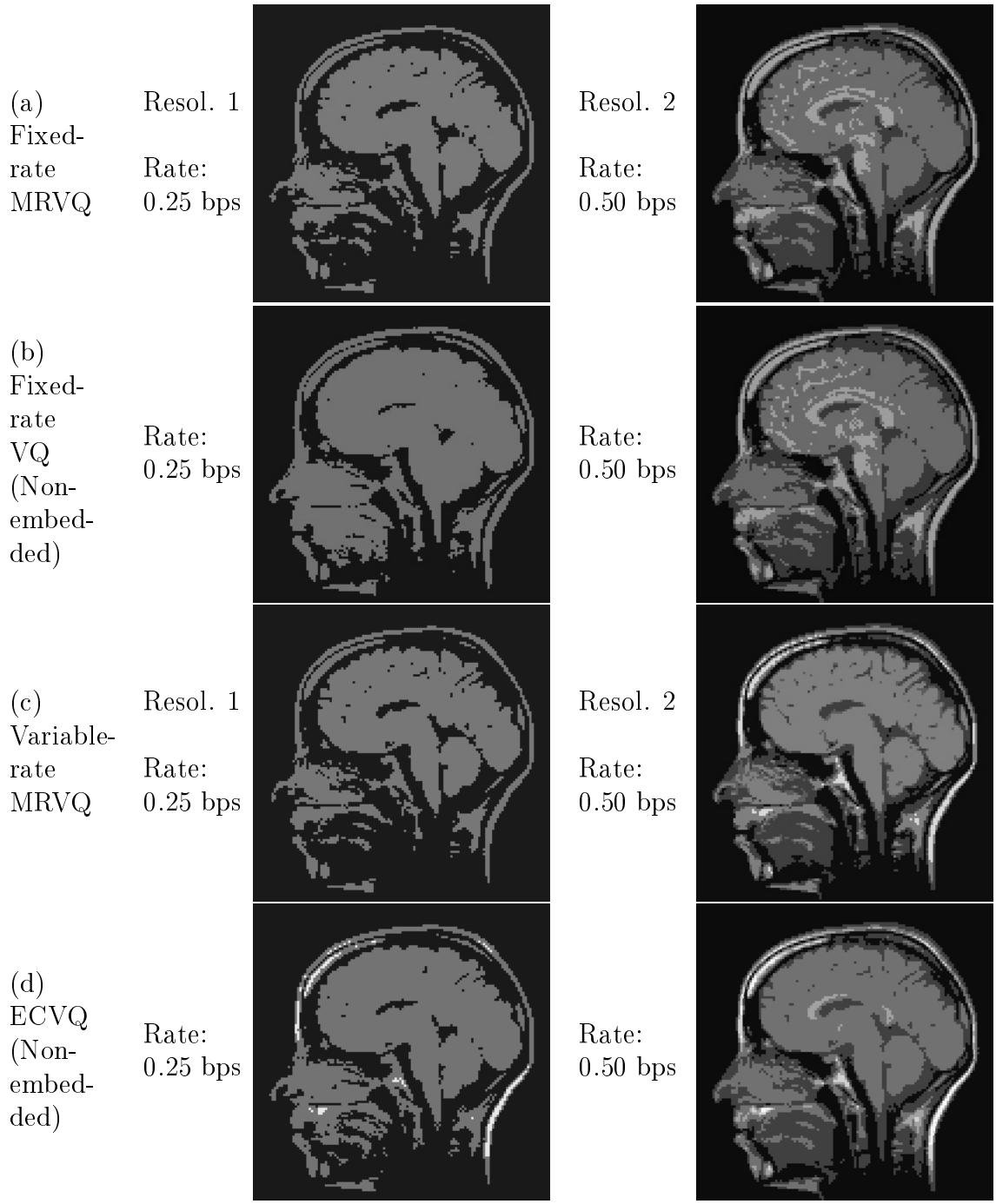



Figure 4.5: Images produced by (a) two resolutions of a single fixed-rate MRVQ, (b) two independent fixed-rate VQs, (c) two resolutions of a single variable-rate MRVQ, and (d) two independent ECVQs. In each case the same image from the test set is shown.

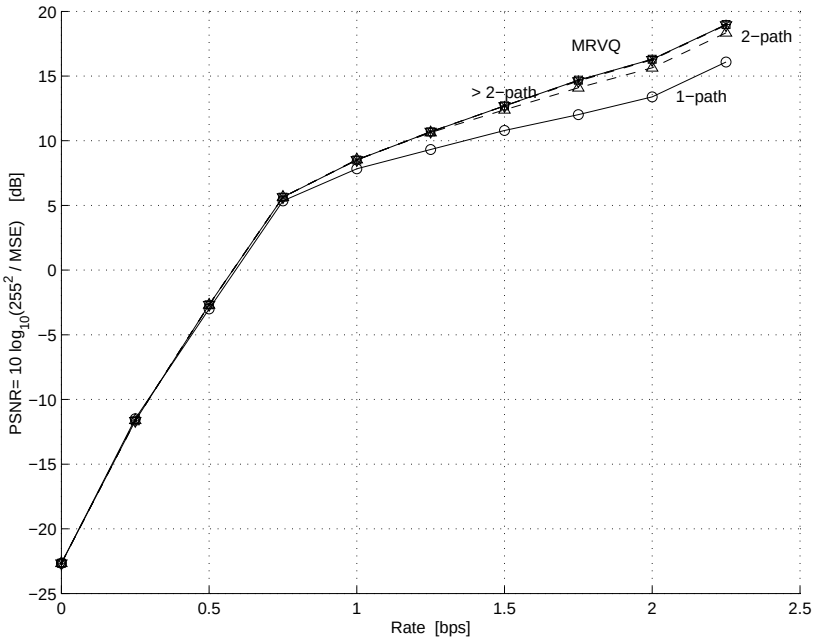



Figure 4.6: PSNR-rate performance of linear complexity MRVQ and MRVQ on the natural data set (brain-scans). On this data set and at this dimension, the linear complexity MRVQ achieves a performance close to that of an MRVQ with just a 2-path greedy search. The performance of the linear complexity MRVQ for 4-paths or more is almost identical to that of an MRVQ. All codes are fixed-rate codes of dimension 4.

previously trained with then MRVQ design algorithm. Experimental results on both synthetic and natural data sets demonstrate significant performance improvements of MRVQ over prior progressive scalar and vector quantizers. The use of the Lagrangian performance measure also yields increased flexibility in code design by allowing the user to design a code that reflects the underlying system priorities.

# Chapter 5 Optimization of Wavelet-based Multi-resolution codes

## 5.1 Introduction

The MRVQs described in Chapter 4, provide a framework for multi-resolution source coding that is theoretically optimal from a rate-distortion perspective. However, in practice optimal design is hard and theoretically optimal performance is only achieved asymptotically in the limit of large vector dimension and computational complexity.

Practical multi-resolution codes generally trade off rate-distortion performance with required complexity.

Recent wavelet-based codes [49, 48, 42, 41, 30, 31] provide computationally efficient multi-resolution algorithms that are not optimal from a rate-distortion perspective but have very good performances at remarkably low computational complexities. In particular, bit-plane coders working in the wavelet domain can exploit dependencies between wavelet coefficients [39, 40]. Several popular methods use heuristic techniques to take advantage of those coefficient dependencies, especially for natural images, achieving excellent rate-distortion performance at low computational expense.

Here we consider the application of a Lagrangian optimization technique to wavelet-based progressive transmission algorithms for image compression. The proposed methods generalize the techniques described in Chapter 4 to wavelet-based codes. In particular, the new techniques incorporate a user's relative priorities over the resolutions of a multi-resolution code into the code's optimization procedure. Prior wavelet-based codes give no way to prioritize over the resolutions. The proposed new algorithms improve on the Set Partitioning in Hierarchical Trees (SPIHT) algorithm [48] and the Group Testing for Wavelets (GTW) algorithm [30, 31]. Some of the results presented here were reported in [17] and [18]. The resulting optimized algo-

rithms achieve gains over SPIHT and GTW at the resolutions of highest interest. In [18], the proposed approach incorporates an adaptive wavelet transform in place of the standard wavelet transform.

## 5.2 Background

### 5.2.1 Set Partitioning in Hierarchical Trees Algorithm (SPIHT)

The SPIHT image compression algorithm [48] is a bit-plane coder working in the wavelet domain. The wavelet coefficients of an image are described one bit plane at a time in order of decreasing significance. The SPIHT algorithm begins with an initial partition of the coefficients into sets. At each bit-plane, it describes the significance or insignificance of the current sets. A set is declared significant at bit-plane $\ell$ if it contains one or more coefficients $C(i,j)$ that are declared significant in that bit-plane; otherwise the set is declared insignificant. A coefficient is significant at bit-plane $\ell$ if the first non-zero bit of its binary description occurs in the $\ell$th most significant position. Each significant set is then partitioned into subsets, and the significance or insignificance of each subset is likewise described. The process continues until each significant subset has exactly one coefficient. The algorithm describes the sign of each newly significant coefficient and then refines the descriptions for all coefficients declared significant in previous bit-planes; the refinement procedure involves describing the $\ell$-th bit in the binary expansion of each such coefficient. The above procedure then repeats using the current sets at bit-plane $\ell - 1$.

Figure 5.1 shows the SPIHT zero-tree on the wavelet pyramid and the corresponding partition hierarchy showing the sets and component subsets used in the SPIHT algorithm. Each leaf node $(i,j)$ of the partition hierarchy represents a single wavelet coefficient $C(i,j)$. Each intermediate node is a set of coefficients used in the partition. The children of each intermediate node give the subsets into which the given set is partitioned when the set becomes significant. The intermediate nodes are labeled according to the convention established by Said and Pearlman [48]. The set $\mathcal{D}(i,j)$

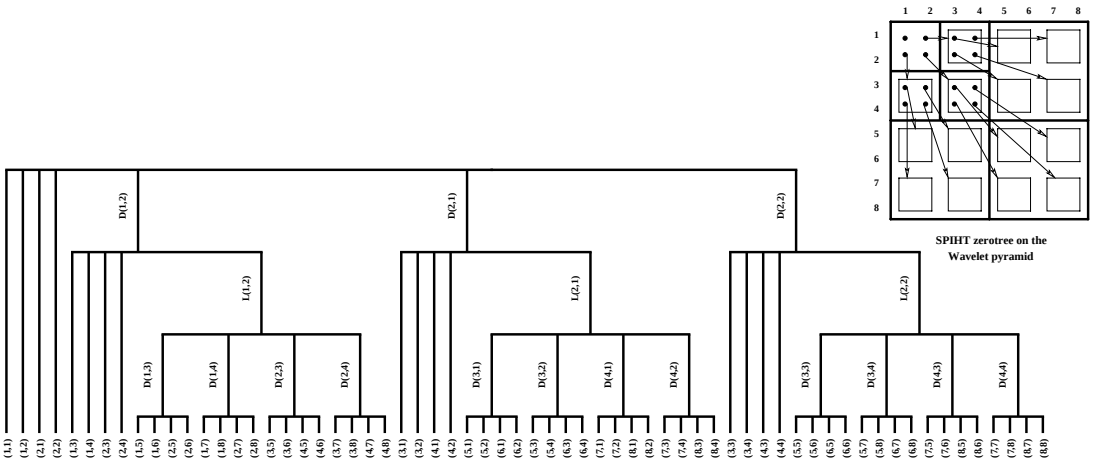



Figure 5.1: SPIHT's decision-tree for a 2-level wavelet decomposition of an 8 x 8 image.

corresponds to the set of "all descendants" of node $(i, j)$ (*all* descendants, including offspring), while the set $\mathcal{L}(i, j)$ is the set of "all descendants-except-children" of node $(i, j)$.

The partition used in the SPIHT algorithm groups together low- and high-frequency coefficients from the same spatial location in the original image. The algorithm is most efficient in images where insignificance of low-frequency coefficients is a good predictor of insignificance in the corresponding coefficients of higher frequency bands. The algorithm is least efficient when a scattering of significant coefficients in the high-frequency bands causes the original partition to be broken into many subsets in order to describe a small number of significant coefficients.

### 5.2.2 Group Testing for Wavelets Algorithm (GTW)

Like SPIHT, GTW is a bit-plane coder. The encoding requires two consecutive passes per bit-plane. First, the significance pass codes all coefficients that were not identified as significant in a prior bit-plane. Second, the refinement pass for a given bit-plane encodes the next bit in the description of those coefficients that were declared significant in prior bit-planes. GTW's refinement pass is the same as SPIHT's, but its significance pass is different.

In GTW the significance pass involves dividing the bits within each bit-plane into classes and then performing a group-testing algorithm on groups within each class. In this case, the groups are not restricted to zero-trees, as in EZW [49] or SPIHT [48], providing a more general framework where groups can be composed of any subset of coefficients.

At each bit-plane, the bit of the binary decomposition of every coefficient is sorted into a class according to three properties: the *sub-band level*, the *pattern type*, and the *significant neighbor metric* (SNM). The level describes the coefficient's position in the wavelet decomposition. The pattern is a function of the coefficient's location, with neighboring coefficients given different pattern values. The SNM describes the number of neighbors that were declared significant at previous bit-planes or in the same bit-plane but before the given coefficient. The classes are ordered first by SNM, then by pattern type, and lastly by level, and the algorithm encodes all classes in order.

The encoding of each class involves sequentially breaking off a group from the class, describing the contents of that group to the decoder, and using those contents to determine the size of the next group. The first group in each class contains one element. The number of elements (i.e., bits) per group doubles with each subsequent group until the encoder describes a group that has at least one non-zero bit. Subsequent group sizes are based on the empirical probability of a zero in the preceding bits in the given class. In particular, the size of each subsequent group equals the unique integer $k$ that satisfies the inequality

$$q^k + q^{k+1} \leq 1 < q^k + q^{k-1}, \tag{5.1}$$

where $q$ is the empirical probability of a zero when the group size is chosen [28, 30, 31].

The first bit of any group description describes whether or not the group is significant. If the group is not significant, then the group description is complete. If the group is significant, then it is split into two parts. Each part is subsequently tested and, if found significant, split again. The process repeats until the first significant

element is found within that group. Once a significant element is found, the code updates $q$ and $k$ and defines the next group to be the $k$ elements immediately following that significant bit. The process repeats on the subsequent groups within the class until every bit in the class has been described or the number of bits remaining is smaller than the current value of $k$. Remaining bits join a list of untested items and the algorithm proceeds with bits from the next class.

### 5.2.3 Essentially Non-oscillatory (ENO) Wavelets

The adaptive ENO wavelet transform [10, 11] reduces the ringing effects of image edge discontinuities by modifying the wavelet coefficients. Performing an ENO wavelet transform involves first performing a standard wavelet transform, then locating the image discontinuities, and finally modifying the coefficients around the discontinuity. Since ENO's overhead in computational complexity is low, ENO retains the advantages of standard wavelets while removing edge artifacts. In particular, ENO provides a multi-resolution decomposition with good energy compaction and uniform high-order accuracy with reduced ringing and smearing effects at region boundaries [10].

For smooth regions of an image, the ENO transform generates the same wavelet coefficients as the corresponding standard wavelet transform. For edge-regions, ENO performs independent transforms on the two sides of each marked discontinuity, using one-sided extrapolation to smoothly extend the data at each side of the edge.* In order to have the same number of coefficients as in the standard wavelet transform, the one-sided extension from the left-hand side of a discontinuity is used to calculate the high-frequency ENO wavelet coefficients, using the extrapolation of coarser-level low-frequency coefficients. Similarly, from the right-hand side of a discontinuity the low-frequency coefficients are calculated using the extension of high-frequency coefficients from a coarser sub-band level.

The result is a reduction in the magnitude of the high-frequency coefficients rela-

*In general, there are several ways to identify an edge. ENO classifies discontinuities by comparing the magnitudes of high-frequency coefficients.

tive to those of the standard wavelet transform. This property is quite useful for embedded zero-tree coding algorithms. By concentrating most of the larger coefficients at the top of the wavelet sub-band pyramid, it allows for a rate-efficient description of the coefficients.

By adjusting the minimum threshold at which a difference between neighboring coefficients is recognized as a discontinuity, ENO provides the flexibility to go from no edge detection (and therefore no adaptation) to maximum edge sensitivity. ENO indicates each detected edge in a binary map of the same size as the image. This map is needed to make the transform reversible. Increasing the edge sensitivity increases the number of non-zero values in the ENO map and, in most cases, also the corresponding description cost. If no ENO decisions are made because no edges are detected, the mapping is all zeros, and can be described at almost no cost. Figure 5.5 shows an example of an ENO edge map for a two-level ENO wavelet transform.

### 5.2.4 Multi-resolution Lagrangian and Coefficient Modifications

The multi-resolution Lagrangian performance measure can be applied in general to a multi-resolution code with an arbitrary number of resolutions. Here we define the number of resolutions as the number of significance levels or coefficient bit-planes for the optimization of wavelet-based bit-plane coders. Using this approach, we define the incremental rate $r_\ell$ as the rate used in describing the $\ell$-th most significant bit-plane, and $D_\ell$ as the MSE of the reproduction achieved through the description of the first $\ell$ bit-planes[†]. Thus the number of resolutions in our code design equals the number of bit-planes in our wavelet calculation. However, the number of resolutions can be expanded, for example, as proposed in [37], increasing the flexibility from the user's point of view as well as the coding complexity from the designer's perspective.

The value of the rate $r_\ell$ spent at resolution $\ell$ and the value of the distortion $D_\ell$ achieved after the description of the first $\ell$ resolutions of a particular image are a

[†]Equivalently, we can also calculate the multi-resolution Lagrangian as a function of the total distortion $D_\ell$ and the *total* rate $R_\ell$, defined as the cumulative rate at end of the $\ell$-th bit-plane.

direct function of the image coefficients. Let $C(i,j)$ denote the wavelet coefficient at spatial coordinates $(i,j)$. We calculate a multi-resolution Lagrangian for each coefficient $C(i,j)$ at every resolution $s \in \{1,...,L\} \cup \{\infty\}$ at which $C(i,j)$ *might be* declared significant, where the case of $s = \infty$ refers to the scenario where $C(i,j)$ is never declared significant.

While describing significance information accurately is necessary if one intends to continue the bit-stream decoding to a lossless description of the wavelet coefficients, the decisions resulting from this desire for high-rate accuracy come at a cost. Our goal is to find the modification to the wavelet coefficient magnitude that yields the best rate-distortion trade-off, subject to the Lagrangian constraints that reflect the priorities over the resolutions. The modifications of wavelet coefficients result in a decrease or increase in the "wavelet-domain" MSE at each resolution and translate to an equivalent decrease or increase in the "image-domain." That is because the Euclidean norm is invariant to unitary hierarchical sub-band transformations, as noted in [48].

Coefficient modifications have the purpose of changing the bit-plane at which a coefficient would be considered significant. These changes cause changes in both $r_\ell$ and $D_\ell$. The goal is to find the choices that minimize the Lagrangian functional. This approach was introduced in [17] and later applied in [37]. Alterations of the sign or refinement information of a coefficient (once the optimal significance plane was determined) can only make the reconstruction worse, and therefore are not considered. However, there are potential performance improvements using more elaborate schemes for the sign information, as demonstrated, for example, in [15]. Credit is due to [37] for indicating that in some cases the minimum Lagrangian is obtained by increasing the magnitude of wavelet coefficients (thus declaring them significant at an "earlier", more-significant, bit-plane). That is, the combination of decreasing or mantaining original values is not always the best approach, as initially proposed in [17].

The resolution $s$ at which we declare $C(i,j)$ significant can be the true significance resolution, a prior resolution, or a later resolution. Finding the Lagrangian

performance

$$\tilde{J}_s(i,j) = \sum_{\ell=1}^{L} [\alpha_\ell D_{s,\ell}(i,j) + \beta_\ell \; r_{s,\ell}(i,j)] \tag{5.2}$$

associated with declaring $C(i,j)$ significant at resolution $s$ requires the calculation of $L$ distortions and rates.

For any $\ell \in \{1, ..., L\}$, the distortion $D_{s,\ell}(i,j)$ equals the squared error between $C(i,j)$ and the resolution-$\ell$ reconstruction of the optimal reproduction for coefficient $C(i,j)$ and significance resolution (or level) $s$. If the true significance level of coefficient $C(i,j)$ equals $s^*$ and $s < s^*$, (that is, $C(i,j)$ is declared to be significant before it actually becomes significant), then the optimal reproduction is the smallest value with significance level $s$. If $s > s^*$, then the optimal reproduction is the largest value with significance level $s$. If $s = s^*$, then the optimal reproduction equals $C(i,j)$.

For example, if a wavelet coefficient is $C = 29$, its binary decomposition in a 7-bit-plane scenario would be 0011101. (We here list bits from most to least significant.) The value $C = 29$ becomes significant in the third bit-plane, i.e., $s^* = 3$. In some cases, though, it might be beneficial from a rate-savings perspective, to declare the coefficient significant at a later bit-plane, for example at $s = 4 > s^*$, giving a range of possible values for the "modified" coefficient, from $8 \equiv (0001000)_2$ to $15 \equiv (0001111)_2$. Since the optimization aims to reduce the weighted sum of rates and distortions, the best choice is the closest value to the original, making 15 the optimal value. On the other hand, if we declare the coefficient significant at resolution $s = 2$ (earlier than the original) then the optimal choice is the smallest value that becomes significant at resolution 2, namely $(0100000)_2 = 32$.

## 5.3 Optimizing SPIHT (Opt-SPIHT)

Opt-SPIHT optimizes the choice of the resolution at which each set of coefficients is declared significant. The significance decision on each coefficient or set of coefficients may affect decisions on other coefficients or sets; thus the optimal decisions must be found in a global fashion.

The Opt-SPIHT algorithm uses dynamic programming on the partition tree, shown in Figure 5.1, to find the collection of significance decisions that globally minimizes the multi-resolution Lagrangian. The dynamic programming algorithm is accomplished in two passes – a backward pass from the leaves to the root of the tree and a forward pass from the root to the leaves. The goal of the backward pass is to calculate and store an array of $\tilde{J}_s$-values, one for every resolution at each node of the tree. The goal of the forward pass is to encode the data using the best significance decisions, based on the values from the backward pass.

The sequence of decisions of set significance in the optimized algorithm is the same as the sequence of decisions in SPIHT. That is, at each resolution the algorithm decides whether or not the sets in its current partition are significant, partitions the significant sets into subsets, and makes decisions about the significance or insignificance of those subsets as well – continuing the process until all significant coefficients are isolated. The difference between the SPIHT algorithm and the approach described here is that the decisions themselves are based on a different criterion. In SPIHT, a set becomes significant at resolution $\ell$ if it contains at least one element with a one in bit-plane $\ell$ and no elements have ones in earlier bit-planes. In Opt-SPIHT, a set becomes significant at resolution $\ell$ if declaring it significant at that resolution yields the best possible multi-resolution rate-distortion trade-off.

The two passes are described in greater detail below.

### 5.3.1 Backward Pass (from the leaves to the root)

At every node, starting at the leaves and working toward the root, calculate a number of partial $\tilde{J}_s$-values equal to the number of resolutions in the data description. Next, store the calculated values, one for each resolution at each node, in a table to be used later in the algorithm.

As shown in Figure 5.1, each node of the tree is either a single coefficient $C(i,j)$ (a leaf node) or a set of coefficients $\mathcal{D}(i,j)$, $\mathcal{L}(i,j)$, or the tree root (an internal node). For each leaf node $(i,j)$ and each resolution $s$, we calculate a partial multi-resolution

Lagrangian $\tilde{J}_s(i,j)$, which requires calculation of $L$ distortions $D_{s,1}(i,j),\ldots,D_{s,L}(i,j)$ and $L-s+1$ rates $r_{s,s}(i,j), r_{s,s+1}(i,j), ..., r_{s,L}(i,j)$ associated with declaring coefficient $C(i,j)$ significant at resolution $s$. Notice that the remaining incremental rates $r_{s,1}(i,j), ..., r_{s,s-1}(i,j)$ associated with coefficient $C(i,j)$ becoming significant at resolution $s$ depend on the resolution at which the parent node (in the tree of Figure 5.1) becomes significant. Because the backward pass goes from leaves toward the root of the tree, such decision on the parent node has not yet been made. Furthermore, we need to calculate for each coefficient (or group of coefficients) the partial multi-resolution Lagrangian for *every one* of the resolutions at which it could become significant.

Finding the best performance for each internal node and each resolution requires making optimal decisions on the significance of that node's children. These optimal decisions are accomplished through the use of the $J$-approximations 5.2 previously calculated for those children and the addition of previously omitted rate values. (These rate values become available once an assumption is made about the resolution at which the parent node becomes significant.)

In particular for any internal node $p$ and any resolution $s$, the aggregate $\tilde{J}_s$-value is the weighted sum of the rate required to declare the given node significant at resolution $s$ plus a collection of values associated with the node's children. Namely, if $\mathcal{C}$ is the set of all (internal or leaf) children of internal node $p$, then

$$\tilde{J}_s(p) = \sum_{c\in\mathcal{C}} \min_{n\geq s}[\tilde{J}_n(c) + \sum_{\ell=s}^{n-1} \beta_\ell r_{s,\ell}(c)].$$

Since the calculation of $\tilde{J}_s$-values begins at the leaves of the tree and works up, layer by layer, to the root, $\tilde{J}_n(c)$ for all $n \in 1, ..., L$ and all $c \in \mathcal{C}$ are available prior to the calculation of $\tilde{J}_s(p)$. Further, the incremental rates $r_{s,s}(c), ..., r_{s,n-1}(c)$ achieved when $p$ becomes significant at resolution $s$ and $c$ becomes significant at resolution $n$ are now known. The resulting minimized values $(\tilde{J}_1(p), ..., \tilde{J}_L(p))$ are stored in the tree, and the resolutions used at the node's children to achieve the given minima are also stored. The process is continued, layer by layer, up to the root of the tree.

Notice that at the outcome of the above procedure, the $L$ Lagrangian values stored at the root of the tree are complete rather than partial values since the tree root has no unknown parent from which to inherit uncertainty. Further, note that once the $\tilde{J}_s$ values for a given layer of the tree have been calculated, the $\tilde{J}_s$-values from the previous (deeper) level are no longer required. As a result, the data structure that originally stored $\tilde{J}_s$ values may be written over with resolution choices as the process works from leaves to root through the tree, allowing a more efficient memory utilization. Finally, notice that while the minimizations performed at each internal node rely on partial $J$-values, the unknown rate values at each calculation are constant for all terms in the minimization and thus do not affect its outcome.

### 5.3.2 Forward Pass (from the root to the leaves)

The result of the backward pass is a table with a number of entries equal to the number of nodes in the tree multiplied by the number of resolutions in the original SPIHT description. The values $J_1(root), ..., J_L(root)$ given at the root of the tree describe the optimal Lagrangian performance achievable if the coding process begins at resolutions *1* through $L$, respectively. The remainder of the table is filled with the decisions used in achieving a particular $J$-value. For example, entries *1* through $L$ associated with some child $c$ of the root node give the resolutions at which child $c$ should become significant if the root node becomes significant at resolutions *1* through $L$, respectively. The forward pass uses these values by first comparing the values $J_1(root), ..., J_L(root)$ and choosing the value of $\ell = s$ that gives the best Lagrangian performance. The corresponding $\ell$-value, here called $s^*$, describes the first bit-plane to be used in the data description, effectively setting the total number of bit-planes to be used by SPIHT.

Starting at the top of the tree, the optimal resolution at which each of the children should become significant can be directly determined by reading entry $s^*$ for each of the root's children. Reading the corresponding information for each of those nodes' children and so on down the tree gives the optimal significance levels for all sets in the

tree. We then encode the data using the SPIHT algorithm, but replacing SPIHT's original decisions about set significance with the optimal collection of decisions.

### 5.3.3 Complexity of Opt-SPIHT

The flexibility benefits of Opt-SPIHT come at the cost of higher computational complexity and greater memory requirements than those needed for the original SPIHT algorithm. The added computation is associated with the calculation of the Lagrangian performance functions during the backward pass, plus the subsequent sequence of decisions. The additional memory is required for the storage of the Lagrangian values (and the resolution choices that are written over them later). Each node of the tree requires as many Lagrangian calculations as the total number of resolutions (or bit-planes). Each calculation uses all Lagrangians of the descendants of that node, making the number of calculations or comparisons for any node proportional to $L^2$, the square of the total number of resolutions used in coding. Depending on the total number of wavelet decomposition levels, $T$, the total number of nodes in the tree is at most 1.3125 times the image size $Z$. The derivation of this bound relies on the structure of the tree shown in Figure 5.1 as described next.

The number of leaf-nodes in the tree is the number of wavelet coefficients, which is equal to $Z$, the number of pixels in the input image. All other nodes are internal nodes of the tree, and correspond to either a $\mathcal{D}(i,j)$ set or an $\mathcal{L}(i,j)$ set.

In the wavelet coefficient pyramid, only 25% of the coefficients have descendants, and only nodes with descendants have associated $\mathcal{D}(i,j)$ set. Of those, 75% are *leaf-parents*, that is, nodes whose descendants in the tree are coefficients. While these *leaf-parent* nodes have an associated $\mathcal{D}(i,j)$ set (for the direct descendants), they have no $\mathcal{L}(i,j)$ set because they have no grandchildren. Thus, the number of $\mathcal{D}$(*leaf-parent*) sets in the decision-tree is $\frac{3}{4} \times \frac{1}{4}Z = \frac{3}{16}Z$.

All other coefficients, that are neither at the bottom of the wavelet decomposition pyramid nor the parents of those, have associated both a $\mathcal{D}(i,j)$ set and an $\mathcal{L}(i,j)$, as long as they have children and grandchildren, respectively. (Notice that

SPIHT convention implies that some of the coefficients at the top sub-band of the wavelet pyramid have no descendants, therefore have only leaf-nodes in the decision tree.) There are $\left[\frac{1}{16} - \left(\frac{1}{4}\right)^T\right] \times Z$ coefficients with both children and grandchildren in the wavelet pyramid. Therefore, these coefficients contribute that number of $\mathcal{D}(i,j)$ sets, as well as the same number of $\mathcal{L}(i,j)$-set entries to the decomposition tree of Figure 5.1.

The number of nodes in the tree, and therefore the number of Lagrangian calculations, increases as the total number of wavelet decomposition levels increases. In the limit of an infinite number of decomposition levels, the maximum possible number of nodes in the decision tree is

$$\text{Number of Nodes} = Z + \frac{3}{16}Z + \frac{2}{16}Z = \frac{21}{16}Z = 1.3125Z\ . \tag{5.3}$$

Thus, the total number of Lagrangians for the complete tree is, at most, $1.3125\,Z\,L^2$.

### 5.3.4 Experimental Results

Here we present experimental results showing the performance of Opt-SPIHT compared with the standard SPIHT algorithm. For the experiments presented here, in all cases we use the same 9-7 tap filters used in [1] for the SPIHT algorithm.

We use an implementation of SPIHT that follows Said and Pearlman's original paper [48], without any modifications or additions. No entropy coding is employed. The detailed description of SPIHT in Said and Pearlman's patent as well as some existing SPIHT software use techniques not specifically mentioned in the original paper. While the performance of both SPIHT and our method could be improved using those techniques, our results correspond to a version of SPIHT that strictly follows the cited paper.

Figure 5.2 compares the performance of the algorithm described here to the performance of SPIHT on a 512 pixel by 512 pixel, 8 bit per pixel gray-scale image scanned from a page of the *IEEE Spectrum Magazine*. The image contains both photographic material and text.

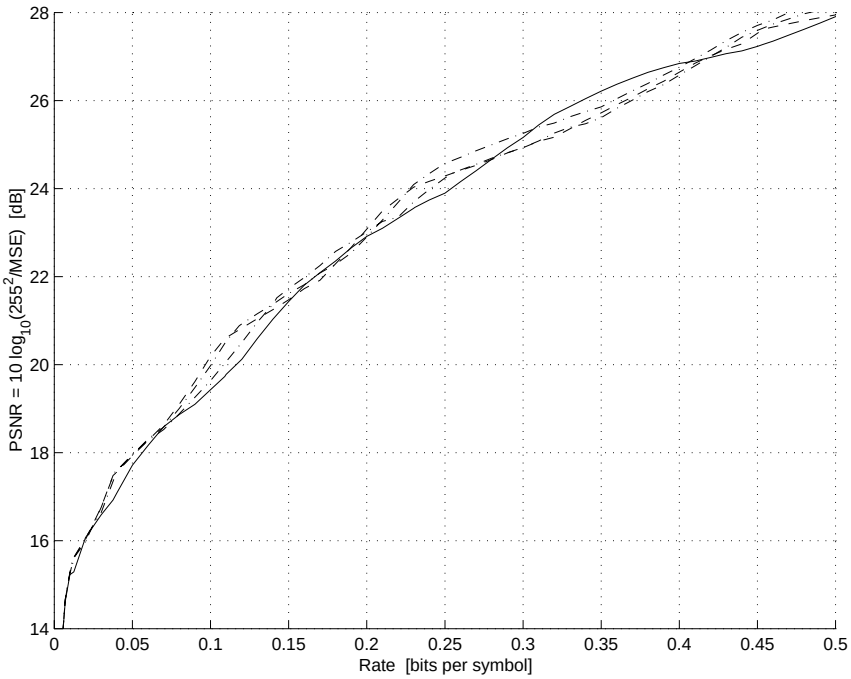



Figure 5.2: PSNR versus rate results fro Opt-SPIHT (dashed lines) using three different priority schedules and SPIHT (solid line), for a $512 \times 512$, 8 bits per pixel gray-scale image.

Figure 5.2 shows the PSNR as a function of rate for both the optimized code and the original SPIHT algorithm. Opt-SPIHT is optimized relative to three different priority functions to show a spectrum of achievable results. By changing the priorities, we achieve better performance at the resolutions of highest priority at the expense of a degradation in performance at low priority resolutions. The advantage of the new method is the algorithm's flexibility, which allows the user to explicitly set the priorities in a manner that reflects the relative importance of the different resolutions. This contrasts with the SPIHT algorithm's implicit prioritization over the resolutions, which places priority on the seldom-used highest rate description. Figure 5.2 shows gains of up to 0.86 dB for the priority functions tested.

## 5.4 Improving GTW (Opt-GTW)

GTW modifies the significance pass of SPIHT. It groups coefficients into classes, and performs group tests on them for significance, where the rules for those groups are different from descendant-based rules used in SPIHT.

To improve on GTW's rate-distortion performance, we modify the wavelet coefficients, thus the composition of the classes that result is a modification in the composition of the groups to be tested. Such modifications are those that minimize an approximated multi-resolution Lagrangian performance measure[18].

Opt-SPIHT's low complexity, globally optimal, dynamic programming solution to the multi-resolution Lagrangian minimization has no natural analog in the GTW framework, where a true multi-resolution Lagrangian optimization is computationally prohibitive.

Roughly, the problem arises from a combination of the class definitions and the empirical distributions used in GTW. In particular, neither of these items can be easily calculated without running the algorithm in full, and both can vary enormously with small changes in the coefficients. As a result, we cannot afford to do global optimization of the multi-resolution Lagrangian over the set of allowed coefficient modifications. Instead, we estimate the class index and corresponding empirical

probability used to code each coefficient, and we make the coefficient modifications that minimize the resulting approximation of the expected value of the Lagrangian. In calculating these approximations, we trade off performance and complexity. The resulting algorithm does not guarantee optimal rate-distortion performance within the GTW framework but does provide a principled method for improving on the existing performance at reasonable computational expense. Details of the Lagrangian approximation techniques follow.

### 5.4.1 Probability and Group-size Estimation

GTW codes symbols in groups of size $k$, where $k$ is a function of the current empirical probability $q$ as described in (5.1). To estimate the expected rate and distortion of our code, we first estimate the members of each group and the corresponding $q$ and $k$ values. Since calculating $q$ and $k$ effectively requires running the full GTW algorithm and since those values can change significantly with coefficient modifications, precise calculation of the $q$ and $k$ values for every possible combination of coefficient modifications is impractical. We therefore estimate $q$ and $k$ for each class using a procedure designed to do well on average across many coefficients.

The significance probability in a given class depends on the proportion of ones and zeros in that class. Recall that each bit of each coefficient is assigned a class defined by its level, pattern, and significant neighbor metric (SNM). While the first two parameters are fixed, the SNM depends on the order in which classes are coded and the modified values chosen for neighboring coefficients. Thus we begin the estimation procedure by estimating SNM.

Let $C(i,j)$ denote the wavelet coefficient at spatial coordinates $(i,j)$. We use a two step procedure to estimate the SNM of $C(i,j)$ at resolution $\ell$. The first step is a coarse estimate. If a neighbor of coefficient $C(i,j)$ is significant at or before resolution $\ell$, then that neighbor is counted as significant for the SNM of $C(i,j)$ at resolution $\ell$. The second step refines the initial coarse estimate by decreasing the SNM estimate by the number of $C(i,j)$'s significant neighbors that are likely to be declared significant

*after* the description of $C(i,j)$. The procedure for refining the SNM estimates relies on the order in which the classes are described, which in turn relies on the first-pass SNM estimate. Only neighbors that have already become significant can be counted in the SNM during GTW encoding since the encoder's SNM calculation must be mirrored at the decoder. While the first SNM estimate for the coefficients can be done in any order, the second step takes the coefficients one by one in zig-zag order through each sub-band of the wavelet decomposition. This approach mimics GTW's adaptive estimation of the SNM.

Given the SNM estimates, we can estimate the class of every bit in the wavelet decomposition. Using these estimates, we calculate the insignificance probability $q$ as the empirical probability of a zero in the estimated class, at a given bit-plane. We use the Kullback-Liebler distance, $D(q||q^\star)$ to compare the original adaptive calculation of the insignificance probability $q^\star$ (in GTW) and our estimation of the insignificance probability $q$ using the estimated SNM and estimated class values, where $q$ equals the empirical probability (the count) of a zero in the estimated class. As in GTW, the estimated group size $k$ is computed using (5.1).

### 5.4.2 Lagrangian Calculation for Opt-GTW

The calculation of the multi-resolution Lagrangian requires the estimation of the associated rates and distortions at each resolution, when declaring coefficient $C(i,j)$ significant at resolution $s$.

The calculation of the distortions, as described in section 5.2, is analogous to SPIHT, Opt-SPIHT, and GTW. On the other hand, the estimation of the expected rate for the description of an element within a group is quite different within the GTW framework. The expected incremental rate $r_{s,m}(i,j)$ is the rate that would be used in the resolution-$m$ description of the coefficient $C(i,j)$, assuming $C(i,j)$ is declared significant at resolution $s$. The value of $r_{s,m}(i,j)$ depends on the $m$-th bit of the chosen reproduction and also on the position of that bit within its group and

| Grp | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
|---|---|---|---|---|---|---|---|---|
| Out | Rate cost per item | | | | | | | |
| 1 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 |
| 0 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 | 1/8 |
| 0 | – | – | – | – | 1/4 | 1/4 | 1/4 | 1/4 |
| 0 | – | – | – | – | – | – | 1/2 | 1/2 |
| Tot | 2/8 | 2/8 | 2/8 | 2/8 | 4/8 | 4/8 | 8/8 | **8/8** |

Table 5.1: Rate costs per item, $U_0(a, b, k)$ and $U_1(b, k)$, for a group (Grp) of size $k = 8$ items with a one at position $b = 8$, with 7 zeros before a one at positions $a = 1, ..., 7$. The corresponding output bit-stream is shown in the Out column.

| # Zeros | Group items | Expected Length of a '1' |
|---|---|---|
| 0 | 1 ? ? ? ? ? ? ? | 4 bits |
| 1 | 0 1 ? ? ? ? ? ? | 2 bits |
| 2 | 0 0 1 ? ? ? ? ? | 2 bits |
| 3 | 0 0 0 1 ? ? ? ? | 1.25 bits |
| 4 | 0 0 0 0 1 ? ? ? | 2.4 bits |
| 5 | 0 0 0 0 0 1 ? ? | 1.3 bits |
| 6 | 0 0 0 0 0 0 1 ? | 1.619 bits |
| 7 | 0 0 0 0 0 0 0 1 | 1 bit |

Table 5.2: Expected description lengths of a '1', $U_1(b, k)$, for groups of size $k = 8$ and $b - 1 = 0, ..., 7$ zeros before the '1'.

within its class. In particular

$$r_{s,m}(i,j) = \begin{cases} Rate(\text{'0'}, k, v, q, N) & \text{if } C_m(i,j) = 0 \\ Rate(\text{'1'}, k, v, q) & \text{if } C_m(i,j) = 1\,, \end{cases} \tag{5.4}$$

where $C_m(i, j)$ is the $m$-th bit in the binary description of coefficient $C(i, j)$. To see why $C(i, j)$ varies with all these parameters, consider the following example.

Table 5.1 shows the expected description lengths, for each item of a group of size $k = 8$, where the first 7 elements are all zeros, and the last element is a one. The tabulated values correspond to the expected rate cost for describing a single '1' at position $b = 8$, denoted by $U_1(b, k)$, as well as the rate cost for describing the $a$-th '0' when the $b$-th element ($b > a$) in a group of size $k$ is a '1', which we denote by

$U_0(a, b, k)$. The first column of Table 5.1 shows the output sequence. The first bit of the output bit-stream indicates that at least one element in the group is a one. The next bit is a zero because the first half of the group (which contains the first 4 items) is insignificant. The following zero specifies that the first half of the remainder is also insignificant. Finally the last zero establishes that of the two items that still remain unidentified, the first is insignificant; therefore the other item has to be a one. The cost of each output bit is amortized over the items that "benefit" from that expenditure of rate. Therefore, bits that only discriminate between later elements are not counted in the rate calculation for earlier items of the group. Since early output bits provide information required for all elements, we split that rate cost across the full group. The total rate for each member of the original group appears in the last row of the table. Table 5.2 shows groups of size $k = 8$, with 0 through 7 zeros before the '1' at position $b$. Notice that the expected length of the description of a one, $U_1(b, k)$, is not monotonic in $b$. The rate costs per item, $U_0(a, b, k)$ and $U_1(b, k)$, are pre-calculated for the range of $a$, $b$, and $k$ values, and stored for table look-up during the calculation of the expected description lengths.

We compute the expected description length of a '1' (a significant element) for any given group as,

$$\begin{aligned} \mathit{Rate}(\text{`1'}, k, v, q) &= q^{v-1} U_1\left((v-1)\%k + 1, k\right) \\ &\quad + p\left(\sum_{g=kM}^{v-2} q^g U_1(g\%k + 1, k) + \sum_{g=0}^{k-1} q^g U_1(g+1, k)\frac{(1-q^{kM})}{1-q^k}\right), \end{aligned} \quad (5.5)$$

where $k$ is the group size, $g$ counts the number of zeros before the '1,' $v$ is the position of the '1' in the whole class that contains the group, $q$ is the probability of a zero, and $p = 1 - q$ is the probability of a one. The procedures for estimating $q$ and $k$ are described in the next sub-section. Here $M = \lfloor (v-2)/k \rfloor$ is the largest integer less than or equal to $(v-2)/k$, and % is the integer mod operator.

Similarly, the estimation of the expected rate needed for the description of a '0'

in a class having $N$ total elements is

$$\begin{aligned} Rate(\text{`0'}, k, v, q, N) \quad = \\ & p\sum_{b=0}^{T} q^{b+v-1} U_0((v-1)\%k+1, (v-1)\%k+b+2, k) \\ & +p^2 \sum_{=0}^{L}\sum_{b=0}^{F} q^{a+b} U_0(a\%k+1, a\%k+b+2, k) \\ & +p\left(q^{k-1}\frac{1-q^{kM}}{1-q^k} + q^{kM+k-1}\frac{v-kM-1}{k}\right) \\ & +q^{kQ+k-1}\frac{1}{k}, \end{aligned} \tag{5.6}$$

where $T = \min\{k-2-(v-1)\%k, N-v-1\}$, $L = \min\{k-2, v-2\}$, $F = \min\{k-2-a\%k, N-v-1\}$ and $Q = [(v-1)/k]$. Here $U_0(a,b,k)$ denotes the expected rate for describing the $a$-th '0' when the $b$-th element ($b > a$) in a group of size $k$ is a '1.'

The above rate and distortion calculations allow us to calculate the Lagrangian $\hat{J}_s(i,j)$ for each coefficient and each value of $s$. We then choose the coefficient value that yields the lowest Lagrangian. After all of the modifications for all of the coefficients are done, we encode the modified coefficients using the standard GTW encoder. The decoder is identical to the regular GTW decoder.

### 5.4.3 Including ENO Adaptive Wavelets in the Optimization

Both SPIHT and GTW normally use a fixed wavelet transform. The experimental results of Section 5.4.5 use the 9-7 tap filters of [1].

Instead of restricting the algorithms to the standard (STD) wavelet transform, we explore the optimization of GTW that allows us to choose between standard wavelet and Essentially Non-Oscillatory (ENO) wavelet coefficients, that yield the best rate-distortion trade-off. This is motivated by the hypothesis that ENO coefficients, which reduce the magnitude of high-frequency coefficients in a wavelet decomposition while preserving the total energy of the image, are often better suited to bit-plane coding

than standard wavelets, as described in Section 5.2.3.

In this section, we use the multi-resolution Lagrangian (5.2) to determine when the trade-off between the rate used to describe the ENO map and the rate saved in the bit-plane description makes using the ENO coefficients worthwhile. In deciding between ENO and STD wavelet coefficients, we calculate the optimal multi-resolution Lagrangian for each and choose the representation with the better Lagrangian performance. The Lagrangian for the ENO coefficients also includes the cost of describing the ENO mapping (compressed using a simple entropy coder). For the STD-MR-Lagrangian, we include the near-negligible cost of indicating that the ENO map is empty.

The process is repeated for all ENO decisions being considered. Because every ENO decision affects a collection of coefficients, we sum up the Lagrangians of all affected coefficients of each single ENO decision. For comparison, we also sum up the individual multi-resolution Lagrangians of the STD wavelet coefficients of those same locations.

To identify the best multi-resolution Lagrangian for a collection of STD or ENO coefficients, we also consider possible modifications of STD and ENO coefficients, as described earlier in this chapter for regular wavelet coefficients. By choosing the lowest Lagrangian, we adopt the corresponding modifications of the coefficients and indicate the final map of ENO decisions. After all of the modifications for all of the coefficients are done, we encode the modified coefficients using the standard GTW encoder. Note that the decoder is the same as the regular GTW decoder except in its use of the ENO inverse wavelet transform, as indicated in the ENO map. Figure 5.5 shows an example of an ENO edge map for the "Cameraman" image (from Figure 5.3), for a two-level ENO wavelet transform.

### 5.4.4 Complexity of Opt-GTW

The added computational complexity of Opt-GTW with respect to Hong's original GTW [30, 31], is mostly due to the calculation of the multi-resolution Lagrangians.

For an image with a total number of $Z$ pixels, we calculate a collection of Lagrangians for each possible significance choice among the possible $L$ resolutions. Therefore, the (maximum) number of Lagrangians needed to decide the best significance for all the wavelet coefficients is $L^2\,Z$

The total number of multiplications and additions for an $L$-resolution Lagrangian $\tilde{J}_s(i,j) = \sum_{\ell=1}^{L} \alpha_\ell D_{s,\ell}(i,j) + \beta_\ell r_{s,\ell}(i,j)$, is $2L^2$ and $(2L-1)L$, respectively. The distortion $D_{s,\ell}(i,j) = \Big(C(i,j) - \widetilde{C}_{s,\ell}(i,j)\Big)^2$ is calculated after the (partial) reconstruction of every coefficient $C(i,j)$ at each one of the $\ell = 1, \cdots, L$ resolutions. Note that $\widetilde{C}_{s,\ell}(i,j)$ is the (partial) reconstruction after the first $\ell$ bit-planes of the modified coefficient. We use $\widetilde{C}_s(i,j)$ to denote the approximation to $C(i,j)$ achieved by declaring this coefficient significant at resolution $s$. Thus for a given $\widetilde{C}_s(i,j,)$ for each possible choice of significance-resolution $s$, and at every (incremental) resolution $\ell$, the distortion calculation involves one addition and one multiplication.

The incremental rate $r_{s,m}(i,j)$ depends on the bit used to represent $\widetilde{C}_{s,m}(i,j)$ at each $m = 1, \cdots, L$ resolution, as described by (5.4).

### 5.4.5 Experimental Results

The optimization is applied on a 7 level decomposition of STD- as well as ENO wavelet coefficients. We use the standard gray-scale image "Cameraman" of size $256 \times 256$ pixel, shown in Figure 5.3. That image and many other *de facto* standards are available on the Internet, for example at [32].

The plot of Figure 5.4 compares the PSNR as a function of rate, for both the optimized code and the original GTW algorithm. The graph shows a performance improvement of approximately 0.7 dB.

Figures 5.6a and 5.6b show the reconstructed images of the gray-scale "Cameraman" at $R = 0.1$ bits per pixel, that is an 80-times compression ratio, using standard GTW with standard 9-7 Antonini wavelets, and the optimized method Opt-GTW with 9-7 ENO wavelets, respectively. The optimization using ENO wavelets allows a perceptual quality improvement that is particularly noticeable at sharp edges in the

image. The combination of the Lagrangian optimization with the application of the ENO wavelet transform proposed in this work yields performance improvements both in PSNR and in visual quality of the reconstructed images.

### 5.4.6 Acknowledgments

We want to thank Ed Hong and R. E. Ladner for providing us with the code for the standard GTW, to which we added the implementation of our algorithm.

Figure 5.3: Original 256 × 256 pixel "Cameraman" gray-scale image.

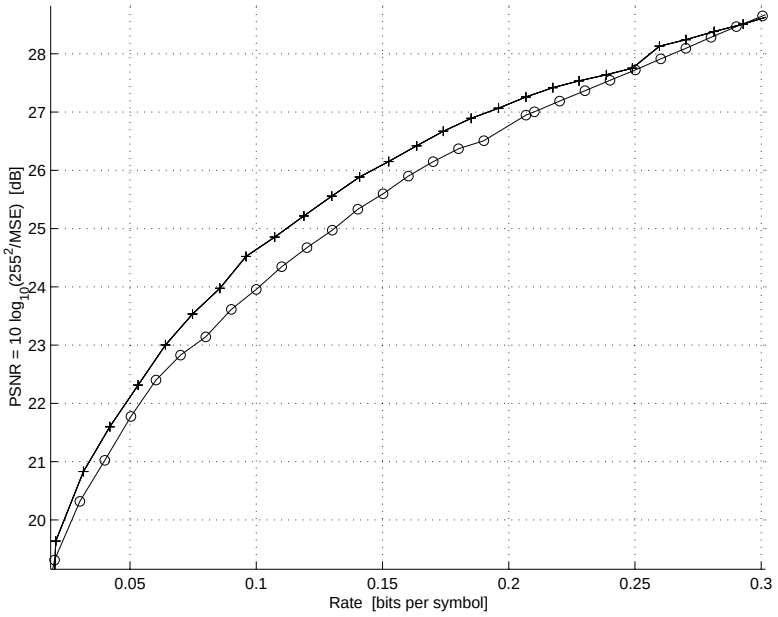



Figure 5.4: Comparison of the PSNR as a function of rate for GTW (circles), and the new optimization method with ENO wavelets. The performance improvement of the new method is approximately 0.7 dB.

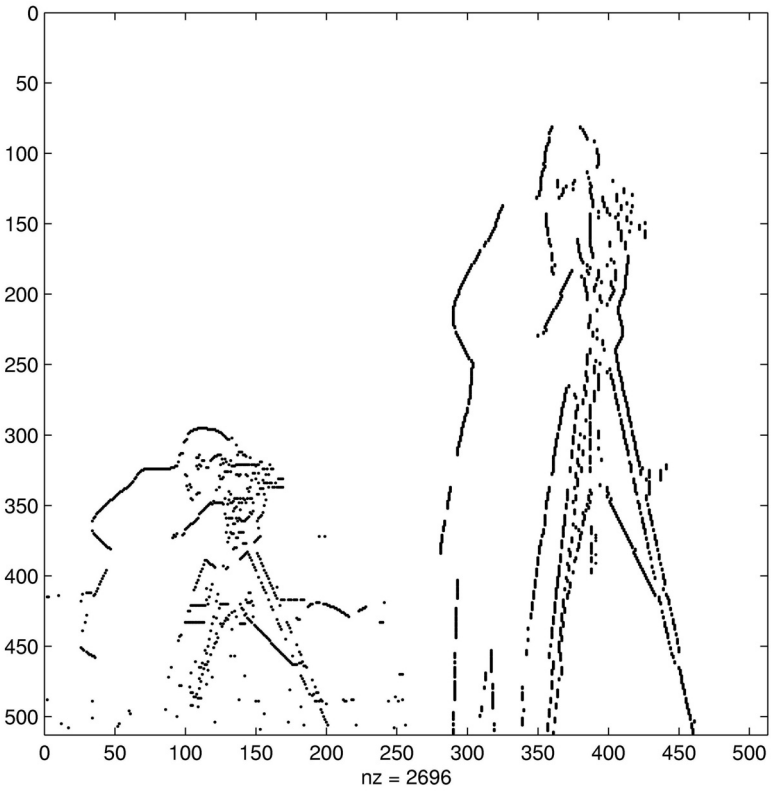



Figure 5.5: Two-level ENO edge map for the "Cameraman" gray-scale image.

(a)

(b)

Figure 5.6: Low-rate reconstructed images of the gray-scale "Cameraman" at $R = 0.1$ bpp, using (a) standard GTW with standard 9-7 Antonini wavelets, and (b) the optimized method Opt-GTW with 9-7 ENO wavelets.

# Chapter 6 Summary and Conclusion

This thesis studies the optimization of multi-resolution source codes. The optimization methods presented here are based on the minimization of a Lagrangian performance measure, where the Lagrangian coefficients are the weights that parameterize the priorities assigned to the resolutions. The relative value of the Lagrangian parameters can be set according to the user's priorities over the resolutions, e.g., depending on the probability of utilization of each resolution. The optimal multi-resolution code is the code that minimizes the weighted performance measure.

In the first part, we have used a Lagrangian to investigate optimality properties implied by the rate-distortion theoretical bounds. We have studied the characterization of the optimal output density functions of a two-resolution source code for any arbitrary set of priorities over the resolutions. The result is a conjecture about when the second-resolution reproduction of a two-resolution code is purely singular, for an iid source.

In the second part, we have presented a new method for converting design constraints into the corresponding Lagrangian parameters for general multi-resolution codes. This technique takes a set of user-defined target rates or target distortions (or any combination of rates and distortions) and their associated priorities, runs a bisection-style search and returns the corresponding Lagrangian coefficients.

We have shown here the performance of practical implementations of the code for several vector dimensions, both for fixed- and variable-rate, comparing the performance and features with other existing algorithms.

Motivated by the computational complexity of MRVQ, which grows exponentially with the number of resolutions, we have proposed a linear complexity MRVQ, that is a hybrid between TSVQ and MRVQ. The linear complexity variant is a multi-path search algorithm, where the number of paths being considered can be adjusted arbitrarily, bridging the gap between the greedy *1-path* (TSVQ), and the *all-path*

(MRVQ) search alternatives.

Also motivated by computational complexity considerations, in the third part of this thesis we have explored the rate-distortion performance optimization of wavelet-based codes. We have introduced a new optimized encoder for the SPIHT algorithm. The new Opt-SPIHT encoder optimizes the code relative to a user-defined priority function, to yield performance improvements at the resolutions of highest importance, at the expense of performance degradation at the resolutions of low priority. The resulting encoded bit-stream is compatible with the standard SPIHT decoder (the decoder requires no knowledge of the priority function) and obtains benefits relative to the given priorities. The achieved low complexity, *globally optimal* solution uses a dynamic programming approach to adjust the magnitudes of the wavelet coefficient, and minimizing a Lagrangian performance measure to optimize the code's performance in a rate-distortion sense.

The optimization of more general bit-plane coders is not straightforward. The GTW algorithm is a generalization of zero-tree codes, like SPIHT, that uses groups of coefficients other than zero-trees. We have presented a new algorithm, named Opt-GTW, that first estimates the Lagrangian performance measure and then modifies the magnitudes of the wavelet coefficients such that the rate-distortion performance is optimized according to the Lagrangian estimates.

While Opt-SPIHT finds a globally optimal solution, there is no direct analog in the GTW framework. Nevertheless, Opt-GTW provides a principled technique to incorporate user-defined priorities into the optimization of the encoded bit-stream, keeping the decoder identical to the original GTW and, again, not requiring it to know the priority function. The new approach proposed here is based on an original *a priori* estimation of several of GTW's adaptive parameters to calculate a Lagrangian cost function that incorporates the priority schedule as well. We have also presented a new method for the joint optimization of GTW and the wavelet transform, by incorporating the adaptive ENO wavelet transform in the optimization procedure.

Through the different algorithms presented here we have demonstrated the practical application of the Lagrangian performance measure to the optimization of different

kinds of multi-resolution source codes.

We end this thesis by describing possible areas for future work. One of the elements that was not explored in this thesis is the joint optimization of rate, distortion, and complexity, in code design. This problem has been partially studied in the literature, for example through variable complexity algorithms but mostly oriented to single-resolution codes, (e.g., [43] for transform codes.) While this work gives examples of multi-resolution codes with different complexities, it does not suggest a method for understanding the fundamental underlying trade-offs between Lagrangian performance and complexity. Furthermore, it would be interesting to consider the joint optimization of rate, distortion, complexity, latency (delay), and memory, in practical multi-resolution code design.

# Bibliography

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

[2] C. F. Barnes. *Residual Quantizers.* Ph. D. Dissertation, Brigham Young University, Provo, UT, 1989.

[3] C. F. Barnes and R. L. Frost. Necessary conditions for the optimality of residual vector quantizers. In *Proceedings of the IEEE International Symposium on Information Theory*, page 34, San Diego CA, January 1990. IEEE.

[4] C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi. Advances in residual vector quantization: a review. *IEEE Transactions on Image Processing*, IP-5(2):226–262, February 1996.

[5] T. Berger. *Rate distortion theory. A mathematical basis for data compression.* Prentice Hall, Englewood Cliffs, N.J., 1971.

[6] H. Brunk and N. Farvardin. Fixed rate successive refining scalar quantizers for progressive transmission. In *Proceedings of the Data Compression Conference*, pages 250–259, Snowbird, UT, April 1996. IEEE.

[7] H. Brunk, H. Jafarkhani, and N. Farvardin. Design of successively refinable scalar quantizers. -, April 1998. Preprint.

[8] A. Buzo, A. H. Gray Jr., R .M. Gray, and J .D. Markel. Speech coding based upon vector quantization. *IEEE Transactions on Information Theory*, 28:562–574, October 1980.

[9] N. Chaddha, P. A. Chou, and R. M. Gray. Constrained and recursive hierarchical table lookup vector quantization. In *DCC-Proc*, pages 220–229, Snowbird, UT, March 1996. IEEE.

[10] T. F. Chan and H. M. Zhou. Adaptive eno-wavelet transforms for discontinuous functions. Technical report, Dept. of Math, UCLA, CAM Report, No. 99-21, June 1999.

[11] T. F. Chan and H. M. Zhou. Eno-wavelet transforms for piecewise smooth functions. *To appear in SIAM Journal of Numerical Analysis*, 2002.

[12] P. A. Chou, T. Lookabaugh, and R. M. Gray. Entropy-constrained vector quantization. *IEEE Transactions on Acoustics Speech and Signal Processing*, 37(1):31–42, January 1989.

[13] P. A. Chou, T. Lookabaugh, and R. M. Gray. Optimal pruning with applications to tree structured source coding and modeling. *IEEE Transactions on Information Theory*, IT-35(2):299–315, March 1989.

[14] J. Chow and T. Berger. Failure of succesive refinement for symmetrical Gaussian mixtures. *IEEE Transactions on Information Theory*, IT-43(1):350–352, January 1997.

[15] A. Deever and S. S. Hemami. What's your sign?: efficient sign coding for embedded wavelet image coding. In *Proceedings of the Data Compression Conference*, pages 273–282, Snowbird, UT, March 2000. IEEE.

[16] D. Dugatkin and M. Effros. Multi-resolution VQ: parameter meaning and choice. In *Conference Record, Thirty-Second Asilomar Conference on Signals, Systems, and Computers*. IEEE, November 1998.

[17] D. Dugatkin and M. Effros. Setting priorities: a new spiht-compatible algorithm for image compression. In *Proceedings of the SPIE International Symposium on Optical Science and Technology*, volume 4119, pages 799–805, San Diego, CA, July 2000. SPIE.

[18] D. Dugatkin, H.M. Zhou, T.F. Chan, and M. Effros. Lagrangian optimization of a group testing for eno wavelets algorithm. In *Proceedings of the 36th Annual Conference on Information Sciences and Systems*, Princeton, NJ, March 2002. CISS.

[19] M. Effros. Practical multi-resolution source coding: TSVQ revisited. In *Proceedings of the Data Compression Conference*, pages 53–62, Snowbird, UT, March 1998. IEEE.

[20] M. Effros. Distortion-rate bounds for fixed- and variable-rate multiresolution source codes. *IEEE Transactions on Information Theory*, IT-45(6):1887–1910, September 1999.

[21] M. Effros and D. Dugatkin. Multi-resolution vector quantization. *Submitted to IEEE Transactions on Information Theory*, October 2000.

[22] W. H. R. Equitz. *Successive refinement of information.* Ph. D. Dissertation, Stanford University, Stanford, CA, 1989.

[23] W. H. R. Equitz and T. M. Cover. Succesive refinement of information. *IEEE Transactions on Information Theory*, IT-37(2):269–275, March 1991. See also [24].

[24] W. H. R. Equitz and T. M. Cover. Addendum to "succesive refinement of information",[23]. *IEEE Transactions on Information Theory*, IT-39(4):1465–1466, July 1993.

[25] H. Feng and M. Effros. Improved bounds for the rate loss of multi-resolution source codes. Submitted to the *IEEE Transactions on Information Theory*, March 2002.

[26] H. Feng and M. Effros. Improved bounds for the rate loss of multi-resolution source codes. In *Proceedings of the IEEE International Symposium on Information Theory*, page 193, Washington, D.C., June 2001. IEEE.

[27] R. L. Frost, C. F. Barnes, and F. Xu. Design and performance of resifual quantizers. In *Proceedings of the Data Compression Conference*, pages 129–138, Snowbird, UT, April 1991. IEEE.

[28] R. G. Gallager and D. C. Van Voorhis. Optimal source codes for geometrically distributed integer alphabets. *IEEE Transactions on Information Theory*, IT-21:228–230, March 1975.

[29] S. Herman and K. Zeger. Variable fanout trimmed tree-structured vector quantization for multirate channels. In *Proceedings of the IEEE International Symposium on Information Theory and Its Applications*, volume 1, pages 417–421, Victoria, British Columbia, Canada, September 1996.

[30] E. S. Hong and R. E. Ladner. Group testing for image compression. In *Proceedings of the Data Compression Conference*, pages 3–12, Snowbird, UT, March 2000. IEEE.

[31] E. S. Hong and R. E. Ladner. Group testing for image compression. *IEEE Transactions on Image Processing*, 11(8):901–911, August 2002.

[32] "Standard" Images. ftp://links.uwaterloo.ca/pub/bragzone/greyset1/camera/. University of Waterloo, Canada.

[33] H. Jafarkhani, H. Brunk, and N. Farvardin. Entropy-constrained successively refinable scalar quantization. In *Proceedings of the Data Compression Conference*, pages 337–346, Snowbird, UT, March 1997. IEEE.

[34] B.-H. Juang and Jr. A. H. Gray. Multiple stage vector quantization for speech coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 597–600, Paris, 1982.

[35] L. Lastras and T. Berger. All sources are nearly successively refinable. In *Proceedings of the IEEE International Symposium on Information theory*, Sorrento, Italy, June 2000. IEEE.

[36] L. Lastras and T. Berger. All sources are nearly successively refinable. *IEEE Transactions on Information Theory*, IT-47(3):918–926, March 2001.

[37] K. K. Lin and R. M. Gray. Rate-distortion optimization for the spiht encoder. In *Proceedings of the Data Compression Conference*, pages 123–132, Snowbird, UT, March 2001. IEEE.

[38] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, January 1980.

[39] J. Liu and P. Moulin. Analysis of interscale and intrascale dependencies between image wavelet coefficients. In *International Conference on Image Processing*, pages 383–386, Vancouver, BC, Canada, September 2000. IEEE. Vol 2.

[40] J. Liu and P. Moulin. Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients. *IEEE Transactions on Image Processing*, IP-10(10):1647–1658, November 2001.

[41] H. Malvar. Fast progressive wavelet coding. In *Proceedings of the Data Compression Conference*, pages 336–343, Snowbird, UT, March 1999. IEEE.

[42] E. Ordentlich, M. Weinberger, and G. Seroussi. A low-complexity modeling approach for embedded coding of wavelet coefficients. In *Proceedings of the Data Compression Conference*, pages 408–417, Snowbird, UT, March 1998. IEEE.

[43] W. Pan and A. Ortega. Complexity-scalable transform coding using variable complexity algorithms. In *Proceedings of the Data Compression Conference*, pages 263–273, Snowbird, UT, March 2000. IEEE.

[44] B. Rimoldi. Succesive refinement of information: characterization of achievable rates. *IEEE Transactions on Information Theory*, IT-40(1):253–259, January 1994.

[45] E. A. Riskin and R. M. Gray. A greedy tree growing algorithm for the design of variable rate vector quantizers. *IEEE Transactions on Signal Processing*, 39:2500–2507, November 1991.

[46] E. A. Riskin, R. Ladner, R.Y. Wang, and L. E. Atlas. Index assignment for progressive transmission of full-search vector quantization. *IEEE Transactions on Image Processing*, 3(3):307–312, May 1994.

[47] K. Rose. A mapping approach to rate-distortion computation and analysis. *IEEE Transactions on Information Theory*, IT-40(6):1939–1952, November 1994.

[48] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–250, June 1996.

[49] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[50] E. Tuncel and K. Rose. Computation and analysis of the n-layer scalable rate-distortion function. *IEEE Transactions on Information Theory*, IT-49(5):1218–1230, May 2003.

[51] M. Vishwanath and P.A. Chou. An efficient algorithm for hierarchical compression of video. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 275–279, Austin, TX, November 1994. IEEE.