Beyond Traditional Transform Coding

by

Vivek K Goyal

B.S. (University of Iowa) 1993
B.S.E. (University of Iowa) 1993
M.S. (University of California, Berkeley) 1995

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering—Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Martin Vetterli, Chair
Professor Venkat Anantharam
Professor Bin Yu

Fall 1998

Beyond Traditional Transform Coding

Copyright © 1998
by
Vivek K Goyal

Printed with minor corrections 1999.

Abstract

Beyond Traditional Transform Coding

by

Vivek K Goyal

Doctor of Philosophy in Engineering—Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Martin Vetterli, Chair

Since its inception in 1956, transform coding has become the most successful and pervasive technique for lossy compression of audio, images, and video. In conventional transform coding, the original signal is mapped to an intermediary by a linear transform; the final compressed form is produced by scalar quantization of the intermediary and entropy coding. The transform is much more than a conceptual aid; it makes computations with large amounts of data feasible. This thesis extends the traditional theory toward several goals: improved compression of nonstationary or non-Gaussian signals with unknown parameters; robust joint source–channel coding for erasure channels; and computational complexity reduction or optimization.

The first contribution of the thesis is an exploration of the use of frames, which are overcomplete sets of vectors in Hilbert spaces, to form efficient signal representations. Linear transforms based on frames give representations with robustness to random additive noise and quantization, but with poor rate–distortion characteristics. Nonlinear, signal-adaptive representations can be produced with frames using a greedy algorithm called matching pursuit. Matching pursuit is a computationally feasible alternative to producing the most sparse approximate representation with respect to a frame. It exhibits good compression performance at low rates even with rather arbitrary frames. Optimal reconstruction is described for both linear and nonlinear frame-based representations.

Within the conventional setting of basis representations, the Karhunen–Loève transform (KLT) is known to solve a variety of problems, including giving optimal compression of Gaussian sources. Its dependence on the probability density of the source, which is generally unknown, limits its application. Including a description of an estimated KLT in the coded data may create significant overhead. In a universal system the overhead is asymptotically negligible. This thesis introduces a method for universal transform coding of stationary Gaussian sources which utilizes backward adaptation. In a backward adaptive system, all parameter updates depend only on data already available at the decoder; hence, no side information is needed to describe the adaptation. Related to this is the development of new transform adaptation techniques based on stochastic gradient descent. These are inspired by an analogy between FIR Wiener filtering and transform coding.

Transform coding is normally used as a source coding method; as such, it is optimized for a noiseless communication channel. Providing a noiseless or nearly noiseless communication channel usually requires channel coding or a retransmission protocol. Rather than using a system with separate source and channel coding,

better overall performance with respect to rate and distortion, with a bound on delay, can often be achieved with joint source–channel coding. This thesis introduces two computationally simple joint source–channel coding methods for erasure channels. Source coding for an erasure channel is analogous to multiple description coding; it is in this context that these techniques are presented.

The first technique uses a square transform to produce transform coefficients that are correlated so lost coefficients can be estimated. This technique uses discrete transforms that are also shown to be useful in reducing the complexity of entropy coding. Analyzed with fine quantization approximations and ideal entropy coding, the rate allocated to channel coding is continuously adjustable with fixed block length. The second technique uses a quantized frame expansion. This is similar to scalar quantization followed by a block channel code, except that quantization and addition of redundancy are swapped. In certain circumstances—marked especially by block length constraints and a lack of knowledge of the channel state—the end-to-end performance significantly exceeds that of a system with separate source and channel coding. The multiple description coding techniques both give graceful performance degradation in the face of random loss of transform coefficients.

The final topic of the thesis is the joint optimization of computational complexity and rate–distortion performance. The basic idea is that higher computational complexity is justified only by better performance; a framework for formalizing this conviction is presented. Sample analyses compare unstructured and harmonic transforms, and show that JPEG encoding complexity can be reduced with little loss in performance.

Contents

List of Figures

List of Tables

Acknowledgements

Though writing a thesis is a solitary act, research is usually collaborative. For this reason, no thesis would be complete without a list of acknowledgements. I have worked significantly with at least half a dozen colleagues at three institutions, so my list may be longer than most. Please bear with me and also note the list of collaborators at the beginning of each chapter.

I owe the most and greatest thanks to my advisor and friend, Martin Vetterli. He has been a tireless and enthusiastic source of good ideas; I only wish I had the energy and dedication to pursue more of them. I would especially like to thank him for leaving Berkeley—in doing so facilitating an adventurous time in Europe—and then for temporarily returning to Berkeley; the final period of working together contiguously was wonderful. *Je voudrais egalment remercier Marie-Laure et Martin pour leur hospitalité en Suisse.*

My latter-day mentor, Jelena Kovačević, arranged my other adventure, a stay at Bell Labs. There I found an open-minded and vibrant research culture unmatched inside or outside of academia. Thanks to all of the Mathematics of Communications Research Department for welcoming me warmly as a student and now as a colleague. I would also like to thank Jelena for sparking my interest in multiple description coding. Finally, her constructive criticism of this document is noteworthy not only for its completeness, but because she was under no obligation whatsoever to read it.

Thanks to my other primary collaborators: Ramon Arean, Christopher Chan, Sundeep Rangan, Nguyen Thao, and Jun Zhuang. Collaborative work with Sundeep is reported in Appendix 2.C. In addition, he helped me solve vexing problems in other diverse areas. Various interactions with Kannan Ramchandran and Antonio Ortega shaped my research and contributed greatly to my education. Mike Orchard, Amy Reibman, Vinay Vaishampayan, and Yao Wang are thanked for providing a preprint that inspired the work in Section 5.3. By sharing their audio coding expertise, Gerald Schuller and Deepen Sinha made the work reported in Section 5.3.4 possible, and it should be noted that Ramon and Jelena are most responsible for this application.

I thank Professors Venkat Anantharam and Bin Yu for being on my committee and providing valuable comments. In addition, innumerable comments and suggestions provided by Alyson Fletcher and Mike Goodwin were indispensable. Thanks to Mary Byrnes, Ruth Gjerde, Rob McNicholas, Phyllis Takahashi, and Ruth Tobey for being the most consistently friendly and adept staff members I encountered in Berkeley. In both Berkeley and Lausanne, I enjoyed the camaraderie of a supportive research group. In particular, Matt Podolsky sacrificed himself for the collective good by learning a bit more about system administration than he may have wanted to, and Mike's thesis prose set a high-water mark that was inspirational.

The past five years have consisted of more than just work. Further thanks to Jérôme Lebrun and Paolo Prandoni for making my time in Lausanne an all-around success. My friends in Berkeley—especially Allie, Richard, Heath, and Al—made there be a lot more good times than bad, and I will miss them. Last, but not least, the love and support of my family is so unfaltering that I often fail to notice or acknowledge it.

Chapter 1

Introduction and Preview

THE WORLD abounds in signals: detectable physical quantities—pressures, forces, voltages, magnetic field strengths, photon energies and counts—that capture conditions at a particular time and place. Communication of these signals allows for sight and sound reproduction at a level taken for granted in modern life. "Signal" is also used to describe a unitless mathematical abstraction of a physical measurement. We will immediately dispense with physics by taking the latter view.

This thesis addresses a class of methods used in communicating signals called transform coding. The primary focus is on producing compact representations of signals so that the subsequent communication requires lesser resources. This is called compression or source coding. Transform coding is the most prevalent compression technique for signals tolerant to some distortion. This is evidenced by its prominent position in image, audio, and video coding standards. This practical success has been the inspiration for a broader look at transform coding that allows adaptation and the use of overcomplete expansions with the goal of designing joint source–channel codes as well as source codes.

This first chapter provides an introduction to the problems considered in this thesis and to needed terminology. For an initiated reader, the meaty chapters that follow should each stand alone. However, unifying themes appear—predominantly here and in the final chapter.

What follows is a description of transform coding: its history, fundamental theory, and application. The work reported here is based on lifting the constraints of the basic theory, in various combinations and settings. The results extend beyond transform coding and touch on various aspects of information theory, signal processing, and harmonic analysis.

1.1 Traditional Transform Coding

Transform coding was invented by Kramer and Mathews [114] as a method for transmitting correlated, continuous-time, continuous-amplitude signals. They showed that the total bandwidth necessary to transmit a set of signals with a prescribed fidelity can be reduced by transmitting a set of linear combinations of the signals instead of the signals themselves. Assuming Gaussian signals and the mean-squared error fidelity criterion, the Karhunen–Loève transform is an optimal transform for this application.

The signals that Kramer and Mathews wished to transmit were the energy signals of a vocoder ("*vo*ice

coder"). The vocoder,[1] developed by Homer Dudley [43], is an analysis-synthesis system for speech. The analysis unit produces a set of signals that represent estimates of the power in various frequency bands. The synthesis unit produces synthesized speech by adding modulated versions of the energy signals. One could argue that the vocoder is itself a transform coder. The distinction to be made is that because of the indifference to phase, the vocoder does not attempt to reproduce the original signal. That was not unintentional; Dudley was taking advantage of the phase insensitivity of human hearing.

Coding for continuous-valued channels is an almost forgotten art, as most effort goes into designing systems for storage and communication of digital data. The application of Kramer and Mathews, though the original, seems alien. The modern use of transform coding is in systems that include quantization. In this context, the theory was established by Huang and Schultheiss [100]. Their results will be discussed in Section 1.1.3 after some additional background material.

1.1.1 Mathematical Communication

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point." With these words,[2] Claude Shannon launched a mathematical discipline devoted to the processing, transmission, storage, and use of information: information theory. Information theory directly addresses problems in communication, but it has also had fundamental impact on fields beyond communication including probability, statistics, computation theory, physics, and economics [34].[3]

One of many successes of Shannon was to formulate a concrete but sufficiently flexible abstraction of a communication system. Shannon's abstraction is shown in Figure 1.1.

- The *information source* produces a message or sequence of messages to be communicated to the destination. A message can be a function of one or more continuous or discrete variables and can itself be continuous- or discrete-valued.
- The *transmitter*, or *encoder*, produces from the message a signal compatible with the channel.
- The *channel* is the physical medium that conducts the transmitted signal. The mathematical abstraction of the transmission is a perturbation by noise. The perturbation process may be considered additive, but since the noise source may be dependent on the transmitted signal, this poses no restriction. Only discrete-time channels will be considered here.
- The *receiver*, or *decoder*, attempts to recreate the message from the received signal.
- The *destination* is the intended recipient of the message.

This abstraction is very general, but the fundamental problems of information theory are already apparent. Given an information source and a complete characterization of the channel, is it possible to determine if there exist a transmitter and a receiver that will communicate the information? If so, how can the transmitter and receiver be designed?

[1] This term is used here narrowly, to refer to the original invention. Its use has since expanded to cover a wide range of algorithms.
[2] This is actually the first sentence of the *second* paragraph of [170].
[3] Shannon [171] and Elias [49] have written timeless editorials on the application and misapplication of information theory.

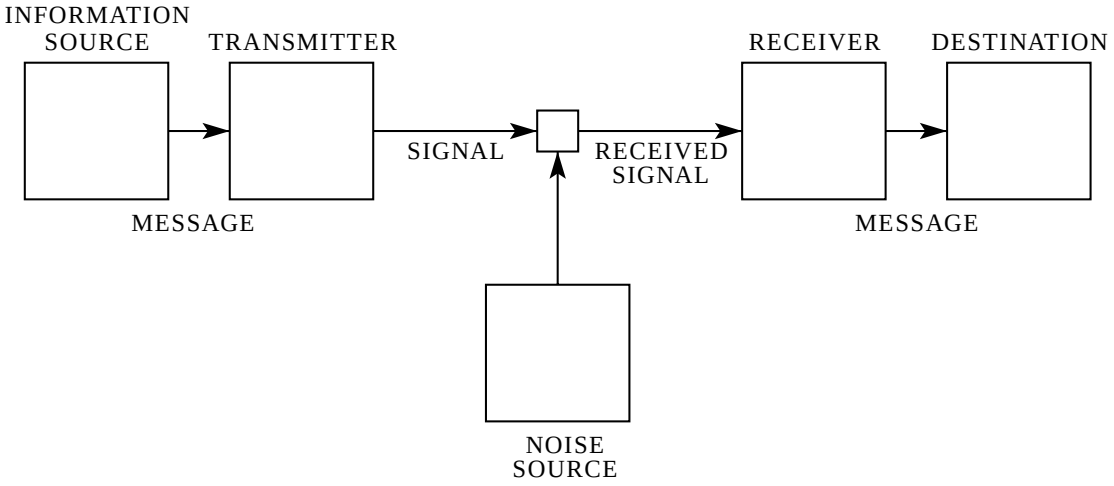


Figure 1.1: Shannon's communication system abstraction [170].

For the moment, let us consider a discrete information source that, at times $n \in \mathbb{N}$, emits a message from a discrete set $\mathcal{X}$. Also assume that the channel is discrete and noiseless, *i.e.*, at each time instant it takes as input a symbol from the discrete set $\mathcal{Y}$ and faithfully conveys it to the decoder. If we allow *any* message sequence in $\mathcal{X}^\infty$, then it is obvious that $|\mathcal{X}| \leq |\mathcal{Y}|$ is a necessary and sufficient condition to allow the communication to take place.

Shannon wanted less restrictive conditions on communication, and thus embarked on a statistical approach.[4] In his work, and almost all subsequent work in the information theory literature, there is some underlying statistical model which governs both the information source and the noise source. More specifically, the information source is a stochastic process, and the noise source is a stochastic process that may depend on the transmitted signal. Whether this is an exact fit to reality is a philosophical question, but it is reasonable and has led to an enormous body of results.

Adjusting the previous example, consider a discrete memoryless source which emits a symbol according to the probability distribution $p(\,\cdot\,)$ (defined on $\mathcal{X}$) at each time $n \in \mathbb{N}$. Suppose that the channel is discrete and noiseless as before. For the communication of a single message (single source symbol), reliable transmission still depends on $|\mathcal{X}| \leq |\mathcal{Y}|$. However, the average performance for coding many messages together now depends on the probability distribution of the source symbols. A number associated with the source, the *entropy rate*, quantifies the difficulty of conveying the messages. The entropy rate of the source is independent of the channel and in this particular case is given by[5]

$$H(p(\,\cdot\,)) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \text{ bits per symbol.} \tag{1.1}$$

Just as the difficulty of transmitting a source can be quantified independent of the channel that will be used, the *capacity* of a channel can be quantified. The capacity is the average of the logarithm of the number of input

[4] As confirmed through an acknowledgement in [170], Shannon was influenced along these lines by Wiener's work on time series. Wiener later strongly asserted that information theory "comes under the theory of time series" [206]. A partial alternative to the statistical view (it applies essentially to source coding with a fidelity criterion) is provided by Kolmogorov's ϵ-entropy [109]. This will not be considered here.

[5] The use of a logarithmic measure was deduced axiomatically by Shannon and had previously been suggested by Hartley [91].

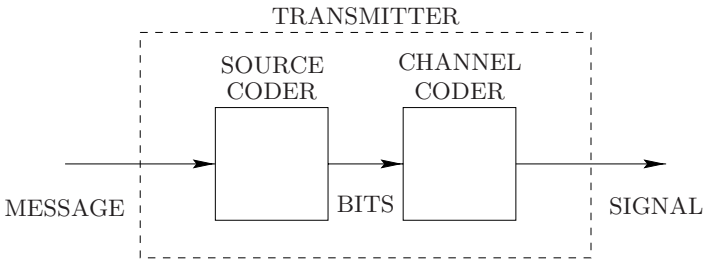


Figure 1.2: Separation of encoding into source and channel coding.

symbols that can reliably be distinguished from observing the output of the channel. For the noiseless channel of this example, all the input symbols can be distinguished by observing the output, so $C = \log_2 |\mathcal{Y}|$.

The Fundamental Theorem for a Discrete Channel with Noise [170, Thm. 11] states that communication with arbitrarily small probability of error is possible (by coding many messages at once) if $H \leq C$ and impossible if $H > C$. In our example, if the source symbols are equally probable, $H = \log_2 |\mathcal{X}|$ and the threshold for reliable communication lies at $|\mathcal{X}| = |\mathcal{Y}|$, as expected. For any other source distribution, the source entropy and the required channel capacity are reduced.

If the goal is coding with a prescribed maximum average distortion D, instead of arbitrarily low probability of error, a different characterization of the source is needed. This is given by *rate–distortion theory*. The *rate–distortion function* gives the minimum rate need to approximate a source up to a given distortion. The reader is referred to [170, 172, 13] for details.

1.1.1.1 Source and channel coding

The separate characterization of the source and the channel, and a unit of measure to connect them, led to separating encoders into two components: a *source coder* and a *channel coder* (see Figure 1.2). The source coder produces an auxiliary random process that is a *compressed version* of the source. For a discrete-time source, the source coder can be considered to produce some number of bits (symbols from $\{0, 1\}$) for each source symbol, without loss of generality. The role of the channel coder is to generate transmitted signals such that the compressed version of the source is conveyed with arbitrarily small probability of bit error.

The most that can be asked of the source coder is to produce a representation a little larger than H or $R(D)$ bits per symbol. On the other hand, the most that can be asked of the channel coder is to communicate C bits per symbol. The Fundamental Theorem above connects these two and is thus sometimes referred to as the "Separation Theorem" for stationary memoryless sources and channels. Extensions to other sources and channels are reviewed in [193].

Separation is very useful. It implies that one can devise a method for compressing a source without having to know anything about the channel, and without having to make adjustments for different channels. Similarly, the design of a system to communicate over a particular channel can be based simply on getting bits to the destination, without regard for the significance of each bit. For this reason, source coding and channel coding have grown into separate fields with rather separate communities.

The Separation Theorem bears further reflection. A very technical discussion of the classes of sources

and channels for which separation holds could follow, but there are more fundamental points. The Separation Theorem addresses only the possibility of communication, not the best way to achieve it. Compressing a source close to the entropy rate and communicating bits at close to the channel capacity both generally require the simultaneous processing of a large amount of data. If not impractical because of computational complexity and memory requirements, this at least increases the cost of implementing such a system. In general, even for a source and channel for which "separation holds," a system which uses a joint source–channel (JSC) code may have the same performance with less delay and less computation. Of course, the separated solution may be easier to design; this is the power of separation.

Communication channels, although subject to various physical phenomena, can be well characterized statistically. The same cannot be said of sources of practical interest. Images, audio, and video are not well-modeled as realizations of stochastic processes. Therefore they certainly do not fall into some class for which a separation theorem has been proven.

Transform coding has been a very popular technique for source coding. Source coding applications will be the primary focus of this thesis. However, two new transform coding approaches to JSC coding be introduced in Chapter 5.

This very quick tour of "the mathematical theory of communication"[6] was intended to provide just a flavor for the power of Shannon's statistical view, the basic terminology of source and channel coding, and a few fundamental results. Information theory is the subject of many books, notably the text by Cover and Thomas [34]; and the specific aspects of channel coding and rate–distortion theory are explained well by Lin and Costello [123] and Berger [13], respectively. Nevertheless, Shannon's lucid, original paper remains the best introduction to the field. We are lucky that Shannon had no motivation to submit manuscripts in least-publishable-increments!

1.1.2 Quantization

Digital transmission of information has come to dominate the design of communications systems to such an extent that information has become synonymous with bits.[7] The generation of bits from a continuous-valued source inevitably involves some form of quantization, which is simply the approximation of a quantity with an element chosen from a discrete set. Assuming the interface between a source coder and a channel coder is discrete, a finite-rate source coder is synonymous with a quantizer.

The communication of any information, continuous or discrete, through quantized values seems very natural today. However, this idea was rather new in Shannon's time. Pulse-code modulation (PCM), which for our purposes is nothing more than quantization for subsequent digital communication, was patented in 1939 [160]. The first fully deployed use of PCM was for a military communication system used in 1945 [154] and described openly in 1947 [17].

The simplest form of quantization is fixed-rate quantization of a scalar source. Here a real-valued source is mapped to one of a finite number of values. Symbolically, the quantizer is a mapping $Q : \mathbb{R} \to \mathcal{C}$ where

$$\mathcal{C} = \{y_1, y_2, \ldots, y_K\} \subset \mathbb{R}, \quad y_1 < y_2 < \cdots < y_K$$

[6] It was with justified confidence that Shannon's paper, "A Mathematical Theory of Communication," became "*The* Mathematical Theory of Communication" in the book with Weaver [173].

[7] The advantages of digital communication are described in many texts; *e.g.*, [115].

is called the *codebook*. This can be decomposed into *encoding* or "quantization"

$$\mathcal{E} : \mathbb{R} \to \{1, 2, \ldots, K\}$$

and *decoding* or "inverse quantization"

$$\mathcal{D} : \{1, 2, \ldots, K\} \to \mathcal{C}$$

operations, though this is usually not necessary. Each inverse image $Q^{-1}(y_i)$ is called a *cell*, and the cells together form the *partition* induced by the quantizer.

Except in the degenerate case where the source takes on no more than K values, the input and output of the quantizer will differ. This difference is called the *quantization error*. Naturally, this error should be made small, so we minimize a nonnegative measure of the error. The most common distortion measure is the mean-squared error (MSE), defined for the random variable X and the quantizer Q by

$$D = E[(X - Q(X))^2],$$

where $E[\,\cdot\,]$ denotes expectation. If unspecified, MSE distortion will be assumed.

With the MSE distortion measure, there are two simple necessary conditions for an optimal quantizer:

- *Nearest neighbor encoding:* Consider the codebook to be fixed. In encoding a source sample x, one should choose the codebook element closest to x. This is written as

  $$Q(x) = \operatorname*{argmin}_{y_i \in \mathcal{C}} |x - y_i|.$$

  In this case the cell is called a *Voronoi cell*.

- *Centroid condition:* With the encoder mapping fixed, the decoder minimizes the distortion by decoding to the average value of the cell:

  $$\mathcal{D}(y_i) = E[x \mid Q(x) = y_i].$$

Except for a few examples with simple probability density functions, optimal quantizer design cannot be done analytically. Lloyd [128] and Max [133] independently suggested that a quantizer be designed iteratively by alternating enforcements of the above conditions. A quantizer designed in this manner is called a *Lloyd–Max quantizer*.

The output of a Lloyd–Max quantizer is a random variable taking values in a discrete set of size K. As discussed in Section 1.1.1, the average number of bits required to transmit such a random variable can usually be reduced by entropy coding. There is no reason to believe that cascading a Lloyd–Max quantizer and an entropy coder would give the best trade-off between average rate and distortion; in fact, a joint design of the quantizer and entropy coder is beneficial. The result of a joint design is called an *entropy-constrained scalar quantizer*. Its first numerical investigation was by Wood [211]. Optimality conditions for MSE distortion were provided by Berger [14].

Vector quantization is "merely" the extension of scalar quantization to multidimensional domains, but the implications of this extension are profound. The source coding of any "finite extent" discrete domain source—like an image, audio segment, or video segment—can be considered a single vector quantization operation. This

	Scalar quantization	Vector quantization
Space complexity:		
codebook size	$N2^R$	2^{NR}
Time complexity:		
distance calculations	$N2^R$	2^{NR}
complexity of distance calculation	$O(1)$	$O(N)$
overall	$O(N2^R)$	$O(N2^{NR})$

Table 1.1: Complexity comparison between scalar and vector quantization. N represents the dimensionality of the source and R the rate in bits per source component. It is assumed that different scalar quantizers are used for each vector component; otherwise, codebook size is reduced from $N2^R$ to 2^R.

is in contrast to having many scalar quantization operations, say one for each sample of an audio signal. Vector quantization is by definition the "ultimate" coding scheme because it includes all forms of source coding.

Vector quantization (VQ) is actually the structure used by Shannon [172] in his theoretical studies of coding with a fidelity criterion, thus it is as old rate–distortion theory. Interest in VQ rose in the 1980's as the use of digital computers made its implementation more practical. It should be noted, however, that unconstrained VQ is feasible for vectors of *much* lower dimension than an entire source realization (image, audio segment, video segment) described above.

The two scenarios for scalar quantization considered above, optimum for a fixed codebook size and entropy-constrained, have been addressed for VQ as well. For the first problem, a design algorithm which generalizes the Lloyd–Max iteration was given by Linde, Buzo, and Gray [124]. It is called the generalized Lloyd, or LBG for the authors, algorithm. An algorithm for entropy-constrained VQ (ECVQ) design was given by Chou, Lookabaugh, and Gray [29].

The main drawback of VQ is its complexity, both in time (number of operations) and space (amount of memory). Unless some structure is imposed on the codebook, each codeword must be stored independently. For a codebook of size K, the nearest neighbor encoding process requires K distance computations and the selection of the smallest of the K distances. Table 1.1 summarizes a complexity comparison between scalar and vector quantization for an N-dimensional source coded at R bits per component.[8] It is assumed that the scalar codebooks used for each component are distinct but equal in size.

Many techniques for reducing time and space complexity by placing constraints on the codebook design (using a suboptimal codebook) or by replacing the nearest neighbor encoding rule (using suboptimal encoding) have been proposed. The most important variants are described in [60], a comprehensive text on VQ.

Viewing a source coder as a vector quantizer can lead to valuable insights. Examples of this will be seen in Chapters 2 and 5, where the geometries of partitionings are used to understand quantized overcomplete expansions and multiple description coders, respectively. In the first case, we will find that a seemingly reasonable reconstruction algorithm can be far from optimal. The second case leads to techniques for JSC coding.

The design of optimal quantizers can rarely be completed analytically. While this is a practical problem in its own right, it poses distinct difficulty in the design of systems consisting of more than just a quantizer. It is useful to be able to model the quantization process in a general way. Three approaches that lead to tractable

[8] It is conventional to measure rates on a per component basis (instead of per vector) because one is often free to manipulate the vector length through (re-)blocking.

analysis are to assume the quantization is fine, uniform, or dithered. These are frequently combined and are described below with emphasis on a few oft-used, simple expressions.

1.1.2.1 Fine quantization approximations

Consider first, Lloyd–Max quantization, *i.e.*, scalar quantization without entropy coding. If the size of the codebook is large and the probability density function (p.d.f.) of the source is smooth, then the p.d.f. is approximately constant over each quantization cell. This approximation, attributed to Bennett [11], facilitates analytical approximations of the distortion as a function of the size of the codebook K and *point density function* (see [60]) of the quantizer. In any cell, the point density function is approximately the width of the cell divided by K.

The optimum point density function is proportional to the cube root of the p.d.f. and yields distortion of

$$D \approx \frac{1}{12K^2}\left(\int_{-\infty}^{\infty} f_X(x)^{1/3}\,dx\right)^3.$$

For a Gaussian random variable with variance σ^2, this becomes

$$D \approx \frac{\sqrt{3}\pi\sigma^2}{2K^2}.$$

Upon relating the rate to the number of cells with $R = \log_2 K$, we obtain

$$D_{\text{Gaussian,Lloyd}--\text{Max}} \approx \frac{\sqrt{3}\pi}{2}\sigma^2 2^{-2R}. \tag{1.2}$$

Bennett's approximation can be used to analyze entropy-coded scalar quantization (ECSQ) as well. The results are rather startling: Gish and Pierce [61] showed that under weak assumptions on the source p.d.f. and on the distortion criterion, the asymptotically optimal point density is constant on the support of the source p.d.f. For a Gaussian source, one obtains

$$D_{\text{ECSQ}} \approx \frac{\pi e}{6}\sigma^2 2^{-2R}. \tag{1.3}$$

For comparison, the distortion–rate function for a *memoryless* Gaussian source with the same variance is

$$D(R) = \sigma^2 2^{-2R}. \tag{1.4}$$

One could call this the limiting performance of infinite-dimensional ECVQ.

1.1.2.2 Uniform quantization

Uniform quantization is usually not optimal, but it is very common in practice because of its simplicity. A *uniform quantizer* is distinguished by having equally spaced codewords. Two types of uniform quantizers are of interest: one with a finite number of codewords (a *bounded* quantizer) and one with a countably infinite number of codewords (an *unbounded* quantizer).

A bounded uniform quantizer is characterized by the number of cells K, the quantization step size Δ, and any single codebook value, say y_1. The remaining codebook entries are given by $y_k = y_1 + (k-1)\Delta$. The interval $\mathcal{I}_{\text{granular}} = [y_1 - \Delta/2, y_K + \Delta/2]$ is called the granular region. Source samples in this interval will be

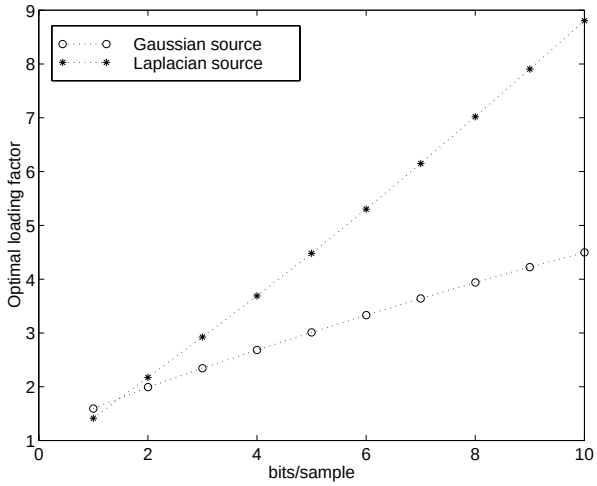


Figure 1.3: Optimal loading factors for bounded uniform quantization of Gaussian and Laplacian sources.

approximated within $\pm\Delta/2$ by their quantized values.[9] Samples in the overload region $\mathcal{I}_{\text{overload}} = \mathbb{R} \setminus \mathcal{I}_{\text{granular}}$ face quantization error greater than $\Delta/2$ in absolute value. Overload distortion and granular distortion refer to the expected value of each type of distortion.

For a fixed value of K, there is a trade-off in selecting the value of Δ. Decreasing Δ diminishes the granular distortion, but also contracts the granular region, enhancing the overload distortion.[10] Similarly, attempting to minimize the overload distortion enhances the granular distortion. The size of Δ can be described by a dimensionless quantity called the loading factor. The *loading factor* is the length of $\mathcal{I}_{\text{granular}}$ divided by twice the standard deviation of the source. For Gaussian and Laplacian sources, the optimal loading factor as a function of the bit rate is shown in Figure 1.3. The difference is due to the heavinesses of the distribution tails.

An unbounded uniform quantizer is described by a quantization step size Δ and an offset $a \in [-\Delta/2, \Delta/2)$. The quantization function is then given by

$$\begin{aligned} Q_{\Delta,a}(x) &= n\Delta - a \text{ if } x \in [(n-\tfrac{1}{2})\Delta - a, (n+\tfrac{1}{2})\Delta - a) \\ &= [x+a]_\Delta - a, \end{aligned} \tag{1.5}$$

where $[\,\cdot\,]_\Delta$ represents rounding to the nearest multiple of Δ. This is most commonly used with a equal to zero or $-\Delta/2$. An unbounded uniform quantizer only makes sense when followed by an entropy code and this is the situation considered.

Except in the small Δ limit, few analytical calculations can be made regarding the performance of entropy-coded unbounded uniform quantization (ECUQ). The use of Bennett approximations for small Δ yields the optimality of ECUQ mentioned previously. At high rates, ECUQ performs within $\frac{1}{2}\log_2(\pi e/6) \approx 0.255$ bits per sample of the rate–distortion function for any *memoryless* source [61]. A numerical study has shown that for a wide range of memoryless sources, ECUQ is within 0.3 bits per sample of the rate–distortion function at all rates [53]. Figure 1.4 provides a comparison for a unit-variance memoryless Gaussian source between ECUQ

[9] Midpoint reconstruction is assumed, though centroid reconstruction could potentially reduce the distortion.
[10] If the overload distortion is not affected, Δ is much too large.

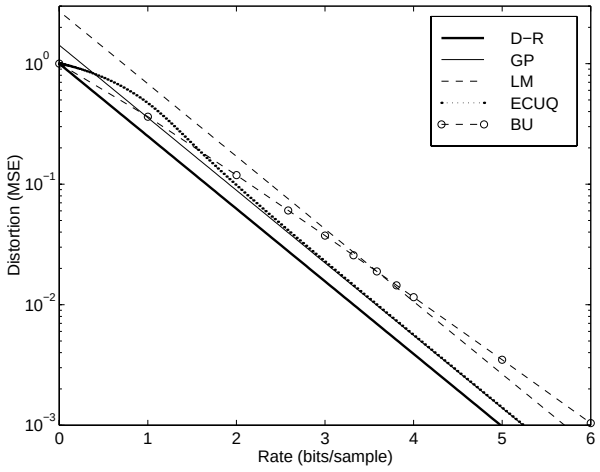


Figure 1.4: Comparison between coding methods, approximations, and rate–distortion bound for memoryless Gaussian source.

with ideal lossless coding, bounded uniform quantization with optimized loading factor, and various bounds and approximations:

- *D-R:* the distortion–rate bound (1.4);
- *GP:* the approximation by Gish and Pierce [61] for the performance of ECUQ (1.3);
- *LM:* the high rate approximation to the performance of a Lloyd–Max quantizer (1.2);
- *ECUQ:* the actual performance of an ECUQ with $a = 0$ where ideal entropy coding is assumed, *i.e.*, rate has been measured by entropy;
- *BU:* the actual performance of a bounded uniform quantizer with optimized loading factor.

Unbounded uniform quantization is convenient for studies of the statistical properties of quantization error. It is common in communications literature to assume that the quantization error is signal-independent, uncorrelated (white), and uniformly distributed on $[-\Delta/2, \Delta/2]$. This assumption is clearly wrong because the quantization error is a deterministic function of the signal. Nevertheless, it is a useful approximation and is justifiable in a precise sense when Δ is small and the probability distribution between pairs of input samples is given by a smooth p.d.f. Results along these lines are originally due to Bennett [11] and are surveyed in [84] and [203]. This type of quantization also makes possible closed-form expressions for the moments of quantized signals [27] (see Chapter 3).

White quantization noise assumptions have been largely avoided in this thesis. Instead, where the statistical properties of quantization noise are significant, in Chapter 3 and in a small part Chapter 2, subtractive dithered uniform quantization has been used. This type of quantization precisely satisfies the white quantization noise model.

1.1.2.3 Dithered quantization

Dithering is the addition of a random signal prior to quantization. The dither signal may or may not be subtracted after quantization, yielding *subtractive* and *nonsubtractive dithered quantization*, respectively. The purpose of this operation is to manipulate the statistical properties of the quantization noise, *e.g.*, to make them independent of the signal.

In subtractive dithered quantization (SDQ), we will assume an unbounded uniform quantizer and a white dither signal uniformly distributed on $[-\Delta/2, \Delta/2)$. This SDQ is equivalent to the use of (1.5) with randomized offset a. The quantization noise is signal-independent, white, and uniformly distributed, regardless of the signal p.d.f. and Δ. These properties will be used to advantage in Chapters 2 and 3.

The first use of SDQ was by Roberts [164] to improve the perceptual quality of PCM-encoded images. Subtractive and nonsubtractive dithered quantization are surveyed in [126, 85].

1.1.3 Series Signal Expansions and Transform Coding

Series expansions are the central operations of signal processing. They date back almost two centuries to the work of Fourier [57] on expansions in terms of sines and cosines. The classical techniques of Fourier analysis occupy a prominent position in engineering curricula and thus will not be reviewed here. Recommended references include [112, 145, 195].

The signals considered in this thesis will be of the discrete-parameter type and have finite extent; thus, they are finite-dimensional vectors. Fourier analysis applies to such signals, specifically the discrete Fourier transform (DFT), but many of the technicalities of harmonic analysis do not apply or are trivially checked. For example, convergence of transforms is not an issue.

Consider the signal space $\mathcal{H} = \mathbb{R}^N$. With the inner product

$$\langle x, y \rangle = \sum_{k=1}^{N} x_i y_i = x^T y = y^T x,$$

this is a separable Hilbert space. Any set of N linearly independent vectors $\{\varphi_k\}_{k=1}^N \subset \mathcal{H}$ specifies a transform $T : H \to H$ through

$$y_k = (Tx)_k = \langle \varphi_k, x \rangle, \qquad \text{for} \quad k = 1, 2, \ldots, N. \tag{1.6}$$

This invertible linear transform is described by the matrix $T = [\varphi_1 \ \varphi_2 \ \cdots \ \varphi_N]^T$. Each y_i is called a *transform coefficient* and the vector y is said to be in the *transform domain*. This is a *series expansion* of the signal because

$$x = \sum_{k=1}^{N} y_k \widetilde{\varphi}_k, \tag{1.7}$$

where $\widetilde{\varphi}_k = (T^T T)^{-1} \varphi_k$, $k = 1, 2, \ldots, N$.

The motivating principle of transform coding is that simple coding may be more effective in the transform domain than in the original signal space. In Kramer and Mathews' case [114], "simple coding" corresponds to retaining only a subset of the signals. When they chose the transform well, keeping $M < N$ components in the transform domain was much more effective than keeping the same number of original vocoder signals. If they allowed arbitrary source coding after the transform, the transform would make no difference; since it is invertible, the information present has not changed.

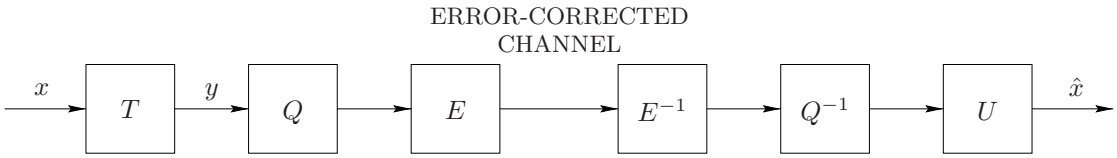


Figure 1.5: Substructure of a communication system with a transform coder. The transform T is a linear function $\mathbb{R}^N \to \mathbb{R}^N$. Q is a scalar quantizer mapping from $\mathbb{R}^N$ to $\mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_N$, where $\mathcal{I}_i$ is the set of codebook indices for the ith component quantizer. E is an entropy coder. The reconstruction generally uses a linear transform $U = T^{-1}$.

The modern use of transform coding is schematically captured in Figure 1.5. The linear transform T is followed by a scalar quantizer Q and possibly by an entropy coder E. If the quantizer were allowed to be a vector quantizer, it could subsume the transform, and the typical problems of design and encoding complexity of VQ would appear.

1.1.3.1 Optimality and Karhunen–Loève transforms

The optimal design of the transform was originally approached under the following conditions:

- The source is Gaussian with mean zero and a known, constant, covariance matrix $R_x = E[xx^T]$.
- Distortion is measured by MSE.
- No entropy coding is used.
- The scalar quantizers are Lloyd–Max quantizers with numbers of levels for each quantizer $\{K_i\}$ controlled by the designer. The total number of cells $\prod_i K_i$ is constrained.[11]

Under these assumptions, Huang and Schultheiss [100] first showed that the decoder should use the inverse of the transform used in the encoder (in Figure 1.5, $U = T^{-1}$). Though this may seem obvious, because of the presence of quantization, it does not go without saying. The proof utilizes statistical properties of Lloyd–Max quantization error.

Next, they showed that the optimal transform matrix T is the transpose of an orthogonal diagonalizing similarity transformation for R_x. In other words, T should satisfy

$$TR_xT^T = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N), \tag{1.8}$$

where the λ_i's are eigenvalues of R_x. For concreteness, we assume $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ (though T is still not unique). A transform satisfying (1.8) is called a *Karhunen–Loève transform* (KLT) of the source. The optimality of this transform does not require specific values for the K_i's, but rather the mild condition $K_1 \geq K_2 \geq \cdots \geq K_N$.[12]

Given a constraint $\prod_i K_i \leq K$, finding the set $\{K_i\}$ that minimizes the distortion is typically difficult because the K_i's must be positive integers. If we lift this constraint, an approximate solution can be found via

[11] Under the constraint of scalar quantizing entropy coding, the choice of optimizing the quantizers individually seems obvious. This is assumed in [100] but proven to be optimal in [167].

[12] Note that in this scenario, the KLT is optimal with realizable quantizers and *without* fine quantization approximations.

calculus. Using the high rate approximation (1.2), one can find the optimal allocation

$$K_i = \left(\frac{K^2 \lambda_i^N}{\prod_{k=1}^N \lambda_k} \right)^{1/2N}.$$

This can be written more conveniently in terms of bits, yielding

$$r_i = R + \frac{1}{2} \log_2 \frac{\lambda_i}{(\prod_{k=1}^N \lambda_k)^{1/N}} \tag{1.9}$$

where r_i bits are allocated to the ith quantizer and $R = (\log_2 K)/N$ is the average bit rate. This bit allocation yields overall performance

$$D_{\text{KLT,LM}} \approx \frac{\sqrt{3}\pi}{2} \prod_{i=1}^N \lambda_i \cdot 2^{-2R}.$$

A similar analysis for high-rate ECSQ or ECUQ yields the same bit allocation since the only difference between (1.2) and (1.3) is a multiplicative constant. The overall performance is given by

$$D_{\text{KLT,ECUQ}} \approx \frac{\pi e}{6} \prod_{i=1}^N \lambda_i \cdot 2^{-2R}. \tag{1.10}$$

The reduction in distortion for a given rate, as compared to a system using the same type of scalar quantization (high-rate ECSQ or ECUQ) is by a factor of

$$\frac{\left(\prod_{i=1}^N \sigma_{x_i}^2\right)^{1/N}}{\left(\prod_{i=1}^N \sigma_{y_i}^2\right)^{1/N}} = \frac{\left(\prod_{i=1}^N \sigma_{x_i}^2\right)^{1/N}}{\left(\prod_{i=1}^N \lambda_i\right)^{1/N}}, \tag{1.11}$$

where $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ refer to the powers of the ith components of x and y, respectively. This is called the *coding gain* of the KLT. If the source vectors are blocks of length N from a wide-sense stationary source, R_x is a Toeplitz matrix, and the coding gain can be put in the more common form

$$\frac{\frac{1}{N}\sum_{i=1}^N \lambda_i}{\left(\prod_{i=1}^N \lambda_i\right)^{1/N}}.$$

The optimality of the KLT is intuitively appealing. The KLT gives transform coefficients which are uncorrelated (and since they are Gaussian, independent): $y = Tx$ and

$$R_y = E[yy^T] = E[Txx^TT^T] = TR_xT^T = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N).$$

Since the processing following the transform is done separately for each coefficient, what could be better than making the coefficients independent? In fact, this independence allows us to extend (1.4) to get the distortion–rate function for a memoryless Gaussian vector source with correlation matrix R_x. Optimally allocating the rate amongst the transform coefficients gives, for high rates,[13]

$$D_{\text{vector}}(R) = \prod_{i=1}^N \lambda_i \cdot 2^{-2R}.$$

The geometric effect of the KLT is to align the principal axes of the source p.d.f. with the standard basis. This is shown in Figure 1.6. On the left is a level-curve representation of the p.d.f. of x. If y is a KLT domain version of x, its p.d.f. will be as shown on the right.

[13] Recall that rates and distortions are measured per component.

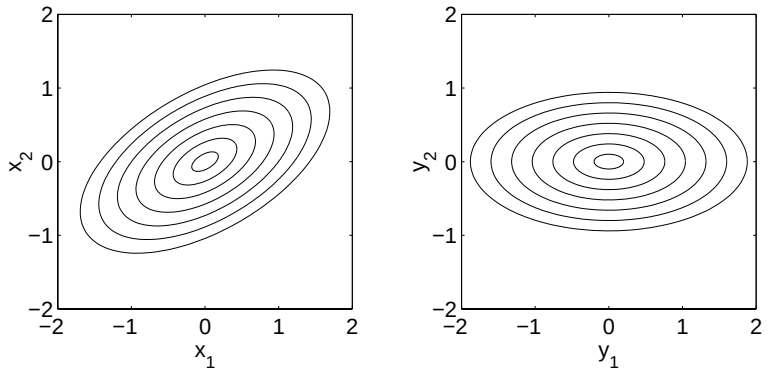


Figure 1.6: Geometric effect of the Karhunen–Loève transform. On the left, level curves of an ellipsoidal source p.d.f. On the right, as a result of the KLT, the principal axes of the p.d.f. are aligned with the standard basis.

1.1.3.2 Karhunen–Loève transforms and $R(D)$ for Gaussian processes

The KLT has a role, not only in practical coding of Gaussian sources as demonstrated in the previous section, but also in obtaining theoretical bounds. Consider again a Gaussian source x with mean zero and correlation matrix R_x. Let T be a KLT for x, so the correlation matrix of $y = Tx$ is given by $R_y = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$.

Directly computing the rate–distortion function of x requires a minimization of mutual information over all conditional distributions $p(\hat{x} \mid x)$ for which the joint distribution $p(x, \hat{x}) = p(x)p(\hat{x} \mid x)$ satisfies a distortion constraint [34]. This computation is complicated by the fact that $p(x)$ is not separable. On the other hand, computing the rate–distortion function of y is relatively easy because it is a product source with independent components. The rate–distortion is given by a simple Lagrangian optimization subject to the availability of the needed slopes [13, Cor. 2.8.3]. The solution is

$$R(D) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \log \frac{\lambda_i}{D_i}, \tag{1.12}$$

where

$$D_i = \begin{cases} \lambda, & \text{if } \lambda < \lambda_i, \\ \lambda_i, & \text{if } \lambda \geq \lambda_i, \end{cases} \tag{1.13}$$

and λ is chosen so that $N^{-1} \sum_{i=1}^{N} D_i = D$. This is the result of "reverse water-filling" rate allocation, whereby components with variances less than λ are allocated no bits and the remaining components are allocated the rate needed to make the component distortion equal λ [34]. The "water level" λ is chosen to make the average distortion equal D.

Since an invertible transform does not change mutual information and a unitary transform does not change Euclidean norm, x and y must have identical rate–distortion functions. Thus (1.12)–(1.13) is also the rate distortion function of x. This is more than an algebraic trick; it leads to an alternative method to asymptotically achieve optimal performance.

Suppose the following: L vectors from the source are coded simultaneously, and one can design a sequence of optimal LN-dimensional fixed-rate vector quantizers to operate directly on x for $L = 1, 2 \ldots$. As $L \to \infty$, the performance approaches arbitrarily close to $R(D)$, but each vector quantizer in the sequence depends on R_x.

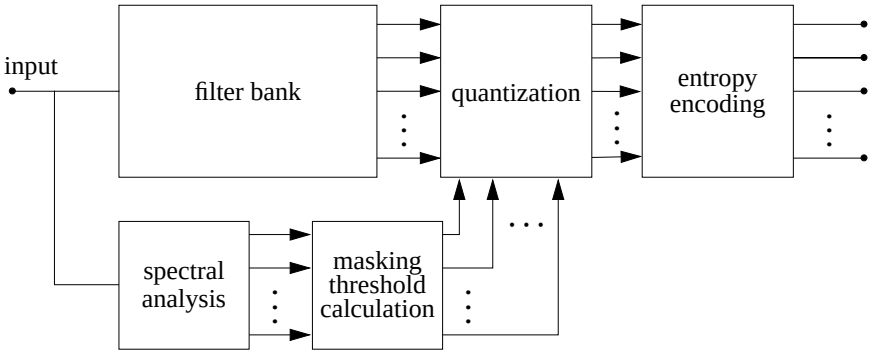


Figure 1.7: A perceptual audio coder.

The representation of x through y provides a conceptually simpler way to approach $R(D)$. All we need is sequences of vector quantizers for a *white* Gaussian source with unit variance. These vector quantizers are then applied to blocks of size L formed from like transform coefficients, with proper scaling and rate allocation.[14] The performance again approaches arbitrarily close to $R(D)$ as $L \to \infty$.

This example reinforces the motivating doctrine of transform coding: though a transform is not necessary to achieve good performance, it may simplify the other processing (quantization and entropy coding, for example) needed to achieve certain prescribed performance. Conversely, with a complexity bound, transform coding may give the best performance. Throughout this thesis, the combination of transform and scalar quantization should not be considered an alternative to vector quantization of the same dimension, but rather an alternative to scalar quantization with no transform. In many situations, a transform coding system can realize the space-filling advantage of vector quantization by using vector quantization of like transform coefficients, while still gaining from the transform.

1.1.4 Applications

It is hard to overstate the ubiquity of transform coding in lossy compression, though the techniques used are often more complicated than the simple structure motivated by theory. Popular methods for compression of audio, images, and video will be described briefly to highlight their use of linear transforms and scalar quantization.

Audio coders that exploit understanding of human hearing to optimize perceived quality, instead of a simple distortion measure like MSE, are very popular [105, 176]. A typical structure for such a *perceptual* audio coder is shown in Figure 1.7. A linear transform, implemented by a filter bank [195], is used to produce a subband representation of the audio signal. The subbands are approximately uncorrelated. The quantization step sizes used in each subband are selected to minimize the perceptibility of the quantization noise. These step sizes depend on a spectral analysis of the signal.

The most common technique for compression of continuous tone images is given by the JPEG standard [198, 199, 153]. A JPEG image coder is typically implemented in the form shown in Figure 1.8. A linear transform, in this case a two-dimensional discrete cosine transform (DCT) [159], is applied to 8×8 blocks from

[14] Here transform coefficients are termed *like* if they occupy the same position in different temporal blocks.

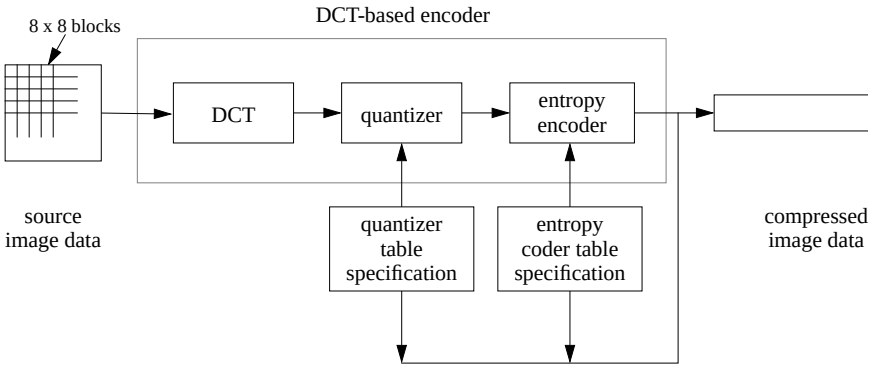


Figure 1.8: A JPEG image coder.

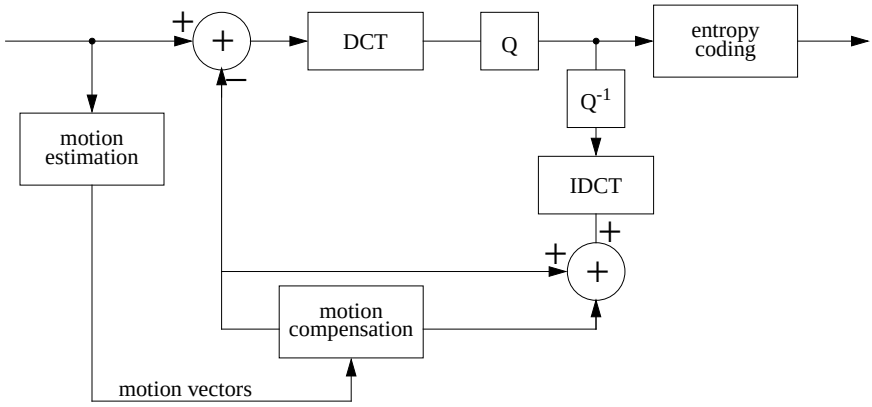


Figure 1.9: A hybrid motion-compensated predictive DCT video coder.

the image. The transform coefficients are scalar quantized and jointly entropy coded.

Many video coding techniques incorporate coding similar to JPEG still image coding. The MPEG standard [116] and other popular algorithms use a hybrid between predictive coding and DCT coding: pixel values are predicted temporally, and the residual from this prediction is DCT coded. A schematic is shown in Figure 1.9.

1.2 Thesis Themes

Rather than focusing on a single problem, this thesis explores a variety of closely related problems which may be solved by extensions of traditional transform coding. There are several common themes that unite the various results. In the course of exposing these themes, some of the main results of the thesis will be previewed.

A prototypical communication system employing a transform coder will have separate transform, quantization, entropy coder, and channel coder blocks. All the blocks are time-invariant, and though they may be designed jointly, they operate independently. The transform itself is linear and square. The topics explored here include the use of nonsquare, nonlinear, and adaptive transforms; allowing interaction between the transform

and quantization blocks; and merging the channel coding function with the other processes.

1.2.1 Redundant Signal Expansion

One interpretation of the optimality of the KLT is that it removes redundancy. Of course, in light of the fact that the transform is invertible, this is an imprecise statement. There is no statistical redundancy if one subset of the scalar components gives no information about a disjoint subset. Figure 1.6 provides a graphical demonstration for a Gaussian source. The correlation between x_1 and x_2 is evidenced by dependence of $E[x_2 \mid x_1]$ on x_1. On the other hand, $E[y_2 \mid y_1] = 0$, independent of y_1. Since the source is Gaussian, we can make a stronger statement: The conditional distribution of y_2 given y_1 is independent of y_1, so there is no redundancy. In applications where the KLT is unknown (perhaps ill-defined) or inconvenient, a discrete cosine transform or subband decomposition may remove redundancy in a similar sense. This is a well-known empirical fact [159] and is also justified for large block sizes by an asymptotic equivalence between frequency-selective unitary transforms [87, 83].

What might be the benefit of leaving redundancy in the signal or introducing redundancy to a signal with independent components? This redundancy can be used to combat channel impairments [89, 21, 175, 165]. In fact, channel coding is the introduction of redundancy such that impairments (in the discrete case: errors or erasures) can be mitigated. This redundancy is different from the previously discussed redundancy in that it is deterministic and in the "bit domain," instead of the "signal domain."

Signal-space redundancy appears in a few ways in this thesis. In Chapter 2, consideration is extended to overcomplete linear transforms. With reference to (1.6), this means using a set $\{\varphi_k\}_{k=1}^{M}$ which spans $\mathbb{R}^N$, but is linearly dependent. (Alternatively, the transform is described by a matrix with more rows than columns.) Overcomplete transforms produce transform coefficients with a deterministic redundancy. The transform coefficients lie in a subspace of $\mathbb{R}^M$ and thus must satisfy one or more nontrivial linear equations. The quantization of the transform coefficients considerably complicates the redundancy.

One fundamental question is: How well does a quantized overcomplete linear expansion describe a signal? This is answered in various specific settings in Chapter 2. It is revealed that careless modeling of the quantization noise leads to suboptimal performance.

Quantized overcomplete expansions arise again in Chapter 5 in the context of joint source–channel (JSC) coding for erasure channels. This provides a specific instance where redundancy introduced in the signal domain can be used as channel coding. Another JSC technique is also introduced which uses signal-domain redundancy of the statistical type.

1.2.2 Adaptive Signal Expansion

In the basic theory of transform coding, it is assumed that the source is a stationary random process and that the coding procedure is linear (excluding quantization noise) and time-invariant. This thesis includes explorations of methods for producing nonlinear and adaptive signal expansions. To most, "adaptive" immediately brings to mind memory and time-variance. While this sense of adaptation is also considered, the first form of adaptive signal expansion considered is actually memoryless and time-invariant.

Suppose instead of forming a signal expansion as in (1.7), one has

$$x \approx \sum_{k=1}^{p} y_k \widetilde{\varphi}_{j_k}, \qquad \text{with} \quad p < n. \tag{1.14}$$

The important distinction is that instead of using all the φ_k's, one uses a subset in an *a priori* unspecified order. Using such an expansion allows signal adaptivity within a memoryless, time-invariant system. Unless the same j_k sequence is used every time, the y_k's are a nonlinear function of the signal.

One method for choosing the y_k's and j_k's (described in Section 2.3.1) is called matching pursuit (MP) [131]. A version of this algorithm including quantization is presented in Section 2.3.2. This algorithm, called quantized matching pursuit (QMP), is the natural way to apply MP to source coding. The melding of transform and quantization in QMP makes its analysis interesting.

Adaptation in the traditional sense is considered in Chapters 3 and 4. In the former, a method for transform adaptation which uses only quantized data is proposed. Adaptation based on quantized data is attractive because it eliminates the need for describing the adaptation over a side information channel. This method leads to a source coder which is universal for memoryless Gaussian vector sources with unknown correlation matrices.

The computation of transforms for an adaptive system is considered in Chapter 4. An analogy between adapting transforms and finite impulse response (FIR) Wiener filters motivates the development of a new class of algorithms for transform adaptation.

1.2.3 Computational Optimization

The goal of a communications engineer is not to just design a system which meets certain performance requirements, but to quickly design an inexpensive system that meets the requirements. Information theory can contribute to the speed of design by suggesting whether or not the specifications are close to fundamental limits. But since most key results in information theory are nonconstructive, the design of an implementable system may still be difficult.

Results which indicate the best possible performance *with constrained resources* are needed. A commonly constrained resource is computational power measured, say, in arithmetic operations per source sample. This type of constraint motivates various system design choices; for example, DCT instead of KLT, or scalar instead of vector quantization. In forming signal-adapted expansions as in (1.14), the use of matching pursuit is motivated by the high computational complexity of finding more sparse expansions. The recursive consistent estimation algorithm of Appendix 2.C is another example of a "suboptimal but good" algorithm. Complexity concerns arise again in Appendix 5.B, where transform coding with discrete transforms is considered. These transforms may be cheap to compute because low precision arithmetic suffices, and they offer the potential for computationally simplified entropy coding.

The decision to replace a theoretically optimal technique with a computationally simple, suboptimal one should be principled. A theory upon which to base these design choices is presented in Chapter 6.

Chapter 2

Quantized Frame Expansions

LINEAR TRANSFORMS and expansions are the fundamental mathematical tools of signal processing, yet the properties of linear expansions in the presence of coefficient quantization are not fully understood. These properties are most intricate when signal representations are with respect to redundant, or overcomplete, sets of vectors. This chapter considers the effects of quantization in overcomplete finite linear expansions. Both fixed and adaptive basis methods are studied. Although the adaptive basis method represents an input vector as a linear combination of elements from a representation set, it is in fact a nonlinear mapping. While many other issues are explored, the unifying theme is that consistent reconstruction methods [184] give considerable improvement over linear reconstruction methods.

2.1 Introduction

Consider the expansion–quantization–reconstruction scenario depicted in Figure 2.1. A vector $x \in \mathbb{C}^N$ is left multiplied by a matrix $F \in \mathbb{C}^{M \times N}$ of rank N to get $y \in \mathbb{C}^M$. The transformed source vector y is scalar quantized, *i.e.*, quantized with a quantizer which acts separately on each component of y, to get $\hat{y}$. As shown in Section 2.2.1.2, this type of representation arises naturally in simple oversampled A/D conversion. In general, this sort of representation may be desirable when many coarse measurements can be made easily, but precise measurements are difficult to make. How can one best estimate x from $\hat{y}$? How does the quality of the estimate $\hat{x}$ depend on the properties of F, in particular its number of rows M? These are the fundamental questions addressed in Section 2.2.

To put this in a solid framework, the basic properties of frames are reviewed and a new result on the tightness of random frames is proven. It will be shown that the quality of reconstruction can be improved by using deterministic properties of quantization; in particular, the boundedness of quantization noise. The alternative is to consider quantization the addition of signal-independent white noise specified only by its variance. The relationship between the redundancy of the frame and the minimum possible reconstruction error is explored.

Without sophisticated coding, a nonadaptive overcomplete expansion can be a very inefficient representation. In the context of Figure 2.1, coding $\hat{y}$ may be an inefficient way to represent x. But could we get a

This chapter includes research conducted jointly with Martin Vetterli and Nguyen T. Thao [77, 78, 79]; Martin Vetterli [71, 74]; and Sundeep Rangan [158].

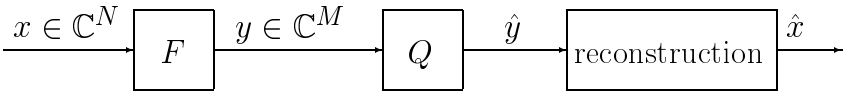


Figure 2.1: Block diagram of reconstruction from quantized frame expansion.

good representation by choosing a few components of $\hat{y}$, *a posteriori*, which best describe x? This question is related to adaptive basis techniques described in Section 2.3.

The use of a greedy successive approximation algorithm for finding sparse linear representations with respect to an overcomplete set is studied in Section 2.3. This algorithm, called matching pursuit (MP) [131], has recently been applied to image coding [12, 74] and video coding [141, 194, 107, 140], which both inherently require coarse coefficient quantization. However, the present work is the first to describe the qualitative effects of coefficient quantization in matching pursuit. In particular, as in Section 2.2, we will find that reconstruction can be improved by consistent reconstruction techniques.

Except where noted, we consider vectors in a finite-dimensional Hilbert space $H = \mathbb{R}^N$ or $\mathbb{C}^N$. For $x, y \in H$, we use the inner product $\langle x, y\rangle = x^T\bar{y}$ and the norm derived from the inner product through $\|x\| = \langle x, x\rangle^{1/2}$. $\mathcal{N}(\mu, \Lambda)$ is used to denote the Normal distribution with mean μ and covariance matrix Λ. The term squared error (SE) is used for the square of the norm of the difference between a vector and an estimate of the vector. The term mean squared error (MSE) is reserved for the ensemble average of SE, or expected SE.

2.2 Nonadaptive Expansions

This section describes frames, which provide a general framework for studying nonadaptive linear transforms. Frames were introduced by Duffin and Schaeffer [44] in the context of non-harmonic Fourier series. Recent interest in frames has been spurred by their utility in analyzing the discretization of continuous wavelet transforms [94, 36, 37] and time-frequency decompositions [137]. The motivation here is to understand quantization effects and efficient representations.

Section 2.2.1 begins with definitions and examples of frames. It concludes with a theorem on the tightness of random frames and a discussion of that result. Section 2.2.2 begins with a review of reconstruction from exactly known frame coefficients. The remainder of the section gives new results on reconstruction from quantized frame coefficients. Most previous work on frame expansions is predicated either on exact knowledge of coefficients or on coefficient degradation by white additive noise. For example, Munch [137] considered a particular type of frame and assumed the coefficients were subject to a stationary noise. This chapter, on the other hand, is in the same spirit as [182, 184, 183, 185, 35] in that it utilizes the deterministic qualities of quantization.

2.2.1 Frames

2.2.1.1 Definitions and basics

This subsection is largely adapted from [37, Ch. 3]. Some definitions and notations have been simplified because we are limiting our attention to $H = \mathbb{R}^N$ or $\mathbb{C}^N$.

Let $\Phi = \{\varphi_k\}_{k\in K} \subset H$, where K is a countable index set. Φ is called a *frame* if there exist $A > 0$ and $B < \infty$ such that

$$A\|x\|^2 \leq \sum_{k\in K} |\langle x, \varphi_k\rangle|^2 \leq B\|x\|^2, \qquad \text{for all } x \in H. \tag{2.1}$$

A and B are called the *frame bounds*. The cardinality of K is denoted by M. The lower bound in (2.1) is equivalent to requiring that Φ spans H. Thus a frame will always have $M \geq N$. Also, notice that one can choose $B = \sum_{k\in K} \|\varphi_k\|^2$ whenever $M < \infty$. We will refer to $r = M/N$ as the *redundancy* of the frame. A frame Φ is called a *tight frame* if the frame bounds can be taken to be equal. It is easy to verify that if Φ is a tight frame with $\|\varphi_k\| = 1$ for all $k \in K$, then $A = r$. In particular, an orthonormal basis is a tight frame consisting of unit vectors with $r = 1$.

Given a frame $\Phi = \{\varphi_k\}_{k\in K}$ in H, the associated *frame operator* F is the linear operator from H to $\mathbb{C}^M$ defined by

$$(Fx)_k = \langle x, \varphi_k\rangle. \tag{2.2}$$

Since H is finite-dimensional, this operation is a matrix multiplication where F is a matrix with kth row equal to φ_k^*. Using the frame operator, (2.1) can be rewritten as

$$AI_N \;\leq\; F^*F \;\leq\; BI_N, \tag{2.3}$$

where I_N is the $N \times N$ identity matrix. (The matrix inequality $AI_N \leq F^*F$ means that $F^*F - AI_N$ is a positive semidefinite matrix.) In this notation, $F^*F = AI_N$ if and only if Φ is a tight frame. From (2.3) we can immediately conclude that the eigenvalues of F^*F lie in the interval $[A, B]$; in the tight frame case, all of the eigenvalues are equal. This gives a computational procedure for finding frame bounds. Since it is conventional to assume A is chosen as large as possible and B is chosen as small as possible, we will sometimes take the minimum and maximum eigenvalues of F^*F to be the frame bounds. Note that it also follows from (2.3) that F^*F is invertible because all of its eigenvalues are nonzero, and furthermore

$$B^{-1}I_N \;\leq\; (F^*F)^{-1} \;\leq\; A^{-1}I_N. \tag{2.4}$$

The *dual frame* of Φ is defined as $\widetilde{\Phi} = \{\widetilde{\varphi}_k\}_{k\in K}$, where

$$\widetilde{\varphi}_k = (F^*F)^{-1}\varphi_k, \qquad \text{for all} \quad k \in K. \tag{2.5}$$

$\widetilde{\Phi}$ is itself a frame with frame bounds B^{-1} and A^{-1}.

Since $\text{Span}(\Phi) = H$, any vector $x \in H$ can be written as

$$x = \sum_{k\in K} \alpha_k \varphi_k \tag{2.6}$$

for some set of coefficients $\{\alpha_k\} \subset \mathbb{R}$. If $M > N$, $\{\alpha_k\}$ may not be unique. We refer to (2.6) as a *redundant representation* even though it is not necessary that more than N of the α_k's are nonzero.

2.2.1.2 Example

The question of whether a set of vectors forms a frame is not very interesting in a finite-dimensional space; any finite set of vectors which span the space form a frame. Thus if $M \geq N$ vectors are chosen randomly

with a circularly symmetric distribution on H, they almost surely form a frame.[1] So in some sense, it is easier to find a frame than to give an example of a set of vectors which do not form a frame. This section gives an example of a structured family of frames. Certain properties of these frames are proven in Section 2.2.2.4.

Oversampling of a periodic, bandlimited signal can be viewed as a frame operator applied to the signal, where the frame operator is associated with a tight frame. If the samples are quantized, this is exactly the situation of oversampled A/D conversion [184]. Let $x = [x_1 \;\; x_2 \;\; \cdots \;\; x_N]^T \in \mathbb{R}^N$, with N odd. Define a corresponding continuous-time signal by

$$x_c(t) = x_1 + \sum_{k=1}^{W} \left[x_{2k}\sqrt{2}\cos\frac{2\pi kt}{T} + x_{2k+1}\sqrt{2}\sin\frac{2\pi kt}{T} \right], \tag{2.7}$$

where $W = (N-1)/2$. Any real-valued, T-periodic, bandlimited, continuous-time signal can be written in this form. Let $M \geq N$. Define a sampled version of $x_c(t)$ by $x_d[m] = x_c(\frac{mT}{M})$ and let $y = \left[x_d[0] \;\; x_d[1] \;\; \cdots \;\; x_d[M-1]\right]^T$. Then we have $y = Fx$, where

$$\begin{aligned} F &= \left[\begin{array}{cccc} \varphi_1 & \varphi_2 & \cdots & \varphi_M \end{array} \right]^T, \quad \text{with} \\ \varphi_k &= \left[\begin{array}{ccccccc} 1 & \sqrt{2}\cos\frac{2\pi k}{M} & \sqrt{2}\sin\frac{2\pi k}{M} & \cdots & \sqrt{2}\cos\frac{2\pi Wk}{M} & \sqrt{2}\sin\frac{2\pi Wk}{M} \end{array} \right]^T. \end{aligned} \tag{2.8}$$

Using the orthogonality properties of sine and cosine, it is easy to verify that $F^*F = MI_N$, so F is an operator associated with a tight frame. Pairing terms and using the identity $\cos^2\theta + \sin^2\theta = 1$, we find that each row of F has norm $\sqrt{N}$. Dividing F by $\sqrt{N}$ normalizes the frame and results in a frame bound equal to the redundancy ratio r. Also note that r is the oversampling ratio with respect to the Nyquist sampling frequency.

2.2.1.3 Tightness of random frames

Tight frames constitute an important class of frames. As we will see in Section 2.2.2.1, since a tight frame is self-dual, it has some desirable reconstruction properties. These extend smoothly to nearly tight frames, *i.e.*, frames with B/A close to one. Also, for a tight frame, (2.1) reduces to something similar to Parseval's equality. Thus, a tight frame operator scales the energy of an input by a constant factor A. Furthermore, it is shown in Section 2.2.2.4 that some properties of "typical" frame operators depend only on the redundancy. This motivates our interest in the following theorem.

Theorem 2.1 (Tightness of random frames) *Let $\{\Phi_M\}_{M=N}^{\infty}$ be a sequence of frames in $\mathbb{R}^N$ such that Φ_M is generated by choosing M vectors independently with a uniform distribution on the unit sphere in $\mathbb{R}^N$. Let F_M be the frame operator associated with Φ_M. Then, in the mean squared sense,*

$$\frac{1}{M}{F_M}^*F_M \longrightarrow \frac{1}{N}I_N \;\; \textit{elementwise as} \;\; M \longrightarrow \infty.$$

Proof: See Appendix 2.A.1. □

Theorem 2.1 shows that a sequence of random frames with increasing redundancy will approach a tight frame. Note that although the proof in Appendix 2.A.1 uses an unrelated strategy, the constant $1/N$ is

[1] An infinite set in a finite-dimensional space can form a frame only if the norms of the elements decay appropriately, for otherwise a finite upper frame bound will not exist.

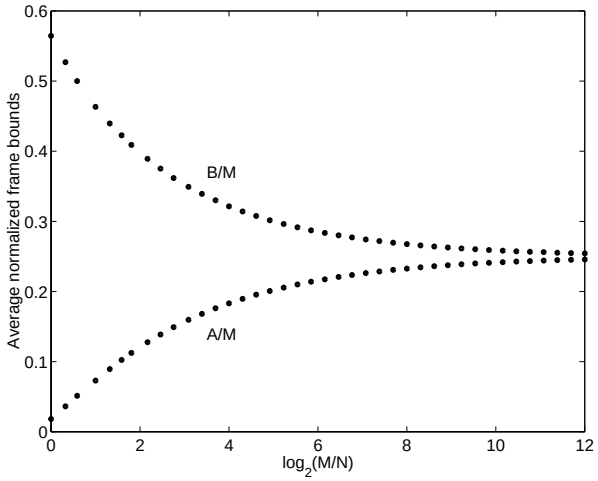


Figure 2.2: Normalized frame bounds for random frames in $\mathbb{R}^4$.

intuitive: If Φ_M is a tight frame with normalized elements, then we have ${F_M}^* F_M = (M/N) I_N$ because the frame bound equals the redundancy of the frame. Numerical experiments were performed to confirm this behavior and observe the rate of convergence. Sequences of frames were generated by successively adding random vectors (chosen according to the appropriate distribution) to existing frames. Results shown in Figure 2.2 are averaged results for 1000 sequences of frames in $\mathbb{R}^4$. Figure 2.2 shows that A/M and B/M converge to $1/N$ and that B/A converges to one.

2.2.2 Reconstruction from Frame Coefficients

One can not rightly call a frame expansion a "signal representation" without considering the viability of reconstructing the original signal. This is the problem addressed presently.

Section 2.2.2.1 reviews the basic properties of reconstructing from (unquantized) frame coefficients. This material is adapted from [37]. The subsequent sections consider the problem of reconstructing an estimate of an original signal from quantized frame coefficients. Classical methods attempt only to minimize the ℓ_2-norm of a residual vector, ignoring bounds on the quantization noise. The approach described here uses deterministic qualities of quantization to arrive at the concept of consistent reconstruction [184]. Consistent reconstruction methods yield smaller reconstruction errors than classical methods.

2.2.2.1 Unquantized case

Let Φ be a frame and assume the notation of Section 2.2.1.1. In this subsection, we consider the problem of recovering x from $\{\langle x, \varphi_k \rangle\}_{k \in K}$. Let $\widetilde{F} : H \to \mathbb{C}^M$ be the frame operator associated with $\widetilde{\Phi}$. It can be shown that $\widetilde{F}^* F = I_N$ [37, Prop. 3.2.3]. Thus, a possible reconstruction formula is given by

$$x \;=\; \widetilde{F}^* F x \;=\; \sum_{k \in K} \langle x, \varphi_k \rangle \widetilde{\varphi}_k.$$

This formula is reminiscent of reconstruction from a Discrete Fourier Transform (DFT) representation, in which case

$$\varphi_k \;=\; \widetilde{\varphi}_k \;=\; \frac{1}{\sqrt{N}}\begin{bmatrix}1 & e^{j2\pi k/N} & \cdots & e^{j2\pi k(N-1)/N}\end{bmatrix}^T \qquad \text{for} \quad k = 0, 1, \ldots, N-1.$$

In the DFT and Inverse DFT, one set of vectors plays the roles of both Φ and $\widetilde{\Phi}$ because it is an orthonormal basis in $\mathbb{C}^N$. Other reconstruction formulas are possible; for details the reader is referred to [37, §3.2].

2.2.2.2 Classical method

We now turn to the question of reconstructing when the frame coefficients $\{\langle x, \varphi_k\rangle\}_{k\in K}$ are degraded in some way. Any mode of degradation is possible, but most practical situations have additive noise due to measurement error or quantization. The latter case is emphasized because of its implications for efficient storage and transmission of information.

Suppose we wish to approximate $x \in \mathbb{C}^N$ given $Fx + \beta$, where $F \in \mathbb{C}^{M\times N}$ is a frame operator and $\beta \in \mathbb{C}^M$ is a zero-mean noise, uncorrelated with x. Notice that $FH = \mathrm{Ran}(F)$ is an N-dimensional subspace of $\mathbb{C}^M$. Hence the component of β perpendicular to FH should not hinder our approximation, and y can be approximated by the projection of $Fx + \beta$ onto $\mathrm{Ran}(F)$. By [37, Prop. 3.2.3], this approximation is given by

$$\hat{x} = \widetilde{F}^*(Fx + \beta). \tag{2.9}$$

Furthermore, because the component of β orthogonal to $\mathrm{Ran}(F)$ does not contribute, we expect $\|x-\hat{x}\| = \|\widetilde{F}^*\beta\|$ to be smaller than $\|\beta\|$. The following proposition makes this more precise.

Proposition 2.2 (Noise reduction in linear reconstruction) *Let $\{\varphi_k\}_{k=1}^M$ be a frame of unit-norm vectors with frame bounds A and B and associated frame operator F. Let $\beta = [\beta_1 \;\; \beta_2 \;\; \cdots \;\; \beta_M]^T$, where the β_i's are independent random variables with mean zero and variance σ^2. Then the MSE of the classical reconstruction (2.9) satisfies*

$$\frac{M\sigma^2}{B^2} \le MSE \le \frac{M\sigma^2}{A^2}.$$

Proof: See Appendix 2.A.2. □

Corollary 2.3 *If the frame in Proposition 2.2 is tight,*

$$MSE \;=\; \frac{N^2\sigma^2}{M} \;=\; \frac{N\sigma^2}{r}.$$

Now consider the case where the degradation is due to quantization. Let $x \in \mathbb{R}^N$ and $y = Fx$, where $F \in \mathbb{R}^{M\times N}$ is a frame operator. Suppose $\hat{y} = Q(y)$, where $Q : \mathbb{R}^M \to \mathbb{R}^M$ is a *scalar* quantization function; *i.e.*, $Q(y) = [q_1(y_1) \;\; q_2(y_2) \;\; \ldots \;\; q_M(y_M)]^T$, where $q_i : \mathbb{R} \to \mathbb{R}$, $1 \le i \le M$, is a quantizer.

Given $\hat{y}$, one approach to approximating x is to treat the quantization noise $\hat{y} - y$ as an arbitrary random noise, independent in each dimension, and uncorrelated with y. The problem can be treated identically to the previous problem, and $\hat{x} = \widetilde{F}^*\hat{y}$ can be used. However, in actuality, the quantization noise $\hat{y} - y$ is a deterministic quantity and is constrained by the subspace to which y must belong.

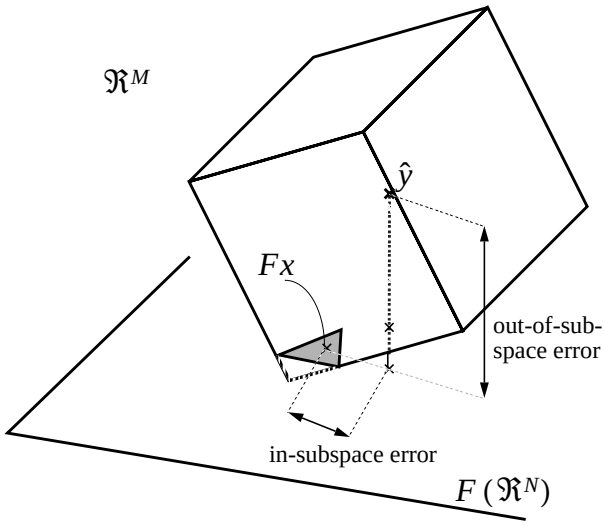


Figure 2.3: Illustration of consistent reconstruction. The quantization noise $\hat{y} - Fx$ has a component in $F(\mathbb{R}^N)$ ("in-subspace error") and an orthogonal component ("out-of-subspace error"). Linear reconstruction $\hat{x} = (F^T F)^{-1} F^T \hat{y}$ removes the out-of-subspace component but may give an estimate that is not in the set $(Q \circ F)^{-1}(\hat{y})$, which is shaded.

2.2.2.3 Consistent reconstruction

Definition 2.1 (Consistency [184]) *Let $f : X \to Y$. Let $x \in X$ and $y = f(x)$. If $f(\hat{x}) = y$ then $\hat{x} \in X$ is called a* consistent estimate of x from y. *An algorithm that produces consistent estimates is called a* consistent reconstruction *algorithm. An estimate that is not consistent is said to be* inconsistent.

The essence of consistency is that $\hat{x}$ is a consistent estimate if it is compatible with the observed value of y, *i.e.*, it is possible that $\hat{x}$ is exactly equal to x. In other words, the set of consistent estimates for x is the inverse image of $f(x)$ under f. It is intuitive to want a reconstruction algorithm to be consistent; the need for the term is precisely because many reconstruction algorithms are not.

In the case of quantized frame expansions, $f = Q \circ F$ and one can give a geometric interpretation. Q induces a partitioning of $\mathbb{R}^M$, which in turn induces a partitioning of $\mathbb{R}^N$ through the inverse map of $Q \circ F$. A consistent estimate is simply one that falls in the same partition region as the original signal vector. These concepts are illustrated for $N = 2$ and $M = 3$ in Figure 2.3. The ambient space is $\mathbb{R}^M$. The cube represents the partition region in $\mathbb{R}^M$ containing $y = Fx$ and has codebook value $\hat{y}$. The plane is $F(\mathbb{R}^N)$ and hence is the subspace within which any unquantized value must lie. The intersection of the plane with the cube gives the shaded triangle within which a consistent estimate must lie. Projecting to $F(\mathbb{R}^N)$, as in the classical reconstruction method, removes the out-of-subspace component of $y - \hat{y}$. As illustrated, this type of reconstruction is not necessarily consistent. Further geometric interpretations of quantized frame expansions are given in Appendix 2.B.

Assuming only that Q has convex partition regions, a consistent estimate can be determined using the projection onto convex sets (POCS) algorithm [215]. In this case, POCS generates a sequence of estimates by alternately projecting on $F(\mathbb{R}^N)$ and $Q^{-1}(\hat{y})$.

When Q is a scalar quantizer and each component quantizer is uniform, a linear program can be used

Table 2.1: Algorithm for consistent reconstruction from a quantized frame expansion.

1. Form $\bar{F} = \begin{bmatrix} F \\ -F \end{bmatrix}$ and $\bar{y} = \begin{bmatrix} \frac{1}{2}\Delta + \hat{y} \\ \frac{1}{2}\Delta - \hat{y} \end{bmatrix}$.
2. Pick an arbitrary cost function $c \in \mathbb{R}^N$.
3. Use a linear programming method to find $\hat{x}$ to minimize $c^T\hat{x}$ subject to $\bar{F}\hat{x} \leq \bar{y}$.

to find consistent estimates. For $i = 1, 2, \ldots, M$, denote the quantization step size in the ith component by Δ_i. For notational convenience, assume that the reproduction values lie halfway between decision levels. Then for each i, $|\hat{y}_i - y_i| \leq \Delta_i/2$. To obtain a consistent estimate, for each i we must have $|(F\hat{x})_i - \hat{y}_i| \leq \Delta_i/2$.. Expanding the absolute value, we find the constraints

$$F\hat{x} \leq \frac{1}{2}\Delta + \hat{y} \quad \text{and} \quad F\hat{x} \geq -\frac{1}{2}\Delta + \hat{y},$$

where $\Delta = [\Delta_1 \;\; \Delta_2 \;\; \ldots \;\; \Delta_M]^T$, and the inequalities are elementwise. These inequalities can be combined into

$$\begin{bmatrix} F \\ -F \end{bmatrix} \hat{x} \leq \begin{bmatrix} \frac{1}{2}\Delta + \hat{y} \\ \frac{1}{2}\Delta - \hat{y} \end{bmatrix}. \tag{2.10}$$

The formulation (2.10) shows that $\hat{x}$ can be determined through linear programming [179]. The feasible set of the linear program is exactly the set of consistent estimates, so an arbitrary cost function can be used. This is summarized in Table 2.1.

A linear program always returns a corner of the feasible set [179, §8.1], so this type of reconstruction will not be close to the centroid of the partition cell. Since the cells are convex, one could use several cost functions to get different corners of the feasible set and average the results. Another approach is to use a quadratic cost equal to the distance from the projection estimate given by (2.9). Both of these variants on the basic algorithm will reduce the MSE by a constant factor. They do not change the asymptotic behavior of the MSE as the redundancy r is increased.

2.2.2.4 Error bounds for consistent reconstruction

In orthogonal representations, under very general conditions, the MSE depends on the quantization step size as $O(\Delta^2)$ for small Δ. For frame expansions, how does the MSE depend on r, for large r, and how does it depend on the reconstruction method? The MSE obtained with any reconstruction method depends in general on the distribution of the source. The evidence suggests that any consistent reconstruction algorithm is essentially optimal—in a sense made clear by the following propositions—and gives $O(1/r^2)$ MSE.

Proposition 2.4 (MSE lower bound) *Let x be a random variable with probability density function $\mathbf{p}$ supported on a bounded subset $\mathcal{B}$ of $\mathbb{R}^N$. Consider any set of quantized frame expansions of x for which*

$$\sup_M \max_{1 \leq i \leq M} \frac{\|\varphi_i\|}{\Delta_i} = d_0 < \infty.$$

Unless $\mathbf{p}$ is degenerate in a way which allows for exact reconstruction, any reconstruction algorithm will yield an MSE that can be lower bounded by b/r^2, where b is a coefficient independent of r and a function of N, $\mathbf{p}$, the diameter D of $\mathcal{B}$, and the maximum density value d_0.

Proof: A proof due to N. T. Thao appears in [79]. □

Proposition 2.5 (Squared error upper bound (DFT case)) *Fix a source vector $x \in \mathbb{R}^N$ and a quantization step size $\Delta \in \mathbb{R}^+$. For a sequence of consistent reconstructions from quantized frame expansions of x, the squared error can be upper bounded by an $O(1/r^2)$ expression under the following conditions:*

i. N odd: The frame operators are as in (2.8) and $\left\|[x_2 \;\; x_3 \;\; \cdots \;\; x_N]^T\right\| > (N+1)\Delta/4$; or

ii. N even: The frame operators are as in (2.8) with the first column removed and $\|x\| > (N+2)\Delta/4$.

Proof: See Appendix 2.A.3. □

Conjecture 2.6 (MSE upper bound) *Under mild conditions on the sequence of frames and source, any algorithm that gives consistent estimates will yield an MSE that can be upper bounded by an $O(1/r^2)$ expression.*

Conjecture 2.6 requires some sort of non-degeneracy condition because we can easily construct a sequence of frames for which the frame coefficients give no additional information as r is increased. For example, we can start with an orthonormal basis and increase r by adding copies of vectors already in the frame. Putting aside such pathological cases, simulations for quantization of a source uniformly distributed on $[-1, 1]^N$ support this conjecture. Simulations were performed with three types of frame sequences:

I. A sequence of frames corresponding to oversampled A/D conversion, as given by (2.8). This is the case in which we have proven an $O(1/r^2)$ SE upper bound.

II. For $N = 3$, 4, and 5, Hardin, Sloane and Smith have numerically found arrangements of up to 130 points on N-dimensional unit spheres that maximize the minimum Euclidean norm separation [90].

III. Frames generated by randomly choosing points on the unit sphere according to a uniform distribution.

Simulation results are given in Figure 2.4. The dashed, dotted, and solid curves correspond to frame types I, II, and III, respectively. The data points marked with +'s correspond to using a linear program based on (2.10) to find consistent estimates. The data points marked with ∘'s correspond to classical reconstruction. The important characteristics of the graph are the slopes of the curves. Note that $O(1/r)$ MSE corresponds to a slope of −3.01 dB/octave, and $O(1/r^2)$ MSE corresponds to a slope of −6.02 dB/octave. The consistent reconstruction algorithm exhibits $O(1/r^2)$ MSE for each of the types of frames. The classical method exhibits $O(1/r)$ MSE behavior, as expected. It is particularly interesting to note that the performance with random frames is as good as with either of the other two types.

The situation we have considered has a random vector source (or a fixed vector x outside of a ball centered at the origin) and deterministic quantization. A dual situation arises with an arbitrary nonzero vector x and randomized quantization. Results along these lines obtained with S. Rangan [158] are summarized in Appendix 2.C; they strongly support Conjecture 2.6. Taking the problem and "dual" together emphasizes that hard bounds on noise greatly affect the quality of reconstruction that can be obtained from an overcomplete expansion.

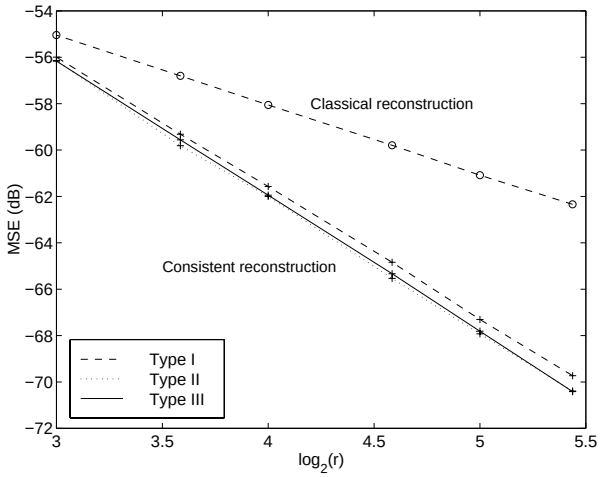


Figure 2.4: Experimental results for reconstruction from quantized frame expansions. Shows $O(1/r^2)$ MSE for consistent reconstruction and $O(1/r)$ MSE for classical reconstruction.

2.2.2.5 Rate-distortion trade-offs

Accept for the moment Conjecture 2.6, that optimal reconstruction techniques give an MSE proportional to $1/r^2$. Since the $O(\Delta^2)$ MSE for orthogonal representations extends to the frame case as well, there are two ways to reduce the MSE by approximately a factor of four: double r or halve Δ. Our discussion has focused on expected distortion without concern for rate, and there is no reason to think that these options each have the same effect on the rate.

As the simplest possible case, suppose a frame expansion is stored (or transmitted) as M b-bit numbers, for a total rate of Mb/N bits per sample. Doubling r gives $2M$ b-bit numbers, for a total rate of $2Mb/N$ bits per sample. On the other hand, halving Δ results in M $(b+1)$-bit numbers for a rate of only $M(b+1)/N$ bits per sample. This example suggests that halving Δ is always the better option, but a few comments are in order. One caveat is that in some situations, doubling r and halving Δ may have very different costs. For example, the higher cost of halving Δ than of doubling r is a major motivating factor for oversampled A/D conversion, Also, if r is doubled, storing the result as $2M$ b-bit values is far from the best thing to do. This is because many of the M additional numbers give little or no information on x. This is discussed further in Appendix 2.B.

2.3 Adaptive Expansions

Transform coding theory, as summarized in Section 1.1.3, is predicated on fine quantization approximations and assuming that signals are Gaussian. For most practical coding applications, these assumptions do not hold, so the wisdom of maximizing coding gain—which leads to the optimality of Karhunen–Loève transforms—has been questioned. More fundamentally, we can leave the usual framework of static orthogonal transform coding and consider the application of adaptive and nonlinear transforms.

The matching pursuit algorithm [131], described in Section 2.3.1, has both adaptive and nonlinear aspects. Given a source vector x and a frame $\{\varphi_k\}_{k=1}^M$, it produces an approximate signal representation $x \approx \sum_{i=0}^{n-1} \alpha_i \varphi_{k_i}$. It is adaptive in the sense that the k_i's depend on x, yet it is time-invariant for transforming a

sequence of source vectors. On the other hand, it has a linear nature because it produces a linear representation, but it is nonlinear because it does not satisfy additivity.[2]

Matching pursuit is a greedy algorithm for choosing a subset of the frame and finding a linear combination of that subset that approximates a given signal vector. The use of a greedy algorithm is justified by the computational intractability of finding the optimal subset of the original frame [38, Ch. 2]. In our finite-dimensional setting, this is very similar to the problem of finding sparse approximate solutions to underdetermined systems of linear equations.

The sparse approximate solution problem is as follows. Denote the signal vector by $x \in \mathbb{R}^N$. Let $A \in \mathbb{R}^{N\times M}$ contain the expansion vectors. Then $\alpha \in \mathbb{R}^M$ is an efficient representation of x if $\|A\alpha - x\|_2$ is small and α has a small number of nonzero entries. For an unconstrained A, finding such an α is hard:

- Given $\epsilon \in \mathbb{R}^+$ and $L \in \mathbb{Z}^+$, determining if there exists α with not more than L nonzero entries such that $\|A\alpha - x\|_2 \le \epsilon$ is an NP-complete problem [139, 38].
- Given $L \in \mathbb{Z}^+$, finding α which minimizes $\|A\alpha - x\|_2$ among all vectors with not more than L nonzero entries is NP-hard [38].

Matching pursuit is equivalent to a well-known greedy heuristic for finding sparse approximate solutions to linear equations [62, 139], without the orthogonalization step.

Quantization of coefficients in matching pursuit leads to many interesting issues; some of these are discussed in Section 2.3.2. A source coding method for $\mathbb{R}^N$ based on matching pursuit is described in Section 2.3.3.

2.3.1 Matching Pursuit

2.3.1.1 Algorithm

Let $\mathcal{D} = \{\varphi_k\}_{k=1}^M \subset H$ be a frame such that $\|\varphi_k\| = 1$ for all k. $\mathcal{D}$ is called the *dictionary*. Matching pursuit is an algorithm to represent $x \in H$ by a linear combination of elements of $\mathcal{D}$. In the first step of the algorithm, k_0 is selected such that $|\langle\varphi_{k_0}, x\rangle|$ is maximized. Then, x can be written as its projection onto φ_{k_0} and a residue R_1x:

$$x = \langle\varphi_{k_0}, x\rangle\varphi_{k_0} + R_1x.$$

The algorithm is iterated by treating R_1x as the vector to be best approximated by a multiple of φ_{k_1}. At step $p+1$, k_p is chosen to maximize $|\langle\varphi_{k_p}, R_px\rangle|$ and

$$R_{p+1}x = R_px - \langle\varphi_{k_p}, R_px\rangle\varphi_{k_p}.$$

Identifying $R_0x = x$, one can write

$$x = \sum_{i=0}^{n-1}\langle\varphi_{k_i}, R_ix\rangle\varphi_{k_i} + R_nx. \tag{2.11}$$

Hereafter, $\langle\varphi_{k_i}, R_ix\rangle$ will be denoted by α_i. Note that the output of a matching pursuit expansion is not only the coefficients $(\alpha_0, \alpha_1, \dots)$, but also the indices $(k_0, k_1, \dots)$. For storage and transmission purposes, we will have to account for the indices.

[2] The usage of additivity is not obvious. Clearly if $x_1 \approx \sum_{i=0}^{n-1}\alpha_i\varphi_{k_i}$ and $x_2 \approx \sum_{i=0}^{n-1}\beta_i\varphi_{k_i}$, then $x_1 + x_2 \approx \sum_{i=0}^{n-1}(\alpha_i+\beta_i)\varphi_{k_i}$. But in general the expansions of x_1, x_2, and $x_1 + x_2$ would not use the same k_i's; for this reason the transform is nonlinear.

Matching pursuit was introduced to the signal processing community in the context of time-frequency analysis by Mallat and Zhang [131]. Mallat and his coworkers have uncovered many of its properties [224, 38, 39].

2.3.1.2 Discussion

Since α_i is determined by projection, $\alpha_i \varphi_{k_i} \perp R_{i+1}x$. Thus we have the "energy conservation" equation

$$\|R_i x\|^2 = \|R_{i+1}x\|^2 + \alpha_i^2. \tag{2.12}$$

This fact, the selection criterion for k_i, and the fact that $\mathcal{D}$ spans H, can be combined for a simple convergence proof that applies for finite-dimensional spaces. In particular, the energy in the residue is strictly decreasing until x is exactly represented.

Even in a finite-dimensional space, matching pursuit is not guaranteed to converge in a finite number of iterations. This is a serious drawback when exact (or very precise) signal expansions are desired, especially since an algorithm which picks dictionary elements jointly would choose a basis from the dictionary and get an exact expansion in N steps. One way to speed convergence is to use an orthogonalized version of MP which at each step modifies the dictionary and chooses a dictionary element perpendicular to all previously chosen dictionary elements. Since orthogonalized matching pursuit does not converge significantly faster than the non-orthogonalized version for a small number of iterations [38, 107, 194], non-orthogonalized matching pursuit is not considered hereafter.

Matching pursuit has been found to be useful in source coding for two related reasons: The first reason—which was emphasized in the original Mallat and Zhang paper [131]—has been termed *flexibility*; the second is that the nonlinear approximation framework allows greater energy compaction than a linear transform.

MP is often said to have flexibility to differing signal structures. The archetypal illustration is that a Fourier basis provides a poor representation of functions well localized in time, while wavelet bases are not well suited to representing functions whose Fourier transforms have narrow, high frequency support [131]. The implication is that MP, with a dictionary including a Fourier basis and a wavelet basis, would avoid these difficulties.

Looking at the energy compaction properties of MP gives a more extensive view of the potential of MP. Energy compaction refers to the fact that after an appropriately chosen transform, most of the energy of a signal can be captured by a small number of coefficients. In orthogonal transform coding, getting high energy compaction is dependent on designing the transform based on knowledge of source statistics; for fine quantization of a stationary Gaussian source, the Karhunen–Loève transform is optimal (see Section 1.1.3.1). Although both produce an approximation of a source vector which is a linear combination of basis elements, orthogonal transform coding contrasts sharply with MP in that the basis elements are chosen *a priori*, and hence at best one can make the optimum basis choice *on average*. In MP, a subset of the dictionary is chosen in a *per vector* manner, so much more energy compaction is possible.

To illustrate the energy compaction property of MP, consider the following situation. A $\mathcal{N}(0, I_N)$ source is to be transform coded. Because the components of the source are uncorrelated, no orthogonal transform will give energy compaction; so in the linear coding case, k coefficients will capture k/N of the signal energy. A k-step MP expansion will capture much more of the energy. Figure 2.5 shows the results of a simulation with $N = 8$. The plot shows the fraction of the signal energy in the residual when one- to four-term expansions are

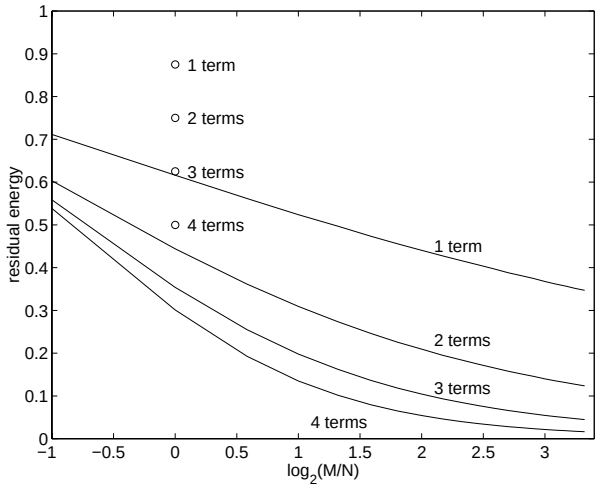


Figure 2.5: Comparison of energy compaction properties for coding of a $\mathcal{N}(0, I_8)$ source. With a k term orthogonal expansion, the residual has $(8-k)/8$ of the energy ($\circ$'s). The residual energy is much less with MP (solid curves).

used. The dictionaries are generated randomly according to a uniform distribution on the unit sphere. For an equal number of terms, the energy compaction is much better with MP than with a linear transform. Notice in particular that this is true even if the dictionary is not overcomplete, in which case MP has no more "flexibility" than an orthogonal basis representation.

2.3.2 Quantized Matching Pursuit

Define *quantized matching pursuit* (QMP) to be a modified version of matching pursuit which incorporates coefficient quantization. In particular, the inner product $\alpha_i = \langle \varphi_{k_i}, R_i x \rangle$ is quantized to $\widehat{\alpha}_i = q(\alpha_i)$ prior to the computation of the residual $R_{i+1}x$. The quantized value is used in the residual calculation: $R_{i+1}x = R_i x - \widehat{\alpha}_i \varphi_{k_i}$. The use of the quantized value in the residual calculation reduces the propagation of the quantization error to subsequent iterations.

Although quantized matching pursuit has been applied to low bit rate compression problems [141, 194, 107, 140], which inherently require coarse coefficient quantization, little work has been done to understand the qualitative effects of coefficient quantization in matching pursuit. Some of these effects are explored in this section. The relationship between quantized matching pursuit and other vector quantization (VQ) methods is discussed in Section 2.3.2.1. The issue of consistency in these expansions is explored in Section 2.3.2.2. The potential lack of consistency shows that even though matching pursuit is designed to produce a linear combination to estimate a given source vector, optimal reconstruction in the presence of coefficient quantization requires a nonlinear algorithm. Such an algorithm is provided. In Section 2.3.2.3, a detailed example of the application of QMP to quantization of an $\mathbb{R}^2$-valued source is presented. This serves to illustrate the concepts from Section 2.3.2.2 and demonstrate the potential for improved reconstruction using consistency.

2.3.2.1 Relationship to other vector quantization methods

A single iteration of matching pursuit is very similar to shape-gain VQ, which was introduced in [23]. In shape-gain VQ, a vector $x \in \mathbb{R}^N$ is separated into a *gain*, $g = \|x\|$ and a *shape*, $s = x/g$. A shape $\hat{s}$ is chosen from a shape codebook $\mathcal{C}_s$ to maximize $\langle x, \hat{s}\rangle$. Then a gain $\hat{g}$ is chosen from a gain codebook $\mathcal{C}_g$ to minimize $(\hat{g} - \langle x, \hat{s}\rangle)^2$. The similarity is clear with $\mathcal{C}_s$ corresponding to $\mathcal{D}$ and $\mathcal{C}_g$ corresponding to the quantizer for α_0, the only differences being that in MP one maximizes the *absolute value* of the correlation and thus the gain factor can be negative. Obtaining a good approximation in shape-gain VQ requires that $\mathcal{C}_s$ forms a dense subset of the unit sphere in $\mathbb{R}^N$. The area of the unit sphere increases exponentially with N, making it difficult to use shape-gain VQ in high dimensional spaces. A multiple-iteration application of matching pursuit can be seen as a cascade form of shape-gain VQ.

2.3.2.2 Consistency

We have thus far discussed only signal analysis (or encoding) using QMP and not synthesis (reconstruction) from a QMP representation. To the author's knowledge, all previous work with QMP has used

$$\hat{x} = \sum_{i=0}^{p-1} \widehat{\alpha}_i \varphi_{k_i}, \tag{2.13}$$

which results from simply using quantized coefficients in (2.11) and setting the final residual to zero. Computing this reconstruction has very low complexity, but its shortcoming is that it disregards the effects of quantization; hence, it can produce inconsistent estimates.

Suppose p iterations of QMP are performed with the dictionary $\mathcal{D}$ and denote the output by

$$QMP(x) = (k_0, \widehat{\alpha}_0, k_1, \widehat{\alpha}_1, \ldots, k_{p-1}, \widehat{\alpha}_{p-1}). \tag{2.14}$$

Denote the output of QMP (with the same dictionary and quantizers) applied to $\hat{x}$ by

$$QMP(\hat{x}) = (k_0', \widehat{\alpha}_0', k_1', \widehat{\alpha}_1', \ldots, k_{p-1}', \widehat{\alpha}_{p-1}').$$

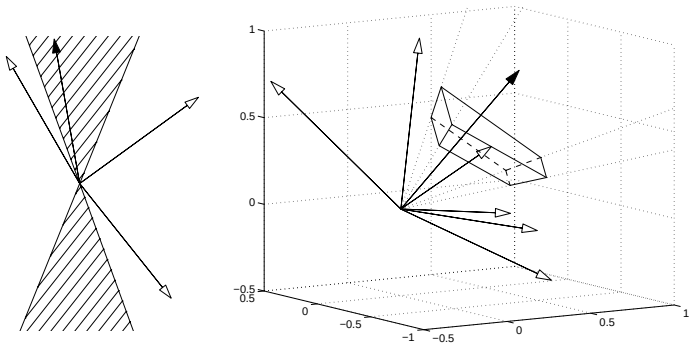
By the definition of consistency (Section 2.2.2.3), $\hat{x}$ is a consistent estimate of x if and only if $k_i = k_i'$ and $\widehat{\alpha}_i = \widehat{\alpha}_i'$ for $i = 0, 1, \ldots, p-1$.

We now develop a description of the set of consistent estimates of x through simultaneous linear inequalities. For notational convenience, we assume uniform scalar quantization of the coefficients with step size Δ and midpoint reconstruction.[3] The selection of k_0 implies

$$|\langle \varphi_{k_0}, x\rangle| \geq |\langle \varphi, x\rangle| \qquad \text{for all} \quad \varphi \in \mathcal{D}. \tag{2.15}$$

For each element of $\mathcal{D} \setminus \{\varphi_{k_0}\}$, (2.15) specifies a pair of half-space constraints with boundary planes passing through the origin. An example of such a constraint in $\mathbb{R}^2$ is shown in Figure 2.6(a). If φ_{k_0} is the vector with the solid arrowhead, chosen from all of the marked vectors, the source vector must like in the hatched area. For

[3] Ambiguities on partition cell boundaries due to arbitrary tie-breaking—in both dictionary element selection and nearest-neighbor scalar quantization—are ignored.



(a) (b)

Figure 2.6: (a) Illustration of consistency constraint (2.15) in $\mathbb{R}^2$; (b) Illustration of consistency constraints (2.15) & (2.16) in $\mathbb{R}^3$.

$N > 2$, the intersection of these constraints is two infinite convex polyhedral cones situated symmetrically with their apexes at the origin. The value of $\widehat{\alpha}_0$ gives the constraint

$$\langle \varphi_{k_0},\, x\rangle \in \left[\widehat{\alpha}_0 - \frac{\Delta}{2}, \widehat{\alpha}_0 + \frac{\Delta}{2}\right]. \tag{2.16}$$

This specifies a pair of planes, perpendicular to φ_{k_0}, between which x must lie. Constraints (2.15) and (2.16) are illustrated in Figure 2.6(b) for $\mathbb{R}^3$. The vector with the solid arrowhead was chosen among all the marked dictionary vectors as φ_{k_0}. Then the quantization of α_0 implies that the source vector lies in the volume shown.

At the $(i-1)$st step, the selection of k_i gives the constraints

$$\left|\left\langle \varphi_{k_i},\, x - \sum_{\ell=0}^{i-1} \widehat{\alpha}_\ell \varphi_{k_\ell} \right\rangle\right| \geq \left|\left\langle \varphi,\, x - \sum_{\ell=0}^{i-1} \widehat{\alpha}_\ell \varphi_{k_\ell} \right\rangle\right| \qquad \text{for all} \quad \varphi \in \mathcal{D}. \tag{2.17}$$

This defines $M-1$ pairs of linear half-space constraints with boundaries passing through $\sum_{\ell=0}^{i-1} \widehat{\alpha}_\ell \varphi_{k_\ell}$. As before, these define two infinite pyramids situated symmetrically with their apexes at $\sum_{\ell=0}^{i-1} \widehat{\alpha}_\ell \varphi_{k_\ell}$. Then $\widehat{\alpha}_i$ gives

$$\left\langle \varphi_{k_i},\, x - \sum_{\ell=0}^{i-1} \widehat{\alpha}_\ell \varphi_{k_\ell} \right\rangle \in \left[\widehat{\alpha}_i - \frac{\Delta}{2}, \widehat{\alpha}_i + \frac{\Delta}{2}\right]. \tag{2.18}$$

This again specifies a pair of planes, now perpendicular to φ_{k_i}, between which x must lie.

By being explicit about the constraints as above, we see that, except in the case that $0 \in \left[\widehat{\alpha}_i - \frac{\Delta}{2}, \widehat{\alpha}_i + \frac{\Delta}{2}\right]$ for some i, the partition cell defined by (2.14) is convex.[4] Thus by using an appropriate projection operator, one can find a consistent estimate from any initial estimate. The partition cells are intersections of cells of the form shown in Figure 2.6(b).

Notice now that contrary to what would be surmised from (2.13), k_i gives some information on the signal even if $\widehat{\alpha}_i = 0$. The experiments in Section 2.3.3.3 show that when $\widehat{\alpha}_i = 0$, it tends to be inefficient in a rate–distortion sense to store or transmit k_i. If we know $\widehat{\alpha}_i = 0$, but do not know the value of k_i, then

[4]The "hourglass" cell that results from $0 \in \left[\widehat{\alpha}_i - \frac{\Delta}{2}, \widehat{\alpha}_i + \frac{\Delta}{2}\right]$ does not make consistent reconstruction more difficult, but is intuitively undesirable in a rate–distortion sense.

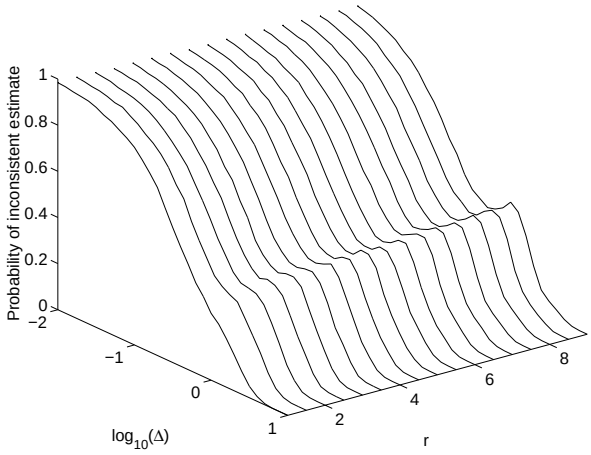


Figure 2.7: Probability that (2.13) gives an inconsistent reconstruction for two iteration expansions of an $\mathbb{R}^4$-valued source.

(2.17)–(2.18) reduce to

$$\left|\left\langle \varphi,\, x - \sum_{\ell=0}^{i-1} \widehat{\alpha}_\ell \varphi_{k_\ell} \right\rangle\right| \le \frac{\Delta}{2} \qquad \text{for all} \quad \varphi \in \mathcal{D}. \tag{2.19}$$

Experiments were performed to demonstrate that (2.13) often gives inconsistent estimates and to assess how the probability of an inconsistent estimate depends on the dictionary size and the quantization. Presented here are results for an $\mathbb{R}^4$-valued source with the $\mathcal{N}(0, I)$ distribution. The consistency of reconstruction was checked for two iteration expansions with dictionaries generated randomly according to a uniform distribution on the unit sphere. Dictionary sizes of $M = 4,\ 8,\ \ldots,\ 20$ were used. The quantization was uniform with reconstruction points $\{m\Delta\}_{m\in\mathbb{Z}}$. The results are shown in Figure 2.7. The probability of inconsistency goes to zero for very coarse quantization and goes to one for fine quantization. The dependence on dictionary size and lack of monotonicity indicate complicated geometric factors. Similar experiments with different sources and dictionaries were reported in [66].

As noted earlier, the cells of the partition generated by QMP are convex or the union of two convex cells that share one point. This fact allows the computation of consistent estimates through the method of alternating projections [215]. One would normally start with an initial estimate given by (2.13). Given an estimate $\hat{x}$, the algorithm given in Table 2.2 performs the one "most needed" projection; namely, the first projection needed in enforcing (2.15)–(2.18). Among the possible projections in enforcing (2.17), the one corresponding to the largest deviation from consistency is performed. For notational convenience and concreteness, we again assume uniform quantization with $q_i(\alpha_i) = m\Delta \iff \alpha_i \in \big[(m - \frac{1}{2})\Delta,\ (m + \frac{1}{2})\Delta\big)$; steps 5 and 6 could easily be adjusted for a general quantizer.

In a broadly applicable special case, the inequalities (2.15)–(2.18) can be manipulated into a set of elementwise inequalities $Ax \le b$ suitable for reconstruction using linear or quadratic programming, where A and b are $2Mp \times N$ and $2Mp \times 1$, respectively, and A and b depend only on the QMP output. This formulation is possible when each QMP iteration either a) uses a quantizer with zero as a decision point; or b) uses a quantizer which maps a symmetric interval to zero, and the value of k_i is discarded when $\widehat{\alpha}_i = 0$.

Consider first the case where q_i has zero as a decision point. For notational convenience, we will assume

Table 2.2: Projection algorithm for consistent reconstruction from a QMP representation.

1. Set $c = 0$. This is a counter of the number of steps of QMP that $\hat{x}$ is consistent with.
2. Let $\bar{x} = \hat{x} - \sum_{i=0}^{c-1} \widehat{\alpha}_i \mathcal{D}_{k_i}$, where it is understood that the summation is empty for $c = 0$.
3. Find $\varphi \in \mathcal{D}$ that maximizes $|\langle \varphi, \bar{x} \rangle|$. If $\varphi = \varphi_{k_c}$, go to step 5; else go to step 4.
4. ($\hat{x}$ is not consistent with k_c.) Let $\widetilde{\varphi}_{k_c} = \operatorname{sgn}(\langle \varphi_{k_c}, \bar{x} \rangle)\varphi_{k_c}$ and $\widetilde{\varphi} = \operatorname{sgn}(\langle \varphi, \bar{x} \rangle)\varphi$.
   Let $\hat{x} = \hat{x} - \langle \widetilde{\varphi}_{k_c} - \widetilde{\varphi}, \bar{x} \rangle (\widetilde{\varphi}_{k_c} - \widetilde{\varphi})$, the orthogonal projection of $\hat{x}$ onto the set described by (2.17). Terminate.
5. ($\hat{x}$ is consistent with k_c.) If $\langle \varphi_{k_c}, \bar{x} \rangle \in \left[\widehat{\alpha}_c - \frac{1}{2}\Delta, \widehat{\alpha}_c + \frac{1}{2}\Delta\right)$, go to step 7; else go to step 6.
6. ($\hat{x}$ is not consistent with $\widehat{\alpha}_c$.) Let
$$\beta = \operatorname{sgn}(\langle \varphi_{k_c}, \bar{x} \rangle - \widehat{\alpha}_c) \cdot \min\left\{ \left| \langle \varphi_{k_c}, \bar{x} \rangle - (\widehat{\alpha}_c + \tfrac{\Delta}{2}) \right|, \left| \langle \varphi_{k_c}, \bar{x} \rangle - (\widehat{\alpha}_c - \tfrac{\Delta}{2}) \right| \right\}.$$
   Let $\hat{x} = \hat{x} - \beta \varphi_{k_c}$, the orthogonal projection of $\hat{x}$ onto the set described by (2.18). Terminate.
7. ($\hat{x}$ is consistent with $\widehat{\alpha}_c$.) Increment c. If $c = p$, terminate ($\hat{x}$ is consistent); else go to step 2.

the decision points and reconstruction values are given by $\{m\Delta_i\}_{m\in\mathbb{Z}}$ and $\{(m + \frac{1}{2})\Delta_i\}_{m\in\mathbb{Z}}$, respectively, but all that is actually necessary is that the quantized coefficient $\widehat{\alpha}_i$ reveals the sign of the unquantized coefficient α_i. Denote $\operatorname{sgn}(\alpha_i)$ by σ_i, and furthermore define the following $2(M-1) \times N$ matrices:

$$\begin{aligned} F_i &= \begin{bmatrix} \varphi_{k_i} & \varphi_{k_i} & \dots & \varphi_{k_i} \end{bmatrix}^T, \\ F_{\underline{i}} &= \begin{bmatrix} \varphi_{k_1} & \dots & \varphi_{k_{i-1}} & \varphi_{k_{i+1}} & \dots & \varphi_{k_M} \end{bmatrix}^T. \end{aligned}$$

First, write (2.17) as

$$\left|\varphi_{k_i}^T (x - c)\right| \geq \left|\varphi^T (x - c)\right| \qquad \text{for all} \quad \varphi \in \mathcal{D},$$

where c is shorthand for $\sum_{\ell=0}^{i-1} \widehat{\alpha}_\ell \varphi_{k_\ell}$. Combining the $M - 1$ nontrivial inequalities gives

$$\sigma_i F_i (x - c) \geq \left| F_{\underline{i}} (x - c) \right|.$$

Expanding the absolute value one can obtain

$$\begin{bmatrix} F_{\underline{i}} - \sigma_i F_i \\ -F_{\underline{i}} - \sigma_i F_i \end{bmatrix} x \leq \begin{bmatrix} (F_{\underline{i}} - \sigma_i F_i)c \\ (-F_{\underline{i}} - \sigma_i F_i)c \end{bmatrix}. \tag{2.20}$$

Writing (2.18) first as

$$\left|\varphi_{k_i}^T (x - c) - \widehat{\alpha}_i\right| \leq \frac{\Delta_i}{2},$$

one easily obtains

$$\begin{bmatrix} \varphi_{k_i}^T \\ -\varphi_{k_i}^T \end{bmatrix} x \leq \begin{bmatrix} \frac{\Delta_i}{2} + \widehat{\alpha}_i + \varphi_{k_i}^T c \\ \frac{\Delta_i}{2} - \widehat{\alpha}_i - \varphi_{k_i}^T c \end{bmatrix}. \tag{2.21}$$

On the other hand, if q_i maps an interval $[-\frac{\Delta_i}{2}, \frac{\Delta_i}{2}]$ to zero and k_i is not coded, then (2.19) leads similarly to the $2M$ inequalities

$$\begin{bmatrix} F \\ -F \end{bmatrix} x \leq \begin{bmatrix} Fc + \frac{\Delta_i}{2} \\ -Fc + \frac{\Delta_i}{2} \end{bmatrix}. \tag{2.22}$$

A formulation of the form $Ax \leq b$ is obtained by stacking inequalities (2.20)–(2.22) appropriately.

2.3.2.3 An example in $\mathbb{R}^2$

Consider quantization of an $\mathbb{R}^2$-valued source. Assume that two iterations will be performed with the four element dictionary

$$\mathcal{D} = \left\{ \left[\cos\frac{(2k-1)\pi}{8} \quad \sin\frac{(2k-1)\pi}{8} \right]^T \right\}_{k=1}^{4}.$$

Even if the distribution of the source is known, it is difficult to find analytical expressions for optimal quantizers. (Optimal quantizer design is considered for a source with a uniform distribution on $[-1, 1]^2$ in [66, §3.3.2].) Since we wish to use fixed, untrained quantizers, we will use uniform quantizers for α_0 and α_1. It will generally be true that $\varphi_{k_0} \perp \varphi_{k_1}$, so it makes sense for the quantization step sizes for α_0 and α_1 to be equal.

The partitions generated by matching pursuit are very intricate. Suppose the quantizer has reconstruction values $\{m\Delta\}_{m\in\mathbb{Z}}$ and decision points $\{(m+\frac{1}{2})\Delta\}_{m\in\mathbb{Z}}$ for some quantization step size Δ.[5] The first quadrant of the resulting partition is shown in Figure 2.8. Heavy lines indicate partition boundaries, except that dotted lines are used for boundaries that are created by choice of k_0 (k_1) but, depending on the reconstruction method, might not be important because $\widehat{\alpha}_0 = 0$ ($\widehat{\alpha}_1 = 0$). In this partition, most of the cells are squares, but there are also some smaller cells. The fraction of cells that are not square goes to zero as $\Delta \to 0$.

This quantization of $\mathbb{R}^2$ gives concrete examples of the inconsistency resulting from using (2.13). The linear reconstruction points are indicated in Figure 2.8 by $\circ$'s. The light line segments connect these to the corresponding optimal[6] reconstruction points. Such a line segment crossing a cell boundary indicates a case of (2.13) giving an inconsistent estimate.

2.3.3 Lossy Vector Coding with Quantized Matching Pursuit

This section explores the efficacy of using QMP as an algorithm for lossy compression. In order to reveal qualitative properties most clearly, simple dictionaries and sources are used in the initial experiments. (Experiments with other dictionaries and sources appear in [66].) Then a few experiments with still images are presented. We do not explore the design of a dictionary or scalar quantizers for a particular application. Dictionary structure has a great impact on the computational complexity of QMP as demonstrated, for example, in [63, 64].

For simplicity, rate and distortion are measured by sample entropy and MSE per component, respectively. The sources used in the initial experiments are multidimensional Gaussian with zero mean and independent components. The inner product quantization is uniform with midpoint reconstruction values at $\{m\Delta\}_{m\in\mathbb{Z}}$. Furthermore, the quantization step size Δ is constant across iterations. This is consistent with equal weighting of error in each direction.

2.3.3.1 Basic experimental results

In the first experiment, $N = 4$ and the dictionary was composed of $M = 11$ maximally spaced points on the unit sphere [90]. Rate was measured by summing the (scalar) sample entropies of k_0, k_1, …, k_{p-1} and $\widehat{\alpha}_0$, $\widehat{\alpha}_1$, …, $\widehat{\alpha}_{p-1}$, where p is the number of iterations. The results are shown in Figure 2.9. The three dotted

[5] The partition is somewhat different when the quantizer has different decision points, *e.g.*, $\{(m+\frac{1}{2})\Delta\}_{m\in\mathbb{Z}}$ [66, §3.3.2]. The ensuing conclusions are qualitatively unchanged.

[6] Optimality is with respect to a uniform source distribution.

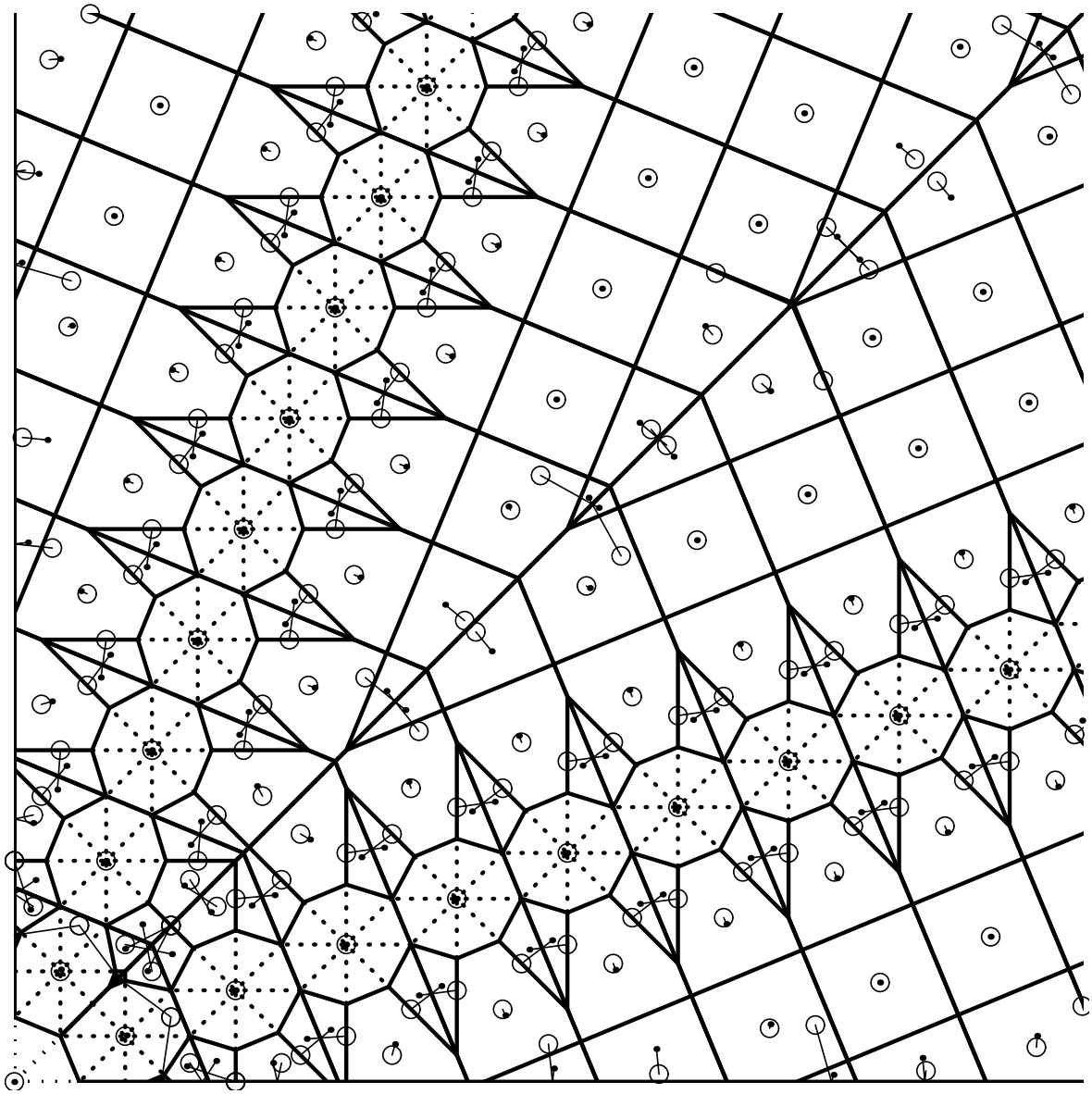


Figure 2.8: Partitioning of first quadrant of $\mathbb{R}^2$ by quantized matching pursuit with four element dictionary (heavy lines). Linear reconstruction points ($\circ$'s) are connected to optimal reconstruction points ($\times$'s) by light line segments.

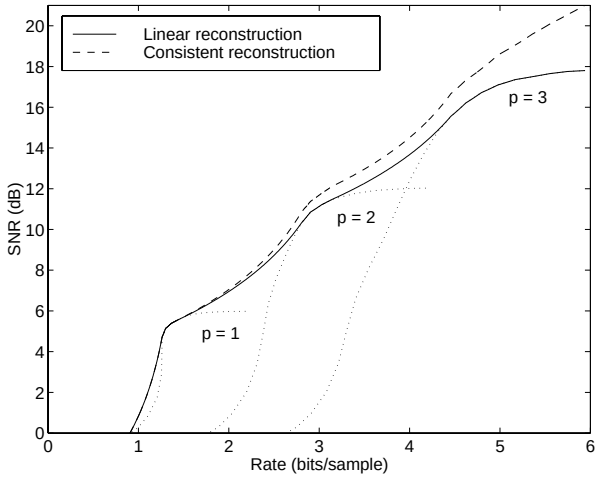


Figure 2.9: Performance comparison between reconstruction based on (2.13) and consistent reconstruction. $N = 4$ and the dictionary is composed of $M = 11$ maximally spaced points on the unit sphere [90].

curves correspond to varying p from 1 to 3 while reconstructing according to (2.13). The points along each dotted curve are obtained by varying Δ. Notice that the number of iterations that minimizes the distortion depends on the available rate. The solid curve is the convex hull of these R-D operating points (converted to a dB scale). In subsequent graphs, only this convex hull performance is shown.

2.3.3.2 Improved reconstruction using consistency

Continuing the experiment above, the degree of improvement obtained by using a consistent reconstruction algorithm was ascertained. Using consistent reconstruction gives the performance shown by the dashed curve in Figure 2.9. Notice that there is no improvement at low bit rates because consistency is not an issue for a single-iteration expansion. The improvement increases monotonically with the bit rate.

2.3.3.3 An effective stopping criterion

Regardless of the reconstruction method, the coding results shown in Figure 2.9 are far from satisfactory, especially at low bit rates. For a p-step expansion, the "baseline" coding method is to apply entropy codes (separately) to k_0, $\widehat{\alpha}_0$, k_1, $\widehat{\alpha}_1$, ..., k_{p-1}, $\widehat{\alpha}_{p-1}$. This coding places a rather large penalty of roughly $\log_2 M$ bits on each iteration, *i.e.*, this many bits must be spent in addition to the coding of the coefficient. In particular, the minimum achievable bit rate is about $(\log_2 M)/N$.

Assume that the same scalar quantization function is used at each iteration and that the quantizer maps a symmetric interval to zero. Based on a few simple observations, we can devise a simple alternative coding method which greatly reduces the rate. The first observation is that if $\widehat{\alpha}_i = 0$, then $\widehat{\alpha}_j = 0$ for all $j > i$ because the residual remains unchanged. Secondly, if $\widehat{\alpha}_i = 0$, then k_i carries relatively little information. Thus we propose that a) $\widehat{\alpha}_i = 0$ be used a stopping criterion which causes a block to be terminated even if the maximum number of iterations has not been reached; and b) k_i be considered conceptually to come after $\widehat{\alpha}_i = 0$, so k_i is not coded if $\widehat{\alpha}_i = 0$.

Simulations were performed with the same source, dictionary, and quantizers as before to demonstrate

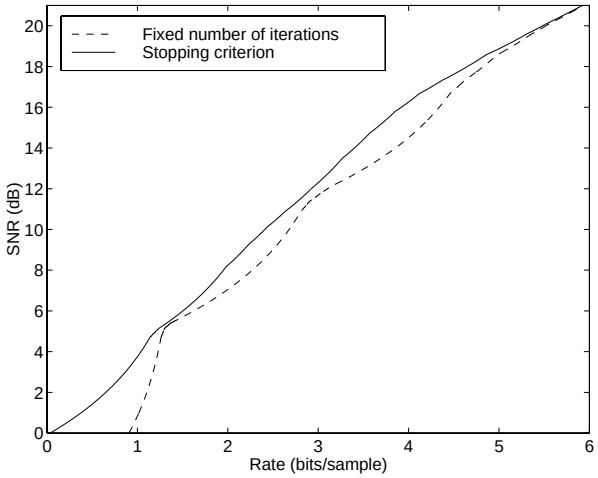


Figure 2.10: Performance comparison between a fixed number of iterations and a simple stopping criterion. $N = 4$ and the dictionary is composed of $M = 11$ maximally spaced points on the unit sphere [90].

the improvement due to the use of a stopping criterion. The results, shown in Figure 2.10, indicate a sizable improvement at low bit rates.

2.3.3.4 Further explorations

Having established the merits of consistent reconstruction and the stopping criterion of Section 2.3.3.3, we now explore the effects of varying the size of the dictionary. Again the source is i.i.d. Gaussian in blocks of $N = 4$ samples, and dictionaries generated randomly according to a uniform distribution on the unit sphere were used. Figure 2.11 shows the performance of QMP with $M = 4, 8, \ldots, 20$ (solid curves); and of independent uniform scalar quantization followed by entropy coding (dotted curve). The performance of QMP improves as M is increased and exceeds that of independent uniform scalar quantization at low bit rates. This result highlights the advantage of a nonlinear transform, since no linear transform would give any coding gain for this source.[7]

In the final experimental investigation, we consider the lowest complexity instance of QMP. This occurs when the dictionary is an orthonormal set. In this case, QMP reduces to nothing more than a linear transform followed by sorting by absolute value and quantization. Here we code an i.i.d. Gaussian source with block sizes $N = 1, 2, \ldots, 8$.[8] The results shown in Figure 2.12 indicate that even in this computationally simple case without a redundant dictionary, QMP performs well at low bit rates. An interesting phenomenon is revealed: $N = 1$ is best at high bit rates and $N = 2$ is best at low bit rates; no larger value of N is best at any bit rate.

2.3.3.5 Image coding experiments

Inspired by the good low bit-rate performance shown in Figure 2.12, QMP was applied to still image coding with a DCT-basis dictionary. Using just a basis contrasts sharply with previous attempts to use MP for

[7] The use of random dictionaries is not advocated. Slightly better performance is expected with an appropriately chosen fixed dictionary.

[8] Of course, $N = 1$ gives independent uniform scalar quantization of each sample.

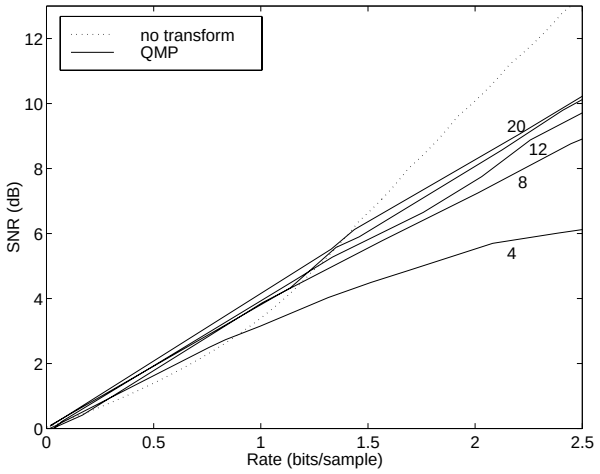


Figure 2.11: Performance of QMP as the dictionary size is varied (solid curves, labeled by M) compared to the performance of independent uniform quantization of each sample (dotted curve).

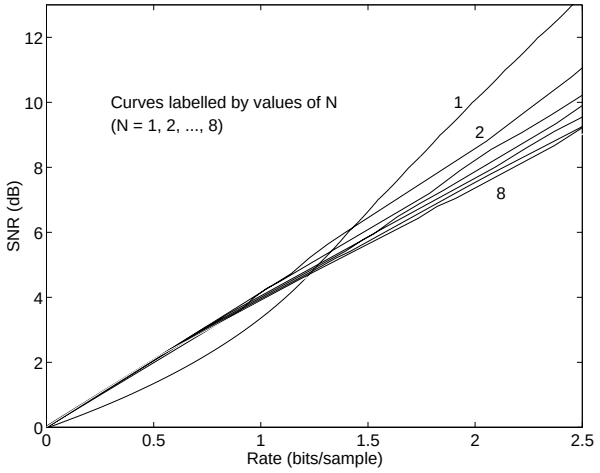
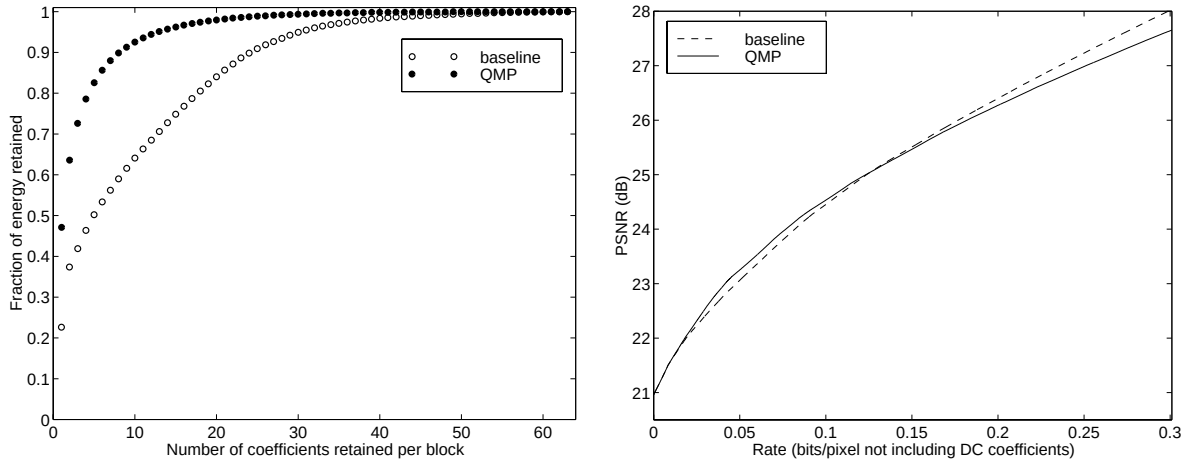


Figure 2.12: Performance of QMP with an orthogonal basis dictionary as the block size N is varied.



(a) (b)

Figure 2.13: Simulation results for coding *Barbara*: (a) Energy compaction; and (b) operational rate–distortion performance.

image coding [12]. The experiments compare a "baseline" system with a QMP-based system. Both work on 8×8 pixel image blocks which have been transformed using a two-dimensional separable DCT and both use uniform scalar quantization with equal step size for each DCT coefficient. The baseline system uses 64 scalar entropy codes, one for each horizontal/vertical frequency pair. The other system uses QMP-based coding as in Section 2.3.3.4.

Experiments were performed on several standard 512×512 pixel, 8-bit deep grayscale test images. To simplify the experiments, the coding of DC coefficients was ignored. Since the DC coefficient was always the largest in magnitude, the relative performances of the two systems was unaffected by this simplification. The peak signal to noise ratio (PSNR) figures and reconstructed images shown are based on exact knowledge of the DC coefficient.

Numerical results on compression of *Barbara* are shown in Figure 2.13. Part (a) shows the greater energy compaction of QMP over a linear transform, as in Figure 2.5, and part (b) gives operational rate–distortion results. The curves were generated by first finding the lower convex hull of rate–distortion operating points and then converting the distortion measure to PSNR. The QMP-based method gave higher PSNR for bit rates up to 0.27 bits/pixel. (Recall that this rate does not include the coding of the DC coefficient.) The peak improvement of 0.133 dB occurs at 0.050 bits/pixel.

Figure 2.14 allows subjective quality comparisons. Two coded versions of *Barbara* are shown at a rate of 0.075 bits/pixel. The difference in the two images is slight at best. Of possible interest is that the QMP coder achieves essentially identical performance while spending slightly more than half of its bits on the index information.

The same coding method was also applied to residual frames in motion-compensated video coding (see Figure 1.9). Results appear in [74].

(a) Baseline method, PSNR = 23.8 dB (b) QMP-based coder, PSNR = 24.0 dB

Figure 2.14: Results for compression of *Barbara* at 0.075 bits/pixel (with exactly known DC coefficients): (a) coded with baseline method; (b) coded with QMP.

2.3.3.6 A few possible variations

The experiments of the previous subsections are the tip of the iceberg in terms of possible design choices. To conclude the discussion of source coding, a few possible variations are presented along with plausibility arguments for their application.

An obvious area to study is the design of dictionaries. For static, untrained dictionaries, issues of interest include not only R-D performance, but also storage requirements, complexity of inner product computation, and complexity of largest inner product search.

There is no *a priori* reason to use the same dictionary at every iteration. Given a p iteration estimate, the entropy of k_p becomes a limiting factor in adding the results of an additional iteration. To reduce this entropy, it might be useful to use coarser dictionaries as the iterations proceed. Another possibility is to adapt the dictionary by augmenting it with samples from the source. (Dictionary elements might also be deleted or adjusted.) The decoder would have to be aware of changes in the dictionary, but depending on the nature of the adaptation, this may come without a rate penalty.

The experimental results that have been presented are based on entropy coding each $\widehat{\alpha}_i$ independently of the indices, which are in turn coded separately; there are other possibilities. Joint entropy coding of indices was explored in [77, 66]. Also, conditional entropy coding could exploit the likelihood of consecutively chosen dictionary vectors being orthogonal or nearly orthogonal.

Finally, for a broad class of source distributions, the distributions of the α_i's will have some common properties because they are similar to order statistics. For example, the probability density of α_0 will be small near zero. This could be exploited in quantizer design.

2.4 Conclusions

This chapter has considered the effects of coefficient quantization in overcomplete expansions. Two classes of overcomplete expansions were considered: fixed (frame) expansions and expansions that are adapted to each particular source sample, as given by matching pursuit. In each case, the possible inconsistency of linear reconstruction was exhibited, computational methods for finding consistent estimates were given, and the distortion reduction due to consistent reconstruction was experimentally assessed.

For a quantized frame expansion with redundancy r, it was proven that any reconstruction method will give MSE that can be lower bounded by an $O(1/r^2)$ expression. Backed by experimental evidence and a proof of a restricted case, it was conjectured that any reconstruction method that gives consistent estimates will have an MSE that can upper bounded by an $O(1/r^2)$ expression. Taken together, these suggest that optimal reconstruction methods will yield $O(1/r^2)$ MSE, and that consistency is sufficient to ensure this asymptotic behavior.

Experiments on the application of quantized matching pursuit as a vector compression method demonstrated good low bit rate performance when an effective stopping criterion was used. Since it is a successive approximation method, matching pursuit may be useful in a multiresolution framework, and the inherent hierarchical nature of the representation is amenable to unequal error protection methods for transmission over noisy channels. Because of the dependencies between outputs of successive iterations, MP might also work well coupled with adaptive and/or universal lossless coding.

Appendices

2.A Proofs

2.A.1 Proof of Theorem 2.1

Let $\Phi_M = \{\varphi_k\}_{k=1}^M$. The corresponding frame operator is given by

$$F = \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_M \end{bmatrix}^T.$$

Thus the $(i,\, j)$th element of $M^{-1}F^*F$ is given by

$$\left(\frac{1}{M}F^*F\right)_{ij} \;=\; \frac{1}{M}\sum_{k=1}^{M}(F^*)_{ik}F_{kj} \;=\; \frac{1}{M}\sum_{k=1}^{M}F_{ki}F_{kj} \;\; \frac{1}{M}\sum_{k=1}^{M}(\varphi_k)_i(\varphi_k)_j,$$

where $(\varphi_k)_i$ is the ith component of φ_k.

First consider the diagonal elements ($i = j$). Since for a fixed i, the random variables $(\varphi_k)_i$, $1 \le k \le M$ are independent, identically distributed, and have zero mean, we find that

$$\begin{aligned} E\left[\left(\frac{1}{M}F^*F\right)_{ii}\right] &= \mu_2 \\ \mathrm{Var}\left[\left(\frac{1}{M}F^*F\right)_{ii}\right] &= \frac{1}{M}\left(\mu_4 - \frac{M-3}{M-1}\mu_2^2\right), \end{aligned} \tag{2.23}$$

where $\mu_2 = E[(\varphi_k)_i^2]$ and $\mu_4 = E[(\varphi_k)_i^4]$ [152, §8-1]. For the off-diagonal elements ($i \neq j$),

$$E\left[\left(\frac{1}{M}F^*F\right)_{ij}\right] = 0 \tag{2.24}$$

$$\mathrm{Var}\left[\left(\frac{1}{M}F^*F\right)_{ij}\right] = \frac{1}{M}E\left[(\varphi_k)_i^2(\varphi_k)_j^2\right]. \tag{2.25}$$

Noting that μ_2 and μ_4 are independent of M, (2.23) shows that $\mathrm{Var}\left[(M^{-1}F^*F)_{ii}\right] \to 0$ as $M \to \infty$, so $(M^{-1}F^*F)_{ii} \to \mu_2$ in the mean-squared sense [152, §8-4]. Similarly, (2.24) and (2.25) show that for $i \neq j$, $(M^{-1}F^*F)_{ij} \to 0$ in the mean-squared sense. This completes the proof, provided $\mu_2 = 1/N$.

We now derive explicit formulas (depending on N) for μ_2, μ_4, and $E\left[(\varphi_k)_i^2(\varphi_k)_j^2\right]$. For notational convenience, the subscript k is omitted; instead, subscripts are used to identify the components of the vector. To compute expectations, we need an expression for the joint probability density of $(\varphi_1, \varphi_2, \ldots, \varphi_N)$. Denote the N-dimensional unit sphere (centered at the origin) by S_N. Since φ is uniformly distributed on S_N, the p.d.f. of φ is given by

$$f(\varphi) = \frac{1}{c_N} \qquad \text{for all} \quad \varphi \in S_N, \tag{2.26}$$

where c_N is the surface area of S_N. Using spherical coordinates, c_N is given by

$$c_N = \left(\int_0^{2\pi} d\theta\right)\left(\int_0^{\pi}\sin\omega_1\, d\omega_1\right)\left(\int_0^{\pi}\sin^2\omega_2\, d\omega_2\right)\cdots\left(\int_0^{\pi}\sin^{N-2}\omega_{N-2}\, d\omega_{N-2}\right). \tag{2.27}$$

Using (2.26), we can make the following calculation:

$$\mu_2 \;=\; E\left[\varphi_i^2\right] \;=\; E\left[\varphi_N^2\right]$$

$$
\begin{aligned}
&= \int_{S_N} \frac{\varphi_N^2}{c_N}\, dA \quad \text{where } dA \text{ is a differential area element} \\
&= \frac{1}{c_N}\left(\int_0^{2\pi} d\theta\right)\left(\int_0^{\pi} \sin\omega_1\, d\omega_1\right)\left(\int_0^{\pi} \sin^2\omega_2\, d\omega_2\right)\cdots\left(\int_0^{\pi} \sin^{N-3}\omega_{N-3}\, d\omega_{N-3}\right)\cdot \\
&\qquad \left(\int_0^{\pi} \cos^2\omega_{N-2}\sin^{N-2}\omega_{N-2}\, d\omega_{N-2}\right) \qquad (2.28) \\
&= \left(\int_0^{\pi} \sin^{N-2}\omega_{N-2}\, d\omega_{N-2}\right)^{-1}\left(\int_0^{\pi} \cos^2\omega_{N-2}\sin^{N-2}\omega_{N-2}\, d\omega_{N-2}\right) \qquad (2.29) \\
&= \frac{1}{N}
\end{aligned}
$$

In this calculation, (2.28) results from using spherical coordinates and (2.29) follows from substituting (2.27) and cancelling like terms. The final simplification is due to a standard integration formula [168, #323]. Similar calculations give $\mu_4 = E\left[\varphi_i^4\right] = 3[N(N+2)]^{-1}$ and, for $i \neq j$, $E\left[\varphi_i^2\varphi_j^2\right] = [N(N+2)]^{-1}$.

2.A.2 Proof of Proposition 2.2

Subtracting $\hat{x} = \sum_{k=1}^{M}\left(\langle x,\, \varphi_k\rangle + \beta_k\right)\widetilde{\varphi}_k$ from $x = \sum_{k=1}^{M}\langle x,\, \varphi_k\rangle\widetilde{\varphi}_k$ gives

$$x - \hat{x} = -\sum_{k=1}^{M}\beta_k\widetilde{\varphi}_k.$$

Then we can calculate

$$
\begin{aligned}
\text{MSE} &= E\left\|x-\hat{x}\right\|^2 = E\left\|\sum_{k=1}^{M}\beta_k\widetilde{\varphi}_k\right\|^2 \\
&= E\left[\sum_{i=1}^{M}\sum_{k=1}^{M}\bar{\beta}_i\beta_k\widetilde{\varphi}_i^*\widetilde{\varphi}_k\right] = \sum_{i=1}^{M}\sum_{k=1}^{M}\delta_{ik}\sigma^2\widetilde{\varphi}_i^*\widetilde{\varphi}_k \qquad (2.30) \\
&= \sigma^2\sum_{k=1}^{M}\left\|\widetilde{\varphi}_k\right\|^2 = \sigma^2\sum_{k=1}^{M}\left\|(F^*F)^{-1}\varphi_k\right\|^2 \qquad (2.31)
\end{aligned}
$$

where (2.30) results from evaluating expectations using the conditions on β, and (2.31) uses (2.5). From (2.4) we can derive $B^{-2}\left\|\varphi_k\right\|^2 \leq \left\|(F^*F)^{-1}\varphi_k\right\|^2 \leq A^{-2}\left\|\varphi_k\right\|^2$, which simplifies to

$$B^{-2} \leq \left\|(F^*F)^{-1}\varphi_k\right\|^2 \leq A^{-2} \qquad (2.32)$$

because of the normalization of the frame. Combining (2.31) and (2.32) completes the proof.

2.A.3 Proof of Proposition 2.5

The proof is based on establishing the hypotheses of the following lemma:

Lemma 2.7 *Assume $x_c(t)$ defined in (2.7) has at least $n = 2W+1$ quantization threshold crossings (QTCs) and consider sampling at a rate of M samples per period. Then there exist constants $c > 0$ and $r_0 \geq 1$ depending only on $x_c(t)$ such that for any $M = rn \geq r_0 n$, whenever $x_c(t)$ and $x_c'(t)$ have the same quantized sampled versions, $T^{-1}\int_T |x_c(t) - x_c'(t)|^2\, dt < c/r^2$.*

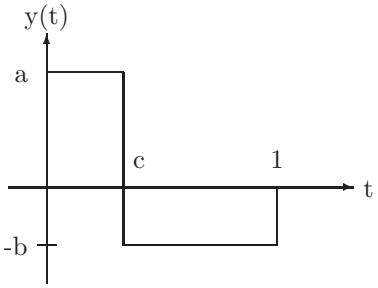


Figure 2.15: One period of the signal used in the proof of Lemma 2.8.

Proof: This is a version of [184, Thm. 4.1] for real-valued signals. □

The following lemma gives a rough estimate which allows us to relate signal amplitude to signal power:[9]

Lemma 2.8 *Among zero-mean, periodic signals with power P, the minimum possible peak-to-peak amplitude is $2\sqrt{P}$.*

Proof: We will construct a signal $y(t)$ with power P of minimum peak-to-peak amplitude. For convenience, let $T = 1$. Without loss of generality, we can assume that there exists c, $0 < c < 1$, such that $y(t) > 0$ for $t < c$ and $y(t) < 0$ for $t > c$. Then, to have minimum amplitude for given power, $y(t)$ must be piecewise constant as in Figure 2.15, with $a > 0$ and $b > 0$. The mean and power constraints can be combined to give $ab = P$. Under this constraint, the amplitude $a + b$ is uniquely minimized by $a = b = \sqrt{P}$. □

This final lemma relates the peak-to-peak amplitude of a continuous signal to its quantization threshold crossings:

Lemma 2.9 *A continuous, periodic signal with peak-to-peak amplitude $> A$ which is subject to uniform quantization with step size Δ has at least $2\lfloor A/\Delta \rfloor$ quantization threshold crossings per period.*

Proof: Consider first a signal $y(t)$ with peak-to-peak amplitude A. The "worst case" is for $A = k\Delta$ with $k \in \mathbb{Z}$, and for $\min y(t)$ and $\max y(t)$ to lie at quantization thresholds. In this case, the most we can guarantee is $k-1$ "increasing" QTCs and $k-1$ "decreasing" QTCs per period. If the peak-to-peak amplitude exceeds A, this worst case cannot happen, and we get at least $2\lfloor A/\Delta \rfloor$ QTCs. □

Proof of the proposition:

i. N odd: Quantized frame expansion of x with M frame vectors is precisely equivalent to quantized sampling of $x_c(t)$ with M samples per period (see Section 2.2.1.2). Denote the quantized frame expansion of x and the corresponding continuous-time signal by x' and $x'_c(t)$, respectively. It is easy to verify that the average time-domain SE $T^{-1}\int_T |x_c(t) - x'_c(t)|^2\,dt$ is the same as the vector SE $\|x - x'\|^2$. Let $\tilde{x}_c(t) = x_c(t) - x_1$. Then $\tilde{x}_c(t)$ is a zero-mean, T-periodic signal with power $\left\|[x_2 \;\; x_3 \;\; \cdots \;\; x_N]^T\right\|^2$, which by hypothesis is greater than $((N+1)\Delta/4)^2$. Applying Lemma 2.8 we conclude that $\tilde{x}_c(t)$ has peak-to-peak amplitude greater than $(N+1)\Delta/2$. Since $x_c(t)$ has precisely the same peak-to-peak amplitude

[9] The notion of power is standard; for $y(t)$ with period T: $T^{-1}\int_T |y(t)|^2\,dt$.

as $\tilde{x}_c(t)$, we can apply Lemma 2.9 to $x_c(t)$ to conclude that $x_c(t)$ has at least $2\lfloor((N+1)\Delta/2)/\Delta\rfloor = 2\lfloor(N+1)/2\rfloor = N+1$ QTCs. Applying Lemma 2.7 with $n = N$ completes the proof.

ii. N even: We need only make slight adjustments from the previous case. Let

$$x_c(t) = \sum_{k=1}^{N/2} \left[x_{2k-1}\sqrt{2}\cos\frac{2\pi kt}{T} + x_{2k}\sqrt{2}\sin\frac{2\pi kt}{T} \right],$$

and define x' and $x'_c(t)$ correspondingly. Again the average time-domain SE is the same as the vector SE. The power of $x_c(t)$ equals $\|x\|^2 > ((N+2)\Delta/4)^2$. Applying Lemmas 2.8 and 2.9 implies that $x_c(t)$ has at least $2\lfloor((N+2)\Delta/2)/\Delta\rfloor = 2\lfloor(N+2)/2\rfloor = N+2$ QTCs. Applying Lemma 2.7, this time with $n = N+1$ to match the form of (2.7), completes the proof.

Note that the bounds in the hypotheses of the proposition are not tight. This is evidenced in particular by the fact that the bound in Lemma 2.8 is not attainable by bandlimited signals. For example, for $N = 2$ the minimum peak-to-peak amplitude is $\sqrt{2}\cdot 2\sqrt{P}$ and for $N = 4$ the minimum is $\approx 1.3657\cdot 2\sqrt{P}$, compared to the bound of $2\sqrt{P}$. Because of Gibbs' phenomenon, the bound is not even asymptotically tight, but a more complicated lemma would serve no purpose here.

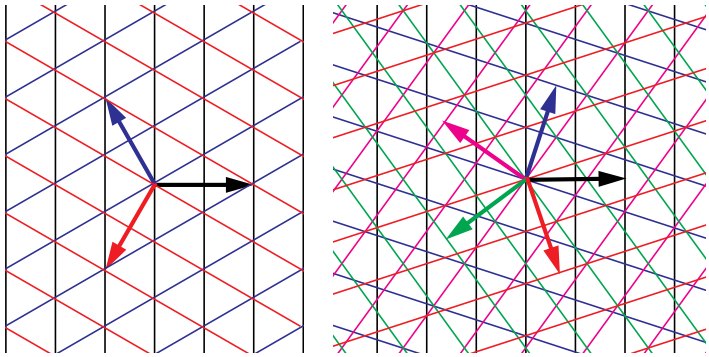
2.B Frame Expansions and Hyperplane Wave Partitions

This appendix gives an interpretation of frame coefficients as measurements along different directions. Given a frame $\Phi = \{\varphi_k\}_{k=1}^M$, the kth component of $y = Fx$ is $y_k = \langle x, \varphi_k\rangle$. Thus y_k is a measurement of x *along* φ_k. We can thus interpret y as a vector of M "measurements" of x in directions specified by Φ. Notice that in the original basis representation of x, we have N measurements of x with respect to the directions specified by the standard basis. Each of the N measurements is needed to fix a point in $\mathbb{R}^N$. On the other hand, the M measurements given in y have only N degrees of freedom.

Now let's suppose y is scalar-quantized to give $\hat{y}$ by rounding each component to the nearest multiple of Δ. Since y_k specifies the measurement of a component parallel to φ_k, $\hat{y}_k = (i+\frac{1}{2})\Delta$ specifies an $(N-1)$-dimensional hyperplane perpendicular to φ_k. Thus quantization of y_k gives a set of parallel hyperplanes spaced by Δ, called a *hyperplane single wave*. The M hyperplane single waves give a partition with a particular structure called a *hyperplane wave partition* [185]. Examples of hyperplane wave partitions are shown in Figure 2.16. In each figure, a set of vectors comprising a frame in $\mathbb{R}^2$ is shown superimposed on the hyperplane wave partition induced by quantized frame expansion with that frame.

Increasing the redundancy r of a frame can now be interpreted as increasing the number of directions in which x is measured. It is well-known that MSE is proportional to Δ^2. Section 2.2.2.4 presents a conjecture that MSE is proportional to $1/r^2$. This conjecture can be recast as saying that, asymptotically, increasing directional resolution is as good as increasing coefficient resolution.

In Section 2.2.2.5 it was mentioned that coding each component of $\hat{y}$ separately is inefficient when $r \gg 1$. This can be explained by reference to Figure 2.16. Specifying $\hat{y}_1$ and $\hat{y}_2$ defines a parallelogram within which x lies. Then there are a limited number of possibilities for $\hat{y}_3$. (In Figure 2.16(a), there are exactly two possibilities. In Figure 2.16(b), there are three or four possibilities.) Then with $\hat{y}_1$, $\hat{y}_2$, and $\hat{y}_3$ specified, there



(a) (b)

Figure 2.16: Examples of hyperplane wave partitions in $\mathbb{R}^2$: (a) $M = 3$. (b) $M = 5$.

are yet fewer possibilities for $\hat{y}_4$. If this is exploited fully in the coding, the bit rate should only slightly exceed the logarithm of the number of partition cells.

2.C Recursive Consistent Estimation

In the main text, reconstruction from quantized overcomplete expansions was considered with deterministic scalar quantization. Though deterministic quantization is most often used in practice, randomized (dithered) quantization is convenient for theoretical study. This approach is described in this appendix. In particular, after assuming that Q in Figure 2.1 is a subtractively dithered uniform scalar quantizer, one can obtain an $O(1/r^2)$ convergence result with no restriction on the source distribution and a very mild restriction on the frame sequence. Moreover, the $O(1/r^2)$ behavior is obtained with a simple, recursive reconstruction algorithm.[10] This reconstruction algorithm is suboptimal, but unlike standard linear reconstruction algorithms it is optimal to within a constant factor.

2.C.1 Introduction

It is common to analyze systems including a quantizer by modeling the quantizer as a source of signal-independent additive white noise. This model is precisely correct only when one uses subtractive dithered quantization [84, 126, 85] (see also Section 1.1.2.3), but for simplicity it is often assumed to hold for coarse, undithered quantization. What can easily be lost in using this model is that the distribution of the quantization noise can be important, especially its boundedness.

A focus of this chapter has been reconstruction from quantized overcomplete linear expansions. Assuming subtractive dither, this can be abstracted as the estimation of an unknown vector $x \in \mathbb{R}^N$ from measurements $y_k \in \mathbb{R}$:

$$y_k = \varphi_k^T x + e_k \qquad \text{for} \quad 0 \le k \le M-1, \tag{2.33}$$

[10] The development and analysis of the recursive reconstruction algorithm are due to Rangan. The ramifications in this context have been explored collaboratively [158].

where $\varphi_k \in \mathbb{R}^N$ are known vectors and $e_k \in \mathbb{R}$ is an independently identically distributed (i.i.d.) noise process with each e_k uniformly distributed on $[-\delta, \delta]$. The maximum noise magnitude $\delta > 0$ is half of the quantization step size and is known *a priori*. Estimation problems of this form may arise elsewhere as well. At issue are the quality of reconstruction that is possible and the efficient computation of good estimates.

The classical method for estimating the unknown vector x is least-squares estimation [127, 92], which attempts to find $\hat{x}$ such that the ℓ_2-norm of the residual sequence $y_k - \varphi_k^T \hat{x}$ is minimized. As in the undithered case, a least-squares estimate may be inconsistent with the bounds on the quantization noise and yields $O(1/M)$ MSE (see Section 2.2.2.3).[11]

A consistent reconstruction algorithm can easily be devised by modifying Table 2.1. Pursuant to the discussion of Section 2.2.2.4, we could expect the distortion to be bounded above by an $O(1/M^2)$ expression, but as of yet we have no way to prove this. In addition, the consistent estimation calculation is computationally expensive. Given M data points, finding a consistent estimate requires the solution of a linear program with N variables and $2M$ constraints. No recursive implementation of this computation is presently known.

This appendix introduces a simple, recursively-implementable estimator with a provable $O(1/M^2)$ MSE. The proposed estimator is similar to a consistent estimator except that the estimates are only guaranteed to be consistent with the most recent data point. The estimator can be realized with an extremely simple update rule which avoids any linear programming. Under suitable assumptions on the vectors φ_k, the simple estimator "almost" achieves the conjectured $O(1/M^2)$ MSE.

Under mild conditions on the *a priori* probability density of x, the MSE decay rate of any reconstruction algorithm is bounded below by $O(1/M^2)$. Thus, the proposed estimator is optimal to within a constant factor. An $O(1/M^2)$ lower bound was shown in [185] under weaker assumptions that do not require uniformly distributed white noise. However, with the uniformly distributed white noise model considered here, one can derive a simple expression for the constant in this lower bound [158]. The lower bound is derived from a recently developed version of the Ziv-Zakai bound [234] presented in [10].

The following section describes the proposed algorithm and its convergence rate. A numerical example is given in Section 2.C.3 to demonstrate the $O(1/M^2)$ MSE of the proposed algorithm, a fully consistent reconstruction algorithm, and the Ziv-Zakai lower bound. To contextualize this work, the ramifications for source coding are discussed in Section 2.C.4. These results may also be of interest to harmonic analysts studying the robustness of various overcomplete representations, such as nonorthonormal discrete wavelet expansions [37].

2.C.2 Proposed Algorithm and Convergence Properties

Suppose $x \in \mathbb{R}^N$ is an unknown vector, and we obtain a set of observations y_k given by (2.33). We wish to find an estimate, $\hat{x}_k$, of the unknown vector x from the data y_i and a_i for $i = 0, 1, \ldots, M-1$. The noise e_k is unknown, but bounded: $|e_k| \leq \delta$ for all k.

Consider the following simple recursive estimation scheme:

$$\hat{x}_{k+1} = \hat{x}_k + \frac{\varphi_k}{\varphi_k^T \varphi_k} \phi(y_k - \varphi_k^T \hat{x}_k), \tag{2.34}$$

[11] Since the dimension N is fixed, $O(1/r)$ and $O(1/M)$ are interchangeable.

where

$$\phi(e) = \begin{cases} 0, & \text{if } |e| \leq \delta, \\ e - \delta, & \text{if } e > \delta, \\ e + \delta, & \text{if } e < -\delta. \end{cases} \tag{2.35}$$

(ϕ is a soft-thresholding function.) Any initial estimate $\hat{x}_0$ may be used.

The motivation behind this estimator is simple. If an observation y_k is consistent with the estimate $\hat{x}_k$, (*i.e.*, $|y_k - \varphi_k^T \hat{x}_k| \leq \delta$), then the estimate is unchanged. That is, $\hat{x}_{k+1} = \hat{x}_k$. If the observation is not consistent, then $\hat{x}_{k+1}$ is taken to be the closest point to $\hat{x}_k$ consistent with the observation.

Two main results concerning this algorithm have been obtained. The first result states that the estimation error decreases monotonically for any noise sequence e_k with $|e_k| \leq \delta$. No statistical assumptions are needed.

Theorem 2.10 *Fix a vector $x \in \mathbb{R}^N$, and consider the algorithm (2.34) acting on a sequence of observations y_k given by (2.33). If $|e_k| \leq \delta$, then*

$$\|x - \hat{x}_{k+1}\| \leq \|x - \hat{x}_k\|.$$

Proof: A proof due to Rangan appears in [158]. □

For the second result, we impose the following assumptions.

Assumptions *For the measurements (2.33) and algorithm (2.34),*

(a) e_k and φ_k are independently distributed random processes, independent from each another;

(b) e_k is uniformly distributed on $[-\delta, \delta]$; and

(c) there exist constants $L > 0$ and $\sigma > 0$ such that for all k, $\|\varphi_k\|^2 \leq L$ and

$$E[|\varphi_k^T z|] \geq \sigma \|z\| \qquad \text{for all} \quad z \in \mathbb{R}^N. \tag{2.36}$$

These assumption provide the simplest scenario in which to examine the algorithm, and are similar to those used in the classical analysis of the least mean squares (LMS) algorithm (see for example [127, 92]). The assumption (2.36), in particular, is a standard and mild persistent excitation condition.

The independence assumption on the vectors a_k is, however, somewhat restrictive, especially for analysis with deterministic a_k. It should be noted that Assumption (a) does not require the vectors φ_k to be identically distributed or have zero mean.

Theorem 2.11 *Fix a vector $x \in \mathbb{R}^N$, and consider the algorithm (2.34) acting on a sequence of observations y_k given by (2.33). If Assumptions (a)–(c) is satisfied then, for every $p < 1$,*

$$\|x - \hat{x}_k\| k^p \to 0 \text{ almost surely.}$$

Proof: A proof by Rangan appears in [158]. □

Theorem 2.11 is the main result on the performance of the algorithm (2.34). The result states that, under suitable assumptions, the estimation error, $\|x - \hat{x}_k\|^2$, converges to zero, and for all $p < 1$, the rate of convergence is $o(k^{-2p})$. In this sense, the mean square error, is "almost" $O(1/k^2)$. As stated in the introduction, this rate is superior to the $O(1/k)$ attained by classical least-squares estimation.

2.C.3 A Numerical Example

As a numerical test, the performance of the proposed recursive algorithm was compared against two other reconstruction methods: a linear programming (LP) algorithm and a classical recursive least squares (RLS) algorithm. For each of the algorithms, the average MSE was measured as a function of the number of samples.

The linear programming (LP) algorithm selects the vector x which minimizes the ℓ_∞ norm of the residual sequence $y_k - \varphi_k^T x$. This estimate corresponds to the maximum likelihood estimate when both x and δ are treated as unknown. The computation of the LP estimate involves the solution of a linear program with $N+1$ variables and $2M$ constraints, where M is the number of samples. This computation cannot be implemented recursively, and the linear program must be recomputed with each new sample. The LP estimate is the most computationally demanding of the three algorithms tested, but is the only one that produces estimates consistent with the noise bounds on all the samples available.

The recursive least squares (RLS) algorithm selects the vector x which minimizes the ℓ_2 norm of the residual sequence $y_k - \varphi_k^T x$. The RLS estimate is not in general consistent with the noise bounds, but can be computed with a simple recursive update [92].

For the test, data in (2.33) was generated with $N = 4$ and $\{\varphi_k\}$ being an i.i.d. process, uniformly distributed on the unit sphere in $\mathbb{R}^4$. We used a noise bound of $\delta = 1$. The algorithms were started with an initial error of $x - \hat{x}_0 = [1, 1, 1, 1]^T$. Figure 2.17(a) shows the results of a single simulation. As expected, the proposed recursive method yields nonincreasing distortion.

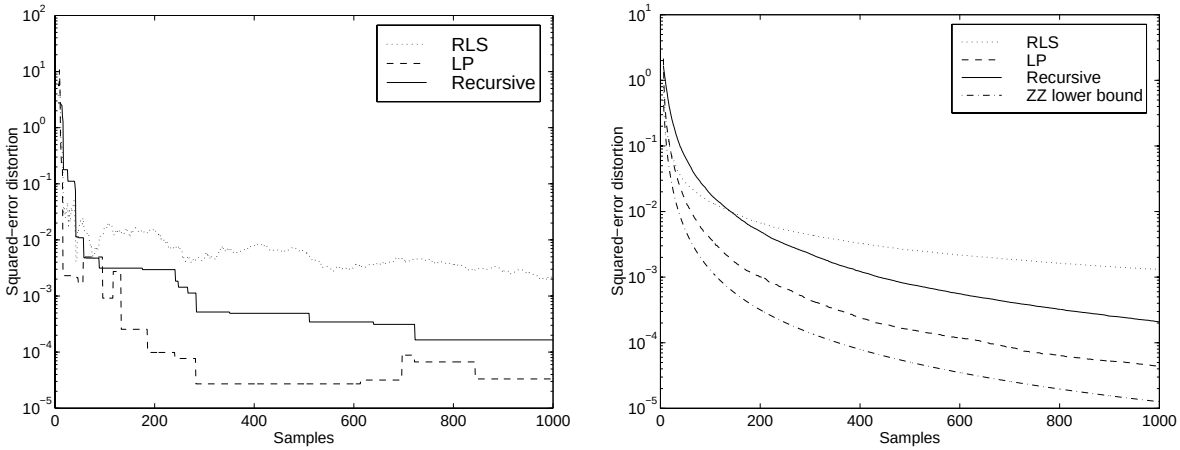
Figure 2.17(b) shows the averaged results of 1000 simulations. Also plotted is the Ziv-Zakai MSE lower bound from [158]. The asymptotic slopes of the curves confirm the $O(1/M)$ MSE for least-squares estimation and the $O(1/M^2)$ MSE for the consistent LP estimation and the proposed algorithm. While very simple and recursive, the proposed algorithm performs only a constant factor worse than the non-recursive consistent reconstruction and the theoretical lower bound.

2.C.4 Implications for Source Coding and Decoding

Thus far our discussion has been limited to the problem of estimating x given the y_k and φ_k sequences. This section presents the implications for using an entropy-coded version of y_k as a source encoding for x. Specifically, let $x \in \mathbb{R}^N$ be an arbitrary source vector. A representation of x can be formed through $y = Q_K(Fx)$, where F is an $M \times N$ matrix and $Q_K(\cdot)$ is an optimal K-dimensional entropy-coded dithered lattice quantizer [231, 220]. (The $K = 1$ case is the uniform scalar quantizer used in previous sections.)

If x comes from sampling a bandlimited, periodic signal at the Nyquist rate and F is a Fourier matrix, this corresponds to the encoding for bandlimited continuous-time signals considered by Zamir and Feder [221]. They showed that—*using least-squares reconstruction*—the MSE for the scheme is fixed as long as the ratio of the oversampling factor to the second moment of the quantizer is kept constant. They also showed that as the dimension of the lattice quantizer is increased, the performance of this scheme for a Gaussian source and MSE distortion measure approaches the rate–distortion bound [222].

Instead of least-squares reconstruction, now consider using algorithm (2.34) for estimating the vector x from the quantized data, $y = Q_K(Fx)$. Although our analysis does not directly apply to the case when F is



(a) Single simulation (b) Average of 1000 simulations

Figure 2.17: Comparison of three reconstruction algorithms. "RLS" refers to the recursive minimum ℓ_2-error reconstruction; "LP" refers to a linear program reconstruction which computes an estimate consistent with the smallest possible noise δ; and "Recursive" refers to the proposed algorithm. "ZZ lower bound" is the Ziv-Zakai theoretical lower bound.

a deterministic matrix, Assumption (c) is satisfied under any arbitrarily small random perturbation of the rows of F. Thus, our analysis for random matrices F, should apply to *generic* deterministic matrices as well.

Also, although we have described the algorithm (2.34) only for the scalar quantization case $K = 1$, it is straightforward to extend the algorithm to the lattice quantization case when $K > 1$. For the $K = 1$ case, we have taken each sample to specify a pair of hyperplane constraints on x. This can be viewed as a rotated and shifted version of the Cartesian product of a one-dimensional lattice quantizer cell (an interval) and $\mathbb{R}^{N-1}$. For general K, each set of K samples specifies a constraint on x that is a rotated and shifted version of the Cartesian product of a K-dimensional cell with $\mathbb{R}^{N-K}$. An iterative reconstruction algorithm could update its estimate every K samples with the nearest point of this set.

Theorem 2.11 shows that the reconstruction algorithm (2.34), or the extension of the algorithm for lattice vector quantization, will attain an $O(1/r^2)$ MSE, where $r = M/N$ is the oversampling ratio. This rate is superior to the $O(1/r)$ rate attained by least-squares reconstruction, so a reconstruction algorithm that utilizes the hard bounds on the quantization error may have rate–distortion performance better than that described in [221, 222]. In the limit of high oversampling, the MSE would remain fixed when the ratio of the *square* of the oversampling ratio to the second moment of the quantizer is kept constant. Furthermore, with the extension to lattice quantization, the performance may approach the rate–distortion bound more quickly as the lattice dimension is increased. The last comment bears further investigation.

2.C.5 Final Comments

These results serve as a reminder that simple linear filtering is not optimal for removing non-Gaussian additive noise, even if it is white. Accordingly, the improvement from consistent reconstruction is not because of the determinism of quantization noise, but because of its boundedness.

Motivated by a source coding application, and for concreteness in the proof of Theorem 2.11, uniformly

distributed noise was assumed. However, the algorithm itself uses only the bound on the noise δ. This raises the broader issue of the value of hard information. The Cramér-Rao theorem implies that, in the presence of i.i.d. noise with an unbounded, smooth distribution, the MSE of any estimator is bounded below by an $O(1/M)$ rate. That this classic limit can be surpassed for certain bounded noise models suggests that hard information may be fundamentally more informative than "soft," or probabilistic, information. In many systems, all the signals—including the noise—can be bounded using certain physical considerations. This sort of "hard" information should be exploited fully.

Chapter 3

On-line Universal Transform Coding

A SOURCE CODING TECHNIQUE that asymptotically (in the length of the data sequence) achieves the best possible performance amongst a class of codes for all sources within some class is called *universal*. The ubiquity of universal lossless coding methods is well-documented, the commonplace example being that most general-purpose data compression is done with some variant of Lempel–Ziv encoding [233]. Universal lossy coding is less common in practice and is an active area of research.

This chapter presents two algorithms for transform coding of sources that emit independent and identically distributed (i.i.d.) Gaussian N-tuples. The algorithms use *on-line adaptation*, meaning that the encoder and decoder adapt in unison based on the coded data without the explicit transmission of coder parameters. The first algorithm uses subtractive dither and is shown to be universal among high rate transform coders of fixed dimension N. Similar results are shown for a second algorithm that does not use dithering.

3.1 Introduction

Zhang and Wei [227] have identified three typical approaches to universal lossy coding. It is instructive to explore where the present algorithm fits in this taxonomy before delving into the details.

The first approach is to use a "universal codebook" which has been trained in advance and is available to the encoder and decoder. Conceptually, the universal codebook contains the union of the optimal codebooks for each source in the class. The coding operation can be described as a two-stage process: in the first stage, a codebook is selected; in the second stage, this codebook is used to code the source. The transmission of the choice of codebook is overhead information; one hopes to make this negligibly small. This approach was used in an existence proof by Neuhoff *et al.* [142] and in universal code design by Chou *et al.* [28].

A second approach does not use codebooks which have been trained in advance. Instead, the encoder constructs a codebook based on a finite length observation of the signal and transmits the codebook to the decoder. Methods that are *(forward) adaptive* fall into this category. Universality of such methods was studied by Ziv [229, 230] and others.

The third, pursued in this work, is to adapt the codebook simultaneously at the encoder and decoder

This chapter includes research conducted jointly with Jun Zhuang and Martin Vetterli [82, 80, 81]. (In addition, Christopher Chan contributed to [82].)

in such a way that the codebook does not need to be transmitted. This is called *on-line adaptation*, *backward adaptation*, or *adaptation without side information*. String matching algorithms (like Lempel–Ziv) used in universal lossless coding are examples of on-line adaptation schemes, and attempts have been made to extend these to lossy coding [178, 129]. Zhang and Wei's "gold washing" algorithm [227] also uses on-line adaptation. The algorithms introduced in this chapter are the first on-line universal *transform coding* methods.

Though several papers on adaptive transform coding have appeared in the literature (*e.g.*, see [42]), it seems that Effros and Chou [46] have done the only work on universal transform coding. Thus this work can be viewed as an "on-line" alternative to [46]. The results of [46] were somewhat inspiring to this study because they indicated superior performance of weighted universal transform coding over weighted universal VQ for reasonable vector dimensions.

The literature contains some more restrictive definitions of "universal." If one requires performance to approach the rate–distortion bound, as opposed to the best performance obtainable in the particular class of coders, then no transform coding method would be universal because of the loss due to *scalar* quantization of transform coefficients. Davisson [40] requires a universal code to be blockwise memoryless, *i.e.*, each block must be coded independent of past and future blocks. To reduce the possibility of confusion, one can refer to the technique introduced in this chapter as "adaptive universal." However, to insist on this distinction is somewhat disingenuous. If the method presented here is used to code K N-tuples, we can view it as a universal code of block length KN where the performance of an optimal N-tuple transform coder is obtained arbitrarily closely as $K \to \infty$. The particular construction of the code happens to have the property that successive N-tuples can be decoded without waiting for the end of an entire KN-length block.

The two algorithms that are analyzed in this chapter are defined in the following section. The main results are stated in Section 3.3 and proven in Section 3.4. Section 3.5 describes ways the algorithms can be modified to reduce computational complexity or to operate on non-i.i.d. sources. Experimental results, on both synthetic sources and images, are presented in Section 3.6. Section 3.7 offers some final thoughts.

3.2 Proposed Coding Methods

The fundamental purpose of source coding is to remove redundancy. When scalar quantization and scalar entropy coding are to be used, correlation between components of a vector is a form of undesirable redundancy. Transform coding (and the choice of the transform therein) is based on removing this simple form of redundancy.

Let $\{x_n\}_{n\in\mathbb{Z}^+}$ be a sequence of independent, identically distributed, zero-mean Gaussian random vectors of dimension N with correlation matrix $R_x = E[xx^T]$.[1] Suppose the source is to be transform coded in a system with unbounded uniform scalar quantization and scalar entropy coding. As discussed in Section 1.1.3, an optimal transform is one that results in uncorrelated transform coefficients; *i.e.*, a Karhunen–Loève transform (KLT) of the source.

The problem considered here is universal transform coding of such i.i.d. Gaussian sources with *unknown* correlation matrix R_x. It is easy to imagine a forward adaptive system which estimates R_x, computes an

[1] Throughout the chapter, R_v will be used to denote the (exact) covariance matrix $E[vv^T]$ of a random vector v. $\widehat{R_v}$ denotes an estimate of R_v obtained from a finite length observation. Aside from this convention, subscripts indicate the time index of a variable, except where two subscripts are given to indicate the row and column indices or where noted.

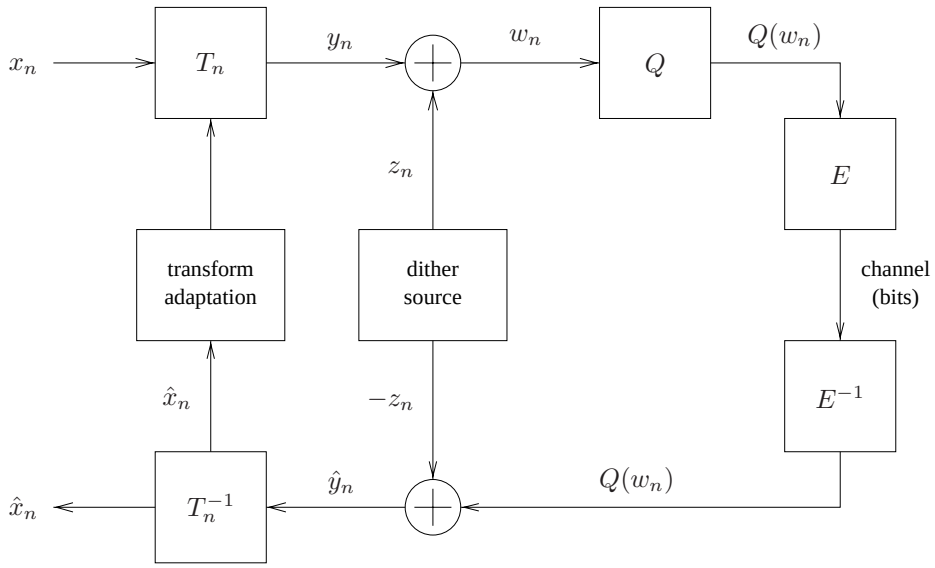


Figure 3.1: Structure of universal transform coding system with subtractive dither. T_n is a time-varying orthogonal transform, Q is a scalar quantizer, and E is a scalar entropy coder. Dithering makes the transform domain quantization error $y_n - \hat{y}_n$ independent of $\{x_k\}_{k\in\mathbb{Z}^+}$. In turn, $x_n - \hat{x}_n$ is uncorrelated with (but not independent of) $\{x_k\}_{k=1}^n$. Thus moments estimated from $\hat{x}_n$ suffice for the purpose of adapting the transform T_n.

approximate KLT (which must be communicated to the decoder), and uses this approximate KLT to code the source. This approach will not be pursued here, though one could presumably extend results of Rissanen [163] to show that such a system could be universal.[2] Instead, on-line approaches are considered as described below. The availability of a universal lossless coder is assumed, but, in contrast to [231], we will apply it only to sequences of scalars.

3.2.1 System with Subtractive Dither

The first coding method is depicted in Figure 3.1. Let T_1 be an arbitrary orthogonal matrix. This will be the initial transform used. Let $\{z_n\}$ be an i.i.d. dither sequence uniformly distributed on the hypercube $[-\Delta/2, \Delta/2]^N$, where $\Delta > 0$. Q represents a uniform scalar quantizer with step size Δ and E represents a universal scalar entropy coder. The dither signal may or may not be known to the entropy coder; we consider mainly the latter case. The diagram completely specifies the operation of the coder at a particular time step:

$$\begin{aligned} y_n &= T_n x_n \\ \hat{y}_n &= Q(y_n + z_n) - z_n \\ \hat{x}_n &= T_n^{-1} \hat{y}_n \end{aligned}$$

It remains to specify the transform update mechanism.

[2] Rissanen considers sources described by finite dimensional parametric models (like ours) which take values in a countable set (unlike ours).

Optimal performance is obtained when the transform diagonalizes R_x. In order for on-line adaptation to be possible, the adaptation must depend only on quantized data. Thus, form the estimate

$$\widehat{R_{\hat{x}}}^{(n)} = \frac{1}{n}\sum_{k=1}^{n} \hat{x}_k \hat{x}_k^T \tag{3.1}$$

from the quantized data and compute the eigendecomposition of $\widehat{R_{\hat{x}}}^{(n)}$ in order to fix T_{n+1} such that $T_{n+1}\widehat{R_{\hat{x}}}^{(n)}T_{n+1}^T$ is diagonal with nonincreasing diagonal elements. The calculation of T_{n+1} will always have sign ambiguities. If the eigenvalues of $\widehat{R_{\hat{x}}}^{(n)}$ are not distinct, there will be additional ambiguities. These can be resolved arbitrarily.

3.2.2 Undithered System

The second coding method is identical to the first except that the dither signal is eliminated. Also, specify the scalar quantizer function used on each component to be $q : \mathbb{R} \to \mathbb{R}$ given by

$$q(x) = i\Delta, \qquad \text{for} \quad \left(i - \frac{1}{2}\right)\Delta \le x < \left(i + \frac{1}{2}\right)\Delta. \tag{3.2}$$

This algorithm is more attractive for practical application but is more difficult to analyze.

The transform update strategy does not use information on the form of the source distribution, *i.e.*, the source is characterized only by its covariance matrix. Appendix 3.A gives a a short discussion of how a more general modeling of the source distribution could be used instead.

3.3 Main Results

The main results are summarized in this section. Proofs are given in Section 3.4.

3.3.1 System with Subtractive Dither

Theorem 3.1 (Convergence of system with subtractive dither) *For any initial transform T_1,*

$$\widehat{R_{\hat{x}}}^{(n)} \text{ converges in mean square to } R_x + \frac{\Delta^2}{12} I \text{ as } n \to \infty.$$

Also, the sequence of transforms $\{T_n\}$ converges in mean square to a KLT for the source.

Notice that the limit of $\widehat{R_{\hat{x}}}^{(n)}$ depends on Δ, but the convergence does not. Also, although we are assuming Gaussian signals throughout, the proof of the theorem does not depend on the distribution of the source. Hence this system will converge to a transform which maximizes coding gain for any i.i.d. source. However, for non-Gaussian sources maximizing coding gain may not be ideal.

When the source is Gaussian, the KLT is the optimal transform and the entropies of the quantized variables are easily estimated. This leads to the following theorem:

Theorem 3.2 (High rate universality of system with subtractive dither) *Denote the eigenvalues of R_x by $\lambda_1, \lambda_2, \ldots, \lambda_N$. Let L_n denote the per component code length for coding the first n vectors using the universal*

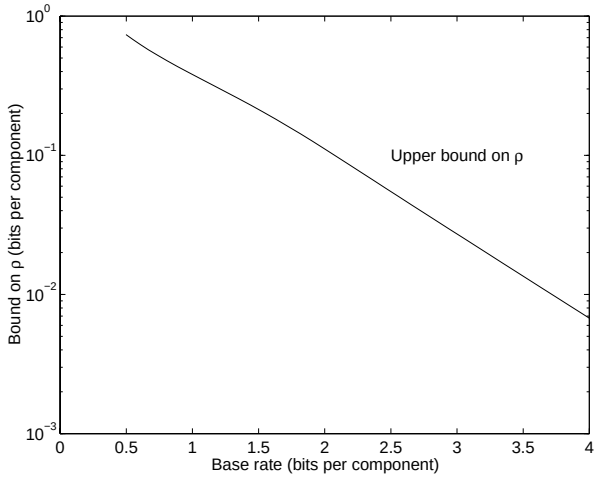


Figure 3.2: Bound (3.3) on the excess rate ρ as a function of the coding rate for a Gaussian source with $(R_x)_{ij} = 0.8^{|i-j|}$.

scheme where the dither signal is not *known to the entropy coder and let* $L_n^\star$ *denote the per component code length for coding the first* n *vectors with a fixed, optimal transform coder (no dither, and transform and scalar entropy coder designed with knowledge of* R_x*). Then the average excess rate* $n^{-1}(L_n - L_n^\star)$ *converges in mean square to a constant* ρ*. Estimating discrete entropies using differential entropies [34],*

$$\rho < \frac{1}{2N} \sum_{i=1}^{N} \log_2 \left(1 + \frac{\Delta^2}{12\lambda_i} \right). \tag{3.3}$$

The constant ρ can be interpreted as the asymptotic redundancy of the system. It is the excess rate, in bits per source component, of the universal system, as compared to a fixed, optimal transform code designed with knowledge of R_x. The origin of this excess rate is the dither signal: $H(Q(w_n)) > H(Q(y_n))$.[3] The bound (3.3) comes simply from the variance of w_n. As $\Delta \to 0$, $\mathrm{Var}(w_n) \to \mathrm{Var}(y_n)$ and accordingly $\rho \to 0$. Thus the system is universal for high rate coding.

At moderate rates, ρ is quite small. For example, consider the coding of an eight-dimensional Gaussian source with $(R_x)_{ij} = 0.8^{|i-j|}$. By computing the bound (3.3) and the correspondence between Δ and the rate of a KLT coder for this particular source, one gets the curve shown in Figure 3.2. This roughly indicates that ρ must decay exponentially with the overall coding rate. In fact, using the high rate approximation

$$\frac{\Delta^2}{12} \approx \frac{\pi e}{6} \left(\prod_{i=1}^{N} \lambda_i \right)^{1/N} 2^{-2R},$$

where R is the rate of the optimal transform coder, (3.3) can be written as

$$\rho < \frac{1}{2N} \sum_{i=1}^{N} \log_2 \left(1 + \frac{\Delta^2}{12\lambda_i} \right) < \frac{1}{2N \ln 2} \sum_{i=1}^{N} \frac{\Delta^2}{12\lambda_i} \approx \frac{\pi e}{12 \ln 2} \left(\prod_{i=1}^{N} \lambda_i \right)^{1/N} \left(\frac{1}{N} \sum_{i=1}^{N} \frac{1}{\lambda_i} \right) 2^{-2R}.$$

At rates of 2 or 3 bits per component, the excess rate is less than 6% or 1%, respectively.

[3] When the dither signal is known at the entropy coder, performance better than the worst case given by (3.3) can be expected [166].

3.3.2 Undithered System

The proof of Theorem 3.1 relies on the fact that dithering makes $\{\hat{x}_n \hat{x}_n^T\}$ an uncorrelated sequence and hence a law of large numbers can be applied to (3.1). Without dithering, it is very hard to make precise statements about the sequence of transforms. Due to the quantization, the distribution of $\hat{x}_n$ depends on T_n, which in turn depends on T_1 and $\{x_k\}_{k=1}^{n-1}$. Because of this complicated interdependence between quantization and stochastic effects, we are only able to prove restricted results.

One way to reduce the complexity of the analysis is to neglect the stochastic aspect, meaning to assume there is no variance in moment estimates despite the fact that moments are estimated from finite length observations. The effect is to replace (3.1) with

$$R_{\hat{x}}^{(n)} = E[\hat{x}_n \hat{x}_n^T] \tag{3.4}$$

and update the transform such that $T_{n+1} R_{\hat{x}}^{(n)} T_{n+1}^T$ is diagonal with nonincreasing diagonal elements. We are left with a deterministic iteration summarized by

$$R_{\hat{x}}^{(n)} \;=\; T_n^T R_{\hat{y}}^{(n)} T_n \;=\; T_n^T \widetilde{Q}(R_y^{(n)}) T_n \;=\; T_n^T \widetilde{Q}(T_n R_x T_n^T) T_n,$$

$$T_{n+1} R_{\hat{x}}^{(n)} T_n = \Lambda_n \text{ (diagonal with nonincreasing diagonal elements)},$$

where $\widetilde{Q} : \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N}$ gives the effect of quantization on the correlation matrix. $\widetilde{Q}$ depends on the source distribution and Δ and can be described by evaluating expressions from [27].

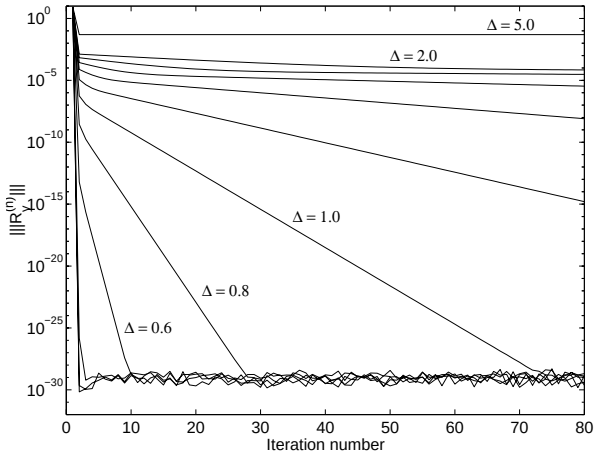
Since R_x and $R_{\hat{x}}^{(n)}$ generally have different eigenvectors, it is not obvious that this iteration will converge. The following theorem, due to Zhuang [228, 81], gives a limited convergence result:

Theorem 3.3 (Deterministic convergence of undithered system) *Let R_x and T_1 be given. Then there exists a sequence of quantization step sizes $\{\Delta_n\} \subset \mathbb{R}^+$ such that the deterministic iteration described above converges to a KLT of the source. Since the KLT is ambiguous if the eigenvalues of R_x are not distinct, convergence is indicated by $R_{\hat{y}}^{(n)}$ approaching a diagonal matrix in Frobenius norm.*

Theorem 3.3 does not preclude the possibility that the iteration will converge only with $\inf \Delta_n = 0$. This limits the practical implications of the theorem. However, numerical calculations suggest that the iteration actually converges for sufficiently small constant sequences of step sizes. Figure 3.3 shows numerical results for a four-dimensional Gaussian source with $(R_x)_{ij} = 0.9^{|i-j|}$, $T_1 = I$ and various values of Δ. To show the degree to which T_n diagonalizes R_x, $|||R_y^{(n)}|||$ is plotted as a function of the iteration number n, where $|||A||| = \sum_{i \neq j} a_{ij}^2$. An approximate correspondence between quantization step size and rate is also given.

Starting from an arbitrary initial transform, $|||R_y^{(n)}|||$ becomes small after a single iteration (note the logarithmic vertical axis). Then, to the limits of machine precision, it converges exponentially to zero with a rate of convergence that depends on Δ. (For $\Delta > 3$, loss of significance problems in the computation combined with very slow convergence make it difficult to ascertain convergence numerically.)

The results shown in Figure 3.3 are representative of the performance with an arbitrary R_x. The convergence, as measured by $|||R_y^{(n)}|||$, is unaffected by the multiplicities of the eigenvalues of R_x. The eigenspace associated with a multiple eigenvalue can be rotated arbitrarily without affecting $|||R_y^{(n)}|||$ or the decorrelation and energy compaction properties of the transform.



Δ	Rate (bits/component)
0.1	4.475
0.2	3.488
0.4	2.536
0.6	2.006
0.8	1.620
1.0	1.322
1.2	1.097
1.4	0.929
1.6	0.802
1.8	0.702
2.0	0.623
5.0	0.218

Figure 3.3: Simulations for various fixed quantization step sizes suggest that the deterministic iteration converges more generally than predicted by Theorem 3.3. The source vector length is $N = 4$ and the initial transform is the identity transform. The accompanying table provides an approximate correspondence between quantization step sizes and rates.

Theorem 3.4 (Deterministic convergence: $N = 2$) *Let $N = 2$ and let R_x be given. There exists $\Delta_{\max} > 0$ such that for any $\Delta < \Delta_{\max}$ the deterministic iteration converges, in the same sense as before, for any initial transform T_1.*

The analysis of the dithered system was relatively straightforward because the dithering statistically separated the variation of the transform from the effects of quantization on moment estimation. In place of the deterministic approach, it is also possible to analyze the undithered system with an "independence assumption" similar to that used in analyzing the LMS algorithm. This forcibly disentangles finite observation effects from quantization effects. A convergence result can be shown, though the situation is rather unrealistic; see Appendix 3.B for details.

3.4 Derivations

3.4.1 Proof of Theorem 3.1

The idea of the proof is to first show that each term of (3.1) has expected value $R_x + \Delta^2 I/12$, has finite variance, and is elementwise uncorrelated with every other term. Applying the Chebyshev law of large numbers [151] then gives the first desired conclusion. The second conclusion follows easily.

First note that

$$\hat{x}_k \;=\; T_k^T \hat{y}_k \;=\; T_k^T (y_k + (\hat{y}_k - y_k)) \;=\; x_k + T_k^T (\hat{y}_k - y_k).$$

Because of the use of subtractive dither, $\hat{y}_k - y_k$ is uniformly distributed on the hypercube $[-\Delta/2, \Delta/2]^N$ and independent of x_k and T_k [126, 85]. (The overall error $\hat{x}_k - x_k$ is uniformly distributed on a rotated hypercube, independent of x_k but not independent of T_k. Its components are uncorrelated but not independent.) Now,

$$\hat{x}_k \hat{x}_k^T = x_k x_k^T + x_k (\hat{y}_k^T - y_k^T) T_k + T_k^T (\hat{y}_k - y_k) x_k^T + T_k^T (\hat{y}_k - y_k)(\hat{y}_k^T - y_k^T) T_k \tag{3.5}$$

and taking expectations gives

$$\begin{aligned} E[\hat{x}_k \hat{x}_k^T] &= E[x_k x_k^T] + E[x_k(\hat{y}_k^T - y_k^T)]T_k + T_k^T E[(\hat{y}_k - y_k)x_k^T] + T_k^T E[(\hat{y}_k - y_k)(\hat{y}_k^T - y_k^T)]T_k \\ &= R_x + 0 + 0 + T_k^T \frac{\Delta^2}{12} I\, T_k \\ &= R_x + \frac{\Delta^2}{12} I \end{aligned}$$

where we have used the independence of x_k and $\hat{y}_k - y_k$ and the fact that each has mean zero. The (i,j) element of $\widehat{R_{\hat{x}}}^{(n)}$ is the average of n random observations of $(\hat{x}_k \hat{x}_k^T)_{ij}$, which we denote $A_{ij}^{(k)}$. The calculation above shows that each $A_{ij}^{(k)}$ has mean $(R_x)_{ij} + \Delta^2 \delta_{ij}/12$. It is shown in Appendix 3.C that each $A_{ij}^{(k)}$ has variance bounded by a constant and that $A_{ij}^{(k)}$ is uncorrelated with $A_{ij}^{(\ell)}$ for $k \neq \ell$. Thus by the Chebyshev law of large numbers we have that $\widehat{R_{\hat{x}}}^{(n)} \to R_x + \Delta^2 I/12$ elementwise in mean square. The second conclusion follows from the fact that R_x and $R_x + \Delta^2 I/12$ have the same eigenvectors.

Note that the dither is essential to the proof because it makes the quality of the estimate $\widehat{R_{\hat{x}}}^{(n)}$ independent of the sequence of transforms.

3.4.2 Proof of Theorem 3.2

Let $N\ell_n$ denote the number of bits used to code x_n. Because of the mean square convergence of the sequence of transforms and the universality of the entropy coder, $E[\ell_n]$ converges to some limit, say $\bar{\ell}$. The number of bits used by the optimal coder $N\ell_n^\star$ will satisfy $E[\ell_n^\star] = \bar{\ell}^\star$. Since the mean of a sequence that converges to a constant must converge to the same limit, the first part of the theorem is proven with $\rho = \bar{\ell} - \bar{\ell}^\star$.

The second part of the theorem pertains to rate estimates. The optimal fixed transform will use the KLT, so the discrete variables to be entropy coded will be quantized Gaussians with variances λ_1, λ_2, ..., λ_N. Assuming the relationship

$$H(Q_\Delta(X)) = h(X) - \log_2 \Delta$$

between differential entropy and discrete entropy [34] and using the differential entropy of a Gaussian random variable

$$h(\mathcal{N}(0,\sigma^2)) = \frac{1}{2}\log_2 2\pi e\sigma^2 \text{ bits}$$

gives the following for the entropy of a Gaussian random variable with variance σ^2, scalar quantized with bin width Δ:

$$H(Q_\Delta(\mathcal{N}(0,\sigma^2))) = \frac{1}{2}\log_2 2\pi e\sigma^2 - \log_2 \Delta = \frac{1}{2}\log_2 \frac{2\pi e\sigma^2}{\Delta^2} \text{ bits.}$$

Averaging the rates for the N transform coefficients gives

$$R_{\text{opt}} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\log_2 \frac{2\pi e\lambda_i}{\Delta^2}. \tag{3.6}$$

The universal scheme converges to an optimal transform. However, because of the dithering, the signal at the input to the quantizer is not Gaussian and does not have component variances equal to the λ_i's. Since $\{z_n\}$ is independent of $\{y_n\}$, the variances simply add, giving $\lambda_i + \Delta^2/12$, $i = 1, 2, \ldots, N$. Since a Gaussian p.d.f. has

the largest differential entropy for a given variance, the asymptotic rate of the universal coder can be bounded as

$$R_{\text{univ}} < \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\log_2\frac{2\pi e(\lambda_i + \Delta^2/12)}{\Delta^2}. \tag{3.7}$$

Subtracting (3.6) from (3.7) and pairing terms gives (3.3).

3.4.3 Proof of Theorem 3.3

The proofs of Theorems 3.3 and 3.4 rely on properties of $\widetilde{Q}$, the function that describes the effects of quantization on the correlation matrix. The proof of Theorem 3.3 presented here is primary due to Zhuang and is minimally modified from [228].

Let η_1 and η_2 be jointly Gaussian with $E[\eta_1] = E[\eta_2] = 0$, $E[\eta_1^2] = \nu_1^2$, $E[\eta_2^2] = \nu_2^2$, and $E[\eta_1\eta_2] = \nu_{12}$. Define $\hat{\nu}_1^2 = E[q(\eta_1)^2]$, $\hat{\nu}_2^2 = E[q(\eta_2)^2]$, and $\hat{\nu}_{12} = E[q(\eta_1)q(\eta_2)]$, where $q(\cdot)$ was defined by (3.2). Then using expressions from [27], one can show

$$\hat{\nu}_1^2 = \nu_1^2 + \frac{\Delta^2}{12} + \sum_{m=1}^{\infty}(-1)^m e^{-2m^2\pi^2\nu_1^2/\Delta^2}\left(\frac{\Delta^2}{m^2\pi^2} + 4\nu_1^2\right) \tag{3.8}$$

(similarly for $\hat{\nu}_2^2$) and

$$\hat{\nu}_{12} = (1+\delta)\nu_{12} + \mu, \tag{3.9}$$

where

$$\delta = 2\left(\sum_{m_1=1}^{\infty}(-1)^{m_1}e^{-2m_1^2\pi^2\nu_1^2/\Delta^2} + \sum_{m_2=1}^{\infty}(-1)^{m_2}e^{-2m_2^2\pi^2\nu_2^2/\Delta^2}\right)$$

and

$$\mu = \sum_{m_1,m_2=1}^{\infty}(-1)^{m_1+m_2}\frac{\Delta^2}{m_1m_2\pi^2}\exp\left(\frac{-2\pi^2(m_1^2\nu_1^2 + m_2^2\nu_2^2)}{\Delta^2}\right)\sinh\left(\frac{4\pi^2\nu_{12}m_1m_2}{\Delta^2}\right). \tag{3.10}$$

For any correlation matrix R, the diagonal elements of $\widetilde{Q}(R)$ are described by (3.8) and the off-diagonal elements are described by (3.9). For the purpose of this theorem, we need only the following simple property of $\widetilde{Q}$:

$$\widetilde{Q}(R) = R + \frac{\Delta^2}{12}I + C, \quad \text{where } C \to 0 \text{ elementwise as } \Delta \to 0. \tag{3.11}$$

To measure the degree to which T_n diagonalizes R_x, define a distance measure $|||\cdot|||$ between a matrix A and the set of diagonal matrices by $|||A||| = \sum_{i\neq j} a_{ij}^2$. The strategy of the proof is to show that for sufficiently small Δ and $n \geq 1$, the inequality $|||R_y^{(n+1)}||| \leq \frac{1}{2}|||R_y^{(n)}|||$ holds.

Combining $R_{\hat{x}}^{(n)} = T_n^T R_{\hat{y}}^{(n)} T_n$ with $R_{\hat{x}}^{(n)} = T_{n+1}^T \Lambda_n T_{n+1}$ gives $T_{n+1}^T\Lambda_n T_{n+1} = T_n^T R_{\hat{y}}^{(n)} T_n$. Define $H_n = T_n T_{n+1}^T$ so that

$$R_{\hat{y}}^{(n)} = H_n\Lambda_n H_n^T. \tag{3.12}$$

Also notice that

$$R_y^{(n+1)} = T_{n+1}R_xT_{n+1}^T = T_{n+1}T_n^TT_nR_xT_n^TT_nT_{n+1}^T = H_n^TR_y^{(n)}H_n.$$

As a final preparation, define $Z_n = R_y^{(n)} - R_{\hat{y}}^{(n)}$.

We can now make the calculation

$$|||R_y^{(n+1)}||| = |||H_n^TR_y^{(n)}H_n||| = |||H_n^T(Z_n + R_{\hat{y}}^{(n)})H_n||| = |||H_n^TZ_nH_n|||,$$

where the last equality follows from $H_n^T R_{\hat{y}}^{(n)} H_n$ being diagonal (see (3.12)). From (3.11), it is clear that if Δ is small enough, $|||Z_n||| \leq \frac{1}{4}|||R_y^{(n)}|||$. It remains now to relate $|||Z_n|||$ and $|||H_n^T Z_n H_n|||$.

Substitute $R_{\hat{y}}^{(n)} = R_y^{(n)} + \Delta^2 I/12 + C_1$, where $\|C_1\| \to 0$ as $\Delta \to 0$, in (3.12) to get

$$R_y^{(n)} + \frac{\Delta^2}{12} I + C_1 = H_n \Lambda_n H_n^T. \tag{3.13}$$

Decrementing the index and rearranging gives

$$H_{n-1}^T R_y^{(n-1)} H_{n-1} + \frac{\Delta^2}{12} I + H_{n-1}^T C_1 H_{n-1} = \Lambda_{n-1}. \tag{3.14}$$

Since $H_{n-1}^T R_y^{(n-1)} H_{n-1} = R_y^{(n)}$, comparing (3.13) and (3.14) gives

$$H_n \Lambda_n H_n^T = \Lambda_{n-1} + C_1 - H_{n-1}^T C_1 H_{n-1}. \tag{3.15}$$

Now let $C_2 = H_n - I$. Substituting in (3.15) and expanding, we conclude that $\|C_2\| \to 0$ as $\Delta \to 0$. Thus by expanding $H_n^T Z_n H_n$ we see that $|||H_n^T Z_n H_n||| - |||Z_n||| \to 0$ faster than $|||Z_n||| \to 0$ as $\Delta \to 0$, so by choosing Δ small enough we have the bound $|||H_n^T Z_n H_n||| \leq 2|||Z_n|||$.

Combining all these calculations gives

$$|||R_y^{(n+1)}||| \;=\; |||H_n^T Z_n H_n||| \;\leq\; 2|||Z_n||| \;\leq\; 2 \cdot \frac{1}{4}|||R_y^{(n)}||| \;=\; \frac{1}{2}|||R_y^{(n)}|||.$$

3.4.4 Proof of Theorem 3.4

Without loss of generality (rotating the coordinate system and initial transform, if necessary), assume $R_x = \mathrm{diag}(\sigma_1^2, \sigma_2^2)$, $\sigma_1 \geq \sigma_2$. The transform iterates are all in $SO_2(\mathbb{R})$ and can be parameterized as

$$T_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

where $\theta \in [-\pi/4, \pi/4]$. We assume $\sigma_1 > \sigma_2$; if $\sigma_1 = \sigma_2$ the situation is uninteresting because R_y is diagonal for any T_θ.

Denote the transform iterate that follows after θ by φ. The proof will be completed by showing that there is a constant $\Delta_{\max}$, independent of θ, such that $\Delta \leq \Delta_{\max}$ implies $\sin^2 2\varphi \leq \sin^2 2\theta$ with equality only when $\theta = 0$. This will show global convergence to the fixed point zero, which is an optimal transform. As a preview of this result—and to motivate the rest of the proof—we compute and plot the "next iterate" map $\theta \mapsto \varphi$. Figure 3.4 shows the map $\theta \mapsto \varphi$ when $\sigma_1 = 1$ and $\sigma_2 = 1/2$ for $\Delta = 1, 2, \ldots, 5$. The iteration globally converges as long as the graph of $\varphi(\theta)$ lies inside the cone $|\varphi| \leq |\theta|$. From the plot, it seems this may be true for any Δ; we endeavor to show this for Δ less than some $\Delta_{\max}$.

The first step is to relate φ to θ. By looking at the general form of $T_\theta R_{\hat{x}} T_\theta^T$, one can show that

$$\varphi = \frac{1}{2} \arctan\left(\frac{-2(R_{\hat{x}})_{1,2}}{(R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2}} \right). \tag{3.16}$$

$R_{\hat{x}}$ is related to θ through R_y and $R_{\hat{y}}$:

$$R_y \;=\; T_\theta R_x T_\theta^T \;=\; \begin{bmatrix} \sigma_1^2 \cos^2\theta + \sigma_2^2 \sin^2\theta & \frac{1}{2}(\sigma_1^2 - \sigma_2^2)\sin 2\theta \\ \frac{1}{2}(\sigma_1^2 - \sigma_2^2)\sin 2\theta & \sigma_1^2 \sin^2\theta + \sigma_2^2 \cos^2\theta \end{bmatrix} = \begin{bmatrix} \nu_1^2 & \nu_{12} \\ \nu_{12} & \nu_2^2 \end{bmatrix}, \tag{3.17}$$

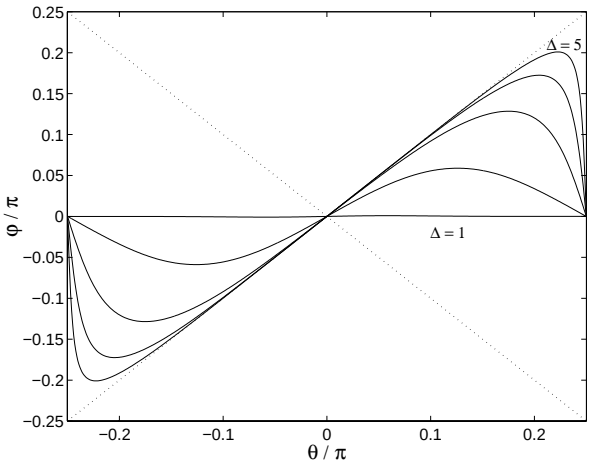


Figure 3.4: Simulations of the deterministic iteration for $N = 2$ suggest convergence for any quantization step size Δ. The eigenvalues of R_x are 1 and 1/4. The step sizes shown are $\Delta = 1, 2, \ldots, 5$. φ is the iterate that follows after θ. Global convergence is indicated by the curves lying inside the cone $|\varphi| \le |\theta|$, which is marked by dotted lines.

$$R_{\hat{y}} \;=\; \widetilde{Q}(R_y) \;=\; R_y + \frac{\Delta^2}{12} I + \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix},$$

where α, β, and γ depend on θ, σ_1, and σ_2 as given by (3.8), (3.9), and (3.17). Now, after computing $R_{\hat{x}} = T_\theta^T R_{\hat{y}} T_\theta$, one finds

$$(R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2} = \sigma_1^2 - \sigma_2^2 + (\alpha - \gamma)\cos 2\theta + 2\beta \sin 2\theta$$

and

$$(R_{\hat{x}})_{1,2} = \frac{1}{2}(\gamma - \alpha)\sin 2\theta + \beta \cos 2\theta. \tag{3.18}$$

Since $\sin^2(\arctan\phi) = \phi^2/(1+\phi^2)$,

$$\begin{aligned} \sin^2 2\varphi \;&=\; \frac{\left(\frac{-2(R_{\hat{x}})_{1,2}}{(R_{\hat{x}})_{1,1}-(R_{\hat{x}})_{2,2}}\right)^2}{1 + \left(\frac{-2(R_{\hat{x}})_{1,2}}{(R_{\hat{x}})_{1,1}-(R_{\hat{x}})_{2,2}}\right)^2} \;=\; \frac{(-2(R_{\hat{x}})_{1,2})^2}{((R_{\hat{x}})_{1,1} - (R_{\hat{x}})_{2,2})^2 + (-2(R_{\hat{x}})_{1,2})^2} \\ &=\; \frac{[(\alpha-\gamma)^2 \sin 2\theta - 2\beta\cos 2\theta]^2}{(\sigma_1^2 - \sigma_2^2)^2 + 2(\sigma_1^2 - \sigma_2^2)[(\alpha-\gamma)^2\cos 2\theta - 2\beta \sin 2\theta] + (\alpha-\gamma)^2 + 4\beta^2} \\ &\le\; \frac{[(\alpha-\gamma)^2 \sin 2\theta - 2\beta\cos 2\theta]^2}{[\sigma_1^2 - \sigma_2^2 - \sqrt{(\alpha-\gamma)^2 + 4\beta^2}]^2}, \end{aligned} \tag{3.19}$$

where (3.19) follows from minimizing the denominator over θ. The following three lemmas allow us to complete the bounding of $\sin^2 2\varphi$.

Lemma 3.5 *$|\alpha - \gamma| < c_1(\Delta, \sigma_1, \sigma_2)$ uniformly in θ, with $c_1(\Delta, \sigma_1, \sigma_2) \to 0$ as $\Delta \to 0$.*

Proof: The series in (3.8) is an alternating series with terms that monotonically decrease in absolute value. Thus it can be bounded (with appropriate sign) by any partial sum [5]. Using simply the first term,

$$-e^{-2\pi^2 (R_{\hat{y}})_{1,1}/\Delta^2} \left(\frac{\Delta^2}{\pi^2} + 4(R_{\hat{y}})_{1,1}\right) < \alpha < 0,$$

and similarly for γ. Finally,

$$|\alpha-\gamma| \;\le\; \max\{|\alpha|,|\gamma|\} \;<\; e^{-2\pi^2\sigma_2^2/\Delta^2}\left(\frac{\Delta^2}{\pi^2}+4\sigma_1^2\right) \;=\; c_1(\Delta,\sigma_1,\sigma_2),$$

since α and γ have the same sign and $\sigma_1<\sigma_2$. □

Lemma 3.6 *$|\beta| \le c_2(\Delta,\sigma_1,\sigma_2)|\sin 2\theta|$ uniformly in θ, with $c_2(\Delta,\sigma_1,\sigma_2)\to 0$ as $\Delta\to 0$.*

Proof: Rearranging (3.9) gives $\beta=\delta(R_{\hat{y}})_{1,2}+\mu$, where the definitions of δ and μ must use $(R_{\hat{y}})_{1,1}$, $(R_{\hat{y}})_{2,2}$, and $(R_{\hat{y}})_{1,2}$ in place of ν_1^2, ν_2^2, and ν_{12}. As in the proof of Lemma 3.5, δ can be bounded by using the first term in each series:

$$|\delta| \;<\; 2\left(e^{-2\pi^2(R_{\hat{y}})_{1,1}/\Delta^2}+e^{-2\pi^2(R_{\hat{y}})_{2,2}/\Delta^2}\right) \;<\; 4e^{-2\pi^2\sigma_2^2/\Delta^2}. \tag{3.20}$$

Assume for the moment that the absolute value of the summand of (3.10) decreases monotonically with both m_1 and m_2. Then computing the double summation (3.10) in either order gives alternating series, so the same bounding technique can be used. We get

$$\begin{aligned}
|\mu| &\le \frac{\Delta^2}{\pi^2}\exp\left(\frac{-2\pi^2((R_{\hat{y}})_{1,1}+(R_{\hat{y}})_{2,2})}{\Delta^2}\right)\sinh\left(\frac{4\pi^2(R_{\hat{y}})_{1,2}}{\Delta^2}\right)\\
&= \frac{\Delta^2}{\pi^2}\exp\left(\frac{-2\pi^2(\sigma_1^2+\sigma_2^2)}{\Delta^2}\right)\sinh\left(\frac{2\pi^2(\sigma_1^2-\sigma_2^2)\sin 2\theta}{\Delta^2}\right)\\
&\le \frac{\Delta^2}{\pi^2}\exp\left(\frac{-2\pi^2(\sigma_1^2+\sigma_2^2)}{\Delta^2}\right)\sinh\left(\frac{2\pi^2(\sigma_1^2-\sigma_2^2)}{\Delta^2}\right)\sin 2\theta\\
&\le \frac{\Delta^2}{2\pi^2}e^{-2\pi^2\sigma_2^2/\Delta^2}\sin 2\theta.
\end{aligned} \tag{3.21}$$

Combining (3.20) and (3.21) gives

$$\begin{aligned}
|\beta| &= |\delta(R_{\hat{y}})_{1,2}+\mu| \;=\; \left|\frac{1}{2}\delta(\sigma_1^2-\sigma_2^2)\sin 2\theta\right|\\
&\le \frac{1}{2}(\sigma_1^2-\sigma_2^2)|\delta||\sin 2\theta|+|\mu|\\
&< \underbrace{\left(2(\sigma_1^2-\sigma_2^2)+\frac{\Delta^2}{2\pi^2}\right)e^{-2\pi^2\sigma_2^2/\Delta^2}}_{c_2(\Delta,\sigma_1,\sigma_2)}|\sin 2\theta|.
\end{aligned}$$

In general, the terms of (3.10) are not monotonically decreasing. However, the terms are monotonically decreasing (in absolute value) outside of $(m_1,m_2)\in\{1,2,\ldots,M\}^2$ for some $M<\infty$. Since each individual term for $1\le m_1,\, m_2\le M$ can be bounded as above, the bound can be extended to the general case. □

Lemma 3.7 *$|\beta|<c_2(\Delta,\sigma_1,\sigma_2)$, uniformly in θ, with $c_2(\Delta,\sigma_1,\sigma_2)\to 0$ as $\Delta\to 0$.*

Proof: This follows immediately from Lemma 3.6. □

By combining Lemmas 3.5 and 3.7, there exists $\Delta_1>0$ such that $\Delta<\Delta_1$ implies $(\alpha-\gamma)^2+4\beta^2\le(\sigma_1^2-\sigma_2^2)^2/4$, uniformly in θ. Thus assuming $\Delta<\Delta_1$ we have

$$\sin^2 2\varphi \le \frac{[(\alpha-\gamma)^2\sin 2\theta-2\beta\cos 2\theta]^2}{\frac{1}{4}(\sigma_1^2-\sigma_2^2)^2}.$$

Applying Lemmas 3.5 and 3.6,

$$\sin^2 2\varphi \leq (c_1 + 2c_2)^2 \sin^2 2\theta,$$

and there exists $\Delta_2 > 0$ such that $\Delta < \Delta_2$ implies $(c_1 + 2c_2)^2 < 1$. The proof is complete with $\Delta_{\max} = \min\{\Delta_1, \Delta_2\}$.

The bounds in this theorem are rather complicated but we can check that the requirements on Δ are reasonable. Suppose $\sigma_1 = 1$ and $\sigma_2 = 1/2$. Then $\Delta_1 > 1.366$ and $\Delta_2 > 1.565$, so the theorem guarantees convergence for any $\Delta < 1.366$. (For this range of Δ, (3.10) can be bounded by the $m_1 = m_2 = 1$ term for any θ.) As we found for Theorem 3.3, numerical calculations (see Figure 3.4) suggest convergence for any Δ.

3.5 Variations on the Basic Algorithms

Certain modifications to the basic algorithms can be made to reduce the computational complexity or to facilitate the coding of non-i.i.d. sources. All of the modifications mentioned in this section apply equally well to the dithered and undithered systems.

The most complicated step in these algorithms is the computation of the updated transform; thus, the complexity can be reduced by suppressing this computation. Instead of computing an eigendecomposition of $\widehat{R_{\hat{x}}}^{(n)}$ at each step, one can compute the eigendecomposition every L steps, holding the transform constant in between. L need not be constant, but if it is to vary it must be computable from coded data. Having constant $L > 1$ does not affect the conclusions in Theorem 3.1.

The coding of a non-i.i.d. source poses many problems. First of all, we must assume that the source is "locally stationary," meaning that $R_{\hat{x}}^{(n)}$ varies slowly. If this is not the case, an on-line algorithm will fail because the coding of x_n is based on an estimate of $R_{\hat{x}}^{(n)}$ from (recent) past samples.[4] Secondly, the correlation matrix estimate $\widehat{R_{\hat{x}}}^{(n)}$ should be local, *e.g.*,

$$\widehat{R_{\hat{x}}}^{(n)} = \frac{1}{K} \sum_{k=n-K+1}^{n} \hat{x}_k \hat{x}_k^T \tag{3.22}$$

or

$$\widehat{R_{\hat{x}}}^{(n)} = \omega \widehat{R_{\hat{x}}}^{(n-1)} + (1-\omega)\hat{x}_n \hat{x}_n^T, \tag{3.23}$$

with appropriate initialization. If the update interval L divides K in (3.22), it is not necessary to store a full window of K past samples [80].

A technique which simultaneously reduces the computational complexity and introduces a correlation estimate equivalent to (3.23) is to replace the eigendecomposition computation with an *incremental* change in the transform based on $\hat{x}_n$. This is explored in Chapter 4.

3.6 Experimental Results

The performance of the undithered system has been tested on synthetic sources and still images. The application to image coding requires some additional design choices. These are specified in Section 3.6.2.

[4] Without local stationarity, a forward adaptive method would presumably be superior. See Chapter 2 and [46].

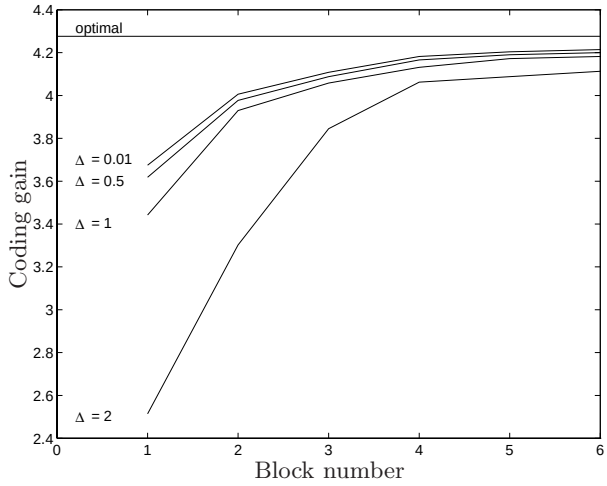


Figure 3.5: Coding gain of undithered universal system compared to optimal coding gain for various coarseness levels.

3.6.1 Synthetic Sources

The undithered system was tested on a source generated by forming blocks of $N = 8$ contiguous samples from a unit-power, first-order autoregressive source with correlation coefficient 0.9. The initial transform is the identity. The transform is updated after every two received vectors based on all of the vectors received thus far. Figure 3.5 gives the coding gain (1.11) after each update for various quantization step sizes. (The optimal coding gain is approximately 4.28.) After a suitable amount of data has become available, close to optimal coding gain is achieved. This experiment shows the relative importance of the number of samples available and the coarseness of the data. A system that uses unquantized data in adaptation (thus requiring side information) would have coding gain approximately as shown by the $\Delta = 0.01$ curve. After a few blocks have been processed, the coding gain of the backward adaptive system is close to that of a system that uses unquantized data in adaptation.

Other experiments with synthetic sources are reported in [82, 228].

3.6.2 Image Coding

To apply the transform adaptation technique to image coding requires first a way to translate the pixels of an image to a "timed" sequence of vectors. This was accomplished by first subdividing the image into 8×8 pixel blocks[5] and then "raster scanning;" *i.e.*, taking the first row of blocks, left to right, then the second row, etc. For a $K \times K$ pixel image, this gives a sequence of $K^2/64$ vectors of length 64. Conceptually, the two-dimensional structure of the image and of each block are maintained since they will be used later.

In block transform coding of images (*e.g.*, in JPEG), separable transforms are typically used. A separable transform can be factored into a processing of the rows and then the columns, or vice versa. For a $B \times B$ pixel block, the transform can be computed as $T_1 X T_2$ with $X \in \mathbb{R}^{B \times B}$ instead of as Tx with $x \in \mathbb{R}^{B^2}$. This reduces the complexity from B^4 multiplications to $2B^3$ multiplications. Aside from complexity, there is

[5] It is assumed that the number of rows and columns in the image are divisible by 8.

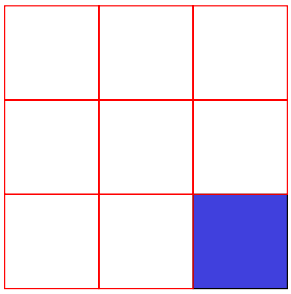


Figure 3.6: The transform used in the coding of the shaded block depends on eight neighboring blocks. Based on a raster scan order, the neighborhood is causal.

an additional advantage to using a separable transform: Two separate B-dimensional unitary transforms have much fewer degrees of freedom than a single B^2-dimensional transform. Thus, if the transform is separable it should require less data to converge to optimal performance or should track a non-stationary source more easily.

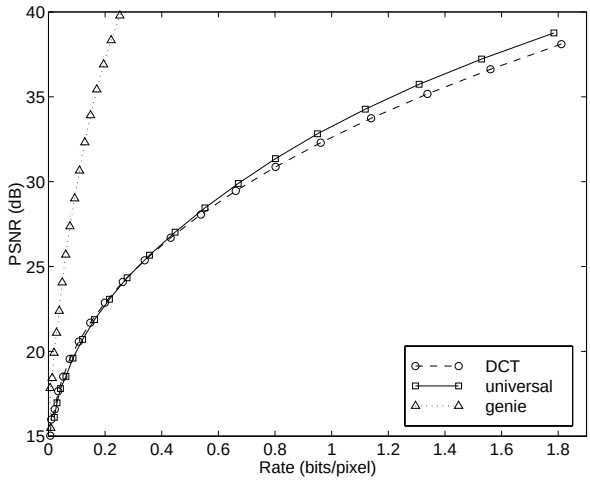
An image coder was implemented with blockwise raster scanning and a separable transform. For the coding of each block, the transform was adapted according to the observed correlation structure from an eight-block neighborhood, as shown in Figure 3.6.[6] Since the raster scan proceeds from top-left to bottom-right, the neighborhood is causal, as required for backward adaptivity. The transform coefficients were quantized with identical uniform scalar quantizers. The column and row correlations could be estimated separately because they were used separately in the row and column transforms. The blocks in the first two rows and columns were transformed with the DCT because they lack a sufficient number of neighbors.

The performance of the adaptive coder was compared to those of a static DCT coder and a hypothetical coder. For each block, the hypothetical coder picks a separable transform based on the statistics of the selfsame block. Such a coder would have to quantize and transmit a description of the transform, but it is assumed here that the transform is conveyed to the decoder by a "genie." The "genie" coder gives an approximate upper bound on the performance possible within the confines of blockwise coding.

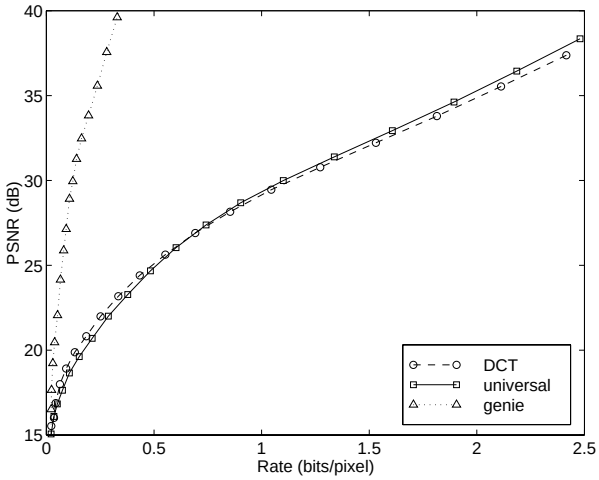
Results for three images are given in Figure 3.7. The first is a 1024×1024 pixel image formed by compositing four uniform texture images from the Brodatz album [20]. The DCT is very effective for uniformly textured regions, but the adaptive coder performs a bit better at moderate rates. This is a somewhat contrived example where information from neighboring blocks is very valuable and the adaptive system is able to converge. The second image is also a sort of composite, but a natural one. It is a page consisting of a photograph and text, scanned from IEEE Spectrum for the purpose of universal image compression experiments [46]. The performance of the adaptive system is slightly better at high rates, but slightly worse at low rates. These two examples suggest that at low rates the information upon which the transform is adapted becomes too coarse to be useful.

The third set of results are for the standard *Lena* image [117]. For this image, the adaptive system performed poorly at all rates. Additional experiments, not directly reported here, indicated that even when the adaptation was based on unquantized data, the performance was poor. The strict causality of the transform adaptation is much more of an impediment than the use of quantized data. *Lena* is simply not a slowly varying stochastic source.

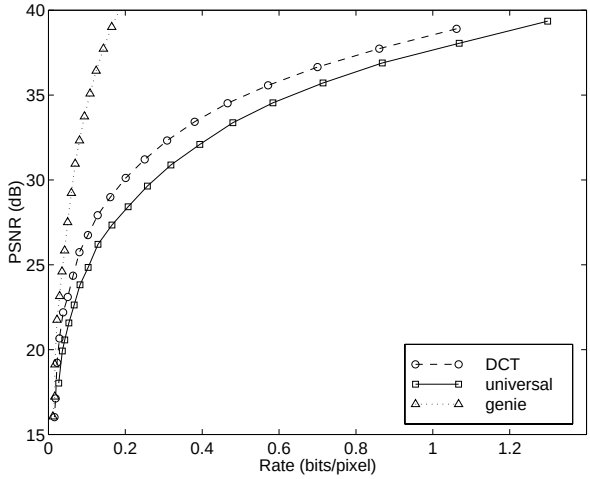
[6] The neighborhood shape is somewhat arbitrary. Experiments with other neighborhoods gave very similar results.



(a) Composite image: four Brodatz textures [20]



(b) Composite image: page scanned from *IEEE Spectrum* [46]



(c) *Lena* image

Figure 3.7: Rate-distortion performance of universal transform coder compared to static DCT coder and to coder employing a "genie" for three grayscale images.

3.7 Conclusions

Two methods for on-line universal transform coding were developed: one using subtractive dither and the other without dither. Both adapt the transform to a strictly causal KLT estimate, derived from quantized data so the decoder can track the encoder state without side information. The method which uses subtractive dither yields to a precise analysis. For coding a stationary Gaussian source, the transform converges in mean square to the optimal transform. The only gap between the performance of the dithered system and a system designed with knowledge of the source distribution is due to the variance increase caused by dithering. This gap can be precisely bounded and vanishes as the quantization step size becomes small. The undithered system does not suffer from this penalty, but is more difficult to analyze; weaker convergence results were shown. Simulations indicate that the undithered system generally converges. The approach taken was purely backward adaptive. The problem of optimally combining forward and backward adaptive modes remains open.

Gaussian sources were assumed throughout the theoretical development. Gaussianity was used in two ways: to justify maximizing coding gain to find the optimal transform and to concretely describe the effect of quantization on moment estimation. Gaussianity is not inherent to either algorithm, *e.g.*, parametric source models were not assumed. Maximizing coding gain is a generally accepted principle even though it is not necessarily optimal for non-Gaussian sources. The convergence, but not necessarily the optimality, of the dithered system thus extends to non-Gaussian sources.

Experiments on synthetic sources consistently confirm the convergence results. However, image coding results were merely middling. On images where neighboring blocks have similarly distributed pixel values, the performance was good. However, the performance on typical photographic images was poor. Such images are simply not well-modeled as slowly varying stochastic sources.

Appendix

3.A Parametric Methods for Adaptation

Consider the coding of a scalar source and suppose that the source can be described by a parametric model. Then, as described by Yu [216], the parameters of the source can in general be consistently estimated from observations of a quantized version of the source as long as the number of quantization cells exceeds the number of parameters. The conditioning of the problem depends on the number and placement of the cells.

A simple example is to observe $X \sim \mathcal{N}(\mu, \sigma^2)$ quantized with three cells: $(-\infty, a)$, $[a, b]$, and (b, ∞). Let $p_1 = P(X \in (-\infty, a))$ and $p_2 = P(X \in [a, b])$. For any $-\infty < a < b < \infty$, the cell probabilities uniquely determine the mean μ and variance σ^2. One can show

$$\left| \frac{\partial(p_1, p_2)}{\partial(\mu, \sigma)} \right| = \frac{b-a}{\pi\sigma^3} \exp\left(- \frac{(a-\mu)^2 + (b-\mu)^2}{\sigma^2} \right),$$

so the conditioning of the problem is poor if a is close to b or if either a or b is far removed from the mean.

A scalar quantized random vector could be treated very similarly, with bins that are Cartesian products of the bins in the scalar case.[7] After finding the parameters describing the source, one can find the moments needed to calculate the KLT.

The approach described above is not entirely satisfactory because it requires the estimation of a large number of bin probabilities.[8] In the case of a Gaussian source and unbounded uniform scalar quantization, the situation is simpler because the parameters of the unquantized signal can be estimated from just the *moments* of the quantized signal as opposed to all of the relative bin probabilities. This is made more precise by the following theorem [82]:

Theorem 3.8 *Let $X = [X_1, X_2, \ldots, X_k]^T$, $X \sim \mathcal{N}(0, \Sigma)$, where Σ is an unknown, non-degenerate covariance matrix. Let $\widehat{X}$ be a scalar quantized version of X such that for $n \in \mathbb{Z}$, either*

(i) $X_i \in [n\Delta_i, (n+1)\Delta_i) \quad \Rightarrow \quad \widehat{X}_i = (n + \frac{1}{2})\Delta_i$; *or*

(ii) $X_i \in [(n - \frac{1}{2})\Delta_i, (n + \frac{1}{2})\Delta_i) \quad \Rightarrow \quad \widehat{X}_i = n\Delta_i$.

Then for any set of positive, finite quantization step sizes $\Delta_1, \ldots, \Delta_k$, all moments of X can be recovered exactly from the first and second order moments of $\widehat{X}$.

The proof is based on finding the mapping between the moments of X and the moments of $\widehat{X}$ (see (3.8)–(3.10) and [27]) and then showing that this mapping is invertible.

The moments of the unquantized signal can be recovered from the moments of the quantized signal. However, this alone does not tell the whole story: the moments of the quantized signal must also be estimated. Since quantization is an irreversible reduction in information, it must be at least as hard to estimate the moments of a signal from a quantized version as it is from the original unquantized signal. This chief disadvantage of a backward adaptive system is quantified below.

[7] It is interesting to note that even very coarse scalar quantization can yield enough information to fit a reasonable parametric model. For example, quantizing with only three bins will yield $3^k - 1$ independent probability estimates, where k is the vector dimension. For any $k \in \mathbb{Z}^+$, $3^k - 1 \geq \frac{1}{2}k^2 + \frac{3}{2}k$, so this quantization is fine enough to fit a multivariate Gaussian signal model.

[8] The number of bins is exponential in the vector dimension.

Let $X_1, X_2, \ldots, X_k$ be an i.i.d. sequence of Gaussian random variables with mean zero and unknown variance σ^2. It is easy to check that the sample variance $s^2 = \frac{1}{k}\sum_{i=1}^{k} X_i^2$ is an unbiased estimator of the variance.[9] The variance of this estimate is given by $E[(s^2 - \sigma^2)^2] = 2\sigma^4/k$.

Now suppose that instead of observing X_1, X_2, ..., X_k, we observe quantized values $\widehat{X}_1$, $\widehat{X}_2$, ..., $\widehat{X}_k$, quantized as in case (ii) of Theorem 3.8. To estimate σ^2, we can first estimate $\hat{\sigma}^2 = E[\widehat{X}_1^2]$ and then invert the mapping which relates σ^2 and $\hat{\sigma}^2$. The quality of the estimate thusly obtained depends on the quality of the estimate of $\hat{\sigma}^2$ (the variance of the sample variance $\hat{s}^2 = \frac{1}{k}\sum_{i=1}^{k} \widehat{X}_i^2$) and the sensitivity of the relationship between σ^2 and $\hat{\sigma}^2$ to errors in $\hat{\sigma}^2$. Using a first order approximation, we obtain

$$\mathrm{Var}(\sigma^2 \text{ estimate}) \approx \frac{\partial[\sigma^2]}{\partial[\hat{\sigma}^2]} \cdot \mathrm{Var}(\hat{s}^2). \tag{3.24}$$

An elementary calculation shows that $\mathrm{Var}(\hat{s}^2) = (E[\widehat{X}_1^4] - E[\widehat{X}_1^2])/k$, where one can obtain expressions for $E[\widehat{X}_1^4]$ and $E[\widehat{X}_1^2]$ by manipulating expressions from [27]. Normalizing the variance of the estimate of σ^2 (approximated through (3.24)) by $2\sigma^4/k$, the variance obtained without quantization) characterizes precisely how much is lost by estimating from quantized data. For example, if the quantization is quite coarse with $\Delta/\sigma = 3$, one needs about twice as much data to estimate σ^2 as well as if the unquantized data were available. A similar analysis can be done for covariance estimates.

3.B Convergence with Independence Assumption

Theorem 3.9 (Convergence with independence assumption) *Let $N = 2$ and assume that for each n, x_n, T_1, T_2, ..., T_{n-1} are independent. With the transforms parameterized as Givens rotations [62] of angles θ_1, θ_2, ..., θ_{n-1}, assume θ_i is uniformly distributed on $[-\pi/4, \pi/4]$. Then T_n converges with probability one to a KLT for the source. With L_n and $L_n^\star$ defined as in Theorem 3.2,*

$$\frac{L_n - L_n^\star}{n} \text{ converges with probability one to } 0.$$

Proof: The independence assumption makes the terms of (3.1) independent. Thus, in light of (3.16) and the law of large numbers, it suffices to show that $(E[\hat{x}_n \hat{x}_n^T])_{1,2} = 0$ and $(E[\hat{x}_n \hat{x}_n^T])_{1,1} - (E[\hat{x}_n \hat{x}_n^T])_{2,2} \neq 0$.[10] The first of these statements will be proven; the second could be proven similarly.

Let $a_n = (\hat{x}_n \hat{x}_n^T)_{1,2}$. By (3.18),

$$E[a_n \mid \theta_n] = \frac{1}{2}(\gamma - \alpha)\sin 2\theta_n + \beta \cos 2\theta_n.$$

Now we take the expectation over θ_n. It is straightforward to verify that α and γ are even functions of θ_n and β is an odd function of θ_n. Thus $E[a_n] = 0$. □

[9] The sum is divided by k because the mean is known; if the mean was unknown and the variance was estimated by summing the squares of the deviations from the *sample* mean, dividing by $k - 1$ would give an unbiased estimator.

[10] This shows that having a uniformly distributed transform has a similar effect as dithering. In practice, the stochastic fluctuations of the transform provide this dithering-like effect. In some sense, the worst case is when the transform does not vary, but this has been covered by Theorem 3.4.

3.C Calculation of $E[A_{ij}^{(k)} A_{ij}^{(\ell)}]$

This appendix presents calculations of $E[A_{ij}^{(k)} A_{ij}^{(\ell)}]$ in order to complete the proof of Theorem 3.1. Recall that $A^{(k)} = \hat{x}_k \hat{x}_k^T$; we wish to show that $A_{ij}^{(k)}$ has variance bounded by a constant and that $A_{ij}^{(k)}$ is uncorrelated with $A_{ij}^{(\ell)}$ when $k \neq \ell$.

Let $e_k = \hat{y}_k - y_k$ and $\tilde{e}_k = T_k e_k$. Then (3.5) can rewritten elementwise as

$$A_{ij}^{(k)} = x_k^i x_k^j + x_k^i \tilde{e}_k^j + \tilde{e}_k^i x_k^j + \tilde{e}_k^i \tilde{e}_k^j,$$

where superscripts on vectors are component indices. A complete, explicit computation of $E[A_{ij}^{(k)} A_{ij}^{(\ell)}]$ is tedious because a full expansion has 16 terms, many of which take on different values depending on whether i equals j or k equals ℓ. A sampling of the calculations is presented, from which the final result can be inferred.

Computation of $E[x_k^i x_k^j x_\ell^i x_\ell^j]$: If $k \neq \ell$, then $x_k^i x_k^j$ is independent of $x_\ell^i x_\ell^j$, so

$$E[x_k^i x_k^j x_\ell^i x_\ell^j] \;=\; E[x_k^i x_k^j] E[x_\ell^i x_\ell^j] \;=\; (R_x)_{ij}^2.$$

When $k = \ell$, $E[x_k^i x_k^j x_\ell^i x_\ell^j] = E[(x_k^i)^2 (x_k^j)^2] = (R_x)_{ii}(R_x)_{jj} + 2(R_x)_{ij}^2$. The final computation, which can be viewed as a mixed moment of a jointly Gaussian vector, can be made easily using a characteristic function [88].

Computation of $E[x_k^i x_k^j x_\ell^i \tilde{e}_\ell^j]$: Let T_k^i denote the ith row of T_k. To compute the expectation of $x_k^i x_k^j x_\ell^i \tilde{e}_\ell^j = x_k^i x_k^j x_\ell^i T_\ell^j e_\ell$, note that e_ℓ is independent of the rest and has zero mean, so the expectation is zero. Similarly, $E[x_k^i x_k^j \tilde{e}_\ell^i x_\ell^j] = 0$.

Computation of $E[x_k^i x_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j]$: Determining $E[x_k^i x_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j]$ would be trivial if it were not for the dependence of T_ℓ on x_k when $\ell > k$. The computation can be simplified by conditioning:

$$E[x_k^i x_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] \;=\; E[E[x_k^i x_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] \mid T_\ell] \;=\; E\Big[x_k^i x_k^j \frac{\Delta^2}{12}\delta_{ij}\Big] \;=\; (R_x)_{ij}\frac{\Delta^2}{12}\delta_{ij}.$$

The second equality follows because the conditioning allows the transform T_ℓ to be treated as constant.

Computation of $E[x_k^i \tilde{e}_k^j x_\ell^i \tilde{e}_\ell^j]$: When $k > \ell$, the expectation of $x_k^i \tilde{e}_k^j x_\ell^i \tilde{e}_\ell^j = x_k^i T_k^j e_k x_\ell^i T_\ell^j e_\ell$ is zero because e_k has zero mean and is independent of the rest. The $k < \ell$ case is similar. When $k = \ell$, x_k^i and $\tilde{e}_k^j$ are independent, so $E[x_k^i \tilde{e}_k^j x_\ell^i \tilde{e}_\ell^j] = E[(x_k^i)^2] E[(\tilde{e}_k^j)^2] = (R_x)_{ii}\Delta^2/12$. A similar calculation shows $E[\tilde{e}_k^i x_k^j \tilde{e}_\ell^i x_\ell^j] = (R_x)_{jj}\Delta^2/12 \cdot \delta_{k\ell}$.

Computation of $E[x_k^i \tilde{e}_k^j \tilde{e}_\ell^i x_\ell^j]$: For $k > \ell$, $E[x_k^i \tilde{e}_k^j \tilde{e}_\ell^i x_\ell^j] = E[x_k^i T_k^j e_k T_\ell^i e_\ell x_\ell^j] = 0$ because e_k has zero mean and is independent of the rest. The $k < \ell$ case is similar. If $k = \ell$, we have $E[x_k^i \tilde{e}_k^j \tilde{e}_\ell^i x_\ell^j] = E[x_k^i x_k^j] E[\tilde{e}_k^i \tilde{e}_k^j] = (R_x)_{ij}\Delta^2/12 \cdot \delta_{ij}$.

Computation of $E[x_k^i \tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j]$: For $k \geq \ell$, x_k^i is independent of $\tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j$, so $E[x_k^i \tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] = E[x_k^i] E[\tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] = 0$. For $k < \ell$, we can again use the trick of conditioning: $E[x_k^i \tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] = E[E[x_k^i \tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] \mid T_k, T_\ell] = 0$. Similarly, $E[\tilde{e}_k^i x_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] = 0$.

Computation of $E[\tilde{e}_k^i \tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j]$: First consider $k = \ell$. Though $\tilde{e}_k^i \tilde{e}_k^j$ and $\tilde{e}_\ell^i \tilde{e}_\ell^j$ are not independent, they are conditionally independent given T_k and T_ℓ, and the conditional expectations are constant. So the desired expectation is given by

$$E[\tilde{e}_k^i \tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j] \;=\; E\left[E[\tilde{e}_k^i \tilde{e}_k^j \tilde{e}_\ell^i \tilde{e}_\ell^j \mid T_k, T_\ell]\right] \;=\; E\left[E[\tilde{e}_k^i \tilde{e}_k^j \mid T_k] E[\tilde{e}_\ell^i \tilde{e}_\ell^j \mid T_\ell]\right] \;=\; \frac{\Delta^2}{12} \cdot \frac{\Delta^2}{12} \delta_{ij}.$$

For the $k = \ell$ case, we need only a constant bound, so by replacing each term with its maximum absolute value, $\sqrt{N}\Delta/2$, we get $E[(\tilde{e}_k^i)^2 (\tilde{e}_k^j)^2] \le N^2 \Delta^4 / 16$.

Combining these calculations for $k = \ell$ gives

$$E[(A_{ij}^{(k)})^2] \le (R_x)_{ii}(R_x)_{jj} + 2(R_x)_{ij}^2 + \frac{\Delta^2}{12}\left(4(R_x)_{ij}\delta_{ij} + (R_x)_{ii} + (R_x)_{jj}\right) + \frac{N^2\Delta^4}{16},$$

so the variance of $A_{ij}^{(k)}$ is bounded by a constant. For $k \neq \ell$,

$$E[A_{ij}^{(k)} A_{ij}^{(\ell)}] \;=\; (R_x)_{ij}^2 + 2\frac{\Delta^2}{12}\delta_{ij}(R_x)_{ij} + \left(\frac{\Delta^2}{12}\right)^2 \delta_{ij} \;=\; \left((R_x)_{ij} + \frac{\Delta^2}{12}\delta_{ij}\right)^2 \;=\; E[A_{ij}^{(k)}] E[A_{ij}^{(\ell)}].$$

Thus $A_{ij}^{(k)}$ and $A_{ij}^{(\ell)}$ are uncorrelated.

Chapter 4

New Methods for Transform Adaptation

IN THE on-line universal transform coding algorithm of Chapter 3, each transform update requires the calculation of a matrix that diagonalizes the measured correlation matrix. This calculation, a determination of all the eigenvectors of the symmetric matrix, is an intensive one. However, since the estimated correlation matrix should vary slowly, the problem is like a typical tracking problem in adaptive signal processing. In this chapter, the analogy between tracking the best transform for transform coding and tracking the best filter for various kinds of filtering problems is used in the development of a new set of algorithms.

For coding an N-dimensional source, these algorithms pose the transform adaptation problem as an unconstrained minimization over $K = N(N-1)/2$ parameters. Performing this minimization through a gradient descent gives an algorithm analogous to LMS. Step size bounds for stability similar in form to those for LMS are proven and linear and fixed-step random search methods are also considered.

4.1 Introduction

Optimal linear transform coding has two striking similarities with optimal finite impulse response (FIR) Wiener filtering: both (often unrealistically) require knowledge of second-order moments of signals; and both require a calculation which is considered expensive if it must be done repeatedly (eigendecomposition and matrix inversion, respectively). In FIR Wiener filtering, it is well-known that these difficulties can be mitigated by adaptation. This chapter establishes new methods in block transform adaptation that are analogous to some of the standard methods in adaptive FIR Wiener filtering.

The basis for many adaptive Wiener filtering methods is to specify independent parameters, define a performance surface with respect to these parameters, and to search the performance surface for the optimal parameter values. The most common method of performance surface search is gradient descent—which leads to the LMS algorithm [204]—but linear and fixed-step random searches [205] also fall into this class. This chapter defines two meaningful performance surfaces (cost functions) for linear transform coding and analyzes various

This chapter includes research conducted jointly with Martin Vetterli [75].

search methods for these surfaces. The result is a set of new algorithms for adaptive linear transform coding.

Subject to a Gaussian condition on the source and fine-quantization approximations,[1] finding an optimal transform for transform coding amounts to finding an orthonormal set of eigenvectors of a symmetric, positive semidefinite matrix; *i.e.*, finding an optimal transform is an instance of the *symmetric eigenproblem*, a fundamental problem of numerical analysis [62]. Thus, in finding a method for transform adaptation we are in fact attempting to approximately solve a sequence of symmetric eigenvalue problems. The idea of using performance surface search (*i.e.*, cost function minimization) for this problem seems to be new, although the cost function which will later be called J_1 has been used in convergence analyses [62]. The algorithms developed here are *not* competitive with cyclic Jacobi methods for computing a *single* eigendecomposition of a large matrix; however, they are potentially useful for computing eigendecompositions of a slowly varying sequence of matrices.

The novelty and potential utility of these algorithms for transform coding comes from the following properties: the transform is always represented by a minimal number of parameters, the autocorrelation matrix of the source need not be explicitly estimated, and the computations are more parallelizable than cyclic Jacobi methods. In addition, further insights may come from drawing together techniques from adaptive filtering, transform coding, and numerical linear algebra.

The reader is referred to [62] for a thorough treatment of the techniques for computing eigendecompositions including the techniques specific to the common special case where the matrix is symmetric. Section 1.1.3 and Appendix 4.A provide brief reviews of transform coding and adaptive FIR Wiener filtering, respectively.

4.2 Problem Definition, Basic Strategy, and Outline

Let $\{x_n\}_{n\in\mathbb{Z}^+}$ be a sequence of $\mathbb{R}^N$-valued random vectors and let $X_n = E[x_n x_n^T]$.[2] We assume that the dependence of X_n on n is mild[3] and desire a procedure which produces a sequence of orthogonal transforms T_n such that $Y_n = T_n X_n T_n^T$ is approximately diagonal for each n. The procedure should be causal, *i.e.*, T_n should depend only on $\{x_k\}_{k=1}^n$. X_n will not be known, but must be estimated or in some sense inferred from $\{x_k\}_{k=1}^n$.

First of all, note that if X_n is known, then a T_n consisting of normalized eigenvectors of X_n solves our problem [99]. A traditional approach would be to construct an estimate $\widehat{X}_n = f(\{x_k\}_{k=1}^n)$ for each n, and then use an "off the shelf" method to compute the eigenvectors of $\widehat{X}_n$. The difficulty with this is that the eigenvector computation may be deemed too complex to be done for each n.

In analogy to the way the LMS algorithm avoids explicitly solving a linear system of equations (see Appendix 4.A), one can avoid using an explicit eigendecomposition algorithm. The first conceptual step is to replace the problem of finding a diagonalizing transform T_n for X_n with a minimization problem for which a diagonalizing transform achieves the minimum. The next step is to derive a gradient descent iteration for the minimization problem. Note that in these two steps it is assumed that X_n is known. The final step is

[1] Without these technical conditions, there is no general principle for determining the optimal transform, so in the remainder of the chapter "optimal" is used without qualification. For more details see Section 1.1.3 and [60].

[2] The use of X in place of R_x, though inconsistent with other chapters, was chosen to reduce the need for double subscripts.

[3] If the dependence of X_n on n is not mild, then it is rather hopeless to use adaptation in the traditional sense of learning source behavior based on the recent past. Better strategies might include classification [46, 45], matching pursuit (see Chapter 2), or other basis selection methods [1].

to apply the gradient descent iteration with X_n replaced by a stochastic approximation $\widehat{X}_n$. The following three sections address these three steps. In Section 4.3, two cost functions are defined which are minimized by a diagonalizing transform. Section 4.4 gives derivations for gradient descents with respect to the two cost functions along with step size bounds which ensure local convergence. Linear and fixed-step random searches are also discussed. Section 4.4 contains the linear algebraic computations which underlie the signal processing algorithms which are ultimately presented in Section 4.5. It is in this final section that stochastic simulations show the applicability to adaptive transform coding.

4.3 Performance Criteria

If two orthogonal transforms only approximately diagonalize X, which of the two is better? In order to use a performance surface search to iteratively find optimal transforms, we need a continuous measure of the diagonalizing performance of a transform. The remainder of the chapter uses two such performance measures.

The most obvious choice for a cost function is the squared norm of the off-diagonal elements of $Y = TXT^T$:

$$J_1(T) = \sum_{i \neq j} Y_{ij}^2. \tag{4.1}$$

This cost function is clearly nonnegative and continuous in each component of T. Also, $J_1(T) = 0$ if and only if T exactly diagonalizes X.

The cost function

$$J_2(T) = \prod_{i=1}^{N} Y_{ii} \tag{4.2}$$

is intimately connected to transform coding theory but is less obviously connected to the diagonalization of X. Under the standard assumptions of transform coding, for a fixed rate, $\sqrt[N]{J_2(T)}$ is proportional to the distortion (see Section 1.1.3.1). Thus minimizing J_2 minimizes the distortion and $J_2(T)$ is minimized by a transform that diagonalizes X. A potential disadvantage of this cost function is that the minimum value is not zero; instead it is $\prod_i \lambda_i$, where λ_i's are the eigenvalues of X.

4.4 Methods for Performance Surface Search

Section 4.4.3 presents two new eigendecomposition algorithms based on gradient descent with respect to the cost functions J_1 and J_2. These algorithms and the random search algorithm of Section 4.4.2 are inspired by and parallel the standard methods in adaptive FIR Wiener filtering [205]. For comparison, standard methods which are computationally attractive for computing single eigendecompositions are presented in Section 4.4.4.

The effects of the time variation of X and estimation noise are left for subsequent sections. Hence, throughout this section we dispense with time indices and consider iterative methods for diagonalizing a fixed matrix X.

4.4.1 Parameterization of Transform Matrices

An $N \times N$ orthogonal matrix has fewer than N^2 independent parameters because of the requirement that the columns (or equivalently the rows) form an orthonormal set. In our search for the best orthogonal transform it will sometimes be useful to represent the matrix in terms of the smallest possible number of parameters.

To determine the number of degrees of freedom in the parameterization of an orthogonal matrix, imagine that one is constructing such a matrix column-by-column. Making the ith column orthogonal to the earlier columns leaves $N-i+1$ degrees of freedom and normalizing gives $N-i$ degrees of freedom plus a choice of sign. Thus overall there are $N(N-1)/2$ degrees of freedom plus N sign choices. The sign choices have no effect on $J_1(T)$ or $J_2(T)$, so we are left with $K = N(N-1)/2$ degrees of freedom. $K = \binom{N}{2}$ matches the number of distinct Givens rotations, and we will see in Lemma 4.1 below that the parameters of interest can be taken to be the angles of Givens rotations.

Definition 4.1 *A matrix of the form*

$$\widetilde{G}_{i,j,\theta} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos\theta & \cdots & \sin\theta & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin\theta & \cdots & \cos\theta & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{array} \\ \begin{array}{ccccccc} & & i & & j & & \end{array} \end{array} \begin{array}{l} \\ \\ i \\ \\ j \\ \\ \\ \end{array} , \tag{4.3}$$

where $-\pi/2 < \theta \leq \pi/2$, is called a Givens (or Jacobi) rotation *[62]. It can be interpreted as a counterclockwise rotation of θ radians in the (i,j) coordinate plane.*

Since we will be interested in Givens rotations with $i < j$, it will be convenient to use the index remapping $G_{k,\theta} = \widetilde{G}_{i,j,\theta}$, where (i,j) is the kth entry of a lexicographical list of $(i,j) \in \{1,2,\ldots,N\}^2$ pairs with $i < j$. For example, the matrix below gives the corresponding value of k in the (i,j) location for $N = 4$:

$$\begin{bmatrix} \star & 1 & 2 & 3 \\ \star & \star & 4 & 5 \\ \star & \star & \star & 6 \\ \star & \star & \star & \star \end{bmatrix}$$

Lemma 4.1 *Let $X \in \mathbb{R}^{N\times N}$ be a symmetric and let $K = N(N-1)/2$. Then there exists $\Theta = [\theta_1, \theta_2, \ldots, \theta_K]^T \in [-\pi/2, \pi/2)^K$ such that $T_\Theta X T_\Theta^T$ is diagonal, where*

$$T_\Theta = G_{1,\theta_1} G_{2,\theta_2} \ldots G_{K,\theta_K}. \tag{4.4}$$

Proof: Since X is symmetric, there exists an orthogonal matrix S such that SXS^T is diagonal [99]. Any orthogonal matrix can be factored as

$$S = (\widetilde{G}_{1,2,\theta_{1,2}} \widetilde{G}_{1,3,\theta_{1,3}} \cdots \widetilde{G}_{1,N,\theta_{1,N}})(\widetilde{G}_{2,3,\theta_{2,3}} \cdots \widetilde{G}_{2,N,\theta_{2,N}}) \cdots (\widetilde{G}_{N-1,N,\theta_{N-1,N}}) D_\epsilon,$$

where $D_\epsilon = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_N)$, $\epsilon_i = \pm 1$, $i = 1, 2, \ldots, N$ *[4]. It is now obvious that taking* $T = SD_\epsilon^{-1}$ *suffices because* $D_\epsilon^{-1} X (D_\epsilon^{-1})^T = X$. □

4.4.2 Random Search

In light of Lemma 4.1 and the discussion of Section 4.3, finding a diagonalizing transform amounts to minimizing J_1 or J_2 (written as J where either fits equally) over $\Theta \in [-\pi/2, \pi/2)^K$. Conceptually, the simplest way to minimize a function—so simple and naive that it is often excluded from consideration—is to guess.

Discretizing the range of interest of Θ, evaluating J at each point on the grid, and taking the minimum of these gives an approximation to the minimum. The accuracy of this approximation will depend on the smoothness of J and the density of the grid. The grid could also be made adaptive to have higher density of points where J is smaller. This exhaustive deterministic approach is not well suited to our application with a slowly-varying sequence of X matrices because information from previous iterations is not easily incorporated. Instead, two approaches that yield random sequences of parameter vectors with expected drift toward the optimum are presented.

In a *fixed-step random search*, a small random change is tentatively added to the parameter vector. The change is adopted if it decreases the objective function; else, it is discarded. Formally, the update is described by

$$\Theta_{k+1} = \begin{cases} \Theta_k + \alpha\eta_k & \text{if } J(\Theta_k + \alpha\eta_k) < J(\Theta_k), \\ \Theta_k & \text{otherwise,} \end{cases}$$

where $\alpha \in \mathbb{R}^+$ and $E[\eta_k \eta_k^T] = I$.

A fixed-step random search makes no progress on an iteration where $\Theta_k + \alpha\eta_k$ is found to be worse than Θ_k. Another possibility is a *linear random search* [205]. In this case, instead of taking no step if η_k seems to be a step in the wrong direction, one takes a step in the opposite direction; the size of each step is proportional to the increase or decrease in J. The update is described by

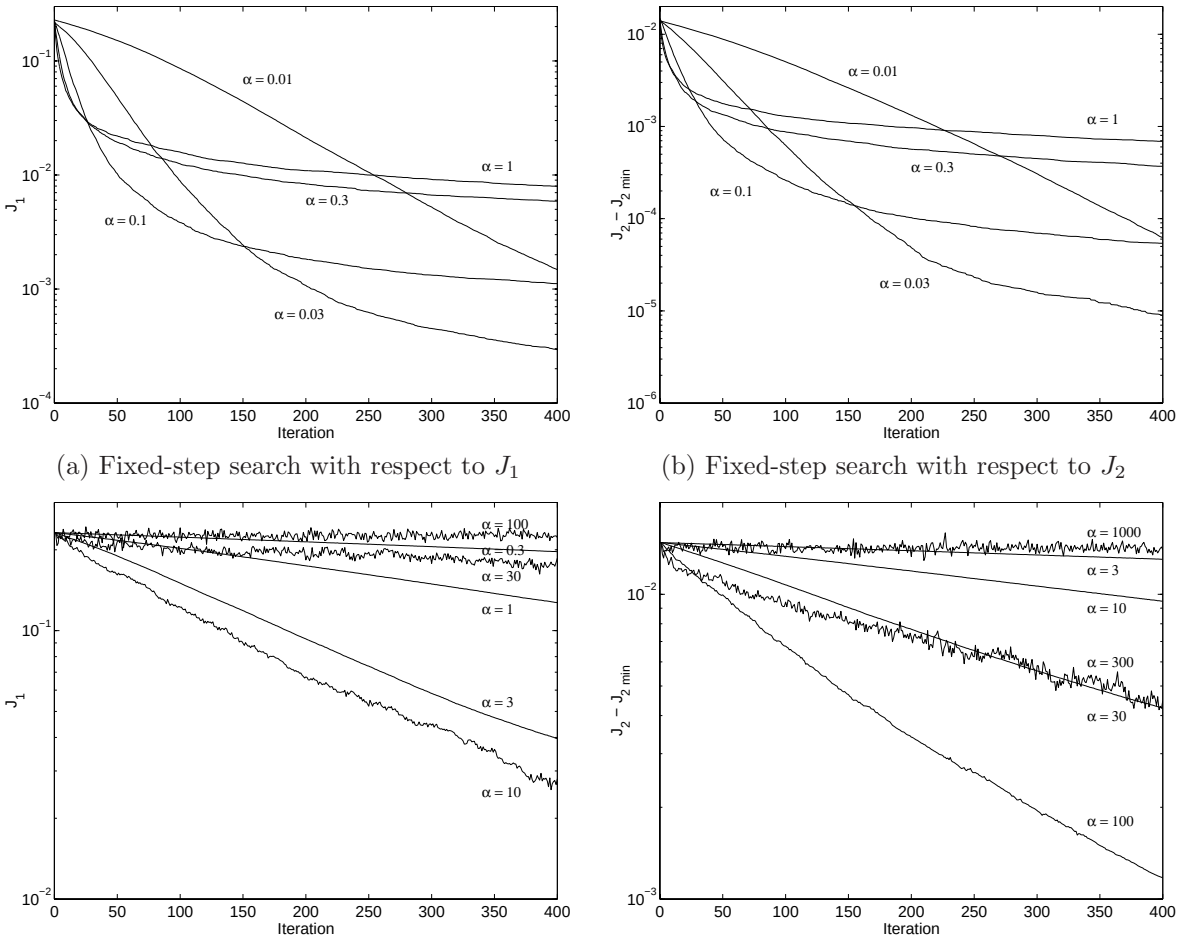
$$\Theta_{k+1} = \Theta_k + \alpha[J(\Theta_k) - J(\Theta_k + \sigma\eta_k)]\eta_k,$$

where $\alpha, \sigma \in \mathbb{R}^+$ and $E[\eta_k \eta_k^T] = I$.

It is intuitively clear that, using either cost function, for sufficiently small α both random search algorithms tend to drift toward local minima of J. The fixed-step and linear random search algorithms were simulated on the problem $X = \mathrm{diag}([1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}])$ with initial guess Θ_0 chosen randomly according to a uniform distribution on $[-\pi/2, \pi/2]^6$. Figure 4.1 gives the averaged results of 400 simulations of 400 iterations each for various values of α. Gaussian η was used for the fixed-step search; for the linear search, η is uniformly distributed on the unit sphere and $\sigma = 0.01$.

As shown in Figure 4.1(a)–(b), the fixed-step searches have the undesirable quality that the best choice of α depends on the number of iterations: for a small number of iterations a large α is preferred while for a large number of iterations the opposite is true. A simple interpretation of this is that for large α the first few steps are more beneficial, but as the optimum Θ is approached, tentative steps are very unlikely to be accepted; close to the optimum Θ, small α is more likely to yield improvements.

While the fixed-step algorithm tends to get stuck when α is large, the performance of the linear search algorithm degrades in a different way. When α is large, many steps are taken which increase J; hence the



(a) Fixed-step search with respect to J_1

(b) Fixed-step search with respect to J_2

(c) Linear search with respect to J_1

(d) Linear search with respect to J_2

Figure 4.1: Simulations of the random search algorithms. $X = \text{diag}([1,\ \frac{1}{2},\ \frac{1}{4},\ \frac{1}{8}])$ and results are averaged over 400 randomly chosen initial conditions Θ_0.

convergence gets more erratic. For very large α there is no negative drift in J. This is shown in Figure 4.1(c)–(d).

The conceptual simplicity of random search algorithms comes from utilizing no knowledge of the function to be minimized. Using gradient descent is one way to utilize knowledge of the function to be minimized. This is discussed in the following section.

4.4.3 Descent Methods

This section explores gradient descent based methods for minimizing J_1 or J_2. The idea of a gradient descent is very simple. Suppose we wish to find Θ which minimizes a function $J(\Theta)$ and we have an initial guess Θ_0. Assuming a first-order approximation of J, changing Θ_0 in the direction of $\nabla J|_{\Theta=\Theta_0}$ produces the maximum increase in J, so taking a step in the opposite direction produces the maximum decrease in J. This leads to the general update formula for gradient descent:

$$\Theta_{k+1} = \Theta_k - \alpha \, \nabla J|_{\Theta=\Theta_k}, \tag{4.5}$$

where $\alpha \in \mathbb{R}^+$ is the step size. We now compute the gradient and the bounds on α for stability for each of the cost function of Section 4.3.

4.4.3.1 Minimization of J_1

Start by computing ∇J_1 elementwise. Firstly,

$$\frac{\partial J_1}{\partial \theta_k} = \sum_{i \neq j} \frac{\partial}{\partial \theta_k} Y_{ij}^2 = \sum_{i \neq j} 2 Y_{ij} \frac{\partial Y_{ij}}{\partial \theta_k}. \tag{4.6}$$

For notational convenience, let $U_{(a,b)} = G_{a,\theta_a} G_{a+1,\theta_{a+1}} \dots G_{b,\theta_b}$, where $U_{(a,b)} = I$ if $b < a$. Also denote $U_{(k,k)}$ by U_k and let $V_k = \frac{\partial}{\partial \theta} U_{k,\theta_k}$. Define $A^{(k)}$, $1 \leq k \leq K$, elementwise by $A_{ij}^{(k)} = \partial Y_{ij} / \partial \theta_k$. Then to evaluate $\partial Y_{ij} / \partial \theta_k$, write $Y = TXT^T$ and use (4.4) to yield

$$A^{(k)} = U_{(1,k-1)} V_k U_{(k+1,K)} X U_{(1,K)}^T + U_{(1,K)} X U_{(k+1,K)}^T V_k^T U_{(1,k-1)}^T. \tag{4.7}$$

Combining (4.6) and (4.7),

$$\frac{\partial J_1}{\partial \theta_k} = 2 \sum_{i \neq j} Y_{ij} A_{ij}^{(k)}. \tag{4.8}$$

Theorem 4.2 *Denote the eigenvalues of X by $\lambda_1, \lambda_2, \dots, \lambda_N$ and let $\Theta_\star$ correspond to a diagonalizing transform for X. Then for Θ_0 sufficiently close to $\Theta_\star$ the gradient descent algorithm described by (4.5) and (4.8) converges to $\Theta_\star$ if*

$$0 < \alpha < \left[2 \max_{i,j} (\lambda_i - \lambda_j)^2 \right]^{-1}.$$

Proof: Without loss of generality, we can assume that $X = \mathrm{diag}([\lambda_1 \ \lambda_2 \ \cdots \lambda_N])$. This amounts to selecting the coordinates such that $\Theta_\star = \mathbf{0}$.

The key to the proof is observing that (4.5) describes an autonomous, nonlinear, discrete-time dynamical system and linearizing the system. Write

$$\Theta_{k+1} = \Theta_k - \alpha f(\Theta_k),$$

which upon linearization about $\mathbf{0}$ *gives*

$$\widehat{\Theta}_{k+1} = (I - \alpha F)\widehat{\Theta}_k,$$

where $F_{ij} = \left[\frac{\partial}{\partial\theta_j} f_i(\Theta)\right]_{\Theta=\mathbf{0}}$. *A sufficient condition for local convergence is that the eigenvalues of* $I - \alpha F$ *lie in the unit circle. The fact that the local exponential stability of the original nonlinear system can be inferred from an eigenvalue condition on the linearized system follows from the continuous differentiability of* f *[196].*

Now evaluate F. *Differentiating (4.8) gives*

$$\begin{aligned}\frac{\partial^2 J_1}{\partial\theta_\ell\,\partial\theta_k} &= 2\sum_{i\neq j}\left(Y_{ij}\frac{\partial A_{ij}^{(k)}}{\partial\theta_\ell} + A_{ij}^{(k)}\frac{\partial Y_{ij}}{\partial\theta_\ell}\right)\\ &= 2\sum_{i\neq j}\left(Y_{ij}\frac{\partial A_{ij}^{(k)}}{\partial\theta_\ell} + A_{ij}^{(k)}A_{ij}^{(\ell)}\right).\end{aligned} \tag{4.9}$$

Evaluating (4.9) at $\Theta = \mathbf{0}$, Y *becomes* X *(diagonal), so the first term makes no contribution; we need not attempt to calculate* $\partial A_{ij}^{(k)}/\partial\theta_\ell$. *By inspection of (4.7),* $A^{(k)}$ *becomes* $V_k X + X V_k^T$. *This simplifies further to a matrix which is all zeros except for having* $\lambda_{j_k} - \lambda_{i_k}$ *in the* (i_k, j_k) *and* (j_k, i_k) *positions, where* (i_k, j_k) *is the* (i, j) *pair corresponding to* k *in the index remapping discussed following Definition 4.1. Noting now that* $A^{(k)}$ *and* $A^{(\ell)}$ *have nonzero entries in the same positions only if* $k = \ell$, *we are prepared to conclude that*

$$F_{k\ell} = \left[\frac{\partial^2 J_1}{\partial\theta_\ell\,\partial\theta_k}\right]_{\Theta=\mathbf{0}} = \begin{cases} 4(\lambda_{i_k} - \lambda_{j_k})^2 & \text{if } k = \ell,\\ 0 & \text{otherwise.}\end{cases} \tag{4.10}$$

The eigenvalues of $I - \alpha F$ *are* $1 - 4\alpha(\lambda_{i_k} - \lambda_{j_k})^2$. *The proof is completed by requiring that these all lie in the unit circle.* □

The nonlinear nature of the iteration makes analysis very difficult without linearization. In the $N = 2$ case, the iteration can be analyzed directly; a stronger result is thus obtained. Similar, stronger results may be true for larger N.

Theorem 4.3 *In the case* $N = 2$, *the result of Theorem 4.2 can be strengthened to an "almost global" exponential stability result,* i.e., *from any initial condition except a maximum, the iteration will converge exponentially to the desired minimum of* J_1.

Proof: Without loss of generality, assume $X = \mathrm{diag}([\lambda_1\ \lambda_2])$. *First notice that when* $N = 2$, *the set of transforms under consideration are described by a single scalar parameter, or* $K = 1$. *Dropping all unnecessary subscripts, (4.7) reduces to*

$$A = VXU^T + UXV^T = \begin{bmatrix} (\lambda_2 - \lambda_1)\sin 2\theta & (\lambda_2 - \lambda_1)\cos 2\theta \\ (\lambda_2 - \lambda_1)\cos 2\theta & -(\lambda_2 - \lambda_1)\sin 2\theta \end{bmatrix},$$

and

$$Y = TXT^T = \begin{bmatrix} \lambda_1\cos^2\theta + \lambda_2\sin^2\theta & \frac{1}{2}(\lambda_2 - \lambda_1)\sin 2\theta \\ \frac{1}{2}(\lambda_2 - \lambda_1)\sin 2\theta & \lambda_1\sin^2\theta + \lambda_2\cos^2\theta \end{bmatrix}.$$

Simplifying (4.8) gives

$$\frac{\partial J_1}{\partial\theta} = 2(Y_{12}A_{12} + Y_{21}A_{21}) = (\lambda_2 - \lambda_1)^2\sin 4\theta.$$

Thus the iteration to analyze is

$$\theta_{k+1} = \theta_k - \alpha(\lambda_2 - \lambda_1)^2 \sin 4\theta_k. \tag{4.11}$$

It is immediately clear that all multiples of $\pi/4$ are fixed points; the even multiples correspond to the desired transforms and the odd multiples are the only initial conditions for which the iteration does not converge to a diagonalizing transform. For convenience, we can consider only $0 < |\theta_0| < \pi/4$; other cases are similar. We would like to show that $\lim_{k\to\infty} \theta_k = 0$. Suppose $0 < \theta_0 < \pi/4$. Then using $\sin 4\theta \le 4\theta$ and $\alpha < (\lambda_2 - \lambda_1)^{-2}/2$ one can show that $\alpha(\lambda_2 - \lambda_1)^2 \sin 4\theta_0 \in (0, 2\theta_0)$. Thus $|\theta_1| < |\theta_0|$. The $-\pi/4 < \theta_0 < 0$ case is similar. Since (4.11) is a strictly contractive mapping on $(-\pi/4, \pi/4)$ the iteration must converge to the only fixed point in the interval, zero. □

4.4.3.2 Minimization of J_2

We will continue to use the notation introduced in Section 4.4.3.2. ∇J_2 is given elementwise by

$$\begin{aligned} \frac{\partial J_2}{\partial \theta_k} &= \frac{\partial}{\partial \theta_k} \prod_{i=1}^{N} Y_{ii} = \sum_{m=1}^{N} \left(\frac{\partial Y_{mm}}{\partial \theta_k} \prod_{i=1,\, i\neq m}^{N} Y_{ii} \right) = \sum_{m=1}^{N} \left(A_{mm}^{(k)} \prod_{i=1,\, i\neq m}^{N} Y_{ii} \right) \\ &= J_2(T) \sum_{m=1}^{N} \left(\frac{1}{Y_{mm}} A_{mm}^{(k)} \right), \end{aligned} \tag{4.12}$$

where $A^{(k)}$ was defined in (4.7). As before, the gradient descent update is specified by (4.5).

Theorem 4.4 *Denote the eigenvalues of X by $\lambda_1, \lambda_2, \ldots, \lambda_N$ and let $\Theta_\star$ correspond to a diagonalizing transform for X. Then for Θ_0 sufficiently close to $\Theta_\star$ the gradient descent algorithm described by (4.5) and (4.12) converges to $\Theta_\star$ if*

$$0 < \alpha < \left[J_{\min} \max_{i,j} \frac{(\lambda_i - \lambda_j)^2}{\lambda_i \lambda_j} \right]^{-1},$$

where $J_{\min} = \prod_{i=1}^{N} \lambda_i$.

Proof: The method of proof is again to linearize the autonomous, nonlinear, discrete-time dynamical system that we have implicitly defined, and again the analysis is simplified by assuming that $X = \mathrm{diag}([\lambda_1\ \lambda_2\ \cdots \lambda_N])$.

Differentiating (4.12) gives

$$\begin{aligned} \frac{\partial^2 J_2}{\partial \theta_\ell\, \partial \theta_k} &= \frac{\partial}{\partial \theta_\ell} \sum_{m=1}^{N} \left(A_{mm}^{(k)} \prod_{i=1,\, i\neq m}^{N} Y_{ii} \right) \\ &= \sum_{m=1}^{N} \left[\left(\frac{\partial A_{mm}^{(k)}}{\partial \theta_\ell} \prod_{i=1,\, i\neq m}^{N} Y_{ii} \right) + \left(A_{mm}^{(k)} \frac{\partial}{\partial \theta_\ell} \prod_{i=1,\, i\neq m}^{N} Y_{ii} \right) \right]. \end{aligned} \tag{4.13}$$

When (4.13) is evaluated at $\Theta = \mathbf{0}$, the second term does not contribute because the diagonal of $A^{(k)}$ is zero for all k. Evaluation of $\partial A_{mm}^{(k)}/\partial \theta_\ell$ is somewhat tedious and is left for Appendix 4.C. The result is summarized as

$$\frac{\partial A_{mm}^{(k)}}{\partial \theta_\ell} = \begin{cases} 2(\lambda_{j_k} - \lambda_{i_k}) & \text{if } k = \ell \text{ and } m = i_k; \\ 2(\lambda_{i_k} - \lambda_{j_k}) & \text{if } k = \ell \text{ and } m = j_k; \\ 0 & \text{otherwise}; \end{cases} \tag{4.14}$$

where (i_k, j_k) is related to k as before. Combining (4.13) and (4.14) gives

$$F_{k\ell} = \left[\frac{\partial^2 J_2}{\partial\theta_\ell\,\partial\theta_k}\right]_{\Theta=\mathbf{0}} = \begin{cases} \frac{2J_{\min}(\lambda_{i_k}-\lambda_{j_k})^2}{\lambda_{i_k}\lambda_{j_k}} & \text{if } k=\ell, \\ 0 & \text{otherwise.} \end{cases}$$

Requiring the eigenvalues of $I-\alpha F$ to lie in the unit circle completes the proof. □

In the $N = 2$ case (but *not* in general) the two gradient descent algorithms are equivalent. Hence Theorem 4.3 applies to the descent with respect to J_2 also. Again, the convergence result of Theorem 4.4 might be strengthened for general N, but the analysis seems difficult.

4.4.3.3 Comparison of descent methods

The linearizations used in the proofs of Theorems 4.2 and 4.4 facilitate easy analyses of the rates of convergence of the two descent methods. Consider the descent with respect to J_1. Using (4.10), the error in the kth component of Θ at the nth iteration is approximately $c[1-4\alpha(\lambda_{i_k}-\lambda_{j_k})^2]^n$. If we assume for the moment that we know $(\lambda_{i_k}-\lambda_{j_k})^2$, we could choose α to make the bracketed quantity equal to zero; then modulo the linearization, the convergence is in one step. The problem is that even if we could do this, the other components of Θ might not converge quickly or converge at all. Thus a quantity of fundamental interest in using the descent with respect to J_1 is the variability of $(\lambda_{i_k}-\lambda_{j_k})^2$, measured by the *pseudo-eigenvalue spread* (designated as such since it is analogous to the eigenvalue spread in LMS adaptive filtering [30]):

$$s_1(X) = \frac{\max_{i,j}(\lambda_i-\lambda_j)^2}{\min_{i,j}(\lambda_i-\lambda_j)^2}.$$

The corresponding quantity for the descent with respect to J_2 is

$$s_2(X) = \frac{\max_{i,j}\frac{(\lambda_i-\lambda_j)^2}{\lambda_i\lambda_j}}{\min_{i,j}\frac{(\lambda_i-\lambda_j)^2}{\lambda_i\lambda_j}}.$$

The difference between $s_1(X)$ and $s_2(X)$ suggests that the superior algorithm will depend on X along with the choice of α. This is confirmed through the following calculations and simulations. Consider the matrices $X_1 = \mathrm{diag}([1,\ \frac{7}{8},\ \frac{5}{8},\ \frac{1}{2}])$ and $X_2 = \mathrm{diag}([1,\ \frac{1}{2},\ \frac{1}{4},\ \frac{1}{8}])$, for which

$$s_1(X_1) \;=\; 16 \;<\; 28 \;=\; s_2(X_1)$$

and

$$s_1(X_2) \;=\; 49 \;>\; \frac{49}{4} \;=\; s_2(X_2).$$

Based on the pseudo-eigenvalue spreads, one expects a descent with respect to J_1 to perform better than a descent with respect to J_2 for diagonalizing X_1, and vice versa for X_2. Simulations were performed with α at half the maximum value for stability and 100 randomly selected initial conditions Θ_0. The averaged results, shown in Figure 4.2, indicate that the performance is as predicted.

4.4.4 Nonparametric Methods

As noted above, finding the optimal transform is equivalent to finding an eigendecomposition of a symmetric matrix. The best algorithms (rated in terms of the number of floating point operations) for the

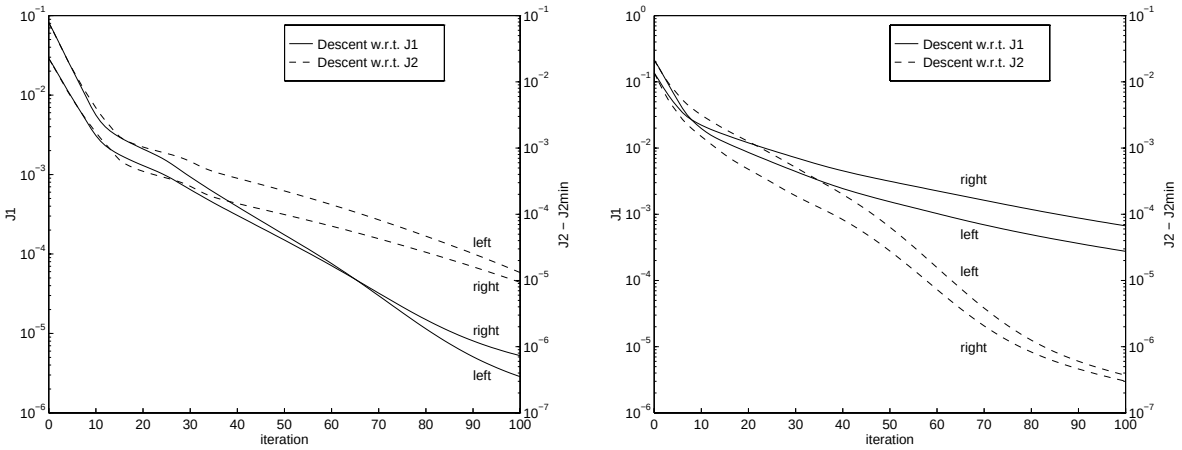


Figure 4.2: Simulations of the gradient descent algorithms. In each case α is set to half the maximum value for stability and results are averaged over 100 randomly chosen initial conditions Θ_0. The relative performances of descents with respect to J_1 and J_2 are as predicted by the pseudo-eigenvalue spreads. Parts (a) and (b) are for matrices X_1 and X_2, respectively. (The curve labels refer to the left and right y-axes.)

symmetric eigenproblem do not use a parameterization of a diagonalizing transform as in the preceding sections. The best algorithms to date for computing the eigendecomposition of a single symmetric matrix are variations of the QR algorithm. However, these algorithms do not allow one to take advantage of knowledge of approximate eigenvectors, as one would have with a slowly-varying sequence of X matrices. This section briefly outlines Jacobi methods, which allow this prior information to be effectively incorporated. Details on QR and Jacobi algorithms can be found in [62].

The idea of the classical Jacobi algorithm is to at each iteration choose a Givens rotation to reduce the off-diagonal energy as much as possible. More specifically, the algorithm produces a sequence $\{T_k\}$ and also keeps track of $A_k = T_k X T_k^T$. If the Givens rotation $\widetilde{G}_{i,j,\theta}$ (see (4.3)) is chosen in computing T_{k+1}, the maximum reduction in the off-diagonal energy (by correctly choosing θ) is $(A_k)_{ij}^2$; thus, the best choice for (i,j) is that which maximizes $(A_k)_{ij}^2$. It is a greedy minimization of J_1, but since Givens rotations do not commute, it is hard to interpret it in terms of the parameterization we used earlier.

A drawback of the classical Jacobi algorithm is that while each iteration requires only $O(N)$ operations for the updates to T_k and A_k, choosing (i,j) requires $O(N^2)$ operations. This can be remedied by eliminating the search step and instead choosing (i,j) in a predetermined manner. This is called the cyclic Jacobi algorithm and each cycle through the $K = N(N-1)/2$ distinct (i,j) pairs is called a *sweep*.

To provide a basis of comparison with the results of Sections 4.4.2 and 4.4.3, simulations of the cyclic Jacobi algorithm were performed on $X = \mathrm{diag}([1,\ \frac{1}{2},\ \frac{1}{4},\ \frac{1}{8}])$ with random initial transforms corresponding to the random initial parameter vectors used before. The averaged results of 400 simulations are shown in Figure 4.3. Note that the x-axis shows the number of rotations, not the number of sweeps.

An attractive feature of the cyclic Jacobi algorithm is that the updates can be partitioned into sets of "noninteracting" rotations, *i.e.*, rotations involving disjoint sets of rows and columns. These noninteracting rotations can be done in parallel. All Jacobi algorithms have the advantage that a good initial transform speeds

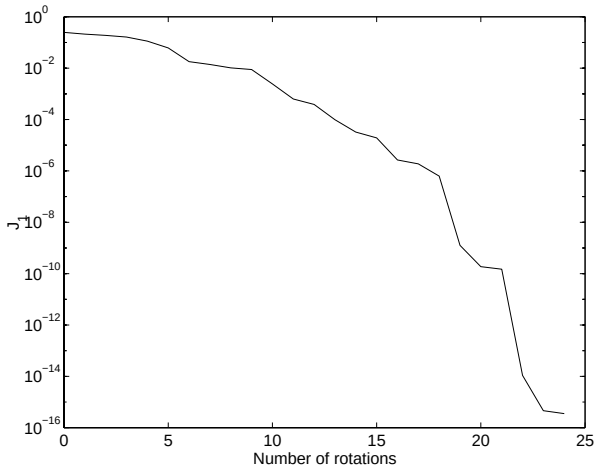


Figure 4.3: Simulations of the cyclic Jacobi algorithm on $X = \text{diag}([1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}])$ with randomly chosen initial transform T_0.

up convergence.

4.4.5 Comments and Comparisons

Comments on the relative merits of random search, gradient descent, and Jacobi methods are in order. By comparing Figures 4.1–4.3, it is clear that the cyclic Jacobi method gives the fastest convergence rate in terms of the number of iterations or rotations. Since the Jacobi method also has the lowest complexity, it is clearly the best choice for computing a single eigendecomposition. Interest in the random search and gradient search methods is due to the analogy to adaptive filtering that is developed more fully in the following section.

A potential benefit of the random search and gradient descent methods is that they operate directly on a minimal parameterization of the transform matrices of interest. Of course, a transform matrix could be determined using a Jacobi method and then parameterized afterward.

4.5 Adaptive Transform Coding Update Methods

In the previous section a set of algorithms for iteratively determining the optimal transform was established, assuming that the source correlation matrix X_n is constant. Recalling our overall strategy, we would now like to apply these algorithms in an adaptive setting.

The traditional implementation approach would be to calculate a sequence of estimates $\{\widehat{X}_n\}$ using a windowed time average and to use these averages in the adaptive algorithms. The extreme case of this approach is to use a time average over only one sample, *i.e.*, $\widehat{X}_n = x_n x_n^T$. This results in a computational savings and—in the case of gradient descent parameter search—gives an algorithm very much in the spirit of LMS. Specifically, in a transform coding application, it may be desirable to eliminate the need for side information by putting quantization inside the adaptation loop, as in Chapter 3. These implementation possibilities are described in detail in the remainder of this section.

In the interest of brevity, simulation results are not provided for each combination of cost function, implementation structure, and search algorithm. The greatest emphasis is placed on gradient descent parameter surface search with no time averaging in the correlation estimation. This is chosen because gradient search outperforms random search and the transform update is simplified by having rank-one autocorrelation estimates.

4.5.1 Explicit Autocorrelation Estimation

The most obvious way to implement an adaptive transform coding system is to use a windowed correlation estimate of the form

$$\widehat{X}_n = \frac{1}{M} \sum_{k=n-M+1}^{n} x_k x_k^T. \tag{4.15}$$

If the true correlation is constant, then $\widehat{X}_n$ is elementwise an unbiased, consistent estimator of X [30]. There will be "estimation noise" (variance in $\widehat{X}_n$ due to having finite sample size) which decreases monotonically with M. If $\{X_n\}$ is slowly varying, there will also be "tracking noise" (mismatch between $\widehat{X}_n$ and X_n caused by the causal observation window) which increases with M. Thus, in the time-varying case there is a trade-off, controlled by M, and one can expect there to be an optimal value of M depending on the rate at which $\{X_n\}$ varies.

To illustrate the ability to track a time-varying source and the dependence on M, construct the following synthetic source: For each time $n \in \mathbb{Z}^+$, x_n is a zero-mean jointly Gaussian vector with correlation matrix

$$X_n = U_n^T \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \cdot U_n,$$

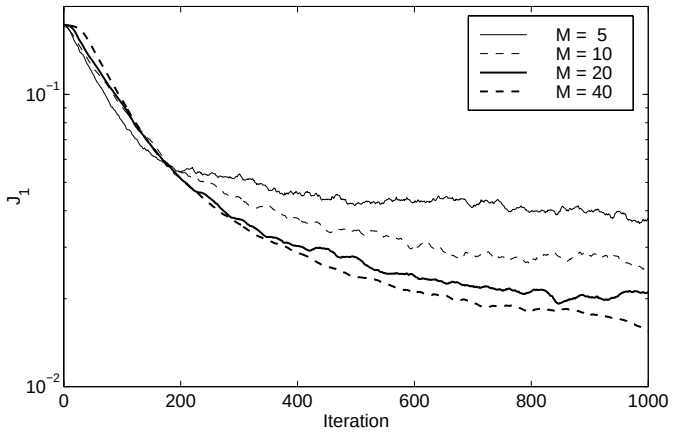
where U_n is a time-varying unitary matrix specified as

$$U_n = G_{1,\omega_1 n+\varphi_1} G_{2,\omega_2 n+\varphi_2} G_{3,\omega_3 n+\varphi_3}.$$

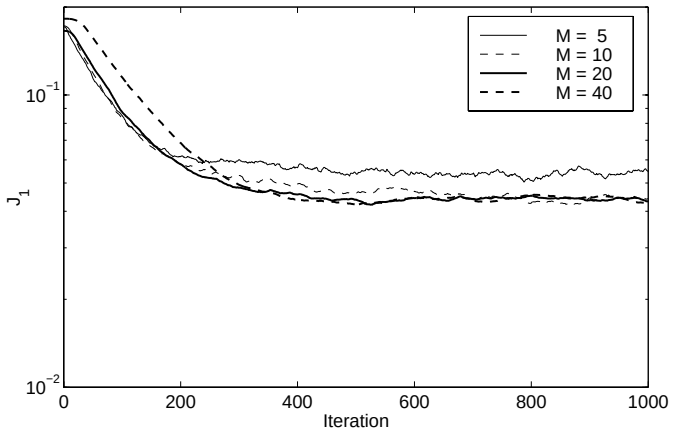
U_n is an ideal transform to be used at time n. The ω_i's are fixed "angular velocities" to be tracked and the φ_i's are independent, uniformly distributed phases. Averaging over randomly selected phases removes any periodic components from simulation results.

This source was used in simulations of the linear search with respect to J_1. For all the simulations $\alpha = 3$ and $\sigma = 0.01$. In the first set of experiments (see Figure 4.4(a)) $\omega_1 = \omega_2 = \omega_3 = 0$. Since the source is not time varying, there is no tracking noise and the estimation noise decreases as M is increased, so the overall performance improves as M in increased. The second and third sets of experiments use $\omega_1 = \omega_2 = \omega_3 = 0.001$ and $\omega_1 = \omega_2 = \omega_3 = 0.002$, respectively. Now since the source is time varying, the performance does not improve monotonically as M is increased because as the estimation noise decreases, the tracking noise increases. For the slower varying source (see Figure 4.4(b)) the performance improves as M is increased from 5 to 20 and then is about the same for $M = 40$. For the faster varying source (see Figure 4.4(c)) the tracking noise is more significant so the best value of M is lower. A faster varying source may also justify a larger value of α.

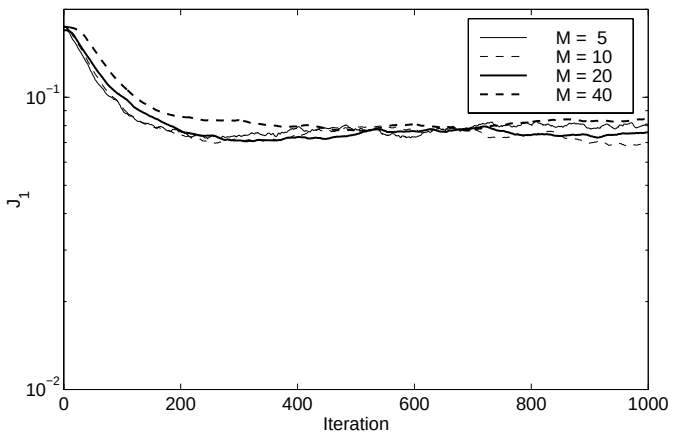
The estimate (4.15) implicitly uses a rectangular window to window the incoming data stream, so each sample vector is equally weighted. One way to more heavily weight the later sample vectors is to use a



(a) $\omega_1 = \omega_2 = \omega_3 = 0$



(b) $\omega_1 = \omega_2 = \omega_3 = 0.001$



(c) $\omega_1 = \omega_2 = \omega_3 = 0.002$

Figure 4.4: Simulations of linear search with respect to J_1 with explicit, windowed correlation estimation. The source is slowly varying as described in the text. M is the length of the data window. Fixed parameters: $\alpha = 3$, $\sigma = 0.01$. Results are averaged over 400 randomly chosen initial conditions Θ_0 and source phases φ_1, φ_2, φ_3.

"forgetting factor" β (as in Recursive Least Squares [104]), which is equivalent to using an exponential window:

$$\widehat{X}_n = \beta \widehat{X}_{n-1} + (1-\beta) x_n x_n^T.$$

This scheme also reduces memory requirements.

4.5.2 Stochastic Update

Taking the autocorrelation estimation of the previous section to the extreme of estimating the autocorrelation based on a single sample vector gives

$$\widehat{X}_n = x_n x_n^T. \tag{4.16}$$

The use of this extremely simple estimate simplifies the calculations associated with parameter surface search. This will be referred to as the *stochastic implementation* because it is the result of replacing an expected value by its immediate, stochastic value.

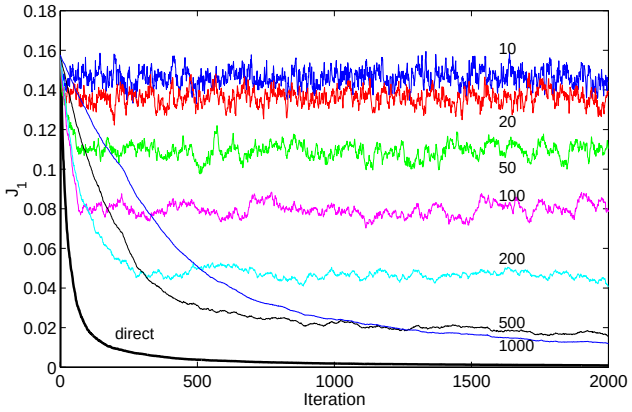
Both random search methods require calculation of J. For a general $X \in \mathbb{R}^{N\times N}$, computing $TXT^T = G_1 G_2 \cdots G_K X G_K^T \cdots G_2^T G_1^T$ requires $8KN$ multiplications and $4KN$ additions because each multiplication by a Givens matrix requires $4N$ multiplications and $2N$ additions. With the rank-one $\widehat{X}_n$ given by (4.16), one can first write

$$\begin{aligned} T\widehat{X}_n T^T &= G_1 G_2 \cdots G_K x_n x_n^T G_K^T \cdots G_2^T G_1^T \\ &= (G_1 G_2 \cdots G_K x_n)(G_1 G_2 \cdots G_K x_n)^T. \end{aligned}$$

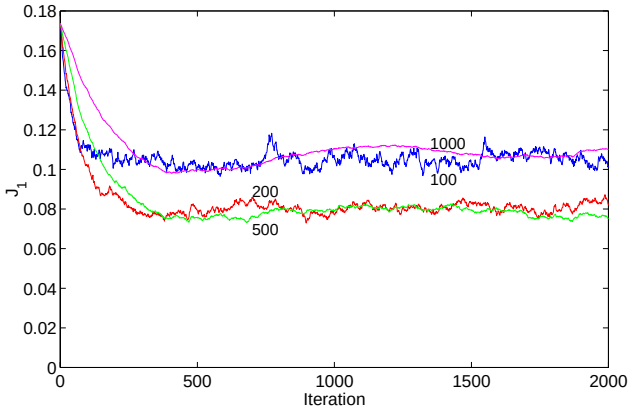
Then since multiplying a vector by a Givens matrix requires 4 multiplications and 2 additions, the bracketed terms can be computed with $4K$ multiplications and $2K$ additions. Now $J_1(T)$ can be computed with K additional multiplications and $K-1$ additional additions or $J_2(T)$ can be computed with N additional multiplications. The computation of ∇J is similarly simplified.

The stochastic implementation of gradient descent parameter search was simulated for the source described in the previous section (see Figure 4.5). There is a single parameter to choose: the step size α. Using Theorems 4.2 and 4.4 gives maximum step sizes of 8/9 and 32/9 for descent with respect to J_1 and J_2, respectively. These theorems apply only to iterative computations with *exact* knowledge of the correlation matrix; however, they provide rough guidelines for step size choice in the stochastic setting.

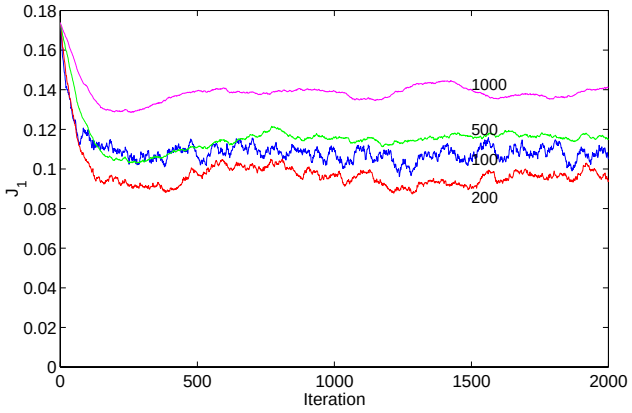
When the source distribution is time-invariant ($\omega_1 = \omega_2 = \omega_3 = 0$ for the source we are considering), the effect of the step size α is easy to discern. A larger step size reduces the adaptation time constants, so steady-state performance is reached more quickly. However, because the parameter vector Θ is adapted based on each source vector, the steady-state performance has a "noisy" stochastic component. This "excess" in J increases as the step size is increased. This has not been characterized analytically, but qualitatively it is similar to the "excess" mean-square error in LMS filtering [205]. Referring to Figure 4.5(a), the steady-state value of J_1 decreases monotonically as α is decreased, but the convergence is slower. Because the source is time-invariant, there is a conceptually simple alternative to the stochastic gradient descent which provides a bound to attainable performance. This is to use all the source vectors observed thus far to estimate the correlation, using (4.15) with $M = n$, and then computing the eigendecomposition of the correlation estimate to full machine precision. This bound is the lowest curve in Figure 4.5(a).



(a) $\omega_1 = \omega_2 = \omega_3 = 0$



(b) $\omega_1 = \omega_2 = \omega_3 = 0.001$



(c) $\omega_1 = \omega_2 = \omega_3 = 0.002$

Figure 4.5: Simulations of stochastic gradient descent with respect to J_1. The source is slowly varying as described in the text. Step sizes are given by $\alpha = \alpha_{\max}/\gamma$, where $\alpha_{\max} = 8/9$ is the maximum step size for stability predicted by Theorem 4.2. Curves are labeled by the value of γ. Results are averaged over 400 random initial conditions Θ_0 and source phases φ_1, φ_2, φ_3. In (a) the performance is compared to computing an exact eigendecomposition of a correlation estimate from all the sample vectors observed thus far.

The situation is more complicated when the source distribution is time-varying. Now the step size affects the ability to track the time variation along with determining the steady-state noise and speed of convergence. Figures 4.5(b) and (c) show the results of simulations with $\omega_1 = \omega_2 = \omega_3 = 0.001$ and $\omega_1 = \omega_2 = \omega_3 = 0.002$, respectively. In the first of these simulations, the best performance is achieved for α between $\frac{8}{9}/500$ and $\frac{8}{9}/200$. The larger of these gives slightly faster convergence and the smaller gives slightly lower steady-state error. For the faster-varying source, $\frac{8}{9}/500$ is too small for effectively tracking the source.

4.5.3 Quantized Stochastic Implementation

In adaptive transform coding, if the transform adaptation is based upon the incoming *uncoded* data stream, then in order for the decoder to track the encoder state, the transform adaptation must be described over a side information channel. This situation, which is commonly called *forward-adaptive*, is depicted in Figure 4.6(a). The need for side information can be eliminated if the adaptation is based on the coded data, as shown in Figure 4.6(b). This *backward-adaptive* configuration again has an analogy in adaptive FIR Wiener filtering: In adaptive linear predictive coding, where the linear predictor is in fact an adaptive FIR Wiener filter, making the adaptation depend on quantized data yields adaptive differential pulse code modulation (ADPCM).[4]

The stochastic gradient descent was simulated in the backward-adaptive configuration. Since quantization is an irreversible reduction in information, it must be at least as hard to estimate the moments of a signal from a quantized version as it is from the original unquantized signal. Thus one would expect the convergence rate to be somewhat worse in the backward-adaptive configuration. Figure 4.7(a) shows simulation results for a time-invariant source ($\omega_1 = \omega_2 = \omega_3 = 0$). The lower set of curves is for direct computation as in Figure 4.5(a) and the upper set of curves is for stochastic gradient descent with step size $\alpha = \frac{8}{9}/500$. With quantization step size $\Delta = 0.125$ or 0.25, the rate of convergence is almost indistinguishable from the unquantized case. As the quantization becomes coarser, the convergence slows. Notice that with direct computation, quantization does not seem to lead to a nonzero steady-state error. Though the quantization is undithered, the random variation of the transform has a similar effect as using a dither signal in conjunction with the quantizer. This may explain this convergence behavior and is suggestive of universal performance of the backward-adaptive scheme, as discussed in Chapter 3.

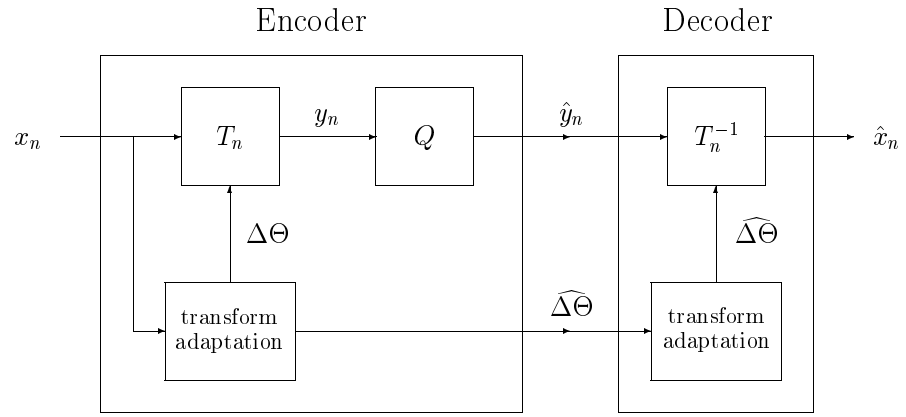
For a slowly varying source ($\omega_1 = \omega_2 = \omega_3 = 0.001$; see Figure 4.7(b)), it is again true that the performance with $\Delta = 0.125$ or 0.25 is indistinguishable from the performance without quantization. The convergence slows as the quantization becomes coarser, but here there may also be a small increase in steady-state error.

4.5.4 Specialization for a Scalar Source

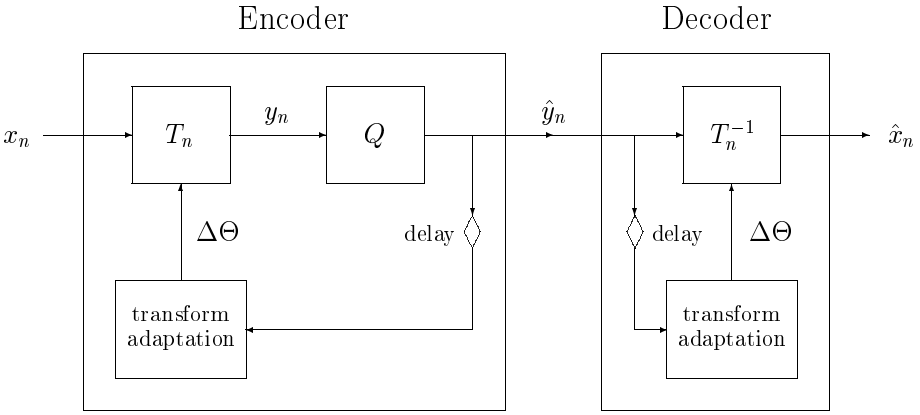
In many applications with processing of vectors, the vectors are actually generated by forming blocks from a scalar-valued source. The methods developed in this chapter are general and hence applicable to this case. However, a few specific refinements facilitate performance better than in the general case.

Suppose the original scalar source is a wide-sense stationary process $\{z_n\}$, which we observe for $n \geq 1$, and that we generate a vector source $\{x_n\}$ by forming blocks of length N. Then the correlation matrix $X =$

[4] Note that ADPCM is often used to refer to a system with adaptive quantization. However, quantization adaptation is beyond the scope of this thesis; see [147, 148].

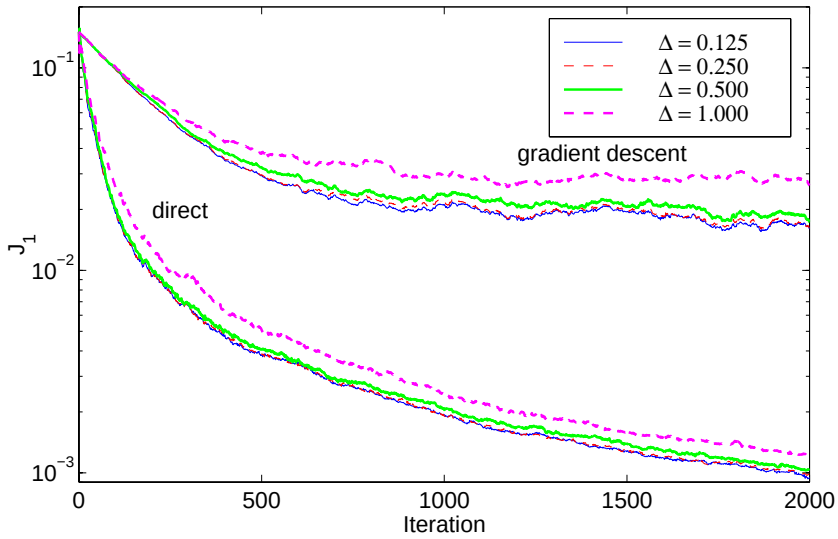


(a) Forward-adaptive transform coding system

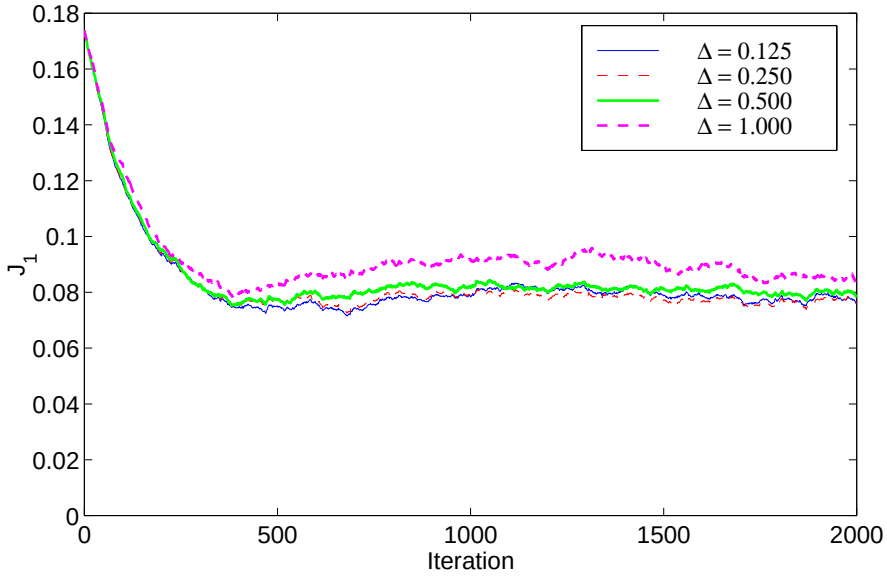


(b) Backward-adaptive transform coding system

Figure 4.6: Structural comparison between forward- and backward-adaptive systems. The backward-adaptive system does not require a side information channel to convey transform state.



(a) Time-invariant source ($\omega_1 = \omega_2 = \omega_3 = 0$)



(b) Slowly-varying source ($\omega_1 = \omega_2 = \omega_3 = 0.001$)

Figure 4.7: Simulations of stochastic gradient descent with respect to J_1 in backward-adaptive configuration. Step sizes are given by $\alpha = \alpha_{\max}/500$, where $\alpha_{\max} = 8/9$ is the maximum step size for stability predicted by Theorem 4.2. Curves are labeled by the value of the quantization step size Δ. Results are averaged over 400 randomly chosen initial conditions Θ_0 and source phases φ_1, φ_2, φ_3. In (a) the performance is also compared to computing an exact eigendecomposition of a correlation estimate based on all the (quantized) sample vectors observed thus far.

$E[x_n x_n^T]$ is a symmetric, Toeplitz matrix with $X_{ij} = r_z(i-j) = E[z_i z_j]$.

One consequence of the symmetric, Toeplitz structure of X is that there are actually less than $K = N(N-1)/2$ independent parameters to estimate to find a diagonalizing transform. For $N = 3$, for example, one can show that

$$X = \begin{bmatrix} a & b & c \\ b & a & b \\ c & b & a \end{bmatrix}$$

has always as an eigenvector $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$ and that it suffices to consider transforms of the form

$$T = \begin{bmatrix} -\sqrt{2}/2 & 0 & \sqrt{2}/2 \\ \zeta & \sqrt{1-2\zeta^2} & \zeta \\ \sqrt{1-2\zeta^2}/\sqrt{2} & -\sqrt{2}\zeta & \sqrt{1-2\zeta^2}/\sqrt{2} \end{bmatrix}.$$

This can be used to derive new performance surface search methods with fewer parameters.

A second consequence is that estimates better than (4.15) can be used. Having observed M N-tuples from the source, (4.15) gives

$$\widehat{X}_{ij} = \frac{1}{M}\sum_{n=1}^{M}(x_n)_i(x_n)_j = \frac{1}{M}\sum_{n=1}^{M} z_{N(n-1)+i} z_{N(n-1)+j}. \tag{4.17}$$

Each of the terms of (4.17) has expected value $r_z(i-j)$ and by averaging over M observations one clearly gets an unbiased, consistent estimate. However, with MN samples we can actually average over $MN-(i-j)$ terms to get a much lower variance estimate:

$$\widehat{X}_{ij} = \widehat{r_z}(i-j) = \frac{1}{MN-(i-j)} \sum_{n=1}^{MN-(i-j)} z_n z_{n+(i-j)}.$$

For a time-varying source, either a finite window or a forgetting factor could be used.

4.6 Conclusions

A new class of algorithms based on parameter surface search was introduced for computing the eigenvectors of a symmetric matrix. These algorithms are potentially useful for adaptive transform coding or on-line principal component analysis. The development is conceptually summarized as follows: A matrix of eigenvectors forms an orthogonal diagonalizing similarity transformation; it suffices to consider orthogonal matrices which are parameterized as a product of Givens rotations; and appropriate parameter values can be found as an unconstrained minimization.

The key is the formulation of unconstrained minimization problems over a minimal number of parameters. Borrowing from the adaptive filtering literature, linear and fixed step random search and gradient descent were applied to the resulting minimization problems. In the gradient descent case, step size bounds were found to ensure convergence in the absence of estimation noise. Simulations demonstrated that in the presence of estimation noise, the gradient descent converges when the step size is chosen small relative to the bound.

In a transform coding application, one may use a rank-one stochastic estimate of the correlation matrix. This simplifies the computations in the gradient descent update. In a backward-adaptive configuration,

the adaptation is driven by quantized data so that the decoder and encoder can remain synchronized without the need for side information. As long as the quantization is not too coarse, the algorithms presented here seem to converge.

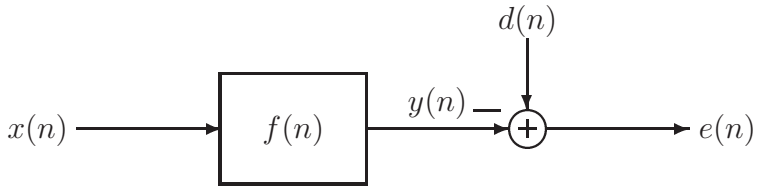


Figure 4.8: Canonical configuration for Wiener filtering. The objective is to design the filter $f(n)$ such that the power of $e(n)$ is minimized.

Appendices

4.A Brief Review of Adaptive FIR Wiener Filtering

The canonical Wiener filtering problem is described as follows.[5] Let $x(n)$ and $d(n)$ be jointly wide-sense stationary, zero-mean, scalar random processes. Design a linear filter $f(n)$ such that the mean-squared error between the desired signal $d(n)$ and the output of the filter $y(n) = x(n) * f(n)$ is minimized (see Figure 4.8). Two common applications are separating signal from noise and channel equalization. For denoising, $x(n) = d(n) + w(n)$ where $w(n)$ is unknown, but has known spectral density and is uncorrelated with $d(n)$. For equalization, $x(n) = d(n) * c(n)$, where $c(n)$ is a channel impulse response. The case where $f(n)$ is constrained to be a causal, L-tap FIR filter is considered here. This and other cases are discussed in detail in [30].

Finding the optimal filter is conceptually simple once we select a convenient vector notation. Let $\bar{f} = [f(0), f(1), \ldots, f(L-1)]^T$ and $\bar{x}_n = [x(n), x(n-1), \ldots, x(n-L+1)]^T$. Then $e(n) = d(n) - \bar{x}_n^T \bar{f}$. The power of $e(n)$ is a quadratic function of the filter vector:

$$J(f) = E[e(n)^2].$$

It is this function which we call the *performance surface*, and finding the optimal filter is to find the parameter vector which yields the minimum of the performance surface.[6] It can be shown that the gradient of J with respect to $\bar{f}$ is given by

$$\nabla J = 2(X\bar{f} - \bar{r}_{dx}),$$

where (consistent with the main text) $X = E[\bar{x}_n \bar{x}_n^T]$ and $\bar{r}_{dx} = E[\bar{x}_n d(n)]$. From this we can conclude that the optimal filter is described by

$$\bar{f}_{\text{opt}} = X^{-1}\bar{r}_{dx}. \tag{4.18}$$

There are two practical problems in applying the analytical solution (4.18). The first is that X and $\bar{r}_{dx}$ may be unknown and may depend on n. A remedy would be to estimate these moments as the incoming data is processed, giving $X(n)$ and $\bar{r}_{dx}(n)$. This leads to the second problem, which is that each update of the filter requires solving a linear system $X(n)\bar{f} = \bar{r}_{dx}(n)$.

The LMS or stochastic gradient algorithm addresses both of these problems. There are two main ideas. Firstly, instead of exactly minimizing J by using (4.18), iteratively update $\bar{f}$ by adding $-\alpha\nabla J$, where $\alpha > 0$ is called the step size. As long as α is chosen small enough, this procedure will converge to $\bar{f}_{\text{opt}}$; however, as

[5] The anonymous designation of "optimal least-squares filtering" is also used.
[6] Under some technical conditions, the minimum is unique.

long as we still require knowledge of X and $\bar{r}_{dx}$ this is not very useful. The second main idea is to replace X and $\bar{r}_{dx}$ by the simplest possible stochastic approximations: $X(n) \approx \bar{x}_n \bar{x}_n^T$ and $\bar{r}_{dx} \approx \bar{x}_n d(n)$. This yields the update equation for LMS:

$$\bar{f}(n+1) = \bar{f}(n) - 2\alpha(y(n) - d(n))\bar{x}_n. \tag{4.19}$$

One normally studies the stability and rate of convergence of (4.19) by analyzing

$$\bar{f}(n+1) = \bar{f}(n) - 2\alpha(X\bar{f} - \bar{r}_{dx}).$$

This can be interpreted as ignoring the stochastic aspect of the algorithm or as looking at the mean of $\bar{f}$ and applying the so-called "independence assumption" [204].

4.B Alternative Gradient Expressions

The gradient expressions given in Section 4.4.3 were intended to facilitate Theorems 4.2 and 4.4. Alternative expressions for ∇J_1 and ∇J_2 are given in this appendix.

We will use the chain rule to compute ∇J_ℓ, $\ell = 1, 2$, through

$$\frac{\partial J_\ell}{\partial \theta_k} = \sum_{i,j} \frac{\partial J_\ell}{\partial T_{ij}} \frac{\partial T_{ij}}{\partial \theta_k}.$$

Recalling the definitions of $U_{(a,b)}$ and $V_{k,\theta}$ from Section 4.4.3.1, if we define $B_{ij}^{(k)} = \partial T_{ij}/\partial\theta_k$, then differentiating (4.4) gives

$$B^{(k)} = U_{(1,k-1)} V_k U_{(k+1,K)}.$$

For both $\ell = 1$ and $\ell = 2$, the following intermediate calculation is useful:

$$\frac{\partial Y_{ab}}{\partial T_{ij}} = \frac{\partial}{\partial T_{ij}} \sum_{r=1}^{N}\sum_{s=1}^{N} T_{as} X_{sr} (T^T)_{rb} = \delta_{i-a} T_{b*} X_{*j} + \delta_{i-b} T_{a*} X_{*j},$$

where T_{b*} is the bth row of T and X_{*j} is the jth column of X.

Now

$$\frac{\partial J_1}{\partial T_{ij}} = \sum_{a\neq b} 2Y_{ab} \frac{\partial Y_{ab}}{\partial T_{ij}} = \sum_{a\neq b} 2Y_{ab}(\delta_{i-a} T_{b*} X_{*j} + \delta_{i-b} T_{a*} X_{*j}) = 4 e_i^T Y (I - e_i e_i^T) T X e_j,$$

where e_i is the column vector with one in the ith position and zeros elsewhere. For J_2 we have

$$\frac{\partial J_2}{\partial T_{ij}} = \frac{\partial}{\partial T_{ij}} \prod_{\ell=1}^{N} Y_{\ell\ell} = \sum_{a=1}^{N} \frac{\partial Y_{aa}}{\partial T_{ij}} \prod_{\ell=1,\ell\neq a}^{N} Y_{\ell\ell} = \sum_{a=1}^{N} \frac{J}{Y_{aa}} 2\delta_{i-a} T_{a*} X_{*j} = 2\frac{J}{Y_{ii}} T_{i*} X_{*j}.$$

4.C Evaluation of $\partial A_{mm}^{(k)}/\partial\theta_\ell$

In this appendix we derive (4.14). First consider the case $\ell = k$. Let $W_k = \partial V_k/\partial\theta$. Differentiating (4.7) gives

$$\begin{aligned}\frac{\partial}{\partial\theta_k} A^{(k)} = {} & U_{(1,k-1)} W_k U_{(k+1,K)} X U_{(1,K)}^T + U_{(1,k-1)} V_k U_{(k+1,K)} X U_{(k+1,K)}^T V_k^T U_{(1,k-1)}^T \\ & + U_{(1,K)} X U_{(k+1,K)}^T W_k^T U_{(1,k-1)}^T + U_{(1,k-1)} V_k U_{(k+1,K)} X U_{(k+1,K)}^T V_k^T U_{(1,k-1)}^T,\end{aligned}$$

which upon evaluation at $\Theta = 0$ reduces to

$$\frac{\partial}{\partial \theta_k} A^{(k)} \Big|_{\Theta=\mathbf{0}} = W_k X + V_k X V_k^T + X W_k^T + V_k X V_k^T.$$

The simple structures of V_k and W_k allow one to now easily show that

$$\frac{\partial}{\partial \theta_k} A^{(k)} \Big|_{\Theta=\mathbf{0}} = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 2(\lambda_{j_k} - \lambda_{i_k}) & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 2(\lambda_{i_k} - \lambda_{j_k}) & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \begin{matrix} \\ \\ i_k \\ \\ j_k \\ \\ \\ \end{matrix}$$
$$\begin{matrix} & & i_k & & j_k & & \end{matrix}$$

Now consider the case $\ell < k$. Differentiating (4.7) and evaluating at $\Theta = \mathbf{0}$ gives

$$\frac{\partial}{\partial \theta_\ell} A^{(k)} \Big|_{\Theta=\mathbf{0}} = V_\ell V_k X + V_k X V_\ell^T + X V_k^T V_\ell^T + V_\ell X V_k^T. \tag{4.20}$$

To satisfy (4.14) we would like to show that the diagonal of (4.20) is zero.

Lemma 4.5 *For $\ell < k$ and $\Theta = 0$, the diagonal of $V_\ell V_k$ is zero.*

Proof: Because $\ell < k$, we have either

(a) $i_\ell < i_k$; or

(b) $i_\ell = i_k$ and $j_\ell < j_k$.

Recall also that $i_\ell < j_\ell$ and $i_k < j_k$.

The only potentially nonzero elements of $V_\ell V_k$ are in the (i_ℓ, i_k), (i_ℓ, j_k), (j_ℓ, i_k), and (j_ℓ, j_k) positions. The (i_ℓ, j_k) element can not be on the diagonal because either $i_\ell < i_k < j_k$ or $i_\ell = i_k < j_k$; similarly for the (j_ℓ, i_k) element. The (i_ℓ, i_k) element is $-\delta(j_\ell - j_k)$ and hence when this element is on the diagonal, it is zero; similarly for the (j_ℓ, j_k) element. □

Corollary 4.6 *For $\ell < k$ and $\Theta = 0$, the diagonals of $V_k V_\ell^T$, $V_k^T V_\ell^T$, and $V_\ell V_k^T$ are zero.*

Since X is diagonal, Lemma 4.5 and Corollary 4.6 can be combined to show that the diagonal of (4.20) is zero.

The $\ell > k$ case is similar to the $\ell < k$ case.

Chapter 5

Multiple Description Coding

SOURCE CODING researchers are demanding consumers of communication systems: They ask for every bit they produce to be reliably delivered. Depending on what is known about the channel, this may be possible in Shannon's sense, but at what cost? At the very least, depending on the acceptable probability of failure and on how close the rate is to the channel capacity, large block sizes and complex encoding and decoding may be needed. In addition, compression may greatly increase the sensitivity to any remaining uncorrected errors.

A simple example is a text file containing a story. If a handful of characters are deleted at random, the reader may be distracted, but the meaning is likely to be fully conveyed. On the other hand, losing a few random bytes of a Lempel–Ziv compressed version of the story could be catastrophic. If the compression is by a factor of, say, ten, the effect is much more pronounced than the loss of ten times as many bytes. The deletions make it nearly impossible to correctly interpret the meanings of the symbols in the file. This effect suggests that if the probability of error cannot be made zero by channel coding, it may be beneficial to leave data uncompressed.

To not compress is a rather extreme reaction to the possibility of a bit error. A more temperate approach is to account for the possibility of (uncorrected) channel impairments in the design of the source coding. For example, this may motivate the use of variable-to-fixed length codes instead of the more common fixed-to-variable length codes, of which the Huffman code is an example.

This chapter addresses the problem of multiple description (MD) source coding, which can be cast as a source coding method for a channel whose end-to-end performance (with channel coding) includes uncorrected erasures. This channel is encountered in a packet communication system that has effective error detection but does not have retransmission of incorrect or lost packets. After a comprehensive introduction, two new transform-based methods for MD source coding are introduced. These are also applied to image and audio coding.

This chapter includes research conducted jointly with Jelena Kovačević [67, 113]; Jelena Kovačević and Martin Vetterli [69, 70, 68, 76]; and Ramon Arean and Jelena Kovačević [6, 7].

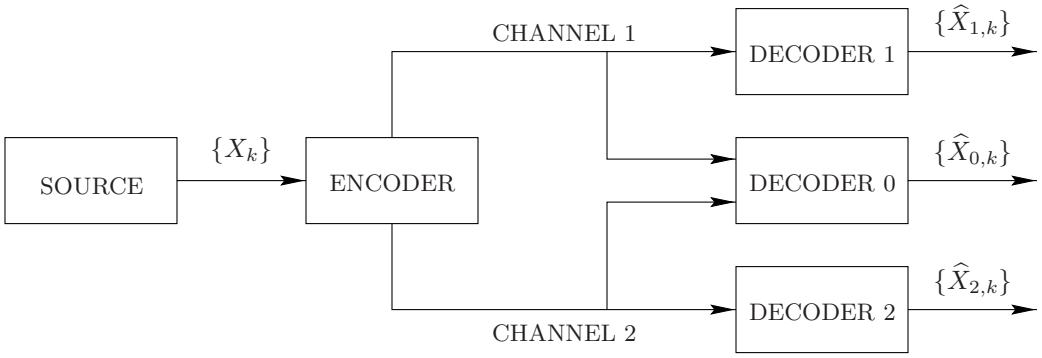


Figure 5.1: Scenario for multiple description source coding with two channels and three receivers. The general case has M channels and $2^M - 1$ receivers.

5.1 Introduction

At the September 1979 Shannon Theory Workshop held at the Seven Springs Conference Center, Mount Kisco, New York, the following question was posed by Gersho, Ozarow, Witsenhausen, Wolf, Wyner, and Ziv [48]:[1] If an information source is described by two separate descriptions, what are the concurrent limitations on qualities of these descriptions taken separately and jointly? This problem would come to be known as the *multiple description problem*. The primary theoretical results in this area were provided by the aforementioned researchers along with Ahlswede, Berger, Cover, El Gamal, and Zhang in the 1980's.

Multiple description coding refers to the scenario depicted in Figure 5.1. An encoder is given a sequence of source symbols $\{X_k\}_{k=1}^N$ to communicate to three receivers over two noiseless (or error-corrected) channels. One decoder (the *central decoder*) receives information sent over both channels while the remaining two decoders (the *side decoders*) receive information only over their respective channels. The transmission rate over channel i is denoted by R_i, $i = 1, 2$; *i.e.*, signaling over channel i uses at most 2^{NR_i} symbols. Denoting by $\{\widehat{X}_{i,k}\}_{k=1}^N$ the reconstruction sequence produced by decoder i, we have three distortions

$$D_i = \frac{1}{N} \sum_{k=1}^{N} E[\delta_i(X_k, \widehat{X}_{i,k})], \quad i = 1, 2, 3,$$

where the $\delta_i(\,\cdot\,,\,\cdot\,)$'s are potentially distinct, nonnegative, real-valued distortion measures.

The central theoretical problem is to determine the set of achievable values (in the usual Shannon sense) for the quintuple $(R_1, R_2, D_0, D_1, D_2)$. Specifically, $(r_1, r_2, d_0, d_1, d_2)$ is achievable if for sufficiently large N there exist encoding and decoding mappings such that

$$\begin{aligned} R_i &\leq r_i, \quad i = 1, 2; \\ D_i &\leq d_i, \quad i = 1, 2, 3. \end{aligned}$$

Decoder 1 receives R_1 bits and hence cannot have distortion less than $D(R_1)$, where $D(\,\cdot\,)$ is the distortion–rate function of the source. Making similar arguments for the other two decoders and using rate–distortion functions

[1] A transitive chain of acknowledgements suggests that the problem originated with Gersho.

instead of distortion–rate functions gives the following bounds on the achievable region:[2]

$$R_1 + R_2 \geq R(D_0), \tag{5.1}$$

$$R_1 \geq R(D_1), \tag{5.2}$$

$$R_2 \geq R(D_2). \tag{5.3}$$

Achieving equality simultaneously in (5.1)–(5.3) would imply that an optimal rate $R_1 + R_2$ description can be partitioned into optimal rate R_1 and rate R_2 descriptions. Unfortunately, this is not true because optimal individual descriptions at rates R_1 and R_2 are similar to each other and hence redundant when combined. Making descriptions individually good and yet "independent" of each other is the fundamental trade-off in this problem.

The counterpart to achievability is the design of explicit codes that operate on finite blocks of the source. The original contributions of this chapter are in this practical area and appear in Sections 5.3 and 5.4. The principal theoretical and practical results on multiple description coding are surveyed in Section 5.2.

The multiple description problem can be generalized to more than two channels and more than three receivers. The natural extension is to M channels and $2^M - 1$ receivers—one receiver for each nonempty subset of channels. This generalization was considered by Witsenhausen [208] for the restricted case where the source has finite entropy rate and lossless communication is required when $e < M$ of the channels are lost. Normalizing the source rate to one and assuming equal usage of each channel, each channel must accommodate a rate of $1/(M - e)$. (The rate cannot be lowered because the sum of the rates of the received channels must be at least one.) This bound is achieved by using (truncated) Reed–Solomon codes. The situation with three channels and seven decoders was studied by Zhang and Berger [225].

The new MD codes constructed in this chapter apply to the generalized MD problem. In fact, the method described in Section 5.4 reduces to a trivial repetition code when there are only two channels. Unfortunately, other than Witsenhausen's result (for a specific source and distortion measure, and requiring zero distortion for specified receivers), no tight achievability bounds are known for generalized MD coding.

5.1.1 Applicability to Packet Networks

Recently the problem of transmitting data over heterogenous packet networks has received considerable attention. A typical scenario might require data to move from a fiber link to a wireless link, which necessitates dropping packets to accommodate the lower capacity of the latter. If the network is able to provide preferential treatment to some packets, then the use of a multiresolution or layered source coding system is the obvious solution. But what if the network will not look inside packets and discriminate? Then packets will be dropped at random, and it is not clear how the source (or source–channel) coding should be designed. If packet retransmission is not an option (*e.g.*, due to a delay constraint or lack of a feedback channel), one has to devise a way of getting meaningful information to the recipient despite the loss. The situation is similar if packets are lost due to transmission errors or congestion.

Drawing an analogy between packets and channels, packet communication with M packets is equivalent to generalized multiple description coding with M channels. Each of the $2^M - 1$ nonempty subsets of the packets

[2] Since the distortion metrics may differ, the use of a single symbol $R(\cdot)$ for the rate–distortion function of the source is a slight abuse of notation.

leads to a potentially distinct reconstruction with some distortion. The case where no packet is received is ignored to maintain the analogy with the classical MD problem and because the source coding in that case is irrelevant. A recent surge of interest in multiple description coding seems to be due primarily to this application (see [102, 189, 201, 146, 202, 169, 7]) and yet the present work is among the first to effectively use more than two packets [67, 69, 70, 68].

If our goal is to effectively communicate over packet networks, we should not discount the use of established techniques. To this end it should be noted that when feasible, retransmission of lost packets is effective.[3] The use of a retransmission protocol, like TCP [200], requires at a minimum that a feedback channel is available to indicate which packets have been successfully received. Even if feedback is available, many factors may preclude the retransmission of lost or corrupted packets. Retransmission adds delay of at least one round-trip transmission time. This may be unacceptable for two-way communication and necessitates additional buffering for streaming multimedia. Retransmission is generally not feasible in broadcast environments because of the so-called feedback implosion problem whereby the loss of a single packet may spark many retransmission requests.

In this chapter, only feedforward coding strategies are considered. The conventional approach is to use separate source and channel coding, with a forward erasure-correcting code (FEC) to mitigate the effect of lost packets. Various comparisons to FEC are presented in Sections 5.3 and 5.4.

From an information-theoretic perspective, an idealized model is to assume that packet losses (erasures) are i.i.d. with a known probability p and that the message sequence is arbitrarily long. Then assuming that the packets have fixed payload of one unit, the capacity of the channel is $1-p$ per channel use. Furthermore, this capacity can be attained by choosing a sequence of good (M,N) block codes with rate $N/M < 1-p$, with $M, N \to \infty$.[4] Attaining error-free transmission at a rate arbitrarily close to the channel capacity is intimately tied to having an arbitrarily long message sequence. "Good" (M,N) codes are ones which allow the N data symbols to be decoded as long as at least N of the M channel symbols are received. (Any time less than N channel symbols are received, there is ambiguity about the transmitted codeword.) The number of received codewords r is the sum of M independent Bernoulli random variables. Thus by the Strong Law of Large Numbers [88], r satisfies

$$\frac{r}{M} \to 1-p \text{ almost surely as } M \to \infty.$$

Since $N/M < 1-p$, asymptotically the probability that at least N channel symbols are received is one. To highlight the asymptotic nature of this result, note that the probability of receiving less than N channel symbols is at least

$$p^{M-N+1}(1-p)^{N-1}\binom{M}{N-1} > 0.$$

(This is obtained by computing the probability that exactly $N-1$ channel symbols are received.) Therefore for a finite block size there is always a nonzero probability that the communication will fail. For fixed M and $p < 1/2$, this probability increases with the code rate N/M, and it can be significant for moderate values of M.

The emphasis in this chapter is on situations in which long block codes cannot be used. Note that the length of the code is limited to the number of packets used in the communication: robustness to erasures

[3] From an information-theoretic view, feedback does not increase capacity (for a discrete memoryless channel) but does make it easier to achieve it [34].

[4] The use of (M,N) in place of the usual (n,k) is to keep notation consistent between this chapter, Chapter 2, and the harmonic analysis and linear algebra literature.

comes from redundancy spread across packets; redundancy within a packet is not useful. For example, consider a network using Internet Protocol, Version 6 (IPv6) [41]. An IPv6 node is required to handle 576-byte packets without fragmentation, and it is recommended that larger packets be accommodated.[5] Accounting for packet headers, a 576-byte packet may have a payload as large as 536 bytes. With packets of this size, a typical image associated with a WWW page may be communicated in a handful of packets; say, ten. One cannot use the law of large numbers to analyze channel codes with only ten output symbols. An explicit evaluation of the performance of a length ten linear block channel code is given in Section 5.4.3. Another reason for using short channel codes is to keep buffering requirements and delay small.

5.1.2 Historical Notes

At the previously mentioned September 1979 meeting, Wyner presented preliminary results on MD coding obtained with Witsenhausen, Wolf, and Ziv for a binary source and Hamming distortion. At that very meeting [33], Cover and El Gamal determined and reported the achievable rate region later published in [48]. Ozarow's contribution was to show the tightness of the El Gamal–Cover region for memoryless Gaussian sources and squared-error distortion [149]. Subsequently, Ahlswede [2] showed that the El Gamal–Cover region is tight in the "no excess rate" sum case (where there is equality in (5.1)), and Zhang and Berger [225] showed that this region is not tight when there is excess rate. The complementary situation, where (5.2)–(5.3) hold with equality, is called the "no excess marginal rate" case and was also studied by Zhang and Berger [226].

The history of the MD problem for a memoryless binary symmetric source with Hamming distortion and no excess sum rate is interesting and allows simple, concrete comparisons to single channel bounds. Let $\{X_k\}$ be a sequence of i.i.d. Bernoulli(1/2) random variables. This is called a memoryless binary symmetric source. Also let

$$\delta_i(x,\hat{x}) = \begin{cases} 0, & \text{if } x = \hat{x}, \\ 1, & \text{if } x \neq \hat{x}, \end{cases} \qquad \text{for} \quad i = 1,2,3.$$

For this source and distortion, consider a restriction of the MD problem to $D_0 = 0$, $D_1 = D_2$, and $R_1 = R_2 = r$. Since the source has entropy rate 1 bit per symbol, we must have $r \geq 1/2$ in order to have $D_0 = 0$. Let us fix $r = 1/2$ and investigate the minimum value for $D_1 = D_2$, which we denote by d.

The rate–distortion function is given by

$$R(D) = \begin{cases} 1 - h(D), & 0 \leq D \leq 1/2, \\ 0, & D > 1/2, \end{cases} \tag{5.4}$$

where

$$h(p) = -p \log_2 p - (1-p)\log_2(1-p)$$

is the binary entropy function [34]. Thus the single-channel rate–distortion bound (5.2) gives

$$\frac{1}{2} \geq 1 + d \log_2 d + (1-d)\log_2(1-d),$$

which calculates to

$$d > 0.110. \tag{5.5}$$

[5] Without the "Jumbo Payload" option, the maximum packet size is 65 575 bytes (65 535-byte payload).

As was discussed earlier, it is unlikely that each individual description can be very good (approaching the rate–distortion bound) while together forming a very good description. So how might we actually encode? Since the source is memoryless and incompressible, one might guess that the best we can do is to send alternate symbols over the two channels. This clearly satisfies $r = 1/2$ and $D_0 = 0$. The performance at the side decoders is $D_1 = D_2 = 1/4$ because half the bits are received and on average half of the remaining bits can be guessed correctly. This gives an initial upper bound

$$d \leq \frac{1}{4}. \tag{5.6}$$

Closing the difference of more than a factor of two between (5.5) and (5.6) took sophisticated techniques and about four years.

El Gamal and Cover lowered the upper bound on d to

$$d \leq (\sqrt{2} - 1)/2 \tag{5.7}$$

as a special case of the following theorem [48]:

Theorem 5.1 (Achievable rates for multiple description coding) *Let $X_1, X_2, \ldots$ be a sequence of i.i.d. finite alphabet random variables drawn according to a probability mass function $p(x)$. Let $\delta_i(\cdot, \cdot)$ be bounded. An achievable rate region for distortions (D_0, D_1, D_2) is given by the convex hull of all (R_1, R_2) such that*

$$\begin{aligned} R_1 &> I(X; \widehat{X}_1), \\ R_2 &> I(X; \widehat{X}_2), \\ R_1 + R_2 &> I(X; \widehat{X}_0, \widehat{X}_1, \widehat{X}_2) + I(\widehat{X}_1; \widehat{X}_2), \end{aligned}$$

for some probability mass function

$$p(x, \hat{x}_0, \hat{x}_1, \hat{x}_2) = p(x)p(\hat{x}_0, \hat{x}_1, \hat{x}_2 \mid x), \tag{5.8}$$

such that

$$\begin{aligned} D_0 &\geq E\delta_0(X; \widehat{X}_0), \\ D_1 &\geq E\delta_1(X; \widehat{X}_1), \\ D_2 &\geq E\delta_2(X; \widehat{X}_2). \end{aligned}$$

To use this theorem, one chooses the distribution of the auxiliary random variables $\widehat{X}_0$, $\widehat{X}_1$, and $\widehat{X}_2$ (jointly with X) to satisfy (5.8) and the distortion criteria. Each set of auxiliary random variables yields an achievable rate region. The convex hull of these regions is also achievable.

The bound (5.7) is easily obtained with a weaker theorem of El Gamal and Cover. This theorem apparently first appeared in print in [225] because it was superceded by Theorem 5.1 before the publication of [48].

Theorem 5.2 (Weak El Gamal–Cover theorem) *The quintuple $(R_1, R_2, D_0, D_1, D_2)$ is achievable if there exist random variables U and V jointly distributed with a generic source random variable X such that*

$$\begin{aligned} R_1 &> I(X; U), \\ R_2 &> I(X; V), \\ R_1 + R_2 &> I(X; U, V) + I(U; V), \end{aligned}$$

and there exist random variables of the forms

$$\begin{aligned}\widehat{X}_0 &= g_0(U,V),\\ \widehat{X}_1 &= g_1(U),\\ \widehat{X}_2 &= g_2(V),\end{aligned}$$

such that $E\delta_i(X,\widehat{X}_i) \le D_i$ *for* $i = 0,1,2$.

A choice of auxiliary random variables that leads to (5.7) is now exhibited.[6] Let U and V be i.i.d. Bernoulli random variables with probability $2^{-1/2}$ of equaling one. With $X = U \cdot V$, X is Bernoulli(1/2) as desired. Let $g_0(U,V) = UV = X$, $g_1(U) = U$, and $g_2(V) = V$. We can now check that Theorem 5.2 gives the desired bound.

Firstly, $\widehat{X}_0 = X$, so $D_0 = 0$ as required. Next,

$$\begin{aligned}E\delta_1(X,\widehat{X}_1) &= P(\widehat{X}_1 \neq X)\\ &= P(X=0,U=1) + P(X=1,U=0)\\ &= P(UV=0,U=1) + P(UV=1,U=0)\\ &= P(V=0,U=1) + 0\\ &= \frac{\sqrt{2}}{2}\cdot\left(1-\frac{\sqrt{2}}{2}\right) = \frac{\sqrt{2}-1}{2}\end{aligned}$$

and similarly $E\delta_2(X,\widehat{X}_2) = (\sqrt{2}-1)/2$. Now for the side rate bounds we compute

$$\begin{aligned}I(X;U) &= H(X) - H(X \mid U) = 1 - H(UV \mid U)\\ &= 1 - P(U=0)H(UV \mid U=0) - P(U=1)H(UV \mid U=1)\\ &= 1 - 0 - P(U=1)H(V)\\ &= 1 - \frac{\sqrt{2}}{2}h\Big(\frac{\sqrt{2}}{2}\Big) \approx 0.383.\end{aligned}$$

$I(X;V)$ is given by the same expression. Finally,

$$I(X;U,V) + I(U;V) = H(X) - H(X \mid U,V) + I(U;V) = 1 - 0 + 0 = 1.$$

It follows now from Theorem 5.2 that $(1/2, 1/2, 0, (\sqrt{2}-1)/2, (\sqrt{2}-1)/2)$ is achievable. Moreover, any quintuple $(R_1, R_2, 0, (\sqrt{2}-1)/2, (\sqrt{2}-1)/2)$ with $R_1 > 0.383$, $R_2 > 0.383$, and $R_1 + R_2 \ge 1$ is achievable. These findings on the achievable rates for distortions $(D_0, D_1, D_2) = (0, (\sqrt{2}-1)/2, (\sqrt{2}-1)/2)$ are summarized in Figure 5.2. The boundaries at $R_i \approx 0.264$ come from evaluating (5.2)–(5.3) with (5.4).

Let us now return to the historical narrative on MD coding of a binary symmetric source with no excess rate. The application of Theorem 5.2 above gives an upper bound on d: $d \le (\sqrt{2}-1)/2$. The first improvement to the lower bound (5.5) was provided by Wolf, Wyner, and Ziv [210] through the following theorem:

Theorem 5.3 *If* $(R_1, R_2, D_0, D_1, D_2)$ *is achievable, then*

$$R_1 + R_2 \ge \begin{cases} 2 - \bar{h}(D_0) - \bar{h}(D_1 + 2D_2)\\ 2 - \bar{h}(D_0) - \bar{h}(2D_1 + D_2),\end{cases}$$

[6] The analysis here follows the explicit discussion by Berger and Zhang [15], though they report (5.7) to be "well-known."

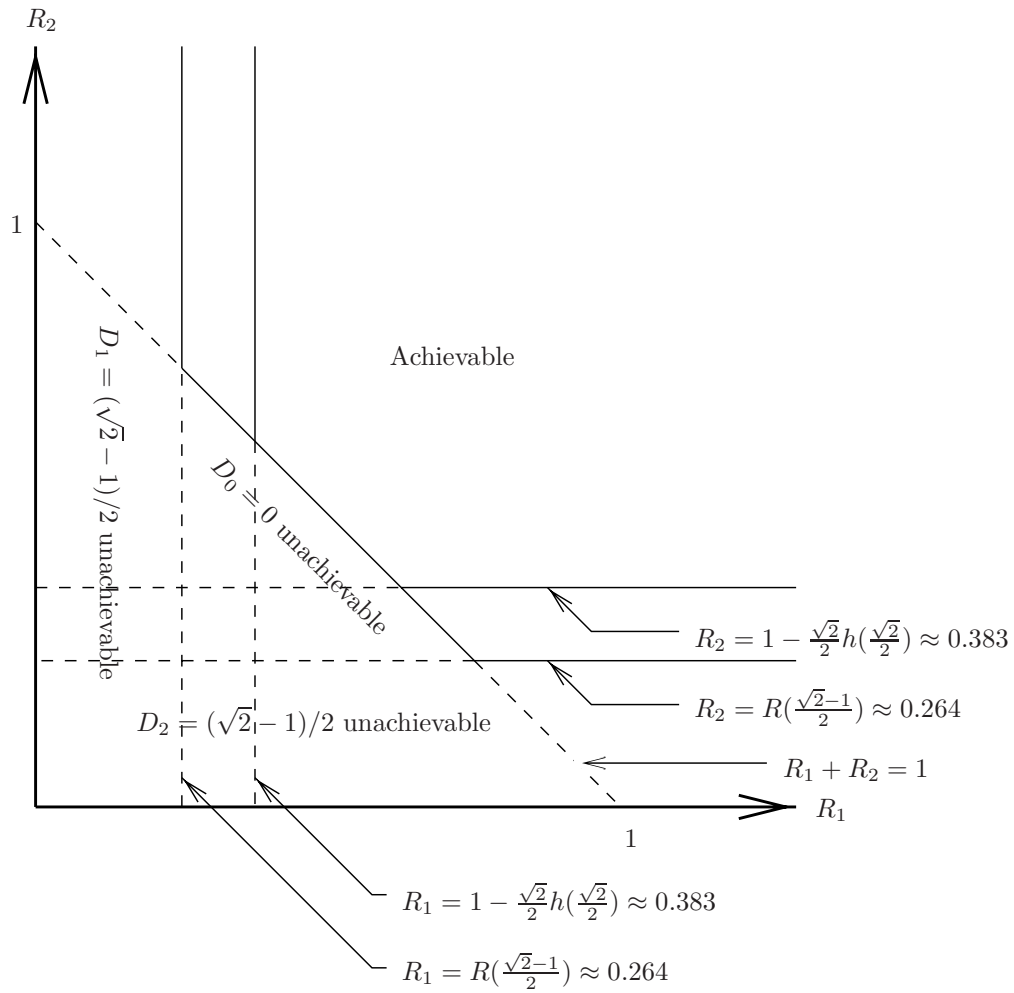


Figure 5.2: Achievable rates for multiple description coding of a binary symmetric source with Hamming distortion $(D_0, D_1, D_2) = (0, (\sqrt{2}-1)/2, (\sqrt{2}-1)/2)$. The excluded regions come from evaluating the rate–distortion bounds (5.1)–(5.3) with (5.4). The included region comes from an application of Theorem 5.2. Note that there is a gap between the regions known to be achievable and unachievable. This could possibly be reduced through applications of Theorem 5.2 with other choices of auxiliary random variables.

where

$$\bar{h}(\lambda) = \begin{cases} 0, & \lambda = 0, \\ -\lambda \log_2 \lambda - (1-\lambda)\log_2(1-\lambda), & 0 < \lambda \le 1/2, \\ 1, & \lambda > 1/2. \end{cases} \tag{5.9}$$

This specializes to give $d \ge 1/6$. Theorem 5.3 was soon improved by Witsenhausen and Wyner [209] to the following:

Theorem 5.4 *If $(R_1, R_2, D_0, D_1, D_2)$ is achievable, then in all cases*

$$R_1 + R_2 \ge 1 - \bar{h}(D_0);$$

furthermore, if $2D_1 + D_2 \le 1$, then

$$R_1 + R_2 \ge 2 - \bar{h}(D_0) - \bar{h}\left(2D_1 + D_2 - \frac{2D_1^2}{1 - D_2}\right),$$

and if $D_1 + 2D_2 \le 1$, then

$$R_1 + R_2 \ge 2 - \bar{h}(D_0) - \bar{h}\left(D_1 + 2D_2 - \frac{2D_2^2}{1 - D_1}\right),$$

where $\bar{h}(\,\cdot\,)$ is given by (5.9).

Applied to the special case at hand, Theorem 5.4 yields $d \ge 1/5$. The gap in the bounds on d was finally closed by Berger and Zhang [15], who showed that $d = (\sqrt{2} - 1)/2$. Further results on binary multiple descriptions in the Shannon setting appear in [225, 226].

Witsenhausen's original paper on MD coding [208]—which did not use the name "multiple description"—has been omitted from this discussion because it requires the central decoder to make no errors whatsoever, as opposed to the less stringent Shannon theory requirement of vanishingly small error probability. In this alternative setting, Witsenhausen showed

$$\left(d_1 + \frac{1}{2}\right)\left(d_2 + \frac{1}{2}\right) \ge \frac{1}{2},$$

which specializes to $d \ge (\sqrt{2} - 1)/2$.

For historical completeness, it should be noted that in an earlier paper [86], Gray and Wyner considered bounds on source coding matched to a simple network model, as shown in Figure 5.3. Instead of having a single source sequence to communicate over two channels to three receivers, they have a sequence of pairs of random variables $\{(X_k, Y_k)\}_{k=1}^{\infty}$ to communicate to two receivers over three channels. Receiver 1 is interested only in $\{X_k\}$ and forms its estimate from Channel 1 and a common channel. Receiver 2 has its own private channel and is interested in the other sequence $\{Y_k\}$. As they suggest, a natural coding scheme is for the common channel to carry a "coarse" version of the pair (X, Y) and for the private channels to add refinement information for the individual components. This paper influenced the development of successive refinement and multiresolution coding (see, *e.g.*, [50]) but is not immediately applicable to the multiple description problem. Since the papers on MD coding in the 1980's did not reference Gray and Wyner, it may be assumed that the later work on this problem was done independently.

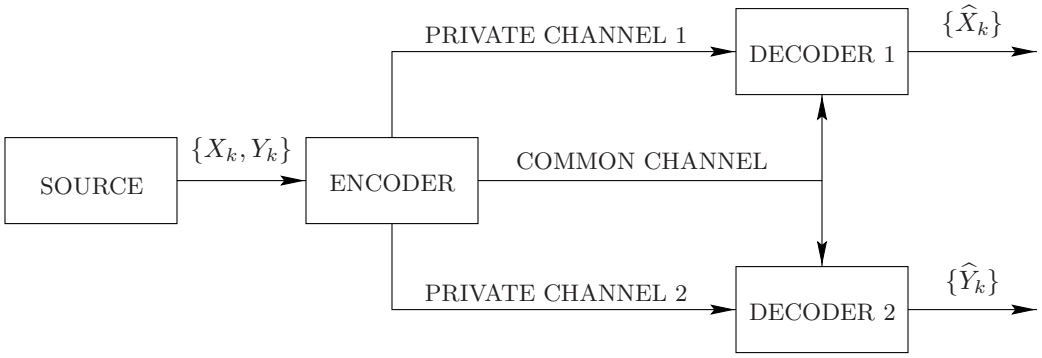


Figure 5.3: Simple network considered by Gray and Wyner [86].

It should also be noted that there is a substantial literature on the error-sensitivity of compressed data and more generally on trade-offs between source and channel coding. The reader is referred to [144, 103, 54, 134, 52, 89, 21, 97, 175, 96] for a sampling of the results.

Finally, MD coding includes as a special case the more well-known *successive refinement* or *multiresolution* coding. The successive refinement problem can also be described by Figure 5.1, but the interest is only in characterizing achievable (R_1, R_2, D_0, D_1). In other words, no attempt is made to estimate the source from Channel 2 alone; or, Channel 1 is always present. The conditions for perfect successive refinement—where (5.1) and (5.2) hold with equality—are described in [50]. The result follows from the tightness of Theorem 5.1 for the no excess rate case, proven by Ahlswede [2]. See also [161].

5.2 Survey of Multiple Description Coding

The early history of multiple description coding was dominated by investigations with a discrete-alphabet source and Hamming distortion measure. This thesis addresses primarily the communication of images and continuous-valued random processes; thus, our attention now turns to continuous-alphabet sources. In accordance with convention, and for convenience, mean-squared error (MSE) distortion is used exclusively.

5.2.1 Theoretical Bounds

As was mentioned in the previous section, the achievable rate–distortion region is completely known only for a memoryless Gaussian source. This result, obtained by Ozarow in 1980 [149], is summarized by the following theorem:

Theorem 5.5 *Let X_1, X_2, ... be a sequence of i.i.d. unit variance Gaussian random variables. The achievable set of rates and mean-squared error distortions is the union of points satisfying*

$$D_1 \geq 2^{-2R_1}, \tag{5.10}$$

$$D_2 \geq 2^{-2R_2}, \tag{5.11}$$

$$D_0 \geq 2^{-2(R_1+R_2)} \cdot \frac{1}{1-(\sqrt{\Pi}-\sqrt{\Delta})^2}, \tag{5.12}$$

where

$$\Pi = (1 - D_1)(1 - D_2) \tag{5.13}$$

and

$$\Delta = D_1 D_2 - 2^{-2(R_1+R_2)}. \tag{5.14}$$

The "forward" part of this theorem, the achievability of any point in the described set, is proven by using Theorem 5.2 with auxiliary variables

$$\begin{aligned} U &= X + N_1, \\ V &= X + N_2, \end{aligned}$$

where N_1 and N_2 are jointly zero-mean Gaussian with covariance matrix

$$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix}.$$

Though this is the standard argument, it should be noted that Theorems 5.1 and 5.2 are proven for discrete-alphabet sources. The "converse" part is proven through an unusual use of an auxiliary random variable.

Since Theorem 5.5 is the key result in coding continuous-valued sources, the region defined therein warrants a close look. The bounds (5.10)–(5.11) are simply the side-channel rate–distortion bounds, a repeat of (5.2)–(5.3). In the final inequality (5.12), $[1 - (\sqrt{\Pi} - \sqrt{\Delta})^2]^{-1}$ is the factor by which the central distortion must exceed the rate–distortion bound. Denote this factor γ.

A few examples will clarify the behavior of γ and the resulting properties of the achievable region. First, suppose that the descriptions are individually very good, yielding $D_1 = 2^{-2R_1}$ and $D_2 = 2^{-2R_2}$. Then $\Delta = 0$ and we may write

$$D_0 \;\geq\; D_1 D_2 \frac{1}{1 - (1 - D_1)(1 - D_2)} \;=\; \frac{D_1 D_2}{D_1 + D_2 - D_1 D_2}.$$

A further chain of inequalities gives $D_0 \geq \min\{D_1, D_2\}/2$, so the joint description is only slightly better than the better of the two individual descriptions.

On the other hand, suppose the joint description is as good as possible, so $D_0 = 2^{-2(R_1+R_2)}$. Then $\gamma = 1$, so $\Pi = \Delta$, and thus

$$D_1 + D_2 = 1 + 2^{-2(R_1+R_2)}. \tag{5.15}$$

Recall that a distortion value of 1 is obtained with no information, simply estimating the source by its mean. For anything but a very low rate, (5.15) implies a very poor reconstruction for at least one of the side decoders.

To gauge the transition between these extreme cases, we may estimate γ under the assumptions $R_1 = R_2 \gg 1$ and $D_1 = D_2 \approx 2^{-2(1-\alpha)R_1}$ with $0 < \alpha \leq 1$. Then

$$\begin{aligned} \frac{1}{\gamma} &= 1 - (\sqrt{\Pi} - \sqrt{\Delta})^2 \\ &= 1 - \left((1 - D_1) - \sqrt{D_1^2 - 2^{-4R_1}} \right)^2 \\ &\approx 1 - \left((1 - D_1) - D_1 \right)^2 \\ &= 4D_1 - 4D_1^2 \;\approx\; 4D_1. \end{aligned}$$

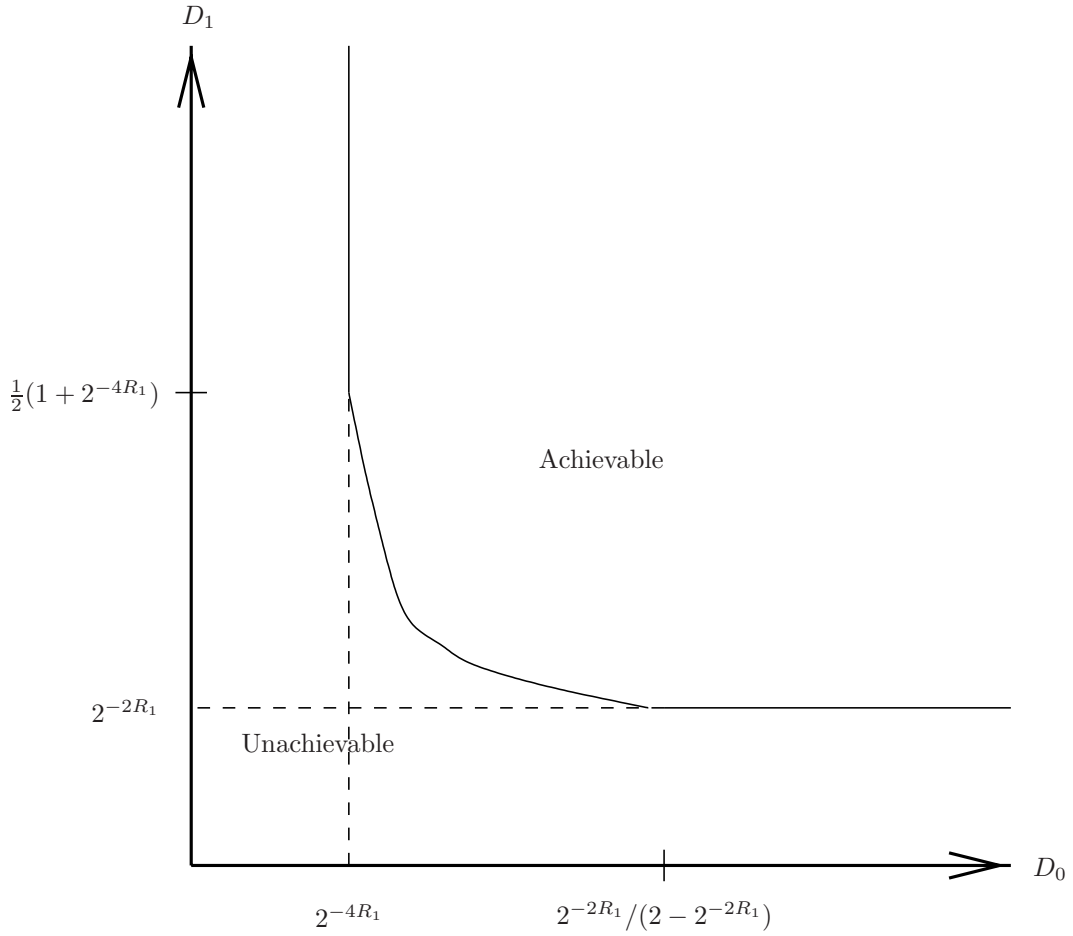


Figure 5.4: Achievable central and side distortions for multiple description coding of a memoryless Gaussian source with squared-error distortion. $D_1 = D_2$ and $R_1 = R_2$ are assumed.

Substituting $\gamma = (4D_1)^{-1}$ in (5.12) gives $D_0 \geq 2^{-4R_1}(4D_1)^{-1}$, so the product of central and side distortions is approximately lower bounded by $4^{-1}2^{-4R_1}$. The best decay of the central distortion is now $D_0 \approx 4^{-1}2^{-2(1+\alpha)R_1}$. This shows that the penalty in the exponential rate of decay of D_1 (the difference from the optimal decay rate, indicated by α being positive) is precisely the increase in the rate of decay of D_0. The region of achievable (D_0, D_1) pairs when $D_1 = D_2$ and $R_1 = R_2$ is shown in Figure 5.4.

At first glance, Theorem 5.5 describes achievable distortions given R_1 and R_2. The bounds (5.10)–(5.12) can be turned around to produce an achievable rate region given D_0, D_1, and D_2.[7] The shape of this region is shown in Figure 5.5, where δ is a function of (D_0, D_1, D_2) that represents the minimum excess (sum) rate necessary to achieve these distortions.

For non-Gaussian sources, no technique for precisely determining the achievable rate–distortion region is known. Zamir [219] has found inner- and outer-bounds to the achievable rate region for MDC of any continuous-valued memoryless source with squared-error distortion. Zamir's result is an extension of Shannon's bounds on rate–distortion functions (see [172, 59, 13]) to MDC. Let X be the generic random variable repre-

[7] The achievable rate region has the advantage of being in a lower dimensional space, and thus is easier to draw.

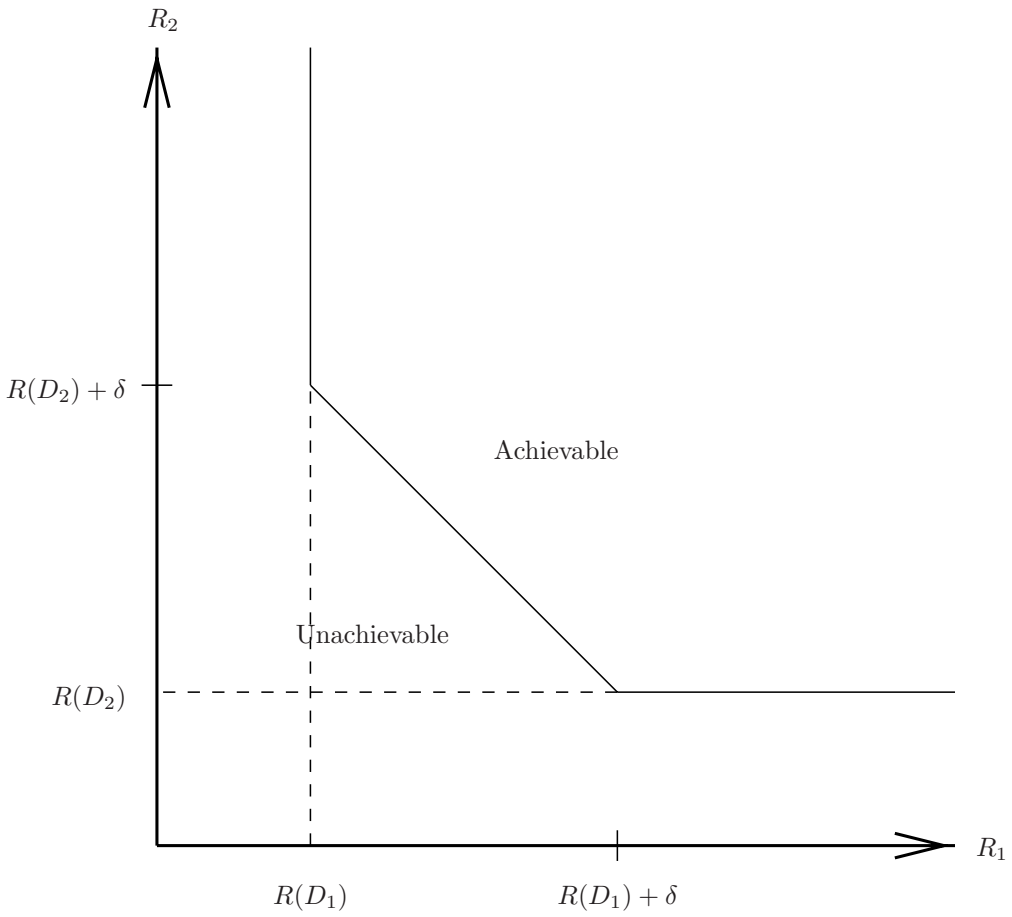


Figure 5.5: Achievable rates for multiple description coding of a unit variance memoryless Gaussian source with squared-error distortion. The minimum excess rate δ is a function of (D_0, D_1, D_2) and may be zero.

senting a memoryless source. If the variance and differential entropy of X are denoted σ_X^2 and h_X, respectively, then

$$\frac{1}{2}\log_2\left(\frac{\sigma_X^2}{D}\right) \geq R_X(D) \geq \frac{1}{2}\log_2\left(\frac{p_X}{D}\right),$$

where

$$p_X = \frac{1}{2\pi e}2^{2h_X}$$

is called the *entropy-power*. Thus we see that the rate–distortion function of X is bounded between the rate–distortion functions of a white Gaussian source with the same power and a white Gaussian source with the same differential entropy. For large classes of difference distortion measures and general source densities, the lower bound is asymptotically tight at high rates [125]. Let $\mathcal{R}(\sigma^2, D_0, D_1, D_2)$ denote the achievable rate region for a memoryless Gaussian source with variance σ^2 and distortion triple (D_0, D_1, D_2). Then the achievable rate region for X, denoted $\mathcal{R}(D_0, D_1, D_2)$, is bounded by

$$\mathcal{R}(\sigma_X^2, D_0, D_1, D_2) \subseteq \mathcal{R}_X(D_0, D_1, D_2) \subseteq \mathcal{R}(p_X, D_0, D_1, D_2).$$

The outer bound is asymptotically tight as (D_0, D_1, D_2) approaches $(0, 0, 0)$ along a straight line [219]. The situation is depicted by Figure 5.6.

5.2.2 Practical Codes

All of the results discussed thus far, for both binary and Gaussian sources, are non-constructive. They are bounds to performance when an infinitely long set of source symbols is coded.

5.2.2.1 Multiple description scalar quantization

A constructive approach is to start from the opposite end and devise schemes for multiple description coding of scalars. This approach was pioneered by Vaishampayan, who introduced multiple description scalar quantization (MDSQ) in [188]. Conceptually, MDSQ can be seen as the use of a pair of independent scalar quantizers to give two descriptions of a scalar source sample. As in all multiple description coding, the design challenge is to simultaneously provide good individual descriptions and a good joint description.

The simplest example is to have scalar quantizers with nested thresholds, as shown in Figure 5.7(a). Each quantizer outputs an index that can be used by itself to estimate the source sample. Using $Q_i : \mathbb{R} \to \{1, 2, \ldots, 6\}$, $i = 1, 2$, to denote the encoding map of quantizer i, the reconstruction knowing $Q_i(x) = k_i$ should be the centroid of the cell $Q_i^{-1}(k_i)$. The central decoder has both $Q_1(x) = k_1$ and $Q_2(x) = k_2$ and thus reconstructs to the centroid of the intersection cell $Q_2^{-1}(k_1) \cap Q_2^{-1}(k_2)$. In the example, the intersection cells are about half as big as the individual quantizer cells, so the central distortion is about a quarter of the side distortions. Asymptotically, if the side rates are $R_1 = R_2 = R$, then D_0, D_1, and D_2 are all $O(2^{-2R})$. This is optimal decay for D_1 and D_2, but far from optimal for D_0.

Recalling the discussion following Theorem 5.5, it should be possible to speed the decay of D_0 at the expense of slowing the decay of D_1 and/or D_2. Let n_i, $i = 1, 2$, denote the number of cells in quantizer Q_i. Let n_0 denote the number of intersections between cells of Q_1 and Q_2 that are nonempty. Notice that in Figure 5.7(a), $n_0 = n_1 + n_2 - 1$. When $n_1 = n_2 = n$, the exponential rate of decay of D_0 is changed only if n_0 grows faster than linearly with n. Accomplishing this requires something never seen in single-description scalar

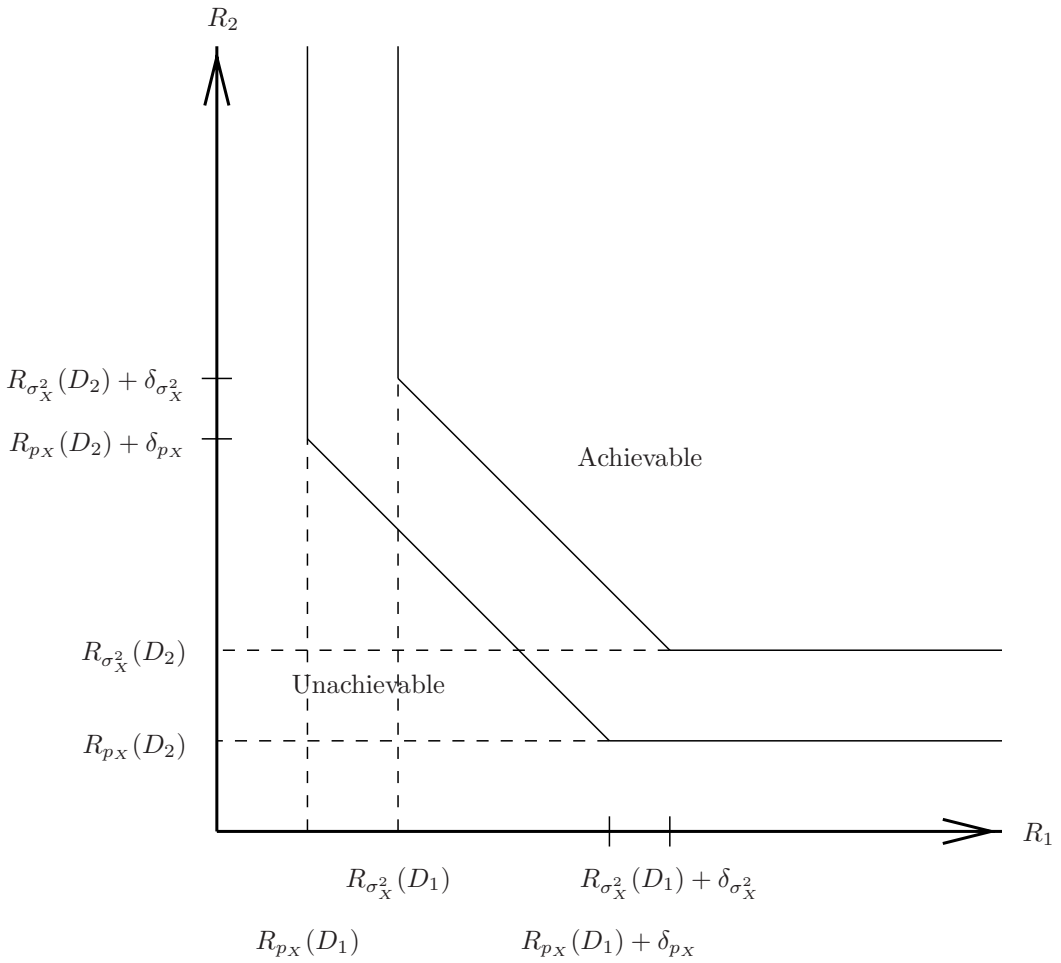


Figure 5.6: Inner- and outer-bounds for the achievable rate region for a memoryless non-Gaussian source with squared-error distortion. The achievable rate region for non-Gaussian source X contains the region for a Gaussian source with the same power and is contained in the region for a Gaussian source with the same differential entropy. $R_{\sigma^2}(\,\cdot\,)$ is the rate–distortion function of a memoryless Gaussian source with variance σ^2 and δ_{σ^2} is the minimum excess rate for this source at the chosen (D_0, D_1, D_2).

or vector quantization: disconnected partition cells. The maximum number of central decoder partition cells is $n_0 = n_1 n_2$. This occurs when each $Q_1^{-1}(k_1) \cap Q_2^{-1}(k_2)$ is nonempty, as shown in Figure 5.7(b). Quantizer Q_2 is individually poor. The asymptotic performance of this scheme with $R_1 = R_2 = R$ is at the opposite extreme of the previous example; $D_0 = O(2^{-4R})$, but at least one of D_1 and D_2 is $O(1)$.

Given a desired partitioning for the central encoder,[8] the crux of MDSQ design is the assignment of indices to the individual quantizers. Vaishampayan's main results in [188] are this observation and an idealized index assignment scheme that gives the optimal combined exponential decay rates for the central and side distortions. An MDSQ designed with Vaishampayan's "modified nested" index assignment is shown in Figure 5.7(c). In contrast to the MDSQ of Figure 5.7(b), the side distortions are approximately equal and the quantizers "refine each other" in a symmetric fashion. For a given number of side levels n, the central distortion is smaller—at the cost of higher side distortions—than for an MDSQ as in Figure 5.7(a).

Additional developments in this area include a design procedure for entropy-constrained codebooks [192], joint optimization of an orthogonal transform and MDSQ [9, 190], and a method for reducing granular distortion [191]. Applications of MDSQ are described in [102, 214, 189, 169].

5.2.2.2 Pairwise correlating transforms

A considerably different approach to MD coding was introduced by Wang, Orchard, and Reibman [201]. Instead of using MDSQ to produce two indices that describe the same quantity, the MD character is achieved with a linear transform that introduces correlation between a pair of random variables; quantization is treated as secondary.

Let X_1 and X_2 be independent zero-mean Gaussian random variables with variances $\sigma_1^2 > \sigma_2^2$. For conventional (single-description) source coding, there would be no advantage to using a linear transform prior to quantization. Assuming high-rate entropy-coded uniform quantization, the distortion at R bits per sample would be given by (see (1.10))

$$D_0 = \frac{\pi e}{6}\sigma_1\sigma_2 2^{-2R}.$$

This is the best single-description performance that can be obtained with scalar quantization.

Now suppose that the quantized versions of X_1 and X_2 are sent on channels 1 and 2, respectively, in a multiple description system. Side decoder 1 cannot estimate X_2, aside from using its mean. Thus

$$D_1 = \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R} + \sigma_2^2,$$

and similarly

$$D_2 = \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R} + \sigma_1^2.$$

Assume for the moment that each channel is equally likely to fail. Then instead of concerning ourselves with D_1 and D_2 separately, we will use the average distortion when one channel is lost:

$$\bar{D}_1 = \frac{1}{2}(D_1 + D_2) = \frac{1}{2}(\sigma_1^2 + \sigma_2^2) + \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R}. \tag{5.16}$$

$\bar{D}_1$ could be reduced if side decoder i had some information about X_j, $i \neq j$. This can be accomplished by transmitting not X_i's, but correlated transform coefficients. The simplest possibility, as proposed in [201],

[8] There is no "central encoder," but Q_1 and Q_2 effectively implement a quantizer with cells given by the intersections of the cells of the individual quantizers. An MDSQ could be viewed as a single quantizer that outputs two-tuple indices.

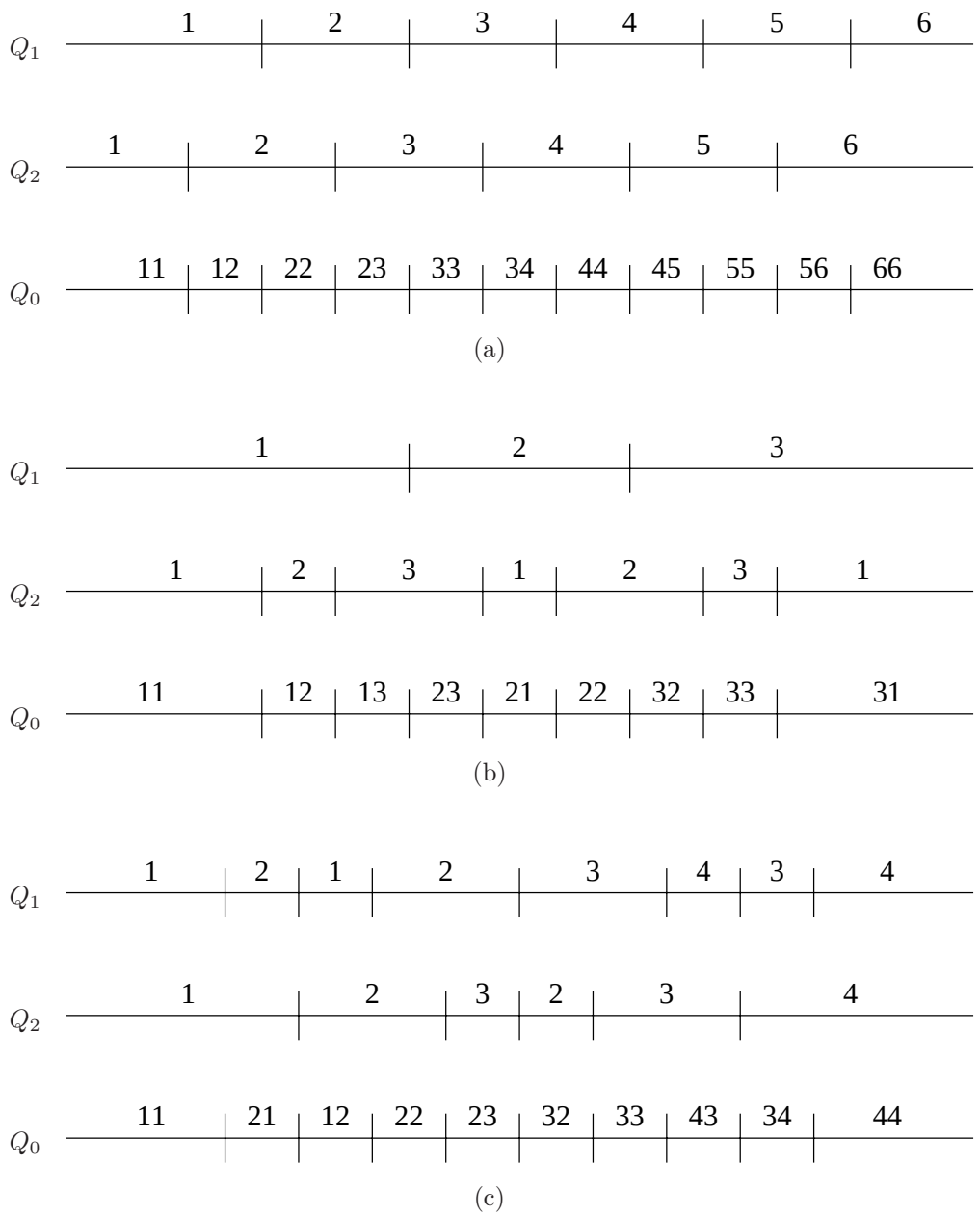


Figure 5.7: Three multiple description scalar quantizers. (a) The simplest form of MDSQ, with nested quantization thresholds. When $R_1 = R_2 = R$, all three distortions, D_0, D_1, and D_2, are $O(2^{-2R})$. (b) An MDSQ which minimizes D_0 for a given rate. Asymptotically, $D_0 = O(2^{-4R})$, but at least one of D_1 and D_2 must be $O(1)$. (c) An MDSQ based on Vaishampayan's "modified nested" index assignment [188]. This construction systematically trades off central and side distortions while maintaining optimal joint asymptotic decay of distortion with rate.

is to transmit quantized versions of Y_1 and Y_2 given by

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}.$$

Now the variances of Y_1 and Y_2 are both $(\sigma_1^2 + \sigma_2^2)/2$, so the central decoder performance is

$$D_0 = \frac{\pi e}{12}(\sigma_1^2 + \sigma_2^2)2^{-2R},$$

which is worse than the performance without the transform by a constant factor of[9]

$$\gamma = \frac{1}{2}(\sigma_1^2 + \sigma_2^2) \,/\, (\sigma_1 \sigma_2).$$

Now consider the situation at side decoder 1. The distortion is approximately equal to the quantization error plus the distortion in estimating Y_2 from Y_1. Since Y_1 and Y_2 are jointly Gaussian, $Y_2 \mid Y_1 = y_1$ is Gaussian and $E[Y_2 \mid Y_1 = y_1]$ is a linear function of y_1. Specifically, $Y_2 \mid Y_1 = y_1$ has mean $(\sigma_1^2 + \sigma_2^2)^{-1}(\sigma_1^2 - \sigma_2^2)y_1$ and variance $2(\sigma_1^2 + \sigma_2^2)^{-1}\sigma_1^2\sigma_2^2$. Thus

$$\bar{D}_1 \approx \frac{2\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} + \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R}. \tag{5.17}$$

Comparing (5.16) and (5.17), the constant term has been reduced by a factor of γ^2. By varying the transform, this method allows a trade-off between the constant factors in D_0 and $\bar{D}_1$.

The high-rate asymptotic behavior of the pairwise correlating transform method is not interesting because, independent of the choice of the transform, $D_0 = O(2^{-2R})$ and $D_1 = D_2 = O(1)$. However, in practice high rates are not necessarily important and constant factors can be very important. As a case in point: this method was introduced in the context of loss-resilient image coding, where it was shown to be successful [201]. Subsequently, this method was extended to nonorthogonal transforms [146]. The contribution of Section 5.3 is to generalize this method to communicating N variables over M channels, $M \leq N$, where the channel failures may be dependent and may have unequal probabilities. A conceptually very different method for $M > N$ is described in Section 5.4.

5.3 Statistical Channel Coding with Correlating Transforms

Channel codes provide protection against impairments by making certain transmitted symbols (or sequences) illegal. If the received symbol corresponds to an illegal transmitted symbol, an error is detected; and if furthermore one legal transmitted symbol is closer than all others, an attempt at correcting the error can be made.[10] The design of channel codes has always been in accordance with minimizing the probability of symbol error, but for communication subject to a smooth distortion measure, we may impose a much less stringent requirement with the hope of achieving satisfactory performance with less coding overhead. To this end, a channel code could be designed so that not even a single erasure can be *corrected*, but instead the effect of erasures is only *mitigated*. This is consistent with the philosophy of multiple description coding, where we

[9] This factor is like the coding gain of (1.11), but it is now the *increase* in distortion from coding correlated quantities.

[10] Note that in general, an error may be undetected and an attempted correction may be incorrect. Erasure channels are easier to handle than channels with errors because what is received is known to be correct. The only remaining problem is to guess at the erased information, if necessary.

generally do not aim for "full quality" when reconstructing from a proper subset of the descriptions. In fact, we could require "full quality," but this would necessitate high overhead.

This section describes a method for multiple description coding based on using transform coefficients or sets of transform coefficients as "descriptions." A square transform is used, so for coding an N-dimensional source at most N descriptions are produced. The method is a generalization of the pairwise correlating transforms of Orchard, Wang, Vaishampayan, and Reibman [146] to $N > 2$. In addition, a more complete analysis of the $N = 2$ case is provided. The reconstruction from a proper subset of the descriptions exploits a statistical correlation between transform coefficients. Thus this technique may be dubbed statistical channel coding for an erasure channel.

5.3.1 Intuition

Before introducing the general structure for multiple description transform coding (MDTC) with a square correlating transform, let us revisit the pairwise correlating transform method of Section 5.2.2.2. The limitations of this method, along with an insight of Vaishampayan, led to the work reported in [146]. This in turn led to the general theory presented here.

As in Section 5.2.2.2, let $X = [X_1, X_2]^T$ where X_1 and X_2 are independent zero-mean Gaussian random variables with variances $\sigma_1^2 > \sigma_2^2$. Let $\{e_1, e_2\}$ denote the standard basis of $\mathbb{R}^2$. Any level curve of the joint p.d.f. of X is an ellipse with principal axis aligned with e_1 and secondary axis aligned with e_2, as in the right side of Figure 1.6. Using the standard basis corresponds to representing X by $(\langle X,\, e_1\rangle,\, \langle X,\, e_2\rangle)$.

Now imagine that, in a multiple description scenario, uniform scalar quantized versions of $\langle X,\, e_1\rangle$ and $\langle X,\, e_2\rangle$ are used as descriptions. It was demonstrated in Section 5.2.2.2 that for a given total rate, the average of the side distortions $\bar{D}_1 = (D_1 + D_2)/2$ can be decreased in exchange for an increase in the the central distortion D_0 by using the representation

$$\left(\left\langle X,\, \frac{1}{\sqrt{2}}[1,1]^T\right\rangle,\, \left\langle X,\, \frac{1}{\sqrt{2}}[-1,1]^T\right\rangle\right). \tag{5.18}$$

But this is only a single operating point, whereas one would like to be able to trade off D_0 and $\bar{D}_1$ in a continuous manner.

We may recognize (5.18) as

$$\left(\langle X,\, G_{\pi/4}\, e_1\rangle,\, \langle X, G_{\pi/4}\, e_2\rangle\right),$$

where G_θ is a Givens rotation of angle θ (see Definition 4.1). Then the natural extension is to consider all representations of the form

$$\left(\langle X,\, G_\theta\, e_1\rangle,\, \langle X,\, G_\theta\, e_2\rangle\right),\quad 0 \le \theta \le \pi/4.$$

This indeed creates a continuous trade-off between D_0 and $\bar{D}_1$. However, it has an undesirable asymmetry. For $0 \le \theta < \pi/4$, the side distortions are not equal. It is an easy exercise to calculate D_1 and D_2, but instead let us look geometrically at why they are unequal. D_1 is the variation of X which is *not* captured by $\langle X,\, G_\theta\, e_1\rangle$, or the variation perpendicular to $G_\theta\, e_1$.[11] Similarly, D_2 is the variation perpendicular to $G_\theta\, e_2$. Now since $G_\theta\, e_1$ and $G_\theta\, e_2$ are not symmetrically situated with respect to the p.d.f. of X (except for $\theta = \pi/4$), D_1 and D_2 are unequal.

[11] We are neglecting quantization error at this point because D_1 and D_2 are equally affected by quantization.

This in itself does not imply that the scheme can be improved, but since we are trying to have two channels of equal importance[12] we might expect equal side distortions. Based on the geometric observation above, it makes sense to represent X by

$$(\langle X,\, G_\theta\, e_1\rangle,\ \langle X,\, G_{-\theta}\, e_1\rangle)\,,\quad 0<\theta<\pi/2. \tag{5.19}$$

Furthermore, in order to be capturing most of the principal component of the source, the basis should be skewed toward e_1, so θ should be between 0 and some maximum value $\theta_{\max} < \pi/2$. This yields $D_1 = D_2$, but introduces a new problem.

The representation of X by (5.19) is (for $\theta \neq \pi/4$) a nonorthogonal basis expansion. The uniform scalar quantization of such a representation produces non-square partition cells.[13] These partition cells have higher normalized second moments than square cells, and thus are undesirable [32]. The insight attributed to Vaishampayan is that a correlating transform can be applied *after* quantization has been performed in an orthogonal basis representation. This allows the introduction of correlation (more generally than with an orthogonal transform) to coexist with square partition cells. The advantage of this approach over the original pairwise correlating method was shown in [146].

If we do not want the two descriptions to be equally important, for example if they are sent over links with different failure probabilities, then we expect the optimal representation to be different from (5.19). Recalling $\sigma_1 > \sigma_2$, we would expect the description over the more reliable link to be closer to the e_1 direction, as this captures most of the energy of the source. This sort of intuition is vindicated by the general framework and optimization that follow.

5.3.2 Design

It is time to attach a bit more formality to multiple description transform coding (MDTC) with correlating transforms. Let $\{X_k\}$ be an i.i.d. sequence of zero-mean jointly Gaussian vectors in $\mathbb{R}^N$ with a known distribution. Without loss of generality, we may assume that the components of X_k are independent with variances $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_N^2$ because we could use a Karhunen–Loève transform at the encoder.

MDTC refers to processing each source vector x as follows:

1. x is quantized with an unbounded uniform scalar quantizer with step size Δ; *i.e.*, $x_{q_i} = [x_i]_\Delta$, where $[\,\cdot\,]_\Delta$ denotes rounding to the nearest multiple of Δ.

2. The vector $x_q = [x_{q_1},\, x_{q_2},\, \ldots,\, x_{q_N}]^T$ is transformed with an invertible, discrete transform $\widehat{T} : \Delta\mathbb{Z}^N \to \Delta\mathbb{Z}^N$, $y = \widehat{T}(x_q)$. $\widehat{T}$ is within a certain quasilinear class described below.

3. The components of y are placed into M sets (in an *a priori* fixed manner). These sets will be sent over M different channels.

4. The M sets of coefficients are independently entropy coded. To improve the efficiency of this stage, many coefficients *within one channel* may be coded jointly.

[12] "Equal importance" comes from the equal weighting of D_1 and D_2 in $\bar{D}_1$. Later the weights will be arbitrary.
[13] In higher dimensions, non-hypercubic cells.

The transform $\widehat{T}$ is a discrete transform derived from a linear transform T, with $\det T = 1$. First T is factored into matrices with unit diagonals and nonzero off-diagonal elements only in one row or column: $T = T_1 T_2 \cdots T_k$. The discrete transform is then given by

$$\widehat{T}(x_q) = \left[T_1 \left[T_2 \ldots [T_k x_q]_\Delta\right]_\Delta\right]_\Delta .$$

The "lifting" construction of the transform ensures that $\widehat{T}$ is invertible on $\Delta\mathbb{Z}^N$. See Appendix 5.A for details.

The coding structure presented here is a generalization of the method proposed by Orchard, Wang, Vaishampayan, and Reibman [146]. They considered coding of two variables with the transform

$$T = \begin{bmatrix} 1 & \beta \\ -(2\beta)^{-1} & 1/2 \end{bmatrix}, \tag{5.20}$$

approximated by

$$\widehat{T}(x) = \left[\begin{bmatrix} 1 & 0 \\ -(2\beta)^{-1} & 1 \end{bmatrix} \left[\begin{bmatrix} 1 & \beta \\ 0 & 1 \end{bmatrix} [x]_\Delta\right]_\Delta\right]_\Delta . \tag{5.21}$$

The mysterious form of (5.20) provided the initial motivation for this work.

The analysis and optimization that follow are based on a high-rate (or fine-quantization, small Δ) assumption. In particular, the following assumptions or approximations are used:

- The scalar entropy of $y = \widehat{T}([x]_\Delta)$ is the same as that of $[Tx]_\Delta$.
- The correlation structure of y is unaffected by the quantization; *i.e.*,

$$E[yy^T] \;=\; E[\widehat{T}(x)\widehat{T}(x)^T] \;=\; E[Tx(Tx)^T].$$

- When one or more components of y are lost, the distortion is dominated by the effect of the erasure, so quantization can be ignored.

These approximations facilitate the analytical treatment of the design of T, or at least the generation of an optimization criterion. A careful accounting of coarse quantization effects would probably make symbolic optimization impossible.

When all the components of y are received, the reconstruction process is to (exactly) invert the transform $\widehat{T}$ to get $\hat{x} = x_q$. The distortion is precisely the quantization error from Step 1. If some components of y are lost, they are estimated from the received components using the correlation introduced by the transform $\widehat{T}$. The estimate $\hat{x}$ is then generated by inverting the transform as before.

Denote the variances of the components of x by σ_1^2, σ_2^2, …, σ_N^2 and denote the correlation matrix of x by $R_x = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$. Under fine quantization approximations, assume the correlation matrix of y is $R_y = TR_xT^T$. By renumbering the variables if necessary, assume that y_1, y_2, …, $y_{N-\ell}$ are received and $y_{N-\ell+1}$, …, y_N are lost. Partition y into "received" and "not received" portions as $y = [\tilde{y}_\mathrm{r},\ \tilde{y}_\mathrm{nr}]^T$ where $\tilde{y}_\mathrm{r} = [y_1,\ y_2,\ \ldots,\ y_{N-\ell}]^T$ and $\tilde{y}_\mathrm{nr} = [y_{N-\ell+1},\ \ldots,\ y_{N-1},\ y_N]^T$. The minimum MSE estimate of x given $\tilde{y}_\mathrm{r}$ is $E[x \mid \tilde{y}_\mathrm{r}]$, which has a simple closed form because x is a jointly Gaussian vector. Using the linearity of the expectation operator gives the following sequence of calculations:

$$\begin{aligned} \hat{x} \;&=\; E[x \mid \tilde{y}_\mathrm{r}] \;=\; E[T^{-1}Tx \mid \tilde{y}_\mathrm{r}] \;=\; T^{-1}E[Tx \mid \tilde{y}_\mathrm{r}] \\ &=\; T^{-1}E\left[\begin{bmatrix} \tilde{y}_\mathrm{r} \\ \tilde{y}_\mathrm{nr} \end{bmatrix} \middle| \tilde{y}_\mathrm{r}\right] \;=\; T^{-1}\begin{bmatrix} \tilde{y}_\mathrm{r} \\ E[\tilde{y}_\mathrm{nr} \mid \tilde{y}_\mathrm{r}] \end{bmatrix}. \end{aligned} \tag{5.22}$$

If the correlation matrix of y is partitioned compatibly with the partition of y as

$$R_y = TR_xT^T = \begin{bmatrix} R_1 & B \\ B^T & R_2 \end{bmatrix},$$

then $\tilde{y}_{\rm nr} \mid \tilde{y}_{\rm r}$ is Gaussian with mean $B^T R_1^{-1}\tilde{y}_{\rm r}$ and correlation matrix $R_2 - B^T R_1^{-1} B$.[14] Thus $E[\tilde{y}_{\rm nr} \mid \tilde{y}_{\rm r}] = B^T R_1^{-1}\tilde{y}_{\rm r}$ and the reconstruction is

$$\hat{x} = T^{-1} \begin{bmatrix} I \\ B^T R_1^{-1} \end{bmatrix} \tilde{y}_{\rm r}. \tag{5.23}$$

5.3.2.1 Optimization criterion

The choice of T, of course, determines the performance of the system. This section develops the relationships between the transform, rates, and distortions necessary to design T.

Estimating the rate is straightforward. Since the quantization is assumed to fine, y_i is approximately the same as $[(Tx)_i]_\Delta$, *i.e.*, a uniformly quantized Gaussian random variable. If y_i is treated as a Gaussian random variable with power $\sigma_{y_i}^2 = (R_y)_{ii}$ quantized with bin width Δ, the entropy of the quantized coefficient is approximately [34, Ch. 9]

$$H(y_i) \approx \frac{1}{2}\log 2\pi e\sigma_{y_i}^2 - \log\Delta = \frac{1}{2}\log\sigma_{y_i}^2 + \frac{1}{2}\log 2\pi e - \log\Delta = \frac{1}{2}\log\sigma_{y_i}^2 + k_\Delta,$$

where $k_\Delta = (\log 2\pi e)/2 - \log\Delta$ and all logarithms are base-two. Notice that k_Δ depends only on Δ. We thus estimate the rate per component as

$$R = \frac{1}{N}\sum_{i=1}^{N} H(y_i) = k_\Delta + \frac{1}{2N}\log\prod_{i=1}^{N}\sigma_{y_i}^2. \tag{5.24}$$

The minimum rate occurs when $\prod_{i=1}^N \sigma_{y_i}^2 = \prod_{i=1}^N \sigma_i^2$ and at this rate the components of y are uncorrelated. Interestingly, $T = I$ is not the only transform which achieves the minimum rate. In fact, an arbitrary split of the total rate among the different components of y is possible. This is a justification for using a total rate constraint in our following analyses. However, we will pay particular attention to the case where the rates sent across each channel are equal.

We now turn to the distortion, and first consider the average distortion due only to quantization. Since the quantization noise is approximately uniform, this distortion is $\Delta^2/12$ for each component. Thus the per component distortion when no components are erased is given by

$$D_0 = \frac{\Delta^2}{12} \tag{5.25}$$

and is independent of T.

For the case when $\ell > 0$ components are lost, the distortion computation was almost completed in the development of (5.23). Let $\eta = \tilde{y}_{\rm nr} - E[\tilde{y}_{\rm nr} \mid \tilde{y}_{\rm r}]$, which is Gaussian with zero mean and correlation matrix $A = R_2 - B^T R_1^{-1} B$. η is the error in predicting $\tilde{y}_{\rm nr}$ from $\tilde{y}_{\rm r}$ and hence is the error caused by the lost coefficients.

[14] This correlation matrix is the Schur complement of R_1 in R_y [99].

However, because we have used a nonorthogonal transform, we must return to the original coordinates using T^{-1} in order to compute the distortion. Substituting $\tilde{y}_{\rm nr} - \eta$ for $E[\tilde{y}_{\rm nr} \mid \tilde{y}_{\rm r}]$ in (5.22) gives

$$\hat{x} \;=\; T^{-1} \begin{bmatrix} \tilde{y}_{\rm r} \\ \tilde{y}_{\rm nr} - \eta \end{bmatrix} \;=\; x + T^{-1} \begin{bmatrix} 0 \\ -\eta \end{bmatrix},$$

so

$$\|x - \hat{x}\|^2 \;=\; \left\| T^{-1} \begin{bmatrix} 0 \\ -\eta \end{bmatrix} \right\|^2 \;=\; \eta^T U^T U \eta, \tag{5.26}$$

where U is the last ℓ columns of T^{-1}. Finally,

$$E\|x - \hat{x}\|^2 \;=\; E[\operatorname{tr} \eta^T U^T U \eta] \;=\; E[\operatorname{tr} \eta\eta^T U^T U] \;=\; \operatorname{tr} A U^T U. \tag{5.27}$$

With M sets of coefficients sent over M channels, there are $2^M - 1$ nontrivial reconstructions, each with a potentially distinct distortion. For the sake of optimization, it is useful to have a scalar distortion measure. Assign to each channel a state $s_i \in \{0, 1\}$ to denote whether the channel is received (1) or not received (0). For any system state $S = s_1 \times s_2 \times \cdots s_M$, the distortion will be denoted $D_{(s_1,s_2,\ldots,s_M)}$ and can be computed with (5.27). We will consider weighted distortions of the form

$$\bar{D} = \sum_{s_i \in \{0,1\}, 1 \le i \le M} \alpha_{(s_1,s_2,\ldots,s_M)} D_{(s_1,s_2,\ldots,s_M)}, \tag{5.28}$$

where

$$\sum_{s_i \in \{0,1\}, 1 \le i \le M} \alpha_{(s_1,s_2,\ldots,s_M)} = 1.$$

In the simplest case, $\alpha_{(s_1,s_2,\ldots,s_M)}$ could be the probability of state $(s_1, s_2, \ldots, s_M)$. In this case, $\bar{D}$ is the overall average MSE. Other meaningful choices are available. For example, if a certain minimum quality is required when k of M channels are received, then the $\binom{M}{k}$ states with $\sum s_i = k$ can be assigned equal weights of $\binom{M}{k}^{-1}$ with the remaining states having no weight. In this case the optimization will make the distortion equal in each of the states with k channels received and this distortion will be an upper bound for the distortion when more than k channels are received.

The problem will be to minimize $\bar{D}$ subject to a constraint on R. The expressions given in this section can be used to numerically determine transforms to realize this goal. Analytical solutions are possible in certain special cases. Some of these are outlined in the following sections.

5.3.2.2 General solution for two variables

Let us now apply the analysis of the previous section to find the best transforms for sending $N = 2$ variables over $M = 2$ channels. In the most general situation, channel outages may have unequal probabilities and may be dependent. Suppose the probabilities of the system states are given by Table 5.1. Let

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

normalized so that $\det T = 1$. Then

$$T^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

		Channel 1	
		not received	received
Channel 2	not received	$p_{0,0}$	$p_{1,0}$
	received	$p_{0,1}$	$p_{1,1}$

Table 5.1: Probabilities of system states in optimal transform determination for MDTC of two variables over two channels.

and

$$R_y \;=\; TR_xT^T \;=\; \begin{bmatrix} a^2\sigma_1^2+b^2\sigma_2^2 & ac\sigma_1^2+bd\sigma_2^2 \\ ac\sigma_1^2+bd\sigma_2^2 & c^2\sigma_1^2+d^2\sigma_2^2 \end{bmatrix}.$$

By (5.24), the rate per component is given by

$$R \;=\; k_\Delta + \frac{1}{4}\log(R_y)_{11}(R_y)_{22} \;=\; k_\Delta + \frac{1}{4}\log(a^2\sigma_1^2+b^2\sigma_2^2)(c^2\sigma_1^2+d^2\sigma_2^2). \tag{5.29}$$

Minimizing (5.29) over transforms with determinant one gives a minimum possible rate of

$$R^* = k_\Delta + \frac{1}{4}\log\sigma_1^2\sigma_2^2. \tag{5.30}$$

As in [146], we refer to $\rho = R - R^*$ as the *redundancy*; *i.e.*, the price we pay in rate in order to potentially reduce the distortion when there are erasures. Subtracting (5.30) from (5.29) gives

$$\rho = \frac{1}{4}\log\frac{(a^2\sigma_1^2+b^2\sigma_2^2)(c^2\sigma_1^2+d^2\sigma_2^2)}{\sigma_1^2\sigma_2^2}.$$

Now we compute the constituents of (5.28). Two terms are obvious: When both channels are received, the distortion (due to quantization only) is $D_{(1,1)} = \Delta^2/12$; and if neither channel is received, the distortion is $D_{(0,0)} = (\sigma_1^2+\sigma_2^2)/2$. These terms do not depend on T and thus do not affect the optimization. The remaining cases require the application of the results of the previous section. Substituting in (5.27),

$$\begin{aligned} D_{(1,0)} &= \frac{1}{2}E\left[\|x-\hat{x}\|^2 \;\middle|\; y_1 \text{ is received and } y_2 \text{ is not}\right] \\ &= \frac{1}{2}\underbrace{\left\|\begin{bmatrix} -b \\ a \end{bmatrix}\right\|^2}_{(U^TU)_{1,1}} \cdot \underbrace{\left((R_y)_{22} - \frac{(R_y)_{12}^2}{(R_y)_{11}}\right)}_{A_{1,1}} \\ &= \frac{1}{2}(a^2+b^2)\cdot\frac{\sigma_1^2\sigma_2^2}{a^2\sigma_1^2+b^2\sigma_2^2}, \end{aligned}$$

where we have used $\det T = ad - bc = 1$ in the simplification. Similarly,

$$D_{(0,1)} = \frac{1}{2}(c^2+d^2)\frac{\sigma_1^2\sigma_2^2}{c^2\sigma_1^2+d^2\sigma_2^2}.$$

The overall average distortion is

$$\bar{D} = \left[p_{1,1}\frac{\Delta^2}{12} + p_{0,0}\frac{1}{2}(\sigma_1^2+\sigma_2^2)\right] + \underbrace{\left[p_{1,0}\frac{(a^2+b^2)\sigma_1^2\sigma_2^2}{2(a^2\sigma_1^2+b^2\sigma_2^2)} + p_{0,1}\frac{(c^2+d^2)\sigma_1^2\sigma_2^2}{2(c^2\sigma_1^2+d^2\sigma_2^2)}\right]}_{\bar{D}_1}, \tag{5.31}$$

where the first bracketed term is independent of T. Thus our optimization problem is to minimize the second bracketed term, $\bar{D}_1$, for a given redundancy ρ.[15]

First note that if the source p.d.f. is circularly symmetric, *i.e.*, $\sigma_1 = \sigma_2$, then $D_{(1,0)} = D_{(0,1)} = \sigma_1^2/2$ independent of T. Henceforth we assume $\sigma_1 > \sigma_2$.

For a given value of ρ, the admissible transforms are simultaneous solutions of

$$\begin{aligned}(a^2\sigma_1^2 + b^2\sigma_2^2)(c^2\sigma_1^2 + d^2\sigma_2^2) &= \sigma_1^2\sigma_2^2 2^{4\rho}, \\ ad - bc &= 1.\end{aligned} \tag{5.32}$$

There are several branches of solutions.[16] First we may consider the case where $a = 0$. In this case, the transforms are of the form $\begin{bmatrix} 0 & b \\ 1/b & d \end{bmatrix}$. This branch is not particularly useful because one channel carries only the *lower* variance component of x. Another special case is where $c = 0$. These transforms are of the form $\begin{bmatrix} a & b \\ 0 & 1/a \end{bmatrix}$, and are not useful for the same reason. Now assuming $a \neq 0$ and $c \neq 0$, we may substitute $d = (1 + bc)/a$ into (5.32) and rearrange to get

$$a^4 \left(\frac{\sigma_1}{\sigma_2}\right)^4 + \frac{a^2}{c^2}\left(b^2c^2 + (1+bc)^2 - 2^{4\rho}\right)\left(\frac{\sigma_1}{\sigma_2}\right)^2 + \frac{b^2(1+bc)^2}{c^2} = 0.$$

Solving this quadratic in a^2 and choosing signs appropriately gives[17]

$$a = \frac{1}{2c}\frac{\sigma_2}{\sigma_1}\left(\sqrt{2^{4\rho}-1} + \sqrt{2^{4\rho} - 1 - 4bc(bc+1)}\right).$$

When this value of a is used, $\bar{D}_1$ depends only on the product $b \cdot c$, not on the individual values of b and c. The optimal value of bc is given by

$$(bc)_{\text{optimal}} = -\frac{1}{2} + \frac{1}{2}\left(\frac{p_{1,0}}{p_{0,1}} - 1\right)\left(\left(\frac{p_{1,0}}{p_{0,1}} + 1\right)^2 - 4\left(\frac{p_{1,0}}{p_{0,1}}\right)2^{-4\rho}\right)^{-1/2},$$

which is interestingly independent of σ_1/σ_2.

It is easy to check that $(bc)_{\text{optimal}}$ ranges from -1 to 0 as $p_{1,0}/p_{1,0}$ ranges from 0 to ∞. The limiting behavior can be explained as follows: Suppose $p_{1,0} \gg p_{0,1}$, *i.e.*, Channel 1 is much more reliable than Channel 2. Since $(bc)_{\text{optimal}}$ approaches 0, ad must approach 1, and hence one optimally sends x_1 (the larger variance component) over Channel 1 (the more reliable channel), and vice-versa. This is the intuitive, layered solution. The multiple description approach is most useful when the channel failure probabilities are comparable, but this demonstrates that the multiple description framework subsumes layered coding.

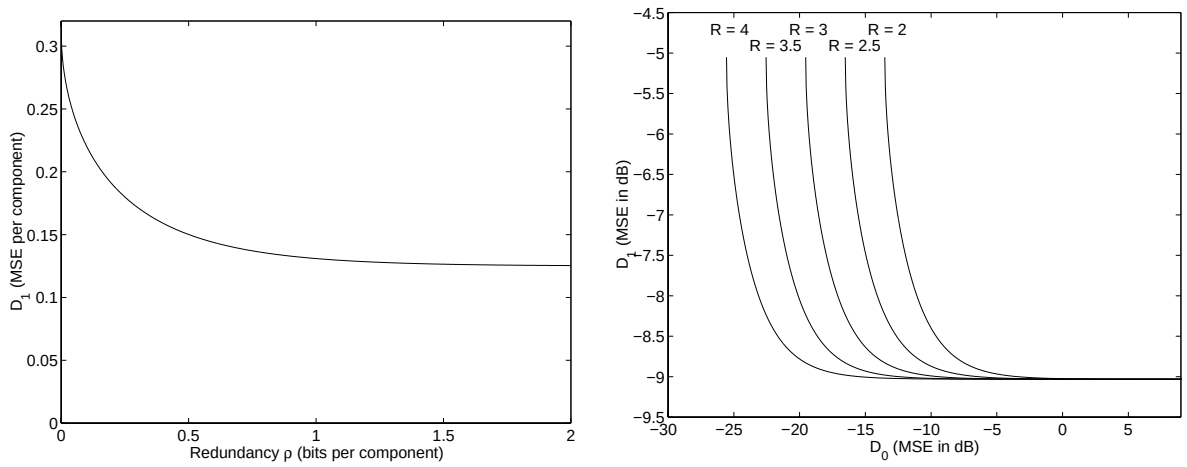
Equal channel failure probabilities If $p_{1,0} = p_{0,1}$, then $(bc)_{\text{optimal}} = -1/2$, independent of ρ. The optimal set of transforms is described by

$$\begin{aligned} a &\neq 0 \text{ (but otherwise arbitrary)}, & b &= \pm a\sigma_2^{-1}\sigma_1\left(2^{2\rho} + \sqrt{2^{4\rho}-1}\right)^{-1}, \\ c &= -1/2b, & d &= 1/2a, \end{aligned} \tag{5.33}$$

[15] The previous usage of $\bar{D}_1$ is generalized here to a *weighted* average of distortions with one erasure.

[16] The multiplicity of solutions is not present in conventional transform coding with orthogonal linear transforms, where for minimum rate ($\rho = 0$) the optimal transform is unique up to reflections and permutations of coordinates. Uses for the extra design freedom in using discrete transforms are discussed further in Appendix 5.B.

[17] In the solution for a^2, one of the two branches of the quadratic is valid. Then in the square root of this expression we arbitrarily choose $a \geq 0$ since the sign of a does not affect ρ or $\bar{D}_1$.



(a) (b)

Figure 5.8: Optimal R-D_0-D_1 trade-offs for $\sigma_1 = 1$, $\sigma_2 = 0.5$: (a) Relationship between redundancy ρ and D_1; (b) Relationship between D_0 and D_1 for various rates.

and using a transform from this set gives

$$D_1 \;=\; D_{(1,0)} \;=\; D_{(0,1)} \;=\; \frac{1}{2}\sigma_2^2 + \frac{\sigma_1^2 - \sigma_2^2}{4 \cdot 2^{2\rho}\left(2^{2\rho} + \sqrt{2^{4\rho} - 1}\right)}. \tag{5.34}$$

This relationship is plotted in Figure 5.8(a). Notice that, as expected, D_1 starts at a maximum value of $(\sigma_1^2 + \sigma_2^2)/4$ and asymptotically approaches a minimum value of $\sigma_2^2/2$. By combining (5.24), (5.25), and (5.34), one can find the relationship between R, D_0, and D_1. For various values of R, the trade-off between D_0 and D_1 is plotted in Figure 5.8(b).

The solution for the optimal set of transforms (5.33) has an "extra" degree of freedom which does not affect the ρ vs. D_1 performance. Fixing $a = 1$ gives the transforms suggested in [146] and allows $\widehat{T}(\,\cdot\,)$ to be implemented with two lifting steps instead of three. This degree of freedom can also be used to control the partitioning of the rate between channels. Other uses are described in Appendix 5.B.

Geometric interpretation The transmitted representation of x is given by $y_1 = \langle x,\, \varphi_1 \rangle$ and $y_2 = \langle x,\, \varphi_2 \rangle$, where $\varphi_1 = [a,\, b]^T$ and $\varphi_2 = [c,\, d]^T$. In order to gain some insight into the vectors φ_1 and φ_2 that result in an optimal transform, let us neglect the rate and distortion that are achieved, and simply consider the transforms described by $ad = 1/2$ and $bc = -1/2$. We can show that φ_1 and φ_2 form the same (absolute) angles with the positive x_1-axis (see Figure 5.9(a)). For convenience, suppose $a > 0$ and $b < 0$. Then $c, d > 0$. Let θ_1 and θ_2 be the angles by which φ_1 and φ_2 are below and above the positive x_1-axis, respectively. Then $\tan\theta_1 = -b/a = d/c = \tan\theta_2$. If we assume $\sigma_1 > \sigma_2$, then the maximum angle (for $\rho = 0$) is $\arctan(\sigma_1/\sigma_2)$ and the minimum angle (for $\rho \to \infty$) is zero. This has the nice interpretation of emphasizing x_1 over x_2—because it has higher variance—as the coding rate is increased (see Figure 5.9(b)).

Optimal transforms that give balanced rates The transforms of [146] do not give channels with equal rate (or, equivalently, power). In practice, this can be remedied through time-multiplexing. An alternative is to

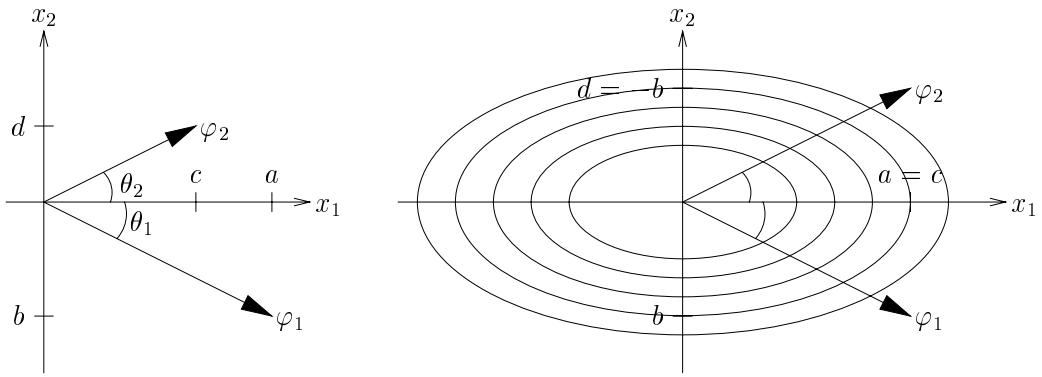


Figure 5.9: Geometric interpretations. (a) When $\sigma_1 > \sigma_2$, the optimality condition ($ad = 1/2$, $bc = -1/2$) is equivalent to $\theta_1 = \theta_2 < \theta_{\max} = \arctan(\sigma_1/\sigma_2)$. (b) If in addition to the optimality condition we require the output streams to have equal rate, the analysis vectors are symmetrically situated to capture the dimension with greatest variation. At $\rho = 0$, $\theta_1 = \theta_2 = \theta_{\max}$; as $\rho \to \infty$, φ_1 and φ_2 close on the x_1-axis.

use the "extra" degree of freedom to make $R_1 = R_2$. Doing this is equivalent to requiring $|a| = |c|$ and $|b| = |d|$, and yields

$$a = \pm\sqrt{\frac{1}{2}\frac{\sigma_2}{\sigma_1}\left(2^{2\rho} + \sqrt{2^{4\rho} - 1}\right)} \tag{5.35}$$

and

$$b = \pm\frac{1}{2a} = \pm\sqrt{\frac{1}{2}\frac{\sigma_1}{\sigma_2}\left(2^{2\rho} - \sqrt{2^{4\rho} - 1}\right)}.$$

These balanced-rate transforms will be used in the applications sections and in the development of a cascade structure. The notation

$$T_\alpha = \begin{bmatrix} \alpha & (2\alpha)^{-1} \\ -\alpha & (2\alpha)^{-1} \end{bmatrix} \tag{5.36}$$

will be used. When there are no erasures, the reconstruction uses

$$T_\alpha^{-1} = \begin{bmatrix} (2\alpha)^{-1} & -(2\alpha)^{-1} \\ \alpha & \alpha \end{bmatrix}. \tag{5.37}$$

For when there are erasures, evaluating (5.23) gives optimal estimates

$$\hat{x} = \frac{2\alpha}{4\alpha^4\sigma_1^2 + \sigma_2^2} \begin{bmatrix} 2\alpha^2\sigma_1^2 \\ \sigma_2^2 \end{bmatrix} y_1 \tag{5.38}$$

and

$$\hat{x} = \frac{2\alpha}{4\alpha^4\sigma_1^2 + \sigma_2^2} \begin{bmatrix} -2\alpha^2\sigma_1^2 \\ \sigma_2^2 \end{bmatrix} y_2 \tag{5.39}$$

for reconstructing from y_1 and y_2, respectively.

5.3.2.3 Techniques for two channels

Hindsight reveals that the simplest generalization of sending two variables over two channels is to keep the number of channels the same, but to increase the number of variables N. The significance of having two

channels is that the transform coefficients must be placed in two sets. The distortion expression (5.28) has just $2^2 = 4$ terms—not 2^N terms—because each set is either received in full or lost in full.

The general solution for sending two variables over two channels can be used to derive methods for sending more variables over two channels. These methods use at most $\lfloor N/2 \rfloor$ transforms of size 2×2 in parallel and thus have complexity that is only linear in the number of variables. For simplicity it is assumed that the channels are equally likely to fail.

Three variables The natural first step is to consider the transmission of three variables. Suppose y_1 is transmitted on Channel 1 and (y_2, y_3) is transmitted on Channel 2. We could start from first principles as before, designing a 3×3 transform with determinant 1 to minimize the distortion given by (5.27)–(5.28). The eight free parameters make this a difficult optimization. A much easier way to determine the optimal performance is to first reduce the number of parameters without risking a loss in performance. It turns out that it is sufficient to send one of the original variables as y_3 and to use an optimal 2×2 transform to produce y_1 and y_2. This assertion is formalized by the following theorem:

Theorem 5.6 *Consider multiple description transform coding where y_1 is sent on Channel 1 and (y_2, y_3) is sent on Channel 2. To minimize the average side distortion D_1 with an upper bound on redundancy ρ, it is sufficient to optimize over transforms of the form*

$$\begin{bmatrix} \widetilde{T} & 0_{1\times 2} \\ 0_{2\times 1} & 1 \end{bmatrix} P, \tag{5.40}$$

with $\widetilde{T} \in \mathbb{R}^{2\times 2}$, $\det \widetilde{T} = 1$, and P a permutation matrix.

Proof: See Appendix 5.C.1. □

Theorem 5.6 reduces the number of design parameters from eight to three and makes the design of an optimal transform a simple application of the results of Section 5.3.2.2. A transform of the form (5.40) shuffles the input variables and then correlates the first two. Since the order of the elements in a correlated pair does not matter, the permutation can be limited to one of the following three:

$$P_1 = I, \quad P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \text{and} \quad P_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Let us consider a generic choice among the three by assuming that the outputs of the permutation have variances ς_1^2, ς_2^2, and ς_3^2; *i.e.*, applying the permutation represented by P to $(\sigma_1^2, \sigma_2^2, \sigma_3^2)$ gives $(\varsigma_1^2, \varsigma_2^2, \varsigma_3^2)$. Recall the original ordering of the variances $(\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2)$ and notice that the three permutations under consideration preserve the ordering of the first two components: $\varsigma_1^2 \geq \varsigma_2^2$.

The component with variance ς_3^2 is sent over Channel 2 without any channel protection. (It is uncorrelated with the other components.) Since Channel 2 is lost half of the time and the distortion is measured per component, this component contributes $\varsigma_3^2/6$ to D_1, independent of $\widetilde{T}$. Now the optimization of $\widetilde{T}$ in (5.40) is precisely the problem of the previous section. Thus $\widetilde{T}$ can be chosen in the form of (5.36) and

$$D_1 = \frac{1}{6}\varsigma_3^2 + \frac{1}{3}\varsigma_2^2 + \frac{\varsigma_1^2 - \varsigma_2^2}{6 \cdot 2^{3\rho}\left(2^{3\rho} + \sqrt{2^{6\rho} - 1}\right)}. \tag{5.41}$$

The second and third terms of (5.41) come from evaluating (5.34) and rescaling the redundancy and distortion to account for the change from two to three components.

Now we can choose the best permutation; *i.e.*, the permutation yielding the lowest side distortion. The three permutations give the following distortions, respectively:

$$\begin{aligned}
(D_1)_1 &= \frac{1}{6}\sigma_3^2 + \frac{1}{3}\sigma_2^2 + \frac{\sigma_1^2 - \sigma_2^2}{6 \cdot 2^{3\rho}\left(2^{3\rho} + \sqrt{2^{6\rho} - 1}\right)}, \\
(D_1)_2 &= \frac{1}{6}\sigma_2^2 + \frac{1}{3}\sigma_3^2 + \frac{\sigma_1^2 - \sigma_3^2}{6 \cdot 2^{3\rho}\left(2^{3\rho} + \sqrt{2^{6\rho} - 1}\right)}, \\
(D_1)_3 &= \frac{1}{6}\sigma_1^2 + \frac{1}{3}\sigma_3^2 + \frac{\sigma_2^2 - \sigma_3^2}{6 \cdot 2^{3\rho}\left(2^{3\rho} + \sqrt{2^{6\rho} - 1}\right)}.
\end{aligned}$$

The best permutation is P_2 because

$$(D_1)_1 - (D_1)_2 = \frac{1}{6}(\sigma_2^2 - \sigma_3^2)\left(1 - \frac{1}{2^{3\rho}\left(2^{3\rho} + \sqrt{2^{6\rho} - 1}\right)}\right) \geq 0$$

and

$$(D_1)_3 - (D_1)_2 = \frac{1}{6}(\sigma_1^2 - \sigma_2^2)\left(1 - \frac{1}{2^{3\rho}\left(2^{3\rho} + \sqrt{2^{6\rho} - 1}\right)}\right) \geq 0.$$

To summarize, transforms of the form

$$\begin{bmatrix} \alpha & 0 & (2\alpha)^{-1} \\ -\alpha & 0 & (2\alpha)^{-1} \\ 0 & 1 & 0 \end{bmatrix}$$

attain the optimal performance

$$D_1 = \frac{1}{6}\sigma_2^2 + \frac{1}{3}\sigma_3^2 + \frac{\sigma_1^2 - \sigma_3^2}{6 \cdot 2^{3\rho}\left(2^{3\rho} + \sqrt{2^{6\rho} - 1}\right)}.$$

This performance is matched by many other transforms, but not surpassed.

Four variables We now move on to communicating four variables over two channels. The problem is the most similar to the one we just solved if one channel carries y_1 and the other carries (y_2, y_3, y_4). In this case, a result analogous to Theorem 5.6 holds, revealing that it is sufficient to consider transforms of the form

$$T = \begin{bmatrix} \widetilde{T} & 0_{2\times 2} \\ 0_{2\times 2} & I_{2\times 2} \end{bmatrix} P,$$

where $\widetilde{T}$ is a 2×2 correlating transform and P is one of six permutations.

The best choice of permutation causes the correlating transform to be applied to the components with the largest and smallest variances. The result is a transform of the form

$$\begin{bmatrix} \alpha & 0 & 0 & (2\alpha)^{-1} \\ -\alpha & 0 & 0 & (2\alpha)^{-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and optimal performance given by

$$D_1 = \frac{1}{8}\sigma_2^2 + \frac{1}{8}\sigma_3^2 + \frac{1}{4}\sigma_4^2 + \frac{\sigma_1^2 - \sigma_4^2}{8 \cdot 2^{4\rho}\left(2^{4\rho} + \sqrt{2^{8\rho}-1}\right)}.$$

Let us now consider the case where each channel carries the same number of coefficients; for concreteness, one carries the odd-numbered coefficients and the other carries the even-numbered coefficients.

The transmission over two channels and the allocation of the coefficients to channels does not place any limitation on the transform. However, we can again place a simplifying limitation on the transform without loss of optimality. It is sufficient to consider pairing the input coefficients, applying a 2×2 correlating transform to each pair, and sending one output of each 2×2 sub-transform over each channel. This is justified by the following theorem:

Theorem 5.7 *Consider multiple description transform coding where (y_1, y_3) is sent on Channel 1 and (y_2, y_4) is sent on Channel 2. To minimize the average side distortion D_1 with an upper bound on ρ, it is sufficient to optimize over transforms of the form*

$$\begin{bmatrix} \widetilde{T}_1 & 0_{2\times 2} \\ 0_{2\times 2} & \widetilde{T}_2 \end{bmatrix} P, \tag{5.42}$$

with $\widetilde{T}_i \in \mathbb{R}^{2\times 2}$, $\det \widetilde{T}_i = 1$, $i = 1, 2$, and P a permutation matrix.

Proof: See Appendix 5.C.2. □

Since the canonical building blocks defined in (5.36) solve the problem of designing $\widetilde{T}_i$, $i = 1, 2$, we may write the transform as

$$T = \begin{bmatrix} T_{\alpha_1} & 0 \\ 0 & T_{\alpha_2} \end{bmatrix} P. \tag{5.43}$$

We only have to select the pairing and two transform parameters.

Since the ordering of a pair does not affect the possible performance, there are three permutations of interest:

$$P_1 = I, \quad P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Let us consider a generic choice among the three by assuming that the inputs to transform T_{α_1} have variances ς_1^2 and ς_2^2, $\varsigma_1^2 \geq \varsigma_2^2$; and the inputs to transform T_{α_2} have variances ς_3^2 and ς_4^2, $\varsigma_3^2 \geq \varsigma_4^2$. Denote the redundancies associated with the pairs ρ_1 and ρ_2, respectively. The redundancy and distortion are both additive between the pairs, so the problem is to minimize

$$d = \frac{1}{2}(d_1 + d_2) \tag{5.44}$$

subject to the constraint[18]

$$\rho_1 + \rho_2 \;\leq\; 2\rho, \tag{5.45}$$

[18]The factor of 2 is present as normalization because redundancy is measured per component. This applies also to the 1/2 in (5.44).

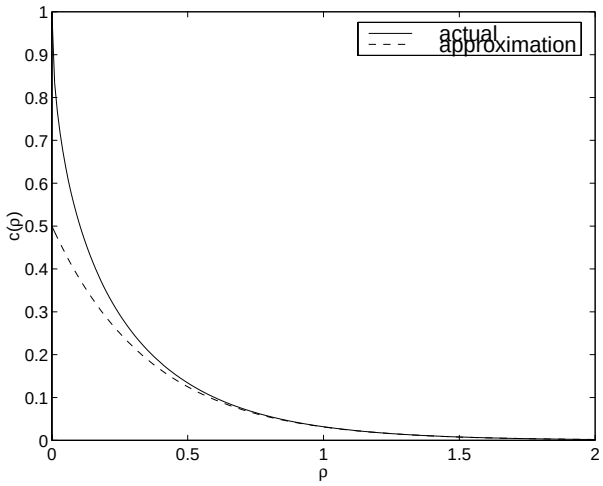


Figure 5.10: Illustration of the accuracy of the approximation of $c(\rho)$ given by (5.48)

where, according to (5.34),

$$d_1 = \frac{1}{2}\varsigma_2^2 + \frac{\varsigma_1^2 - \varsigma_2^2}{4 \cdot 2^{2\rho_1}(2^{2\rho_1} + \sqrt{2^{4\rho_1} - 1})}, \tag{5.46}$$

$$d_2 = \frac{1}{2}\varsigma_4^2 + \frac{\varsigma_3^2 - \varsigma_4^2}{4 \cdot 2^{2\rho_2}(2^{2\rho_2} + \sqrt{2^{4\rho_2} - 1})}. \tag{5.47}$$

Since d_1 and d_2 are strictly decreasing functions of ρ_1 and ρ_2, respectively, (5.45) can be replaced by an equality. The optimal split occurs when both pairs operate at the same distortion-redundancy slope. However, since $\partial D_{\alpha_i}/\partial \rho_i$ is complicated, there is no simple closed form for operating at the same slope.[19]

Let

$$c(\rho) = \frac{1}{2^{2\rho}(2^{2\rho} + \sqrt{2^{4\rho} - 1})}.$$

For large ρ, the 1 in the denominator becomes negligible, so

$$c(\rho) \approx \frac{1}{2^{2\rho}(2^{2\rho} + \sqrt{2^{4\rho}})} = \frac{1}{2 \cdot 2^{4\rho}}. \tag{5.48}$$

The error made in approximation (5.48) is shown in Figure 5.10.

Using (5.48) it becomes feasible to optimally allocate redundancies. Operating at equal slopes means that

$$\frac{\partial D_{\alpha_1}}{\partial \rho_1} = \frac{\partial D_{\alpha_2}}{\partial \rho_2}, \tag{5.49}$$

[19] Matching the slopes gives

$$\alpha_2^4 = \frac{\gamma(16\alpha_1^8\varsigma_1^4 - \varsigma_2^4) + \sqrt{\gamma^2(16\alpha_1^8\varsigma_1^4 - \varsigma_2^4)^2 + 64\alpha_1^8\varsigma_3^4\varsigma_4^4}}{32\alpha_1^4\varsigma_3^4},$$

where

$$\gamma = \frac{\varsigma_3^2\varsigma_4^2(\varsigma_3^2 - \varsigma_4^2)}{\varsigma_1^2\varsigma_2^2(\varsigma_1^2 - \varsigma_2^2)}.$$

This exact relationship is used in generating Figure 5.11, but is too complicated to use in drawing general conclusions about best pairings.

yielding, together with (5.45), two linear equations with two unknowns:

$$\begin{aligned}\rho_1 - \rho_2 &= \frac{1}{4}\log\frac{\varsigma_1^2-\varsigma_2^2}{\varsigma_3^2-\varsigma_4^2},\\ \rho_1 + \rho_2 &= 2\rho.\end{aligned}$$

Solving the system gives the optimal split of redundancies as

$$\rho_1 = \rho + \frac{1}{8}\log\frac{\varsigma_1^2-\varsigma_2^2}{\sigma_3^2-\varsigma_4^2}, \tag{5.50}$$

$$\rho_2 = \rho - \frac{1}{8}\log\frac{\varsigma_1^2-\varsigma_2^2}{\sigma_3^2-\varsigma_4^2}. \tag{5.51}$$

Substituting (5.50)–(5.51) into (5.44) gives

$$d = \frac{1}{4}(\varsigma_3^2+\varsigma_4^2) + \frac{1}{8}\left((\varsigma_1^2-\varsigma_2^2)(\varsigma_3^2-\varsigma_4^2)\right)^{1/2} 2^{-4\rho}.$$

To minimize d for large ρ, we can immediately conclude that ς_3 and ς_4 should be the smallest of the σ's. This eliminates permutation P_1. The following sequence of inequalities shows that $\varsigma_1 > \varsigma_3 > \varsigma_4 > \varsigma_2$ is the ideal sorting, *i.e.*, the largest-variance and smallest-variance components should be paired:

$$\begin{aligned}\varsigma_1^2 &> \varsigma_2^2\\ \varsigma_1^2(\varsigma_4^2-\varsigma_3^2) &< \varsigma_2^2(\varsigma_4^2-\varsigma_3^2)\\ \varsigma_1^2\varsigma_4^2-\varsigma_1^2\varsigma_3^2 &< \varsigma_4^2\varsigma_2^2-\varsigma_2^2\varsigma_3^2\\ \varsigma_1^2\varsigma_2^2-\varsigma_1^2\varsigma_3^2-\varsigma_4^2\varsigma_2^2+\varsigma_4^2\varsigma_3^2 &< \varsigma_1^2\varsigma_2^2-\varsigma_1^2\varsigma_4^2-\varsigma_2^2\varsigma_3^2+\varsigma_4^2\varsigma_3^2\\ (\varsigma_1^2-\varsigma_4^2)(\varsigma_2^2-\varsigma_3^2) &< (\varsigma_1^2-\varsigma_3^2)(\varsigma_2^2-\varsigma_4^2)\end{aligned}$$

In other words, P_3 is the best permutation. We will see shortly that this "nested" pairing method generalizes to K pairs as well.

$2K$ paired variables Let us now consider transmission of $2K$ variables over the two channels with the odd and even indexed coefficients sent over Channels 1 and 2, respectively. The extension of Theorem 5.7 to K pairs of variables would seem to naturally hold, but no proof of this has been found. Consider transforms of the following form, though we have not proven that this restriction is innocuous:

$$T = \begin{bmatrix} T_{\alpha_1} & & & \\ & T_{\alpha_2} & & \\ & & \ddots & \\ & & & T_{\alpha_K} \end{bmatrix} P,$$

where P is a permutation matrix. Again let $\varsigma_1^2, \varsigma_2^2, \ldots, \varsigma_{2K}^2$ denote the variances of the components after the permutation, with $\varsigma_{2i-1}^2 \geq \varsigma_{2i}^2$ for $i = 1, 2, \ldots, K$. Denote the redundancy associated with T_{α_i} by ρ_i. We then have a similar problem as before: Minimize

$$d = \frac{1}{K}\sum_{i=1}^{K} d_i, \tag{5.52}$$

with

$$d_i = \frac{1}{2}\varsigma_{2i}^2 + \frac{\varsigma_{2i-1}^2 - \varsigma_{2i}^2}{4 \cdot 2^{2\rho_i}\left(2^{2\rho_i} + \sqrt{2^{4\rho_i} - 1}\right)}, \qquad \text{for} \quad i = 1, 2, \ldots, K, \tag{5.53}$$

subject to the constraint

$$\rho = \frac{1}{K}\sum_{i=1}^{K} \rho_i. \tag{5.54}$$

Using the approximation (5.48) for large ρ and imposing the equal-slope conditions gives a system of K linear equations with K unknowns ($K-1$ equations coming from equal-slope conditions and an additional one from (5.54)). The solution of this system is the optimal redundancy allocation of

$$\rho_i = \rho + \frac{1}{4K}\sum_{j=1,\, j\neq i}^{K}\left[\log\frac{\varsigma_{2i-1}^2 - \varsigma_{2i}^2}{\varsigma_{2j-1}^2 - \varsigma_{2j}^2}\right], \qquad \text{for} \quad i = 1, 2, \ldots, K.$$

The resulting average distortion with half the coefficients lost is

$$d = \frac{1}{2K}\sum_{i=1}^{K}\varsigma_{2i}^2 + \frac{1}{8}\left[\prod_{i=1}^{K}(\varsigma_{2i-1}^2 - \varsigma_{2i}^2)\right]^{1/K} 2^{-4\rho}. \tag{5.55}$$

This distortion is minimized by the "nested" pairing under the conditions of the following theorem.

Theorem 5.8 (Optimal pairing) *Consider the minimization problem in (5.52)-(5.54), where in addition to choosing the ρ_i's one can choose the pairing by permuting ς_i's. The ς_i's are a permutation of $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{2K}$. At high redundancies the minimum distortion is achieved with the nested pairing $\varsigma_{2i-1} = \sigma_i$, $\varsigma_{2i} = \sigma_{2K+1-i}$, $i = 1, 2, \ldots, K$.*

Proof: See Appendix 5.C.3. □

Applying the nested pairing of Theorem 5.8, (5.55) becomes

$$d = \frac{1}{2K}\sum_{i=1}^{K}\sigma_{K+i}^2 + \frac{1}{8}\left[\prod_{i=1}^{K}(\sigma_i^2 - \sigma_{2K+1-i}^2)\right]^{1/K} 2^{-4\rho}.$$

Whereas using (5.48) helped us derive the optimal pairing and the optimal redundancy allocation, there are two problems with using this approximation: First, (5.48) is not a good approximation when ρ is small (see Figure 5.10). Second, the Lagrangian redundancy allocation solution (5.50)–(5.51) may ask for a negative redundancy, which is impossible. However, numerical calculations (see Figure 5.11) verify that the nested pairing is best over all redundancies.

Other allocations of coefficients to channels Theorems 5.6 and 5.7 are suggestive of the following more general result:

Conjecture 5.9 (Generality of pairing) *Consider a multiple description transform coding system with N variables sent over two channels. Suppose the transform coefficients are assigned to channels with $(y_1, y_3, \ldots, y_{2K-1})$, $K \leq N/2$, sent on Channel 1 and the remaining coefficients sent on Channel 2. Then for*

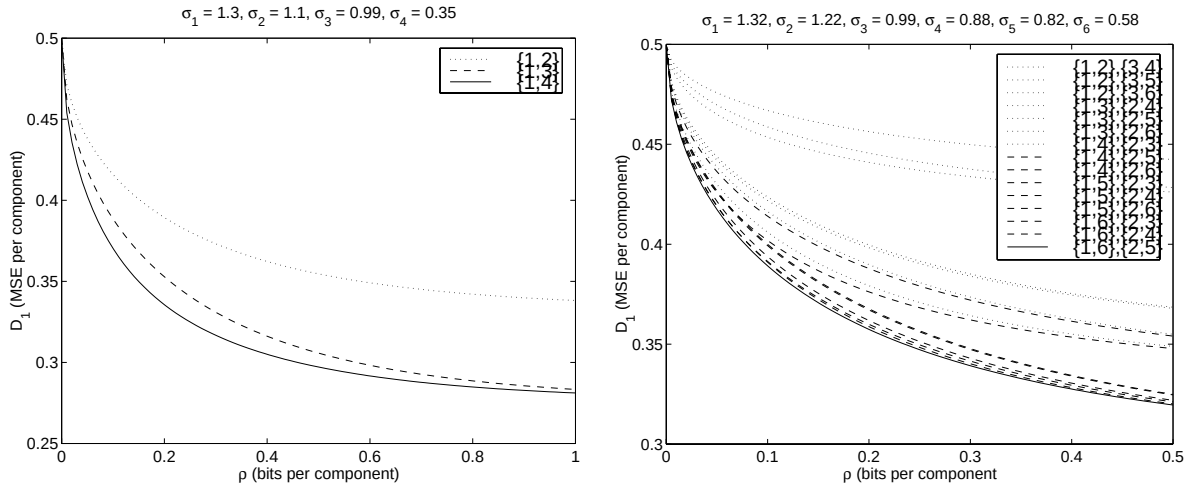


Figure 5.11: Numerical calculations using the exact redundancy allocation solution (without approximation (5.48)) confirm that the nested pairing is optimal for all rates: (a) A random instance with two pairs; (b) A random instance with three pairs.

any redundancy ρ, a transform that minimizes the average side distortion D_1 can be found in the form

$$T = \begin{bmatrix} T_{\alpha_1} & & & & \\ & T_{\alpha_2} & & & \\ & & \ddots & & \\ & & & T_{\alpha_K} & \\ & & & & I_{N-2K} \end{bmatrix} P,$$

where each T_{α_i} is of the form (5.36) and P is a permutation matrix. The permutation maps

$$[x_1, x_2, \ldots, x_N]^T \qquad \textit{to} \qquad [x_1, x_N, x_2, x_{N-1}, \ldots, x_K, x_{N+1-K}, x_{K+1}, \ldots, x_{N-K}]^T.$$

In attempting to prove this conjecture using the techniques of Appendices 5.C.1 and 5.C.2, one is faced with the following problem: Let

$$R_y = \begin{bmatrix} \Lambda_1 & A_1 & A_2 \\ A_1^T & \Lambda_2 & A_3 \\ A_2^T & A_3^T & \Lambda_3 \end{bmatrix} \begin{matrix} K \\ K \\ N-2K \end{matrix}$$
$$\begin{matrix} K & K & N-2K \end{matrix}$$

be a positive definite matrix with block dimensions as marked and Λ_1, Λ_2, and Λ_3 positive diagonal matrices. The problem is to find $V_1 \in \mathbb{R}^{K\times K}$ and $V_2 \in \mathbb{R}^{N-K\times N-K}$, each with determinant 1, such that

$$\begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} R_y \begin{bmatrix} V_1^T & 0 \\ 0 & V_2^T \end{bmatrix} = \begin{bmatrix} \Gamma_1 & B & 0_{K\times N-2K} \\ B^T & \Gamma_2 & 0_{K\times N-2K} \\ 0_{N-2K\times K} & 0_{N-2K\times K} & \Gamma_3 \end{bmatrix},$$

with Γ_1, Γ_2, Γ_3, and B all diagonal matrices.

Choosing V_1 and V_2, each with determinant 1, gives $K^2 + (N-K)^2 - 2$ degrees of freedom. The number of independent constraints is $N(N-1)/2 - K$.[20] For all $N \geq 2$ and $K \geq 1$, the number of variables is greater than the number of constraints. This suggests that a solution can be found, but obviously does not guarantee it. The proofs of the earlier theorems use explicit determinations of suitable V_1 and V_2; unfortunately, these depend on R_y in a nonlinear fashion, so they cannot be generalized in an obvious way.

5.3.2.4 Techniques for more than two channels

The extension of the correlating transform method for MDTC to more than two channels is hindered by design complexity. Applying the results of Section 5.3.2.1 to the design of 3×3 transforms is considerably more complicated than what has been presented thus far because there are eight degrees of freedom remaining after fixing $\det T = 1$. Even in the case of equal channel failures, a closed form solution would be much more complicated that (5.33). When $\sigma_1 > \sigma_2 > \sigma_3$ and erasure probabilities are equal and small, a set of transforms which gives good performance is described by

$$\begin{bmatrix} a & -\dfrac{\sqrt{3}\sigma_1 a}{\sigma_2} & -\dfrac{\sigma_2}{6\sqrt{3}\sigma_1^2 a^2} \\ 2a & 0 & \dfrac{\sigma_2}{6\sqrt{3}\sigma_1^2 a^2} \\ a & \dfrac{\sqrt{3}\sigma_1 a}{\sigma_2} & -\dfrac{\sigma_2}{6\sqrt{3}\sigma_1^2 a^2} \end{bmatrix}.$$

The design of this set was based on enforcing equal rates for the three channels, equal "importance" of the channels (measured through the distortion when there are erasures), and a dependence on a similar to the two-channel case. The performance when two of three components are received is almost as good as with a numerically optimized transform.

This heuristic design has reduced the eight available degrees of freedom in designing T to a single parameter a. However, it sacrifices in performance. Just as the parallel use of two-by-two transforms gave a method for sending $2K$ variables over two channels, a cascade combination of these transforms gives a method for sending 2^K variables over 2^K channels. The cascade structure simplifies the encoding, decoding (without erasures), and design when compared to using a general $2^K \times 2^K$ transform.

The simplest instance of the cascade structure is shown in Figure 5.12, where four variables are sent over four channels. This is equivalent to the use of a transform of the form

$$T = \begin{bmatrix} T_\gamma & 0 \\ 0 & T_\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_\alpha & 0 \\ 0 & T_\beta \end{bmatrix}. \tag{5.56}$$

Though this has only three degrees of freedom—in place of 15 in a general determinant 1 transform of this size—empirical evidence suggests that this class of transforms is sufficiently rich to give nearly optimal performance. Numerical optimizations of the cascade structure for one, two, and three component erasures give performance nearly as good as numerically-optimized unstructured transforms.

[20] "Independent" is used loosely here to indicate constraints that are not obviously identical due to the symmetry of the product VR_yV^T.

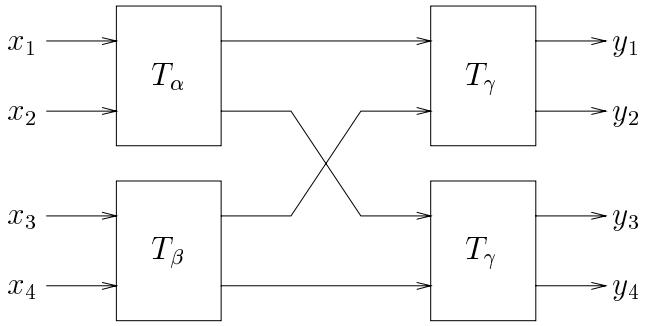


Figure 5.12: Cascade structure for MDTC of four variables to be transmitted over four channels. The cascade structure simplifies the design procedure be reducing 15 free parameters to 3. The use of γ in both second-stage blocks is because each first-stage block produces streams of equal rate and equal importance.

Since the correlating transform method described in this section is a generalization of the technique for two variables proposed by Orchard *et al.*, the improvement afforded by this generalization should be noted. The generalization consists of facilitating optimization for an arbitrary p.d.f. on the system state and for an arbitrary number of channels. Let us look specifically at the second part. To transmit four variables over four channels that are equally likely to fail using the technique from [146], one would form two sub-problems of sending two variables over two channels. Each pair would be transmitted after the application of the transform (5.21). The cascade transform (5.56) can give strictly better performance simultaneously for one, two, and three erasures. A numerical example is shown in Figure 5.13. The source is described by $R_x = \text{diag}([1, 0.64, 0.36, 0.16])$, and redundancy $\rho = 0.125$ bits/component is added to mitigate erasure effects. The signal-to-noise ratio (SNR) with the cascade transform is simultaneously higher for both one erasure and two erasures.

5.3.3 Application to Image Coding

We now turn our attention to the communication of still images. The most common way to communicate an image over the Internet is to use a progressive encoding system and to transmit the coded image as a sequence of packets over a TCP connection. When there are no packet losses, the receiver can reconstruct the image as the packets arrive; but when there is a packet loss, there is a large period of latency while the transmitter determines that the packet must be retransmitted and then retransmits the packet. The latency is due to the fact that the application at the receiving end uses the packets only after they have been put in the proper sequence. Changing to UDP does not solve the problem: because of the progressive nature of the encoding, the packets are useful only in the proper sequence. (The problem is more acute if there are stringent delay requirements, for example, for fast browsing or for streaming video. In this case retransmission is not just undesirable but impossible.) To combat this latency problem, it is desirable to have a communication system that is robust to arbitrarily placed packet erasures and that can reconstruct an image progressively from packets received in any order. The MDTC method described earlier seems suitable for this task.

As a proof of concept, let us design a system which uses four channels and is robust to the loss of any one of the channels. We consider the channels to be equally likely to fail and use transforms which can be implemented with the cascade combination of four 2×2 transforms as in Figure 5.12.

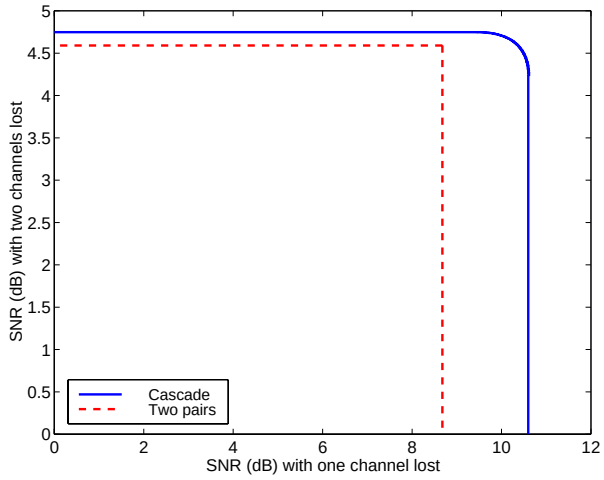


Figure 5.13: Comparison between the cascade transform (5.56) and pairing for sending four variables over four channels. The source is given by $R_x = \text{diag}([1, 0.64, 0.36, 0.16])$, and in both cases redundancy $\rho = 0.125$ bits/component. With the zero-erasure performance held equal, the cascade transform can simultaneously give better performance with one or two component lost.

This method is designed to operate on source vectors with uncorrelated components with different variances. We approximately obtain such a condition by forming vectors from DCT coefficients separated both in frequency and in space. A straightforward application proceeds as follows:

1. An 8×8 block DCT of the image is computed.
2. The DCT coefficients are uniformly quantized.
3. Vectors of length 4 are formed from DCT coefficients separated in frequency and space. The spatial separation is maximized, *i.e.*, for 512×512 images, the samples that are grouped together are spaced by 256 pixels horizontally and/or vertically.
4. Correlating transforms are applied to each 4-tuple.
5. Entropy coding akin to that of JPEG is applied.

The system design is completed by determining which frequencies are to be grouped together and designing a transform (an (α, β, γ)-tuple for use as in Figure 5.12) for each set. This can be done based on training data. Even with a Gaussian model for the source data, the transform parameters must be numerically optimized.[21]

An abstraction of this system was simulated. If we were to use precisely the strategy outlined above, the importance of the DC coefficient would dictate allocating most of the redundancy to the group containing the DC coefficient. Thus for simplicity we assume that the quantized DC coefficient is communicated reliably through some other means. We separate the remaining coefficients into those that are placed in groups of four and those that are sent by one of the four channels only. The optimal allocation of redundancy between groups

[21] In the case of pairing transforms as in [146], the optimal pairing and allocation of redundancy between the pairs can be found analytically as shown in Section 5.3.2.3.

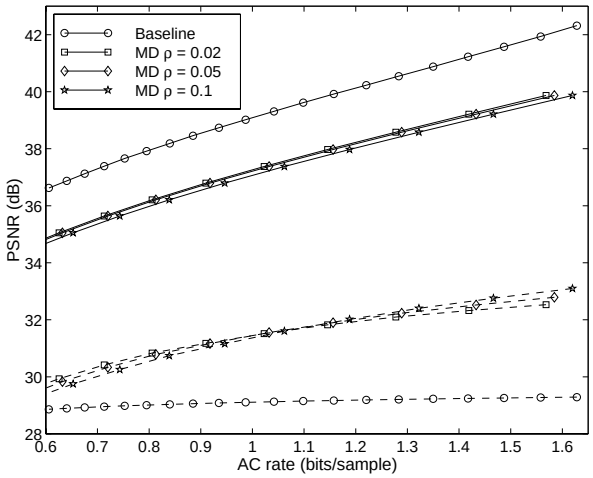


Figure 5.14: Rate-distortion results for coding of 512×512 *Lena* image with the correlating transform method. Top (solid) and bottom (dashed) curves are for when all and three-fourths of the transmitted data arrives at the decoder, respectively.

is difficult, so we allocate approximately the same redundancy to each group. For comparison we consider a baseline JPEG system that also communicates the DC coefficient reliably. The AC coefficients for each block are sent over one of the four channels. The rate is estimated by sample scalar entropies.

Simulation results for the standard 512×512 *Lena* image [117] are given in Figure 5.14. The curve type indicates whether all (solid) or three-fourths (dashed) of the transmitted data arrives at the decoder. The marker type differentiates the MD and baseline JPEG systems. For the MD systems, the objective redundancy ρ in bits per pixel is also given. As desired, the MD system gives a higher-quality image when one of four packets is lost at the expense of worse rate–distortion performance when there are no packet losses. Size 128×128 sections of sample images are given in Figure 5.15.

The results presented here are only preliminary because the MDTC techniques from this section have been applied without much regard for the the structure and properties of images. The transform design is based on high-rate entropy estimates for uniformly quantized Gaussian random variables. Effects of coarse quantization, dead zone, divergence from Gaussianity, run length coding, and Huffman coding are neglected. Incorporating these will require a refinement of the theory and/or an expansive numerical optimization. Aside from transform optimization, this coder could be improved by using a perceptually tuned quantization matrix as suggested by the JPEG standard. Here we have used a constant quantization matrix for simplicity. With this type of tuning it should be possible to design a system which would perform precisely as well as the system in [201] when two or four of four packets arrive, but which performs better when one or three packets arrive. The results in [201] are ahead of those presented here because they have, quite correctly, used a classification of image blocks which allows for better statistical modeling and redundancy allocation.

A full image communication system would probably require packetization. We have not explicitly considered this, so we do not produce four streams with precisely the same number of bits. The expected number of bits for each stream is equal because of the form of (5.36). In contrast, with the transforms used in [146] one must multiplex the streams to produce packets of approximately the same size.

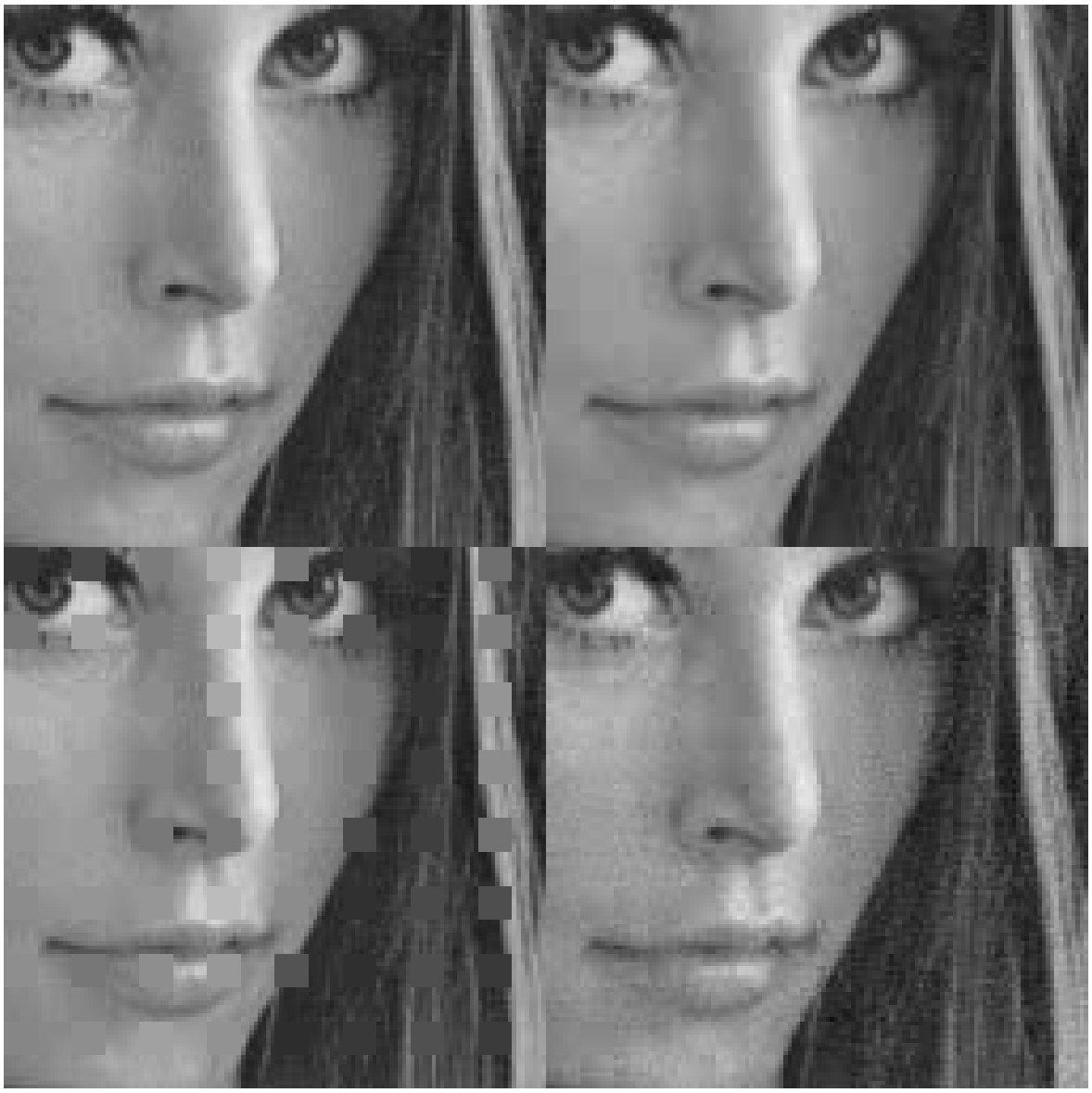


Figure 5.15: Results for correlating transform method at 1 bpp. Top row: no packet losses; bottom row: one packet lost. Left column: baseline system; right column: MD system.

5.3.4 Application to Audio Coding

As a second application of statistical channel coding, consider the problem of robust transmission of audio over a lossy packet network such as the Internet. Rather than building a joint source–channel audio coder from scratch, the correlating transform method was introduced in an existing coder to provide the multiple description feature. The experiments confirm that indeed, in the presence of channel erasures—lost packets—the new MD audio coder is robust and gracefully degrades as the number of erasures increases.

Multiple description transform coding was applied to a state-of-the-art audio coder developed at Bell Labs, the *Perceptual Audio Coder (PAC)* [177]. This is a successful and well-known audio coder. Section 5.3.4.1 reviews the basic principles of perceptual audio coding and describes Bell Labs' PAC. The design of a *multiple description PAC (MD PAC)* coder is described in Section 5.3.4.2. Section 5.3.4.3 discusses experimental results.

5.3.4.1 Perceptual audio coding

In audio compression one wishes to find digital representations of audio signals that provide maximum signal quality with a given bit rate and delay. Whereas in speech coding both the source model (speech production model) and the destination model (hearing properties) can be used to lead to efficient compression, audio coding has to rely mainly on the destination model due to the high complexity of modeling audio sources. Human perception plays a key role in compression of audio material. As a result, recent audio standards work has concentrated on a class of audio coders known as *perceptual coders*. Rather than trying to understand the source, perceptual coders model the listener and attempt to remove *irrelevant* information contained in the input signal. For a given rate, a perceptual coder will typically have a lower SNR than a lossy source coder designed to maximize SNR, but will provide superior perceived quality to the listener. By the same token, at equivalent perceived quality, the perceptual coder will need a lower bit rate. The combination of an appropriate *signal representation*, by means of a transform or a filter bank, and a *psychoacoustic model* of the destination provides the means to achieve efficient compression.

Bell Labs' PAC coder Like most perceptual coders, PAC combines both source coding techniques to remove signal *redundancy* and perceptual coding techniques to remove signal *irrelevancy*, as shown in Figure 5.16. PAC divides the input signal into 1024-sample-blocks of data—*frames*—that will be used throughout the encoding process. It consists of five basic parts:

- The *analysis filter bank* converts the time-domain data to frequency domain. First, the 1024-sample block is analyzed and, depending on its characteristics, such as stationarity and time resolvability, a *Modified Discrete Cosine Transform (MDCT)* or a discrete wavelet transform is applied [177]. The total given bit rate and the sampling rate also play a role in the design of the transform.

  The MDCT, the wavelet filter bank, and their combinations produce either 1024- or 128-sample blocks of frequency-domain coefficients. In both cases, the base unit for further processing is a block of 1024 samples.

- The *perceptual model* determines the frequency-domain thresholds of masking from both the time-domain signal and the output of the analysis filter bank. A threshold of masking is the maximum noise one can add to the audio signal at a given frequency without perceptibly altering it. Depending on the transform that

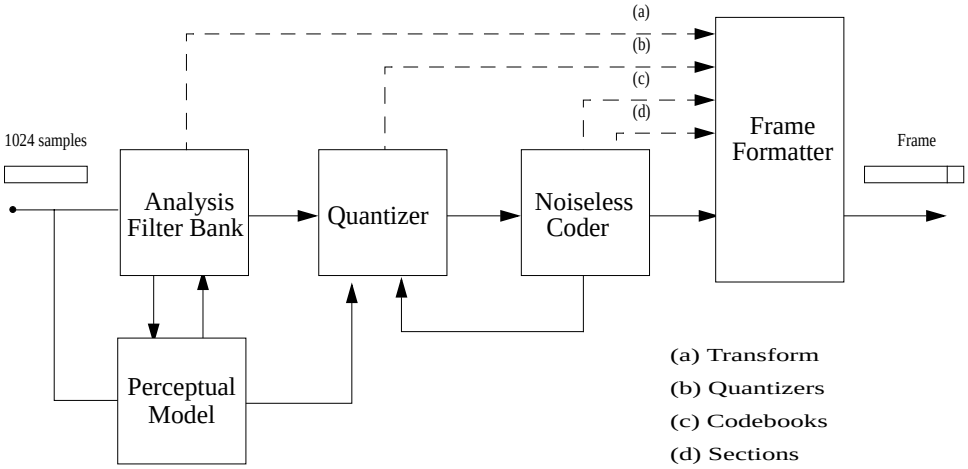


Figure 5.16: PAC coder block diagram.

was used previously, each 1024-sample block is split into a predefined number of groups of bands—*gain factor bands*. Within each gain factor band, a perceptual threshold value is computed.

- *Quantizer* – Within each factor band the quantization step sizes are adjusted according to the computed perceptual threshold values in order to meet the noise level requirements. The quantization step sizes may also be adjusted to comply with the target bit rate, hence the feedback from the noiseless coder to the quantizer.

- *Noiseless coding* – Huffman coding is used to provide an efficient representation of the quantized coefficients. A set of optimized codebooks is used; each codebook codes sets of two or four integers. For efficiency, consecutive factor bands with the same quantization step size are grouped into sections, and the same codebook is used within each section. Failure to meet the target bit rate may trigger a recomputation of quantization step sizes.

- The *frame formatter* forms the bit stream, adding to the coded quantized coefficients the side information needed at the decoder to reconstruct the 1024-sample-block. This block is defined as the *frame* and contains, along with one 1024-sample or eight 128-sample blocks, the following side information for each of them: the transform used in the analysis filter bank, section boundaries, codebooks, and quantization step sizes for sections. Side information accounts for 15 to 20% of the total bit rate of the coded signal.

At the decoder, the entropy coding, quantization, and transform blocks are inverted and an error mitigation block is added between the inverse quantization and the synthesis filter bank. In this block, lost frames are interpolated based on the previous and following frames.

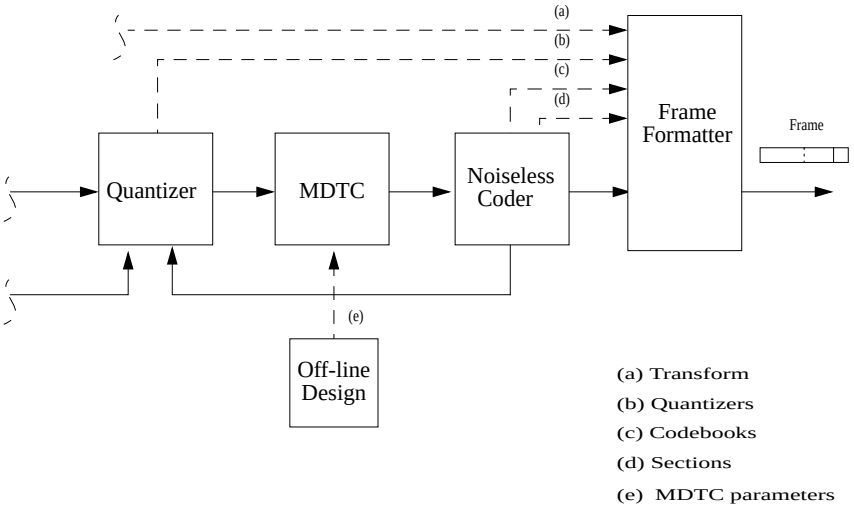


Figure 5.17: MD PAC encoder block diagram.

5.3.4.2 A multiple description perceptual audio coder

Figure 5.17 depicts the MD version of the PAC coder. The only difference when compared to the PAC coder is the addition of the MDTC block together with the off-line design.

- An *MD transform* block is inserted between the quantizer and the noiseless coder. Within each 1024-sample block or eight 128-sample blocks contained in the 1024-sample-unit-block, MDTC is applied to the quantized coefficients (integers) coming out of the quantizer. The transform is applied to pairs of quantized coefficients and produces pairs of *MD-domain quantized coefficients*, using the off-line designed side information. Within each pair, MD-domain quantized coefficients are then assigned to Channel 1 (quantized coefficient with higher variance) or Channel 2 (quantized coefficient with smaller variance).[22] The side information must now contain both the way quantized coefficients have been paired and the parameter of the transform for each pair. Then, the MD-domain quantized coefficients are passed to the noiseless coder.

As in the encoder, in the decoder we add an inverse MDTC block and off-line side information (see Figure 5.18).

- *Inverse MD transform* – This block is the core of the MDTC scheme, since it performs the estimation and recovery of lost MD-domain quantized coefficients. The inverse MDTC block is inserted between the noiseless decoder and the inverse quantizer. Within each 1024-sample block or eight 128-sample blocks contained in the 1024-sample unit, the inverse MDTC function is applied to the MD-domain quantized

[22] One channel consistently having higher variance than the other is an unexplained empirical fact. If the source is Gaussian, uncorrelated, and zero-mean, and if the quantizer preserves the zero mean, then the outputs of the MDTC block should have the same variance. The unequal variances indicate that the quantizer does not preserve the zero mean, the source is too far from Gaussian, or some flaw remains in the software. This investigation is left for future work.

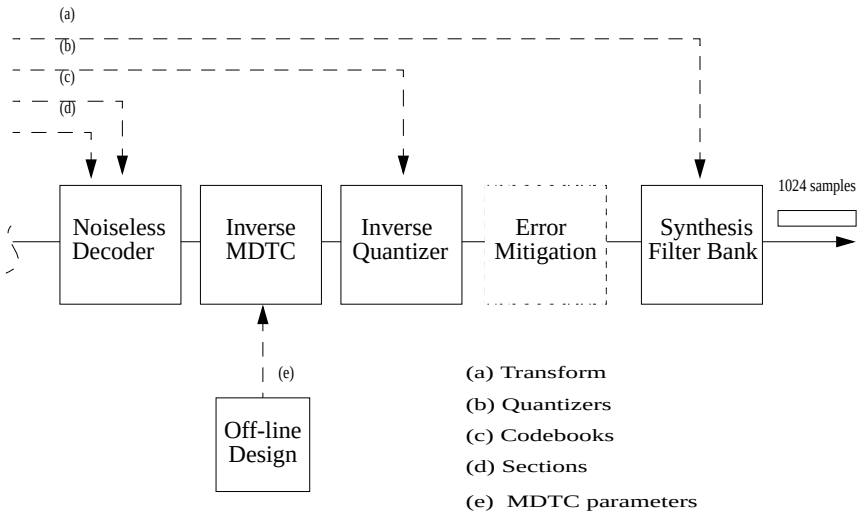


Figure 5.18: MD PAC decoder block diagram.

coefficients (integers) coming out of the noiseless decoder block. Then, one of the following MDTC inversion strategies is applied:

- When both channels are received, the MD transform is inverted by applying a discrete version of (5.37), recovering perfectly the quantized coefficients.
- When Channel 1 is lost, the lost coefficients are estimated from their counterparts in Channel 2 and the MDTC is inverted; or directly, (5.38) is used.
- When Channel 2 is lost, (5.39) is used.
- When both channels are lost, the built-in loss mitigation feature of the PAC is used.

As in the encoder, side information provides the way quantized coefficients have to be paired, the parameter α of the inverse transform for each pair, and the variances to be used in the estimation of lost MD-domain quantized coefficients. Once the MDTC has been inverted according to one of these four strategies, the output quantized coefficients are simply passed to the inverse quantizer.

Audio file statistics When applying MDTC, knowledge of the second-order statistics of the source is needed, not only for designing the optimal pairing and transform, but also for the estimation of lost coefficients. In the PAC structure, the bit rate affects the MDCT or filter bank choice and thus the coefficient variances. This can be seen in Figure 5.19, which gives the frequency-domain coefficient variances for an audio segment at three different bit rates. As the rate is increased, more of the original spectrum is retained.

Since we are interested in Internet audio applications, a bit rate of 20 kbps was selected. Five files recommended by the European Broadcast Union [51] and four additional files were analyzed. Although audio signals are inherently nonstationary and their statistics may change completely from one file to another, as a first estimate, variances were computed based on whole files. Our analysis was conducted on three different

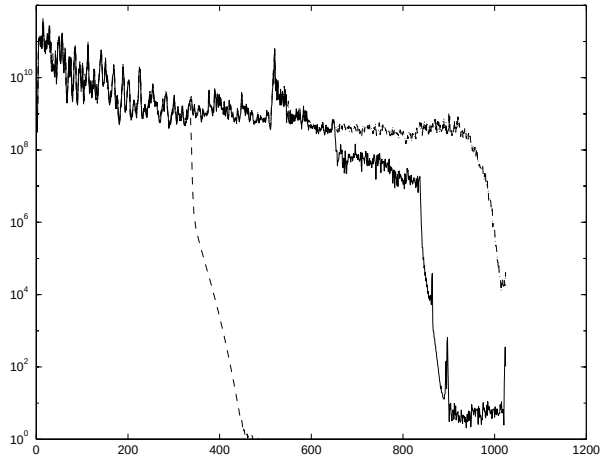


Figure 5.19: Coefficient variances as a function of frequency for bit rates of (from left to right) 20 kbps, 30 kbps and 48 kbps.

file	description	duration[s]
casta.44	castagnette	7.66
pipe.48	pipe	14.42
track59.44	violin (Ravel) [51]	29.00
track60.44	piano (Schubert) [51]	92.00
track65.44	classical (Strauss) [51]	112.00
track67.44	classical (Mozart) [51]	82.00
track68.44	classical (Baird) [51]	164.00
comedance.22	pop/rock (*Come Dancing* by The Kinks)	42.71
underthestars.22	pop/rock (*Underneath The Stars* by Mariah Carey)	48.40

Table 5.2: Descriptions of the analyzed audio files. The file extensions give the sampling frequencies, where 44 stands for 44.1 kHz, 48 for 48.0 kHz and 22 for 22.05 kHz.

kinds of music: solo instruments, classical orchestra and pop/rock. Table 5.2 lists the files used in this project and gives brief descriptions of musical content. The file extensions give sampling frequencies, where 44 stands for 44.1 kHz, 48 for 48.0 kHz, and 22 for 22.05 kHz. Among these files are the standard highly nonstationary *castagnette* and very stationary *pipe* files.

We analyzed the frequency-domain coefficients at the output of the analysis filter bank of the PAC coder. Figure 5.20 shows the empirical variances of two audio files for two MDTC lengths (1024 and 128). The power spectrum of the *castagnette* file exhibits two distinct peaks. This file and the *pipe* file have significantly different variance distributions from the rest of audio files; the majority of audio files have power spectra similar to that of the *track65* file.

Pairing design As described in Section 5.3.2.3, when there are $2N$ variables and two channels, the optimal pairing consists of pairing the variable with the highest variance with the one with the lowest variance, the second highest variance variable with the second lowest variance one, etc. There may be either 1024 or 128 variables to pair, leading to either 512 or 64 pairs. Since factor bands may have different quantization steps,

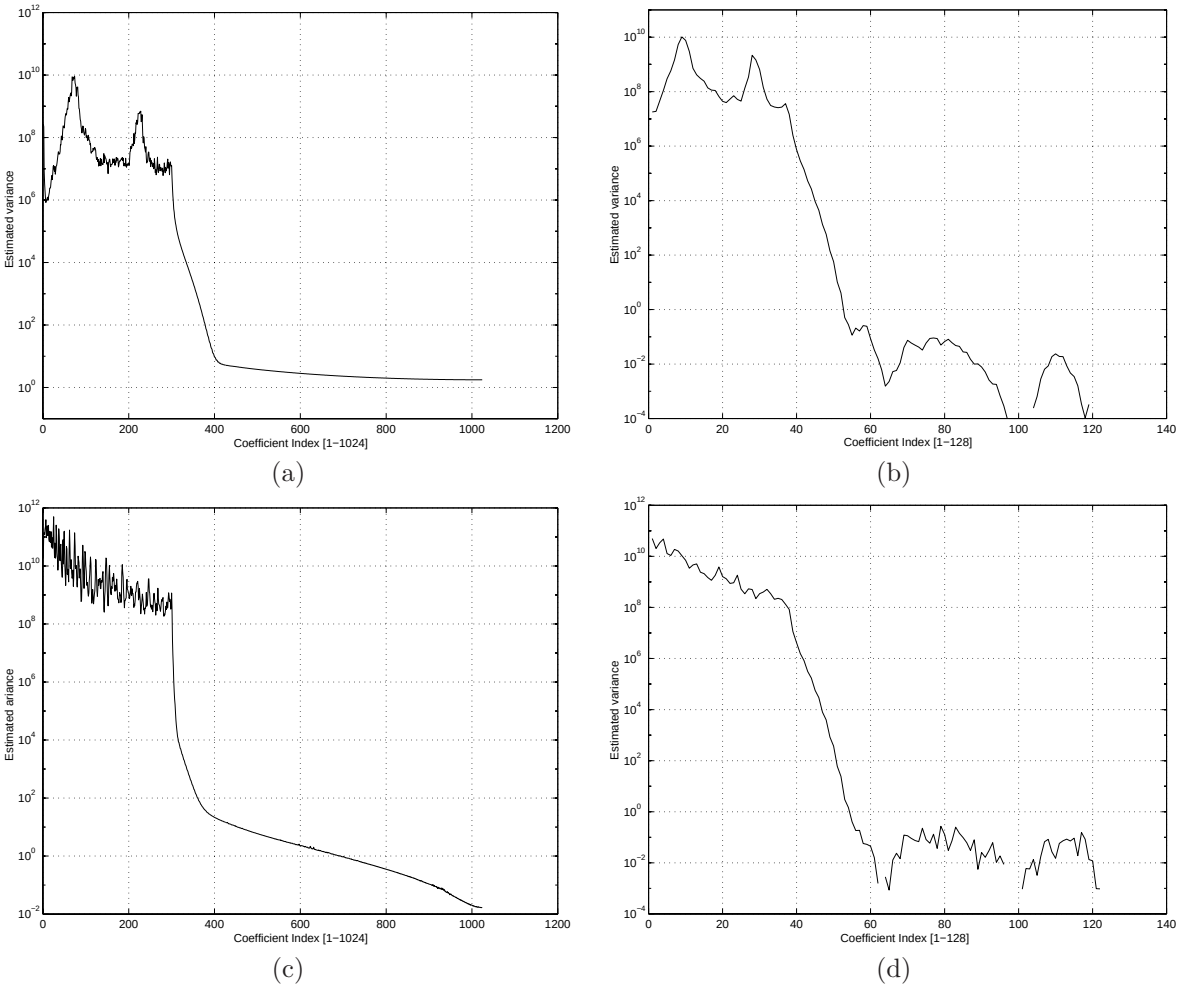


Figure 5.20: Empirical variances frequency-domain transform coefficients: (a) *casta.44* with MDCT transform of length 1024; (b) *casta.44* with MDCT transform of length 128; (c) *track65.44* with MDCT transform of length 1024; (d) *track65.44* with MDCT transform of length 128. Empirical variances of most of the audio files follow that of the *track65.44* file.

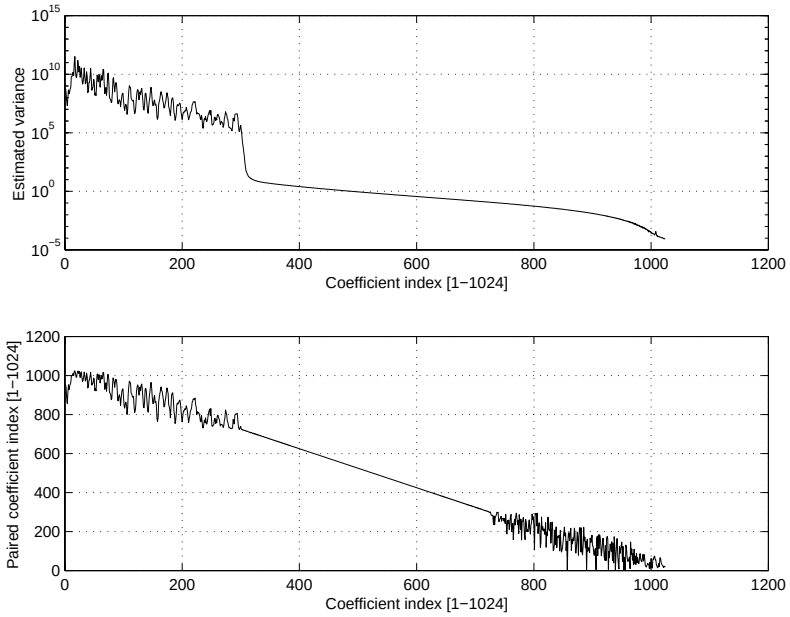


Figure 5.21: Pairing design across all bands for audio file *track67.44.c20M.dec*. The pairing is performed according to the optimal scheme, as described by Theorem 5.8.

this approach implies a rescaling of the domain spanned by the variables prior to the application of MDTC, by multiplying variables by their respective quantization steps.

As will be explained in Section 5.3.4.3, the theoretically optimal pairing across all bands did not work well. A second approach was to take the factor bands into account, and pair variables belonging to the same factor band. For the audio file *track67.44.c20M.pac* (c20M.pac meaning that the file was compressed to a mono bit rate of 20 kbps with PAC), Figure 5.21 depicts the optimal pairing across all bands while Figure 5.22 depicts the optimal pairing within factor bands.

Transform design Given a set of pairings, we next must design the correlating transform T_α defined by (5.36) for each pair. This requires a redundancy allocation between pairs as described in Section 5.3.2.3. The transform parameter α is then given by (5.35). Figure 5.23 depicts, for audio file *track67.44.c20M.dec*, both the optimal redundancy allocation between pairs and the optimal transform parameter α for each pair when the mean redundancy is 0.1 or 0.5 bits per variable. A pairing of variables within factor bands has been used.

Entropy coding After the application of MDTC, the two channels (the MD-domain quantized coefficients) must be entropy coded. Because of difficulties in modifying the PAC software, the entropy coder of the single description (SD) PAC was used without modification. This provides a crude rate estimate. Since this entropy coder processes more than one sample at a time it cannot really be used in MD PAC unless the channels are separated. One may have the impression that the joint processing gives artificially low rate estimates. In actuality, the entropy coder uses static Huffman code tables designed for uncorrelated samples. The performance would almost certainly improve with Huffman codes optimized for MDTC. Thus the comparisons between SD PAC and MD PAC are fair. In any case, this is a serious limitation of the current test bed.

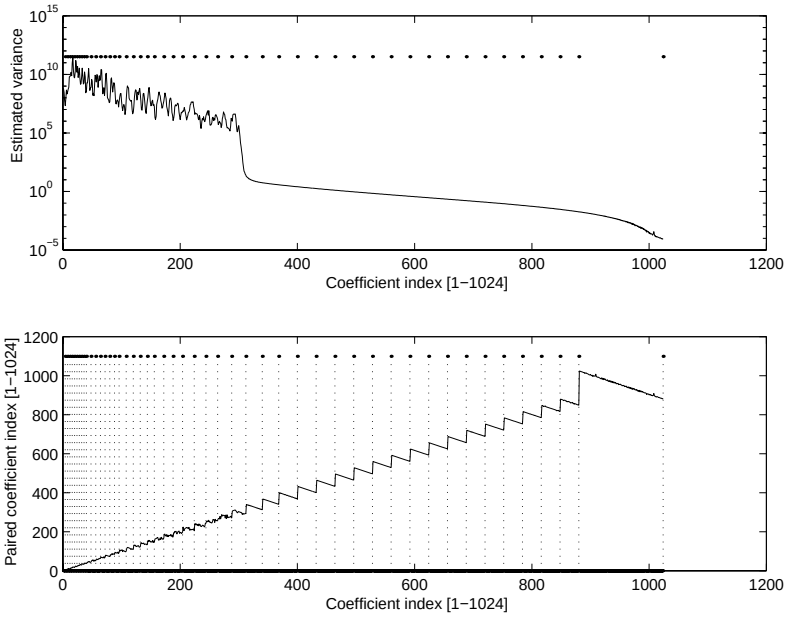
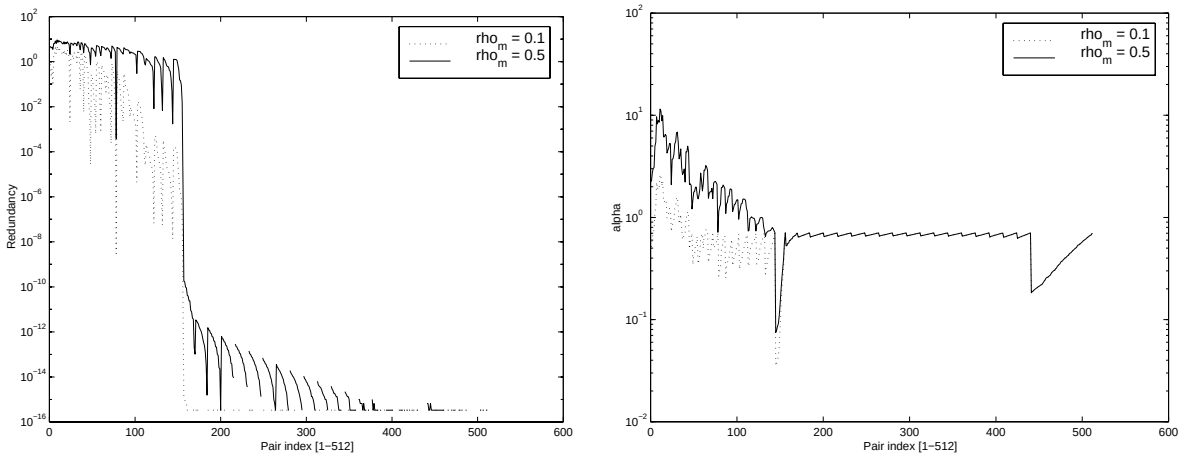


Figure 5.22: Pairing design within factor bands for audio file *track67.44.c20M.dec*. Factor band boundaries are indicated by small dots on tops of the plots. The optimal nested pairing is used within each factor band. This pairing scheme is suboptimal but avoids pairing coefficients quantized with different quantization step sizes.



(a) (b)

Figure 5.23: Transform design for audio file *track67.44.c20M.dec*: (a) optimally allocated redundancies (ρ's); and (b) transform parameters (α's) for each of the 512 pairs.

Side information From one set of variables, the MDTC scheme produces two distinct channels that have to be sent separately through a network. In our case, from each 1024- or 128-sample block, the MDTC produces two 512- or 64-sample sets of coefficients. As described previously, the set with higher variances is called Channel 1 and the other Channel 2. Since these two channels have to be sent separately, side information of the original frame has to be doubled and sent with each channel. This leads to an increase in the total bit rate of the coded audio output and is accounted for in all reported rates. In the monophonic case, the side information amounts to up to 20% of the total target bit rate.

Side information containing the MDTC parameters has to be transmitted to the receiver as well. This could be done at the beginning of the transmission. The size of such a file is only a few tens of kbits. This is small compared to the size of the compressed files, representing less than 3 seconds of compressed data. Moreover, the transform need not be adjusted for each audio file.

5.3.4.3 Experimental results

This section includes first a description of experiments to compare the MD PAC to the original single description version (SD PAC). A explanation is then given for why pairing within factor bands performs better than the theoretically optimal pairing across all bands. Both subjective and objective means are used to interpret the results. All the experiments with the MD PAC were done with a small amount of redundancy, $\rho = 0.1$ bits per variable.

- **Experiment 1:** The first experiment is to compare SD PAC and MD PAC at the same bit rate when no frames are lost. Since we are introducing redundancy, the MD version should sound slightly worse.
- **Experiment 2:** Then, still without network impairments, we increase the bit rate in the MD PAC until we reach the same quality as in the SD PAC. This shows the price paid in bits for robustness; these bits are wasted when no data is lost.
- **Experiment 3:** Finally, we compare the MD PAC and SD PAC at various loss rates. The SD PAC uses frame interpolation to recover lost frames. If frame information from only one channel is lost, the MD PAC uses statistical redundancy as explained earlier. If frame information from both channels is lost, *i.e.*, the whole frame is lost, the SD PAC error recovery scheme is used.

In what follows, P_i is the loss rate of Channel i, $i = 1, 2$. The overall average loss rate is defined as $P_{\text{tot}} = (P_1 + P_2)/2$. For example, if $P_1 = 20\%$, then 20% of half-frames corresponding to Channel 1 are lost.

Pairing across all bands As described in Section 5.3.2.3, the optimal pairing of $2N$ variables is obtained by pairing the highest-variance variable with the lowest-variance one, the second highest with the second lowest and so on. According to this scheme, the optimal pairing may form pairs between coefficients in different factor bands. Doing this we have to take into account the fact that variables in a pair may have had different quantization step sizes.

Using pairing across all bands, the results of Experiment 1 were quite disappointing. The quality degradation in the MD PAC was extreme. Here is an explanation why: After applying MDTC to the quantized coefficients, the MD-domain outputs were simply passed to the original PAC noiseless coder. Since the correlating transforms have been designed to produce two equal-rate outputs, we are basically introducing nonzero

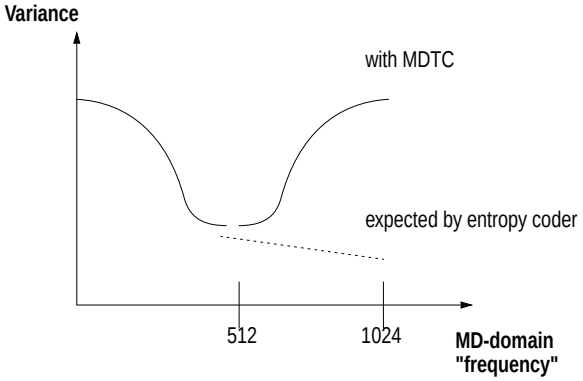


Figure 5.24: MD-domain "spectrum" modification when pairing is across all factor bands. The figure demonstrates that what the entropy coder expects is far from what the MDTC block provides; thus, the coder runs out of bits. The problem stems from a lack of a "frequency" interpretation of the MD-domain coefficients.

values at the positions where the noiseless coder is expecting zeros. Thus, modifying the input to the noiseless coder in such a way led to ineffective coding, resulting in quality degradation. This effect is depicted in Figure 5.24. A solution to this problem would be to design and optimize new entropy codes to be used for the MD-domain quantized coefficients. This is left for future work.

The MDTC analysis developed in Section 5.3.2 assumes equal quantization step sizes for each variable. To properly handle variables paired from different factor bands, the theory would have to be further generalized. As a first approximation to accounting for this discrepancy, a rescaling described in [6] was used. Further investigation is needed.

Pairing within factor bands We now restrict ourselves to pairing variables belonging to the same factor band. The results were much better. First, we did not face the problem of pairing variables quantized with different step sizes. Also, the MD-domain spectrum is only slightly modified by applying the correlating transform, as can be seen in Figure 5.25. Note that after applying the MDTC transform, the outputs lose their "frequency"-domain meaning. For convenience, when ordering the MD-domain coefficients, we keep the first MDTC output at the frequency-domain position of its first input, and the second MDTC output at the frequency-domain position of its second input; the ordering within a pair is arbitrary.

Throughout the rest of this section, the reader will be referred to the WWW for experimental results. Audio files are provided in three formats, *aiff*, *wave*, and *next*, and can be accessed on-line at

`http://cm.bell-labs.com/who/jelena/Interests/MD/AudioDemo/DemoList.html`.

Listening to these files is the only way to genuinely assess the results. In [6], one can find plots of power spectra of reconstructed signals. These give some impression of the robustness to packet losses.

Experiment 1 was performed at 20 kbps. The quality degradation due to the redundancy was very low, though noticeable to expert listeners (listen to Files 1 and 2 under "No losses").

In Experiment 2, the difference between the SD PAC at 20 kbps and the MD PAC at 22 kbps is hardly noticeable (listen to Files 1 and 3 under "No losses").

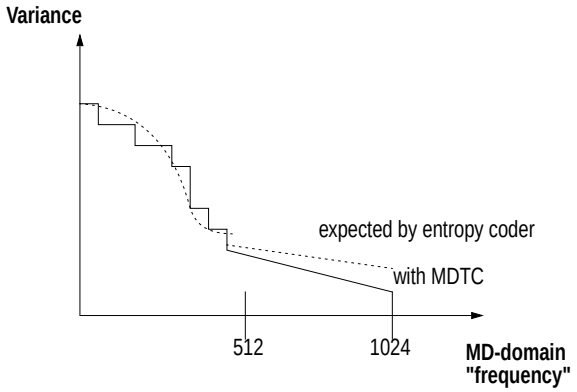


Figure 5.25: MD-domain "spectrum" modification when pairing is within factor bands. Now what the entropy coder expects is close to what the MDTC block provides, resulting in more efficient coding.

Experiment 3 was performed at various loss rates. For the MD PAC coder, results are available on the Internet for loss rates in the following set:

$$(P_1, P_2) \in \{(5\%, 5\%), (10\%, 0\%), (20\%, 20\%), (40\%, 0\%), (50\%, 50\%), (0\%, 100\%), (100\%, 0\%)\}.$$

Results for the SD PAC coder are available at the corresponding average loss rates.

The SD PAC error recovery scheme is very effective below $P_{\text{tot}} = 10\%$. However, for higher loss rates, the interpolation process becomes insufficient, and gaps appear in the music (*e.g.*, listen to File 4 under "Total loss rate = 50%"). The MD PAC estimation process seems to work well, even at high loss rates. However, we should remember that when both halves of a particular frame are lost, the system reverts to the SD PAC frame interpolation scheme. Thus, if the probability of losing both halves of a frame reaches about 10%, the insufficiency of frame interpolation will be audible. Also, some of the audible artifacts may be due to jumping between perfectly received frames, frames estimated using the MD scheme, and interpolated frames. Further descriptions of these experiments appear in [6, 7].

Reanalyzing the reconstructed audio revealed a slight bias from the original spectrum. Moreover, audio is inherently nonstationary, yet we are using only one set of estimated variances for an entire audio file. A possibility for future work is to implement an adaptive scheme where the variances are estimated on shorter audio segments.

5.4 Signal-Domain Channel Coding with Frame Expansions

Let us revisit the operation of a conventional channel code over an erasure channel. For concreteness, consider a linear systematic block code over $\mathbb{Z}_{2^k}$ that maps N-tuples to M-tuples. If the code is good, it is possible to recover the N information symbols from any N of the M transmitted symbols. As detailed in Section 5.1.1, this is all that is needed from a set of codes to asymptotically achieve the capacity of a memoryless channel with large N and M. The capacity is achieved if N and M are chosen such that exactly N symbols are received with high probability. But what happens when more or less than N symbols are received? When M is not huge or the erasures are bursty or time-varying, this is frequently the case.

When more than N transmitted symbols are received, those in excess of N are useless; thus, the full throughput of the channel has not been utilized. One plausible reaction is to wish that the value of M had been lower, but this would certainly increase the probability of receiving *less* than N symbols. Another reaction is to wish that the received symbols contained more information about the source. With N symbols from a discrete source this does not make any sense, but if there is an underlying continuous source, then it may be advantageous for each received symbol (even in excess of N) to contain some independent information about the source.

On the other hand, when less than N symbols are received, it is usually not possible to determine all of the information symbols.[23] This is generally considered a failed communication attempt, without further ado. One might try to recover as many of the information symbols as possible. Techniques for this are limited, and in practice, recovering the received portion of the systematic part of the code is often all that is done.

In order to achieve good average performance and robustness to channel variation without very large block lengths, it is desirable to have channel protection that works well with any number of received packets. This is precisely in the spirit of multiple description coding: we are interested in the performance when any subset of the M transmitted symbols is received, instead of just subsets of size N. A method for joint source–channel coding that is very similar to an (M, N) block code is presented in this section. The difference is that the expansion from N to M dimensions is done in the original, continuous domain of the source. The resulting representation is then scalar quantized. This new technique partially remedies the weaknesses of using a discrete block code when more or less than N symbols are received.

5.4.1 Intuition

Consider the communication of a source taking values in $\mathbb{R}^N$ across an erasure channel. Denote the channel alphabet by $\mathcal{X}$ and suppose $|\mathcal{X}| = 2^{(N/M)R}$ where M is the number of channel uses per N-tuple and R is the overall rate per component (including channel coding). The objective of this section is to compare the following two techniques:

- **Conventional system:** Each component of the source vector is quantized using an $(N/M)R$-bit quantizer, giving N codewords. A linear systematic block code $\mathcal{X}^N \to \mathcal{X}^M$ is applied to the quantizer output, and the M resulting codewords are sent on the channel.
- **Quantized frame system:** The source vector is expanded using a linear transform $F : \mathbb{R}^N \to \mathbb{R}^M$. Each transform coefficient is quantized using an $(N/M)R$-bit quantizer. The M resulting codewords are sent on the channel.

Since a linear block code is a linear transform, the difference between the systems is the swapping of transform and quantization operations. In contrast to Section 5.3 and Appendix 5.B, the transform is rectangular with more rows than columns. The second system uses a quantized frame expansion, as analyzed in Chapter 2.

The conventional system works by producing a linear dependence between the transmitted symbols. A valid transmitted M-tuple must lie in a specified N-dimensional subspace of $\mathcal{X}^M$. When N or more symbols are received, they are consistent with exactly one valid element of $\mathcal{X}^M$, so the information symbols are known.

[23] It is definitely not possible for *all* transmitted N-tuples to be distinguishable from less than N received symbols. Whether it is ever possible to determine the information N-tuple from less than N received symbols will not be considered here.

This works very well when exactly N of the M transmitted codewords are received. However, when more or less than N codewords are received, there either is no benefit from the extra information or it is difficult to recover partial information about the source vector.

The quantized frame (QF) system has a similar way of adding redundancy. Denote the signal vector by x. The expanded signal $y = Fx$ has a linear dependence between its components. Thus, if N or more components of y are known, x can be recovered exactly. However, the components of y are not directly transmitted; it is the quantization that makes the two systems different. Quantization makes the components of $\hat{y} = Q(y)$ linearly independent, so each component of $\hat{y}$—even in excess of N—gives distinct information on the value of x. It is known that from a source-coding point of view, a quantized frame expansion (with linear reconstruction) is not competitive with a basis expansion (see [221, 222] and Section 2.2.2.5). Here the "baseline" fidelity is given by a basis expansion, and the noise reduction property of frames (see Proposition 2.2) improves the fidelity when we are "lucky" to receive more than N components.

One should not get the impression that the QF system is automatically as good as the conventional system when N components are received and better when more than N are received. The comparison is more subtle because all basis expansions are not equally good. The conventional system can use the best orthogonal basis (the KLT) or at least an orthogonal basis. On the other hand, it is not possible to make all N element subsets of the frame associated with F orthogonal.[24] Quantizing in a nonorthogonal basis is inherently suboptimal [32]. Comparisons appear in Section 5.4.3.3.

When less than N components are received, the QF representation fails to localize x to a finite cell. Neglect quantization error for the moment, and assume $k < N$ components are received. $\mathbb{R}^N$ can be decomposed into a k-dimensional subspace and an $(N-k)$-dimensional perpendicular subspace such that the component of x in the k-dimensional subspace is completely specified and the component in the perpendicular subspace is unknown. In many applications the source is known to have mean zero, so the component in the perpendicular subspace can be estimated as zero. Thus, the reconstruction process may follow the same linear algebraic calculations for any $k < N$ received components without a distinction between systematic and parity parts of the code.

The QF system will be analyzed in detail in two steps. The first analysis, presented in Section 5.4.2, assumes that the quantization error is an additive white noise, independent of the source, and that a linear reconstruction is used. Expressions for the mean-squared error are determined which depend on the number of erased components and angles between erased components of the frame. These expressions are valid only when the received components comprise a frame. Section 5.4.3 considers the communication of a white Gaussian source and fixes the quantization to be unbounded and uniform. This facilitates a specific numerical comparison between the conventional and QF systems using the earlier analysis.

5.4.2 Effect of Erasures in Tight Frame Representations

Let $\Phi = \{\varphi_k\}_{k=1}^M \subset \mathbb{R}^N$ be a tight frame with $\|\varphi_k\| = 1$ for all k, and let F be the frame operator associated with Φ. A source vector $x \in \mathbb{R}^N$ is represented by $\hat{y} = Q(y)$, where $y = Fx$ and Q is a scalar quantizer. Since the components of $\hat{y}$ will be used as "descriptions" in a multiple description system, we are interested in the distortion incurred in reconstruction from a subset of the components of $\hat{y}$.

[24] Frame terminology from Section 2.2.1.1 will be used frequently. The reader may wish to review Section 2.2 before continuing.

We will model $\eta = \hat{y} - y$ as a white noise independent of x with component variances σ_η^2. This is a common model, though η is actually completely determined by x when Q is a deterministic quantizer. If subtractive dithered uniform quantization with step size Δ is used, the model is precisely valid with $\sigma_\eta^2 = \Delta^2/12$. We will ignore the distribution of the quantization error and use linear reconstruction strategies that minimize the residual norm $\|\hat{y} - F\hat{x}\|_2$. This is tantamount to assuming that the p.d.f. of η has unbounded support (see Appendix 2.C).

In the multiple description system, there are M channels, each of which carries one coefficient $\langle x, \varphi_k\rangle + \eta_k$. If there are no erasures, the minimum MSE reconstruction uses the dual frame $\widetilde{\Phi} = (N/M)\Phi$; the resulting reconstruction error is given by[25]

$$\mathrm{MSE}_0 \;=\; E[N^{-1}\|x - \hat{x}\|^2] \;=\; \frac{N}{M}\sigma_\eta^2. \tag{5.57}$$

(The MSE with e erasures has been denoted MSE_e.)

If there are erasures, the reconstruction strategy should be different. Let E denote the index set of the erasures; *i.e.*, suppose $\{\langle x, \varphi_k\rangle + \eta_k\}_{k\in E}$ are erased. If $\Phi' = \Phi \setminus \{\varphi_k\}_{k\in E}$ is a frame, the minimum MSE estimate is obtained with the dual frame of Φ'; otherwise, x can only be estimated to within a subspace and distributional knowledge is needed to get a good estimate. Until Section 5.4.3, we assume that Φ' is a frame; hence, $e \le M - N$.

An estimate of the MSE can be obtained by bounding the frame bounds of Φ' and applying Proposition 2.2. However, this gives a very pessimistic estimate. Deleting a single element of Φ can reduce the lower frame bound from M/N by 1. Then Proposition 2.2 gives

$$\mathrm{MSE}_1 \le \frac{(M-1)N}{(M-N)^2}\sigma_\eta^2,$$

which may be much worse than the exact value

$$\mathrm{MSE}_1 = \left(1 + \frac{1}{M-N}\right)\frac{N}{M}\sigma_\eta^2. \tag{5.58}$$

The exact value is derived below by making computations similar to those in the proof of Proposition 2.2 (see Appendix 2.A.2).

5.4.2.1 One erasure

Rather then starting immediately with the general case, a detailed analysis is given which yields (5.58). Since the numbering of the frame elements is arbitrary, we can assume for notational simplicity that the erased component is $\langle x, \varphi_1\rangle + \eta_1$. Denote the frame operator associated with $\Phi_1 = \Phi \setminus \varphi_1$ by F_1 and use the shorthand $\varphi = [\varphi_1]$. (When more than one erasure is considered, φ will be matrix with e columns.) We will assume that Φ_1 is a frame. This will always be the case when M is larger than N because the lower frame bound of Φ_1 is $M/N - 1$, which is positive.

The frame operator associated with the new frame is

$$F_1 = [\varphi_2, \varphi_3, \ldots, \varphi_M]^*,$$

[25] See Corollary 2.3 and note that the distortion is measured per component.

while the elements of the dual frame of Φ_1 are

$$\widetilde{\varphi}_{1_k} = (F_1^* F_1)^{-1}\varphi_k, \quad k = 2, 3, \ldots, M.$$

The expansion that was, in effect, used was

$$x = \sum_{k=2}^{M} \langle x, \varphi_k \rangle \widetilde{\varphi}_{1_k},$$

and the minimum MSE reconstruction is

$$\hat{x} = \sum_{k=2}^{M} (\langle x, \varphi_k \rangle + \eta_k) \widetilde{\varphi}_{1_k}.$$

The error is

$$\hat{x} - x = \sum_{k=2}^{M} \eta_k \widetilde{\varphi}_{1_k}. \tag{5.59}$$

Taking the norm of (5.59) and using the independence of the η_k's leads to

$$\text{MSE}_1 \;=\; N^{-1}\sigma_\eta^2 \sum_{k=2}^{M} \|\widetilde{\varphi}_{1_k}\|^2 \;=\; N^{-1}\sigma_\eta^2 \sum_{k=2}^{M} \|(F_1^* F_1)^{-1}\varphi_k\|^2. \tag{5.60}$$

To evaluate further we will need to simplify $(F_1^* F_1)^{-2}$. First, note the following identity:

$$(A - BCD)^{-1} = A^{-1} + A^{-1}B(C^{-1} - DA^{-1}B)^{-1}DA^{-1}. \tag{5.61}$$

Now using

$$F_1^* F_1 \;=\; F^* F - \varphi_1 \varphi_1^* \;=\; \frac{M}{N} I - \varphi_1 \varphi_1^*$$

and identity (5.61), with $A = N^{-1} M\, I_N$, $B = \varphi_1$, $C = 1$, and $D = \varphi_1^*$, yields

$$(F_1^* F_1)^{-1} \;=\; \left(\frac{M}{N} I - \varphi_1 \varphi_1^*\right)^{-1} \;=\; \frac{N}{M} I + \frac{N^2}{M(M-N)} \varphi_1 \varphi_1^*.$$

Bearing in mind that $\varphi_1^* \varphi_1 = 1$, we can now compute

$$\begin{aligned}
(F_1^* F_1)^{-2} &= \left(\frac{N}{M} I + \frac{N^2}{M(M-N)} \varphi_1 \varphi_1^*\right)\left(\frac{N}{M} I + \frac{N^2}{M(M-N)} \varphi_1 \varphi_1^*\right) \\
&= \frac{N^2}{M^2} I + \frac{2N^3}{M^2(M-N)} \varphi_1 \varphi_1^* + \frac{N^4}{M^2(M-N)^2} \varphi_1 \varphi_1^* \varphi_1 \varphi_1^* \\
&= \frac{N^2}{M^2} I + \frac{N^3(2M-N)}{M^2(M-N)^2} \varphi_1 \varphi_1^*.
\end{aligned}$$

Substituting this into (5.60) gives

$$\begin{aligned}
\text{MSE}_1 &= N^{-1}\sigma_\eta^2 \sum_{k=2}^{M} \|(F_1^* F_1)^{-1}\varphi_k\|^2 \\
&= N^{-1}\sigma_\eta^2 \sum_{k=2}^{M} \varphi_k^* (F_1^* F_1)^{-2} \varphi_k
\end{aligned}$$

$$
\begin{aligned}
&= N^{-1}\sigma_\eta^2 \sum_{k=2}^{M} \varphi_k^* \left(\frac{N^2}{M^2} I + \frac{N^3(2M-N)}{M^2(M-N)^2} \varphi_1 \varphi_1^* \right) \varphi_k \\
&= \sigma_\eta^2 \left(\frac{N(M-1)}{M^2} + \frac{N^2(2M-N)}{M^2(M-N)^2} \varphi_1^* F_1^* F_1 \varphi_1 \right) \\
&= \sigma_\eta^2 \left(\frac{N(M-1)}{M^2} + \frac{N^2(2M-N)}{M^2(M-N)^2} \varphi_1^* \left(\frac{M}{N} I - \varphi_1 \varphi_1^* \right) \varphi_1 \right) \\
&= \sigma_\eta^2 \left(\frac{N(M-1)}{M^2} + \frac{N^2(2M-N)}{M^2(M-N)^2} \left(\frac{M}{N} - 1 \right) \right) \\
&= \frac{\sigma_\eta^2 N}{M} \left(1 + \frac{1}{M-N} \right) \\
&= \left(1 + \frac{1}{M-N} \right) \mathrm{MSE}_0.
\end{aligned}
$$

The effect of any single erasure on the MSE is simply multiplication by the bracketed term. Note that dividing by $M-N$ is not a problem because when $M = N$, Φ_1 is not a frame and the entire analysis using the dual frame is invalid.

5.4.2.2 e Erasures

Assume now that there are e erasures and that the elements have been renumbered so that the erasures occurred at positions 1, 2, ..., e. The matrix φ is now defined as $\varphi = [\varphi_1, \ldots, \varphi_e]$. All the notation from Section 5.4.2.1 carries over with subscript 1 replaced by e. The MSE is now

$$
\mathrm{MSE}_e = N^{-1}\sigma_\eta^2 \sum_{k=e+1}^{M} \|(F_e^* F_e)^{-1} \varphi_k\|^2 = N^{-1}\sigma_\eta^2 \sum_{k=e+1}^{M} \varphi_k^* (F_e^* F_e)^{-2} \varphi_k.
$$

Using $F_e^* F_e = N^{-1} M\, I_N - \varphi\varphi^*$ and (5.61) we can write

$$
\begin{aligned}
(F_e^* F_e)^{-1} &= \frac{N}{M} I_N + \frac{N}{M} I_N \varphi \left(I_e - \varphi^* \frac{N}{M} I_N \varphi \right)^{-1} \varphi^* \frac{N}{M} I_N, \\
&= \frac{N}{M} I_N + \frac{N^2}{M^2} \varphi \left(I_e - \frac{N}{M} \varphi^* \varphi \right)^{-1} \varphi^*. \qquad (5.62)
\end{aligned}
$$

Note that the matrix to be inverted is $e \times e$ and that it depends only on the inner products between erased components of the original tight frame, not on the surviving elements.

There are two ways one could evaluate the above expression. The first is to iteratively compute $F_i^* F_i = F_{i-1}^* F_{i-1} - \varphi_i \varphi_i^*$ and then invert $(F_i^* F_i)^{-1}$ using the identity (5.61). However, as e increases it seems simpler to directly use (5.62); this is what we will do below.

Let

$$
\begin{aligned}
A^{(e)} &= \left(I_e - \frac{N}{M} \varphi^* \varphi \right)^{-1}, \\
B^{(e)} &= 2A^{(e)} + \frac{N}{M} A^{(e)} \varphi^* \varphi A^{(e)}, \\
C^{(e)} &= \left(\frac{M}{N} \varphi^* \varphi - \varphi^* \varphi \varphi^* \varphi \right)^T.
\end{aligned}
$$

Since

$$(F_e^* F_e)^{-2} = \frac{N^2}{M^2} I + \frac{N^3}{M^3} \varphi B^{(e)} \varphi^*$$

and

$$\sum_{k=e+1}^{M} \varphi_k^* \varphi B^{(e)} \varphi^* \varphi_k = \sum_{i,j=1}^{e} B_{ij}^{(e)} C_{ij}^{(e)},$$

we can compute

$$\sum_{k=e+1}^{M} \varphi_k^* (F_e^* F_e)^{-2} \varphi_k \;=\; \frac{N^2}{M^2} \cdot (M-e) + \frac{N^3}{M^3} \sum_{i,j=1}^{e} B_{ij}^{(e)} C_{ij}^{(e)}.$$

With the required normalizations we finally obtain

$$\mathrm{MSE}_e = \left[1 - \frac{e}{M} + \frac{N}{M^2} \sum_{i,j=1}^{e} B_{ij}^{(e)} C_{ij}^{(e)} \right] \mathrm{MSE}_0. \tag{5.63}$$

Computations for $e = 2, 3, 4$ suggest that (5.63) can be written as

$$\mathrm{MSE}_e = \left[1 + \frac{e(m-n)^{e-1} + Y_e}{(m-n)^e - Z_e} \right] \mathrm{MSE}_0, \tag{5.64}$$

where Y_e and Z_e depend only on the inner products between erased vectors.

Two Erasures The expression (5.63) is applicable to any number of erasures, as long as the remaining vectors Φ_e form a frame. This computation is made here explicitly for $e = 2$. Let $\alpha = \varphi_1^* \varphi_2$. Then

$$\varphi^* \varphi \;=\; \begin{bmatrix} \varphi_1^* \varphi_1 & \varphi_1^* \varphi_2 \\ \varphi_2^* \varphi_1 & \varphi_2^* \varphi_2 \end{bmatrix} \;=\; \begin{bmatrix} 1 & \alpha \\ \bar{\alpha} & 1 \end{bmatrix}$$

and so

$$A^{(2)} \;=\; \left(I_e - \frac{N}{M} \varphi^* \varphi \right)^{-1} \;=\; \frac{M}{(M-N)^2 - N^2|\alpha|^2} \begin{bmatrix} M-N & N\alpha \\ N\bar{\alpha} & M-N \end{bmatrix}.$$

Furthermore,

$$B^{(2)} = \frac{M}{((M-N)^2 - N^2|\alpha|^2)^2} \begin{bmatrix} b_1 & b_2 \\ \bar{b_2} & b_1 \end{bmatrix}, \tag{5.65}$$

where

$$\begin{aligned} b_1 &= 2M^3 - 5M^2 N + 4MN^2 - N^3 + N^3|\alpha|^2, \\ b_2 &= N\alpha(3M^2 - 4MN + N^2 - N^2|\alpha|^2), \end{aligned}$$

and

$$C^{(2)} = \begin{bmatrix} N^{-1}M - 1 - |\alpha|^2 & \bar{\alpha}\left(N^{-1}M - 2\right) \\ \alpha\left(N^{-1}M - 2\right) & N^{-1}M - 1 - |\alpha|^2 \end{bmatrix}. \tag{5.66}$$

Substituting (5.65) and (5.66) in (5.63) yields

$$\mathrm{MSE}_2 = \left(1 + \frac{2(M-N) + 2N^2|\alpha|^2}{(M-N)^2 - N^2|\alpha|^2} \right) \mathrm{MSE}_0.$$

This matches (5.64) with $Y_2 = 2N^2|\alpha|^2$ and $Z_2 = N^2|\alpha|^2$.

Three Erasures Denote the inner products between three erased frame elements by α, β, and γ. The MSE expression (5.64) again holds, with

$$\begin{aligned} Y_3 &= (2M-3N)N(|\alpha|^2+|\beta|^2+|\gamma|^2)+3N^2(\bar{\alpha}\beta\bar{\gamma}+\alpha\bar{\beta}\gamma), \\ Z_3 &= (M-N)N^2(|\alpha|^2+|\beta|^2+|\gamma|^2)+N^3(\alpha\bar{\beta}\gamma+\bar{\alpha}\beta\bar{\gamma}). \end{aligned}$$

Four Erasures As a final example, consider the case of four erasures. Denote the six inner products in $\varphi^*\varphi$ by α, β, γ, δ, ϵ, and ζ. A long computation yields

$$\begin{aligned} Y_4 &= N[2K(M-N)(M-2N)+4(-\bar{\gamma}\bar{\delta}\gamma\delta+\bar{\beta}\bar{\epsilon}\gamma\delta-\bar{\beta}\alpha\delta+\bar{\gamma}\alpha\delta\zeta-\bar{\gamma}\alpha\epsilon+\bar{\gamma}\bar{\delta}\beta\epsilon+\bar{\alpha}\bar{\epsilon}\beta\zeta \\ &\quad -\bar{\gamma}\beta\zeta-\bar{\beta}\gamma\bar{\zeta}+\bar{\alpha}\bar{\delta}\gamma\bar{\zeta}-\bar{\alpha}\bar{\delta}\beta-\bar{\alpha}\alpha\zeta\bar{\zeta}-\bar{\delta}\epsilon\bar{\zeta}-\bar{\beta}\bar{\epsilon}\beta\epsilon+\bar{\beta}\alpha\epsilon\bar{\zeta}-\bar{\alpha}\bar{\epsilon}\gamma-\bar{\epsilon}\delta\zeta)N^2 \\ &\quad +3(\bar{\gamma}\alpha\epsilon+\bar{\beta}\alpha\delta+\bar{\gamma}\beta\zeta+\bar{\delta}\epsilon\bar{\zeta}+\bar{\epsilon}\delta\zeta+\bar{\alpha}\bar{\delta}\beta+\bar{\alpha}\bar{\epsilon}\gamma+\bar{\beta}\gamma\bar{\zeta})MN], \\ Z_4 &= N^2[K(M-N)^2+(-\bar{\gamma}\bar{\delta}\gamma\delta+\bar{\beta}\bar{\epsilon}\gamma\delta-\bar{\beta}\alpha\delta+\bar{\gamma}\alpha\delta\zeta-\bar{\gamma}\alpha\epsilon+\bar{\gamma}\bar{\delta}\beta\epsilon+\bar{\alpha}\bar{\epsilon}\beta\zeta-\bar{\gamma}\beta\zeta \\ &\quad -\bar{\beta}\gamma\bar{\zeta}+\bar{\alpha}\bar{\delta}\gamma\bar{\zeta}-\bar{\alpha}\bar{\delta}\beta-\bar{\alpha}\alpha\zeta\bar{\zeta}-\bar{\delta}\epsilon\bar{\zeta}-\bar{\beta}\bar{\epsilon}\beta\epsilon+\bar{\beta}\alpha\epsilon\bar{\zeta}-\bar{\alpha}\bar{\epsilon}\gamma-\bar{\epsilon}\delta\zeta)N^2 \\ &\quad +(\bar{\gamma}\alpha\epsilon+\bar{\beta}\alpha\delta+\bar{\gamma}\beta\zeta+\bar{\delta}\epsilon\bar{\zeta}+\bar{\epsilon}\delta\zeta+\bar{\alpha}\bar{\delta}\beta+\bar{\alpha}\bar{\epsilon}\gamma+\bar{\beta}\gamma\bar{\zeta})MN]. \end{aligned}$$

There is no obvious pattern in $\{Y_1, Y_2, Y_3, Y_4\}$ or $\{Z_1, Z_2, Z_3, Z_4\}$, so (5.63) may be the best general expression for the MSE.

Orthogonal Erasures A simple special case is when the erased components are pairwise orthogonal. In this case, $\varphi^*\varphi = I_e$, and $A^{(e)}$, $B^{(e)}$, and $C^{(e)}$ reduce to

$$\begin{aligned} A^{(e)} &= \frac{M}{M-N}I_e, \\ B^{(e)} &= \frac{M(2M-N)}{(M-N)^2}I_e, \\ C^{(e)} &= \frac{M-N}{N}I_e. \end{aligned}$$

Substituting in (5.63) gives

$$\mathrm{MSE}_e = \left(1+\frac{e}{M-N}\right)\mathrm{MSE}_0.$$

5.4.2.3 Comments

A linear reconstruction method has been assumed throughout. In light of the many results of Chapter 2, consistent reconstruction may reduce the MSE by a factor of N/M. However, most of the consistent reconstruction results are for large M/N. For the multiple description coding application, the interesting values of M/N are small and the improvement from consistent reconstruction is less than the asymptotic predictions.

The analysis presented thus far makes no assumptions about the source and instead makes strong assumptions about the quantization error. In effect, it is a distortion-only analysis; since the source has not entered the picture, there is no relationship between σ_η^2 and the rate. This deficiency is remedied in the following section.

5.4.3 Performance Analyses and Comparisons

Let x be a zero-mean, white, Gaussian vector with covariance matrix $R_x = \sigma^2 I_N$. This source is convenient for analytical comparisons between the QF system and a conventional system with a block channel code. Entropy-coded uniform quantization (ECUQ) will be used in both systems. The distortion–rate performance of ECUQ on a Gaussian variable with variance σ^2 will be denoted $D_{\sigma^2}(R)$. This function directly describes the performance of the conventional communication system when the channel code is successful in eliminating the effect of erasures and is also useful in describing the performance of the QF system. (A high rate approximation of $D_{\sigma^2}(R)$ is given by (1.3) and a plot is given in Figure 1.4.)

5.4.3.1 Performance of conventional system

Let us first analyze the conventional system. For coding at a total rate of R bits per component of x (including channel coding), NR bits are split among M descriptions. Thus the overall average distortion per component with e erasures is

$$\bar{D}_e = D_{\sigma^2}\Big(\frac{NR}{M}\Big), \qquad \text{for} \quad e = 0,\, 1,\, \ldots,\, M-N. \tag{5.67}$$

When $e > M - N$, the channel code cannot correct all of the erased information symbols. Since the code is systematic, the decoder will have received some number of information symbols and some number of parity symbols. Assume that the decoder discards the parity symbols and estimates the erased symbols by their means. Denoting the number of erased information symbols by e_s, the average distortion is then

$$\widetilde{D}_{e_s} = \frac{e_s}{N}\sigma^2 + \frac{N-e_s}{N} D_{\sigma^2}\Big(\frac{NR}{M}\Big), \qquad \text{for} \quad e = M-N+1,\, \ldots,\, M-1,\, M. \tag{5.68}$$

As it is, (5.68) does not completely describe the average distortion because the relationship between e and e_s is not specified. In fact, given that there are e total erasures, e_s is a random variable. There are $\binom{M}{e}$ ways that e erasures can occur and we assume these to be equally likely. The probability of k erased information symbols is then

$$P\Big(e_s = k \,\Big|\, e-(M-N) \le k \le \min(e,N)\Big) = \binom{M}{e}^{-1}\binom{N}{k}\binom{M-N}{e-k}. \tag{5.69}$$

Using this with (5.68) gives the average distortion per component as

$$\begin{aligned}
\bar{D}_e &= \sum_{k=e-(M-N)}^{\min(e,N)} P(e_s = k \mid e \text{ total erasures})\widetilde{D}_k \\
&= \binom{M}{e}^{-1} \sum_{e_s=e-(M-N)}^{\min(e,N)} \binom{N}{e_s}\binom{M-N}{e-e_s}\left[\frac{e_s}{N}\sigma^2 + \frac{N-e_s}{N} D_{\sigma^2}\Big(\frac{NR}{M}\Big)\right], \\
&\qquad\qquad \text{for} \quad e = M-N+1,\, \ldots,\, M-1,\, M,
\end{aligned} \tag{5.70}$$

because the received components of x are subject to quantization error and the erased components have variance σ^2.

There is no denying that discarding the parity symbols is not the optimal reconstruction strategy—to minimize MSE or probability of error. However, it comes close to minimizing the MSE, and actually minimizing the MSE seems computationally difficult. Investigation of a couple of cases provides a credible, though

incomplete, justification for discarding the parity. Consider $e = M - N + 1$, one more erasure than can be corrected. One extreme case is $e_s = 1$, where all the parity symbols are erased. In this case there is no parity information, so estimating the erased information symbol by its mean is clearly the best that can be done. In the other extreme case, $e_s = e$ and all the parity is received. For convenience, number the erased components so that $\hat{x}_1$, $\hat{x}_2$, …, $\hat{x}_e$ are lost. If a single one of these was known, then the rest could be determined because the code can correct $e - 1$ erasures. So a possible decoding method is as follows: For each possible value of $\hat{x}_1$, determine $\hat{x}_2$, $\hat{x}_3$, …, $\hat{x}_e$. Since the $\hat{x}_i$'s are independent, it is easy to compute the probabilities of each of the $[\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_e]^T$ vectors. The centroid of the vectors gives the minimum MSE estimate.

There are two main difficulties with this computation. Firstly, the number of possibilities to enumerate is exponential in $e - (M - N)$—namely $|\mathcal{X}|^{e-(M-N)}$, where $\mathcal{X}$ is the alphabet for $\hat{x}$—and may be very large. More importantly, it may simply not be useful to compute the probability density of the possible vectors. The nature of the channel code is to make values in each possible $[\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_e]^T$ vector more or less uniform. Thus the minimum MSE estimate is often close to simply estimating each component by its mean.

5.4.3.2 Performance of quantized frame system

When F is the frame operator associated with a normalized tight frame Φ, each component of $y = Fx$ is Gaussian with mean zero and variance σ^2. Thus D_{σ^2}, as defined previously, can again be used to describe the distortion–rate characteristics of the quantized coefficients (y_i's). These distortions, however, do not equal the component distortions in the reconstruction of x because of the use of frames and nonorthogonal bases.

We assume the frame is designed such that all subsets of at least N elements form a frame. Then when there are at most $M - N$ erasures, we can approximate the distortion using the expressions from Section 5.4.2. Specifically, using (5.57) and noting that D_{σ^2} connects the coding rate to the quantization noise power σ_η^2, we obtain

$$\bar{D}_0 = \frac{N}{M} D_{\sigma^2}\Big(\frac{NR}{M}\Big), \tag{5.71}$$

which is better than the performance of the conventional system. With erasures, there is no simple closed form for the distortion, but it can be written as

$$\bar{D}_e = c_e D_{\sigma^2}\Big(\frac{NR}{M}\Big), \qquad \text{for} \quad e = 1, 2, \ldots, M - N.$$

The constant c_e is M/N times the average of the bracketed term of (5.63), where the average is taken over all possible erasure patterns. With a given frame, additional measurements always reduce the average reconstruction error, so $\{c_e\}_0^{M-N}$ is an increasing sequence.

When there are more than $N - M$ erasures, the decoder has less than a basis representation of x. Let E be the index set of the erasures; *i.e.*, suppose $\{\hat{y}_k\}_{k\in E}$ are erased. The source vector x can be orthogonally decomposed as

$$x = x_S + x_{S^\perp} \qquad \text{where} \quad x_S \in S = \operatorname{span}(\{\varphi_k\}_{k\notin E}).$$

Since the source is white and Gaussian, x_S and $x_{S^\perp}$ are independent. Thus not only does the decoder have no direct measurement of $x_{S^\perp}$, but it has absolutely no way to estimate it aside from using its mean. Estimating $x_{S^\perp} = 0$ introduces a distortion of $N^{-1}(e-(M-N))\sigma^2$ because the dimension of $S^\perp$ is $e-(M-N)$. The received coefficients $\{\hat{y}_k\}_{k\notin E}$ provide a quantized basis representation of x_S. The basis will generally be a nonorthogonal

	Number of erasures	
	$e \leq M-N$	$e > M-N$
Conventional system	Block code corrects all erasures Orthogonal basis expansion $\bar{D}_e = D_{\sigma^2}(\frac{NR}{M})$	e_s-dim. subspace lost, $e_s \geq e-(M-N)$ Orthogonal basis expansion $\widetilde{D}_{e_s} = \frac{e_s}{N}\sigma^2 + \frac{N-e_s}{N}D_{\sigma^2}(\frac{NR}{M})$
Quantized frame system	Representation covers full space Frame expansion $\bar{D}_e = c_e D_{\sigma^2}(\frac{NR}{M})$	k-dim. subspace lost, $k = e-(M-N)$ Nonorthogonal basis expansion $\bar{D}_e = \frac{k}{N}\sigma^2 + \frac{N-k}{N}c_e D_{\sigma^2}(\frac{NR}{M})$

Table 5.3: Comparison of systems for communicating a white Gaussian N-tuple source across an erasure channel. The conventional system uses scalar quantization in an orthonormal basis and an (M, N) block code. The quantized frame system uses an $M \times N$ tight frame operator followed by scalar quantization. Reconstruction methods are described in the text. The constants c_e depend on the noise reduction of frames and non-cubic cell shapes.

basis, so the per component distortion will exceed $D_{\sigma^2}(NR/M)$ by a constant factor which depends on the skew of the basis. Thus we conclude

$$\bar{D}_e = \frac{e-(M-N)}{N}\sigma^2 + \frac{M-e}{N}c_e D_{\sigma^2}\Big(\frac{NR}{M}\Big), \quad \text{for} \quad e = M-N+1, \ldots, M-1, M.$$

The constant factor c_e can be computed through calculations similar to those in Appendix 2.A.2. It is always larger than 1, since it is not possible for all subsets of a given size of the frame to be orthogonal.

5.4.3.3 Numerical comparison

The findings of Sections 5.4.3.1 and 5.4.3.2 are summarized in Table 5.3. These expressions are not necessarily very enlightening by themselves, so let us construct a simple example with a specific frame.

Let $N = 4$ and $M = 5$ and let Φ be a real harmonic tight frame similar to those constructed in Section 2.2.1.2. The frame operator is given explicitly by

$$F = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ \cos\frac{\pi}{5} & \cos\frac{3\pi}{5} & \sin\frac{\pi}{5} & \sin\frac{3\pi}{5} \\ \cos\frac{2\pi}{5} & \cos\frac{6\pi}{5} & \sin\frac{2\pi}{5} & \sin\frac{6\pi}{5} \\ \cos\frac{3\pi}{5} & \cos\frac{9\pi}{5} & \sin\frac{3\pi}{5} & \sin\frac{9\pi}{5} \\ \cos\frac{4\pi}{5} & \cos\frac{12\pi}{5} & \sin\frac{4\pi}{5} & \sin\frac{12\pi}{5} \end{bmatrix}.$$

The channel code for the conventional system can simply form a parity symbol by the $\mathbb{Z}_2$ addition of the four information symbols.

The performance of the conventional system is easy to calculate using (5.67) and (5.70). For the quantized frame system, $\bar{D}_0$ and $\bar{D}_1$ are given by (5.71) and (5.58), respectively. The remaining distortions require a computation of c_e averaged over all erasure patterns. The results of these computations are given in Table 5.4.

At high rates, $D_{\sigma^2}(4R/5) \ll \sigma^2$. Thus, except when there is one erasure, the quantized frame system has lower distortion. It is not surprising that the conventional system is best when $e = M-N$ because in this

	Conventional system	Quantized frame system
D_0	$D_{\sigma^2}(\frac{4}{5}R)$	$\frac{4}{5}D_{\sigma^2}(\frac{4}{5}R)$
$\bar{D}_1$	$D_{\sigma^2}(\frac{4}{5}R)$	$\frac{8}{5}D_{\sigma^2}(\frac{4}{5}R)$
$\bar{D}_2$	$\frac{2}{5}\sigma^2 + \frac{3}{5}D_{\sigma^2}(\frac{4}{5}R)$	$\frac{1}{4}\sigma^2 + \frac{9}{10}D_{\sigma^2}(\frac{4}{5}R)$
$\bar{D}_3$	$\frac{3}{5}\sigma^2 + \frac{2}{5}D_{\sigma^2}(\frac{4}{5}R)$	$\frac{2}{4}\sigma^2 + \frac{8}{15}D_{\sigma^2}(\frac{4}{5}R)$
$\bar{D}_4$	$\frac{4}{5}\sigma^2 + \frac{1}{5}D_{\sigma^2}(\frac{4}{5}R)$	$\frac{3}{4}\sigma^2 + \frac{1}{4}D_{\sigma^2}(\frac{4}{5}R)$
$\bar{D}_5$	σ^2	σ^2

Table 5.4: Comparison between conventional and quantized frame systems for $N = 4$, $M = 5$. The average distortion–rate performance is given for each possible number of erasures.

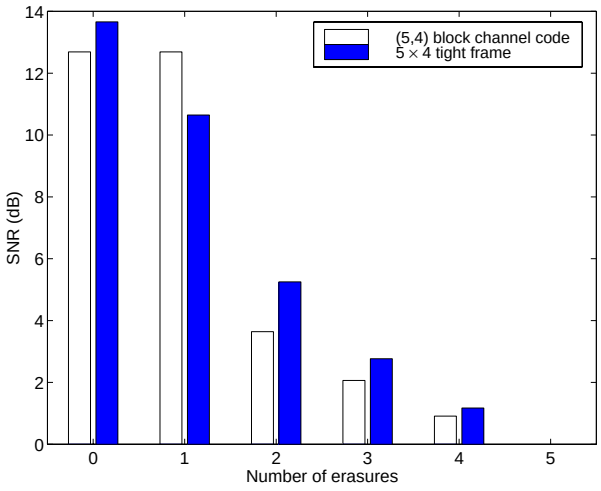


Figure 5.26: Comparison of conventional system and quantized frame system with $N = 4$, $M = 5$, and $R = 3$. The average signal-to-noise ratio is given for each system for possible number of erasures.

case it makes perfect use of the channel, conveying an orthogonal basis representation of the source with no wasted received data.

At lower rates, the systems can be compared only by numerical calculation. Figure 5.26 shows each of the distortions from Table 5.4 for rate $R = 3$. (Actually, signal-to-noise ratios are shown in order to normalize for σ^2.) At this rate, the SNR is higher for the quantized frame system except when there is one erasure. This is consistent with the high rate approximation. The flat performance of the conventional system with zero or one erasure, followed by a sharp drop-off with more than one erasure, is sometimes called the "cliff effect" of channel coding. The performance of the QF system degrades gracefully as the number of erasures increases.

To have a single comparison between the two systems, the performance with each number of erasures can be given weights. A natural weighting is to simply use the probability of that number of erasures, though other weights may be used. If the channel is memoryless with probability p of erasure, the probability of e erasures is

$$P(e \text{ erasures}) = \left(\begin{array}{c} M \\ e \end{array} \right) p^e (1-p)^{M-e}. \tag{5.72}$$

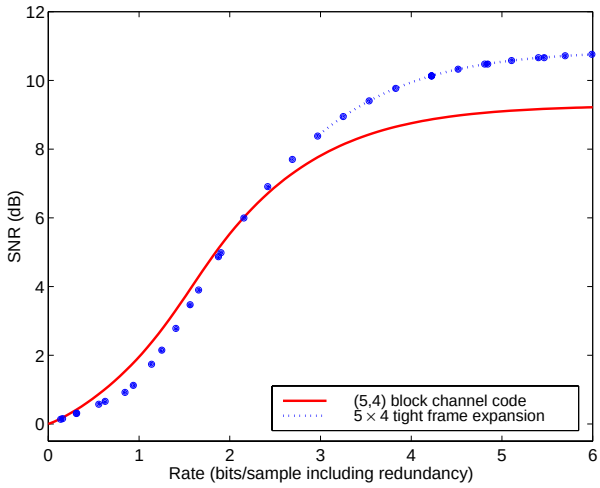


Figure 5.27: Comparison of conventional system and quantized frame system with $N = 4$, $M = 5$, and memoryless erasures with probability 1/5.

	Probability					
Symbols erased	0 of 5	1 of 5	2 of 5	3 of 5	4 of 5	5 of 5
Memoryless channel ($p = 1/5$)	0.32768	0.4096	0.2048	0.0512	0.0064	0.00032
Bursty channel ($p = 1/5$, $\beta = 6$)	0.52488	0.1944	0.1316	0.08	0.0432	0.02592

Table 5.5: Probabilities of the various numbers of erasures for a memoryless channel and a bursty channel with the same overall probability of erasure.

Using this weighting with $p = 1/5$ gives the overall average SNR's shown in Figure 5.27. In this figure, the performance of the conventional system and the high-rate performance of the quantized frame system are computed analytically using the expressions above. The low-rate performance of the QF system is simulated because the computations assuming uniformly distributed noise overestimate the distortion.

Of course, the erasures need not be independent. Many channels exhibit some sort of burstiness. A simple two-state bursty channel has β times higher probability of an erasure following an erasure than following a received symbol. Consider a channel with overall probability of erasure $p = 1/5$ as before, but with burstiness described by $\beta = 6$. The probabilities of different numbers of erasures for this channel are compared to those given by (5.72) in Table 5.5. Weighting by these probabilities gives the performance shown in Figure 5.28. The QF system works well on the bursty channel because the number of erasures is less clustered around the expected number of erasures.

As a final numerical comparison, let us again assume that the channel is memoryless but now fix the rate and allow the erasure probability to vary. Figure 5.29(a) shows the performance gain of the QF system as a function of the erasure probability on a memoryless channel. Results are given for a few rates. At higher rates, the QF system is better than the conventional system at all probabilities of erasure. At lower rates, there is a connected interval of erasure probabilities for which the conventional system is better. This comparison shows that the QF system is less sensitive to choosing the correct coding rate (N/M) to match the channel. For the bursty channel (see Figure 5.29(b)), the advantage of the QF system is more pronounced.

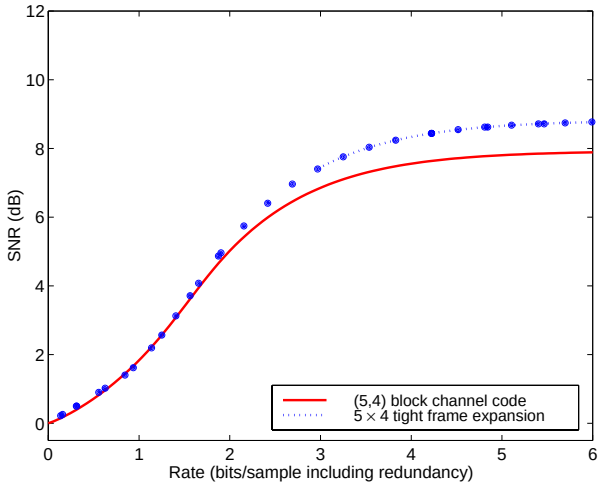


Figure 5.28: Comparison of conventional system and quantized frame system with $N = 4$, $M = 5$, and bursty erasure channel with $p = 1/5$, $\beta = 6$.

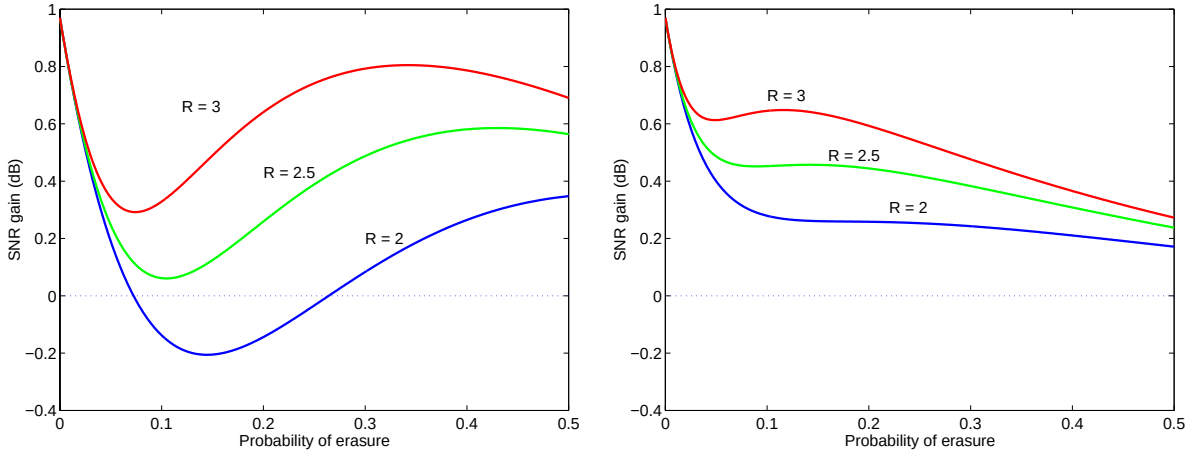


Figure 5.29: Comparison of conventional system and quantized frame system with $N = 4$, $M = 5$, and memoryless erasure channel. The plot shows the SNR advantage of the QF system as a function of the channel erasure probability: (a) memoryless channel; (b) bursty channel with $\beta = 6$.

5.4.3.4 Asymptotic behavior

The comparisons presented in Section 5.4.3 provide some insight into the performance of the quantized frame system, but are based on just a single example. The primary obstacle to making general statements about the value of the QF system is that the cell-shape constants (c_e's) depend on the frame, but there is no design procedure for the frame. This section makes a few comments on the asymptotic behavior of the QF system. Both high rate and large block length performance are considered.

In the limit of high rate, quantization error is negligible in comparison to the distortion caused by completely missing one orthogonal component of the source. The distortion goes to zero when there are at most $M-N$ erasures, but for larger numbers of erasures the distortion approaches $N^{-1}(e-(M-N))\sigma^2$. Compared to an unconstrained multiple description source coding scheme, the asymptotic performance with more than $M-N$ erasures is very poor. One could use M independent vector quantizers to form the M descriptions. In this case every side distortion would asymptotically approach zero. Such a scheme would presumably have high encoding complexity and high decoding complexity (in time, memory, or both); this is why we are interested in linear transform-based approaches.

Comparing the QF system to the conventional system at high rate, the QF system is better when there are more than $M-N$ erasures. In this case the QF system loses an $e-(M-N)$-dimensional part of the signal while the conventional system loses at least this much; averaging over all erasure patterns, the conventional system loses even more. For lower numbers of erasures, the relative performance depends on the factor c_e. This constant generally depends on the tight frame, but has a simple form in two cases: $c_0 = N/M$ and $c_1 = M^{-1}N(1+(M-N)^{-1})$. When the tight frame is not a basis ($M > N$), c_{M-N} must be larger than 1; it is not possible to design a tight frame such that all subsets of size N are orthogonal bases. $\{c_e\}_{e=0}^{M-N}$ is monotonic and crosses 1 somewhere between $e=0$ and $e=M-N$.

In information theory it is typical to look at performance limits as the block size grows without bound. In channel coding for a memoryless channel this makes the number of erasures predictable, by the law of large numbers. Using multiple description coding as an abstraction for coding for an erasure channel is in part an attempt to avoid large block sizes and to cope with unpredictable channels. Nevertheless, it is useful to understand the performance of the QF system with large block sizes.

A performance analysis must depend in some part on a choice of a set of frames. Intuition suggests that the best frame, at least for a white source, is one that uniformly covers the space. The following theorem shows that asymptotically, the most uniform frame approaches an orthonormal basis in a certain sense.

Theorem 5.10 *Suppose that $M/N = r$, with $1 < r < 2$. Let $\Phi = \{\varphi_k\}_{k=1}^{M}$ be a frame in $\mathbb{R}^N$. If the design of Φ is the packing of M lines in $\mathbb{R}^N$ such that the minimum angular separation is maximized, then as $N \to \infty$ (M increasing accordingly as $M = \lfloor rN \rfloor$) the elements of Φ become pairwise orthogonal.*

Proof: See Appendix 5.C.4. □

An upper bound on the constant c_{M-N} close to 1 would be useful in bounding the worst case performance of the QF system with respect to the conventional system. Theorem 5.10 suggests that if a frame is designed to maximize uniformity in the specified manner and any $M-N$ elements of the frame are deleted, the remaining set is approximately an orthonormal basis. Unfortunately, the convergence in Theorem 5.10 does not lead to small bounds on the c_e's. Numerical computations show that as M and N are increased with M/N

held constant, the constant factor c_{M-N} increases. This holds for harmonic frames as well as frames designed as in Theorem 5.10.

What we would like is for an arbitrary N element subset of the frame to not only be a basis, but to be a good basis for a quantized representation. This is related to bounding the eigenvalues of A^TA away from zero, where A is a matrix of basis vectors. For random bases generated according to a uniform distribution, the eigenvalues of A^TA cannot be bounded away from zero [106, 132]. This negative result is not surprising given that optimally "uniform" frames also fail to give this property. These negative results should not discourage the use of the QF system with small numbers of descriptions.

5.4.4 Application to Image Coding

As an example, we construct an image communication system that uses a quantized frame expansion as an alternative to a $(10, 8)$ block code. For the 10×8 frame operator F we use a matrix corresponding to a length-10 real Discrete Fourier Transform of a length-8 sequence (see Section 2.2.1.2). This can be constructed as $F = [F^{(1)}\ F^{(2)}]$, where

$$F^{(1)}_{ij} = \frac{1}{2} \cos \frac{\pi(i-1)(2j-1)}{10}, \quad 1 \le i \le 10,\ 1 \le j \le 4,$$

and

$$F^{(2)}_{ij} = \frac{1}{2} \sin \frac{\pi(i-1)(2j-1)}{10}, \quad 1 \le i \le 10,\ 1 \le j \le 4.$$

In order to profit from the psychovisual tuning that has gone into JPEG coding, we apply this technique to DCT coefficients and use quantization step sizes as in a typical JPEG coder. The coding proceeds as follows:

1. An 8×8 block DCT of the image is computed.
2. Vectors of length 8 are formed from DCT coefficients of like frequency, separated in space.
3. Each length 8 vector is expanded by left-multiplication with F.
4. Each length 10 vector is uniformly quantized with a step size depending on the frequency.

The baseline system against which we compare uses the same quantization step sizes, but quantizes the DCT coefficients directly and then applies a systematic $(10, 8)$ block code which can correct any two erasures. As before, it is assumed that if there are more than two erasures, only the systematic part of the received data is used.

The two systems were simulated with quantization step sizes conforming to a *quality* setting of 75 in the Independent JPEG Group's software.[26] For the *Lena* image, this corresponds to a rate of about 0.98 bits per pixel plus 25% channel coding. In order to avoid issues related to the propagation of errors in variable length codes, we consider an abstraction in which sets of coefficients are lost. The alternative would require explicitly forming ten entropy coded packets. The reconstruction for the frame method follows a least-squares strategy. For the baseline system, when eight or more of the ten descriptions arrive, the block code ensures that the image is received at full fidelity. The effect of having less than eight packets received is simulated using the probabilities (5.69).

[26] Version 6b of `cjpeg`. The current version is available at `ftp://ftp.uu.net/graphics/jpeg/`.

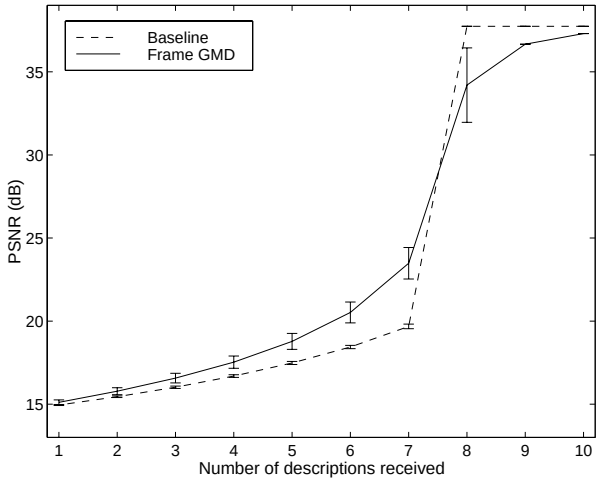


Figure 5.30: Numerical image coding results for quantized frame system. Results are for compressing *Lena* at about 1 bit per pixel, plus 25% channel coding. The mean performance is shown for each number of packets received along with the standard deviation.

Numerical results are shown in Fig. 5.30 for one through ten received packets. As expected, the frame system has better performance when less than eight packets are received. It is disappointing to see that the frame system did not have better performance when all ten packets are received, as was expected. Sample images are given in Fig. 5.31. (From the 512×512 images, 64×64 pixel detail images are shown.) Numerically and visually it is apparent that the performance of the MD system degrades gracefully as the number of lost packets increases.

A probable reason for the poor performance of the quantized frame system with all ten packets received is as follows: With the exception of the DC terms, the DCT coefficients are well-approximated by independent Laplacian random variables. These are particularly well suited to quantization in the standard basis. (Gaussian random variables, on the other hand, are invariant to basis changes.) The performance should be improved by designing an appropriate frame that operates directly on the pixel intensities.

5.4.5 Discussion

The use of overcomplete expansions in source coding for robustness to erasures seems to be a novel development. In the latter stages of this work, the author became aware of two related works.

Recently—after the dissemination of the present work, but likely not inspired by it—Wang, Orchard, and Reibman [202] have extended the framework of [146] in a way that sometimes uses an overcomplete expansion. They recognized that the correlating transform method exhibits good performance when a small amount of redundancy is added to a pair, but the return on this investment diminishes very quickly (see Figure 5.8(a)). They have proposed a scheme for two-channel multiple description coding which is a hybrid between pairwise correlating and a 4×2 frame expansion. When a small amount of redundancy is to be added, the correlating transform is used alone, but for larger redundancies each channel individually carries information based on a basis expansion.

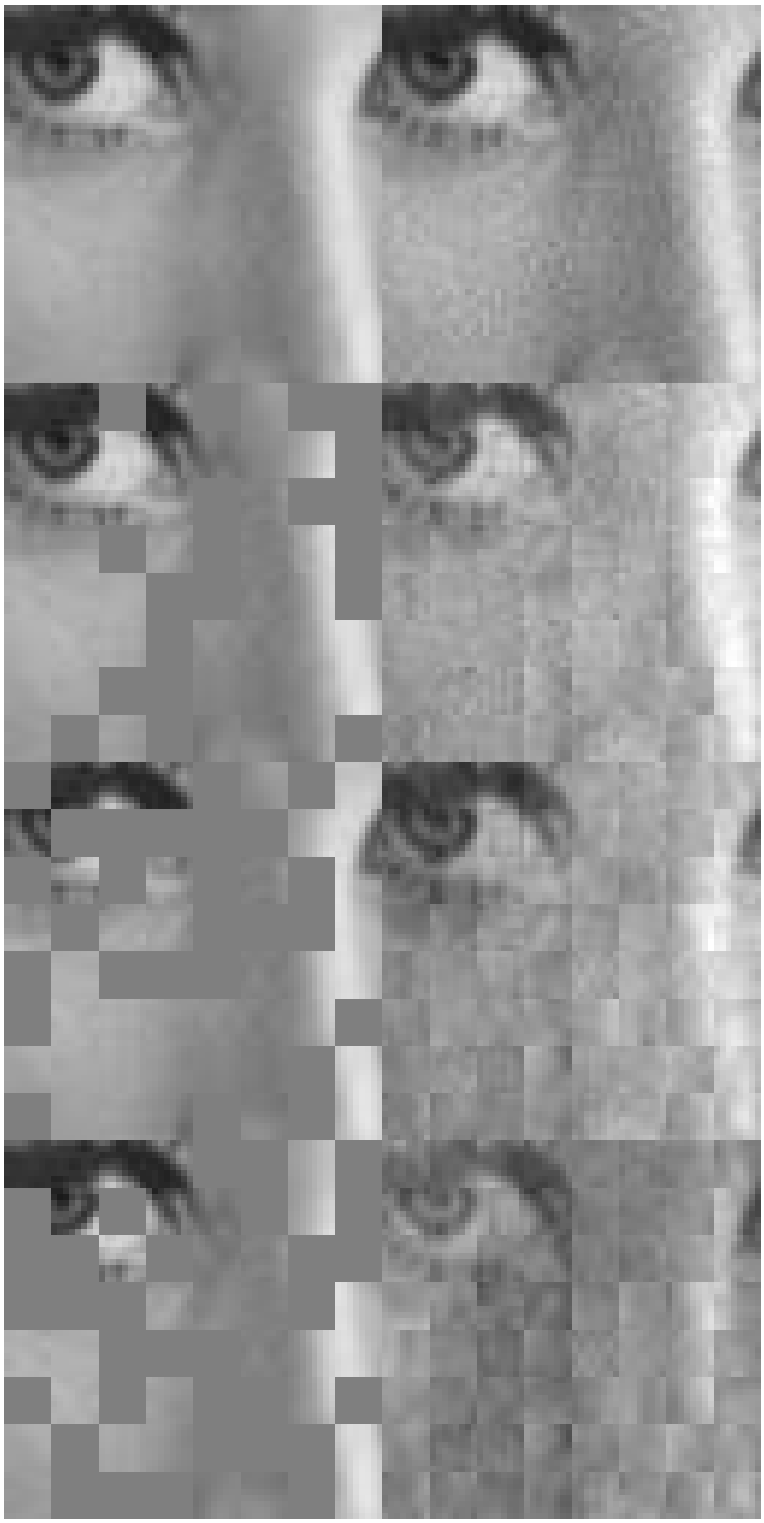


Figure 5.31: Visual image coding results for quantized frame system. Results are for compressing 512×512 *Lena* at about 1 bit per pixel. 64×64-pixel detail images are shown. Left column: baseline system; right column: MD system. From top to bottom, number of packets received is 8, 7, 6, and 5.

An overcomplete representation was also used by Yu, Liu, and Marcellin [217, 218] in an error concealment algorithm for packet video. Many video coding standards—including H.261, H.263, and MPEG [116]—use block transform coding of residual images after motion compensation. Yu *et al.* considered the problem of increasing the robustness to block loss. They proposed the use of DCT-domain coding of overlapping 9×9 blocks of the residual images, with one pixel overlap on each side. Because of the overlap, corner pixels are included in four DCT blocks and non-corner edge pixels are included in two DCT blocks. In parts of the image affected by lost blocks, the extra information is successfully used to eliminate visible error propagation in certain circumstances. In undegraded parts of the image, the overlap gives the decoder overcomplete representations of the blocks; an instance of the projection onto convex sets (POCS) algorithm is used to compute consistent reconstructions, which lowers the quantization noise (see Section 2.2.2.3).

5.5 Conclusions

This chapter has provided a comprehensive introduction to the multiple description problem and two new methods for practical multiple description coding. Both methods combine transforms and scalar quantization, and thus are computationally simpler than vector quantization. The first method uses a basis representation; but in contrast to transform coding systems designed purely for compression, not robustness, the transform coefficients are correlated. This method is very effective in increasing robustness with a small amount of redundancy. The second method uses an overcomplete expansion. It is similar to the use of a block channel code except that the redundancy is added to continuous-valued data prior to quantization. This method works well when the actual number of received descriptions is likely to vary significantly from its mean.

Practical applications of these methods, to image and audio coding, were also presented. All of the applications are very promising, but are at the "proof of concept" stage.

The multiple description scenario provides a good analogy to communication over a lossy packet network. For this reason, "description," "channel," and "packet" have been used interchangeably. However, this is not the only communication environment in which MD coding may be useful. Effros and Goldsmith [47] have studied the various notions of capacity for general time-varying channels. One of their results is that more information can be reliably received than can be reliably transmitted. With some thought, this is an intuitive result: It is less demanding to ask for every bit that gets across the channel to be correct than to ask for every bit that is transmitted to correctly get across the channel. For such a general channel it may be useful to use a multiple description source code since all the received information will be useful, but the loss of some of the transmitted information is not catastrophic.

Appendices

5.A Pseudo-linear Discrete Transforms

Recently, several researchers have proposed using invertible discrete-domain to discrete-domain transforms [98, 223, 24]. They appear under various names (lossless transforms, integer-to-integer transforms, lifting factorizations) and in various flavors (finite dimensional matrices, or Fourier or wavelet domain operators). All these transforms are based on factorizations of matrices which make information flow in a simple, regular way. Inversion can then be achieved by reversing the information flow.

For example, one can factor any 2×2 matrix with determinant 1 into three lower- and upper-triangular matrices with unit diagonals:

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ (d-1)/b & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 1 & 0 \\ (a-1)/b & 1 \end{bmatrix}}_{T_3} \text{ or}$$

$$= \underbrace{\begin{bmatrix} 1 & (a-1)/c \\ 0 & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 1 & (d-1)/c \\ 0 & 1 \end{bmatrix}}_{T_3}.$$

Since the inverse of a block

$$\begin{bmatrix} 1 & 0 \\ x & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & y \\ 0 & 1 \end{bmatrix}$$

is simply

$$\begin{bmatrix} 1 & 0 \\ -x & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & -y \\ 0 & 1 \end{bmatrix},$$

respectively, the inverse of T can be found by reversing the order of the factors and changing the signs of the off-diagonal elements.

The more profound fact is that the simplicity of inversion remains if the off-diagonal elements represent nonlinear functions. Let $[\,\cdot\,]_\Delta$ represent rounding to the nearest multiple of Δ and let

$$T_1 = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}.$$

If $x \in \Delta\mathbb{Z}^2$, then

$$[T_1 x]_\Delta = \left[\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right]_\Delta = \left[\begin{bmatrix} x_1 + ax_2 \\ x_2 \end{bmatrix}\right]_\Delta = \begin{bmatrix} x_1 + [ax_2]_\Delta \\ x_2 \end{bmatrix}.$$

Thus $[T_1\,\cdot\,]_\Delta$ is an identity operator except for a nonlinear function of x_2 being added to x_1. Direct computation shows that *on the domain* $\Delta\mathbb{Z}^2$, $[T_1^{-1}\,\cdot\,]_\Delta$ is the inverse operator. A cascade of such operations is invertible in the same manner, so a factorization $T = T_1 T_2 T_3$ yields an invertible discrete transform $\widehat{T} : \Delta\mathbb{Z}^2 \to \Delta\mathbb{Z}^2$ "derived from T" through

$$\widehat{T}(x) = \big[T_1\,[T_2\,[T_3 x]_\Delta]_\Delta\big]_\Delta. \tag{5.73}$$

The discrete transform $\widehat{T}$ depends not only T, but the factorization of T. Among the possible factorizations, one can minimize a bound on $\|\widehat{T}(x) - Tx\|$. Let

$$T_1 = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad T_3 = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}.$$

For $x \in \Delta\mathbb{Z}^2$, the computation (5.73) involves three rounding operations. Using δ_i's to denote the roundoff errors gives

$$\widehat{T}(x) = T_1 \left(T_2 \left(T_3 x + \begin{bmatrix} 0 \\ \delta_1 \end{bmatrix} \right) + \begin{bmatrix} \delta_2 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ \delta_3 \end{bmatrix}.$$

Expanding and using $T_1 T_2 T_3 = T$, one can compute

$$\begin{aligned} \left\| \widehat{T}(x) - Tx \right\|_\infty &= \left\| T_1 T_2 \begin{bmatrix} 0 \\ \delta_1 \end{bmatrix} + T_1 \begin{bmatrix} \delta_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \delta_3 \end{bmatrix} \right\|_\infty \\ &= \left\| \begin{bmatrix} b\delta_1 \\ (1+ab)\delta_1 \end{bmatrix} + \begin{bmatrix} \delta_2 \\ a\delta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \delta_3 \end{bmatrix} \right\|_\infty \\ &\le (1 + \max\{|b|, |a| + |1+ab|\}) \frac{\Delta}{2}. \end{aligned}$$

This shows that $\widehat{T}$ approximates T in a precise sense; in particular, $\widehat{T}(x) \approx Tx$ when Δ is small.

For $N \times N$ matrices, the process is similar. T is factored into a product of matrices with unit diagonals and nonzero off-diagonal elements only in one row or column: $T = T_1 T_2 \cdots T_k$. The discrete version of the transform is then given by

$$\widehat{T}(x) = \left[T_1 \left[T_2 \ldots \left[T_k x \right]_\Delta \right]_\Delta \right]_\Delta. \tag{5.74}$$

The lifting structure ensures that the inverse of $\widehat{T}$ can be implemented by reversing the calculations in (5.74):

$$\widehat{T}^{-1}(y) = \left[T_k^{-1} \ldots \left[T_2^{-1} \left[T_1^{-1} y \right]_\Delta \right]_\Delta \right]_\Delta.$$

The existence of such a factorization follows from the fact that any nonsingular matrix can be reduced to an identity matrix by multiplication with elementary matrices [180]. Since our original matrix has determinant 1, it is sufficient to consider the following three types of elementary matrices:

- $E_{ij}^{(\lambda)}$, to subtract a multiple λ of row j from row i.
- P_{ij}, to exchange rows i and j.
- $D_{ij}^{(\lambda)}$, to multiply row i by λ and row j by $1/\lambda$.

$E_{ij}^{(\lambda)}$ is already in the desired form. The remaining two can be factored as desired using the factorization of 2×2 matrices above.

5.B Transform Coding with Discrete Transforms

The phrase "transform coding" evokes a structure consisting of a linear transform, scalar quantization, and entropy coding—in that order. Zero-tree structures [174] and similar developments mix the quantization

and entropy coding to some degree, but it remains that the transform is calculated on continuous-valued (or "full precision") data.

Since the data will ultimately be represented coarsely, it seems that it should be sufficient to compute the transform coarsely.[27] Another approach that may reduce complexity is to compute the transform on a discrete domain of about the same "size" as the final representation. Computing the transform on a "smaller" domain implies that the source is first quantized and then undergoes a transform, as in the correlating transform method for multiple description coding described in Section 5.3.

This appendix analyzes a few systems which combine—in the unusual specified order—scalar quantization, transform, and entropy coding. The use of discrete transforms provides more design freedom than we can handle, but by restricting attention to a particular family of discrete transforms, we can describe forward and inverse transforms simply and follow principled design rules. At the same time, some things are possible with discrete transforms that cannot be done with continuous-valued orthogonal transforms. The resulting systems provide opportunities for complexity reduction and reducing sensitivity to erasures.

The appendix is organized as follows: Section 5.B.1 provides a review of transform coding. Sections 5.B.2–5.B.4 give the results on achieving coding gain, reduction in entropy-coding complexity, and robustness to erasures using a class of discrete transforms. This class of transforms is described in detail in Appendix 5.A.

5.B.1 A Perspective on Transform Coding

In its simplest incarnation, transform coding is the representation of a random vector $x \in \mathbb{R}^N$ by the following three steps:

- A transform coefficient vector is computed as $y = Tx$, $T \in \mathbb{R}^{N \times N}$.
- Each transform coefficient is quantized by a scalar quantizer:

$$\hat{y}_i = q_i(y_i), \qquad i = 1, 2, \ldots, N.$$

  The overall quantizer is denoted $Q : \mathbb{R}^N \to \mathbb{R}^N$.
- An entropy code is applied to each quantized coefficient: $E_i(\hat{y}_i)$, $i = 1, 2, \ldots, N$.

The decoder reverses the steps to produce an estimate $\hat{x}$.

We will consider only the coding of i.i.d. jointly Gaussian sources. Under either high rate or optimal fixed-rate quantizer assumptions, the transform should be a Karhunen–Loève transform (KLT) of the source; *i.e.*, a transform that produces uncorrelated transform coefficients. An analysis of this arrangement including optimal design of the transform is given in Section 1.1.3.1.

Why does transform coding work? An algebraic answer is that the transform makes $\prod \sigma_{y_i}^2 < \prod \sigma_{x_i}^2$, but this is not very enlightening. The geometry of the situation is shown in Figure 5.32. The ellipses represent level curves of the p.d.f. of the source. Quantization in the original coordinates is shown on the left, and quantization after the KLT is shown on the right. Is the second partition any better than the first? Put in another way, is $Q(T(\cdot))$ a better vector quantization encoder mapping than $Q(\cdot)$? In the high rate limit, the

[27] This is explored in Section 6.3.

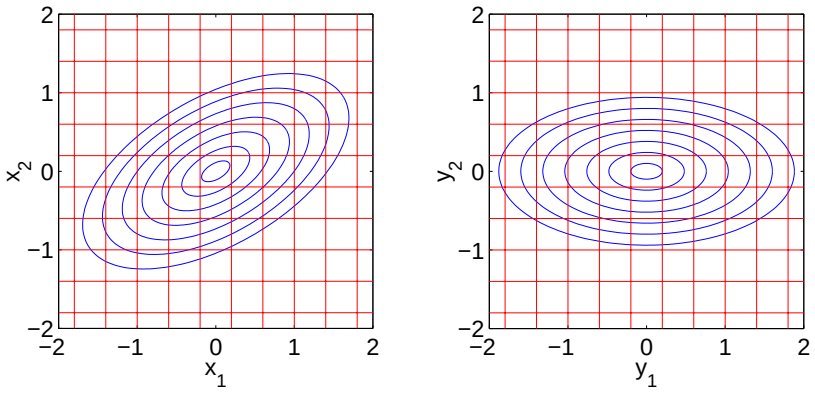


Figure 5.32: Partitioning induced by uniform scalar quantization with and without the application of the Karhunen–Loève transform. The ellipses are level curves of the p.d.f. of the source.

answer is no, since either gives distortion $D = \Delta^2/12$. Transform coding does not improve the quantization, but rather makes the *scalar* entropy coding that follows it work well; if the entropy coding processed an entire vector at a time, the transform would give no advantage.[28]

Since the quantization performance is not affected much by the transform, we may try to replace the "T-Q-E" structure of transform–quantization–entropy coding with a "Q-T-E" structure. Considering only linear transforms from $\Delta\mathbb{Z}^n$ to $\Delta\mathbb{Z}^n$ would be too restrictive, but placing no restriction on the transform gives more design freedom than we can deal with well. Thus, to have easily implemented transforms and to simplify the design process we use only transforms which are derived from (continuous) linear transforms, as described in Appendix 5.A.

Discrete domain transforms can nearly achieve the coding gain of traditional transform coding, but at the same time introduce other possibilities. Though the remainder of the appendix addresses the coding of Gaussian sources, the use of discrete transforms extends the applicability of transform coding to discrete sources, and perhaps to abstract alphabet (non-numerical) sources. For simplicity, most expressions and all simulations are for two-dimensional sources.

5.B.2 Rate Reduction

In transform coding with quantization preceding the transform, the distortion is fixed (at approximately $\Delta^2/12$) independent of the transform. The role of the transform is to reduce the coded rate, but an invertible transform cannot affect the entropy for a discrete random variable [34]. This seeming contradiction is resolved by again remembering that we wish for the entropy coding to operate on scalars.

Denote the source by (x_1, x_2), the transform by $\widehat{T}$, and the transform coefficients by $(y_1, y_2) = \widehat{T}(x_1, x_2)$. In the best case scenario, y_1 and y_2 are independent, so we take advantage of

$$H(x_1) + H(x_2) \;\geq\; H(x_1, x_2) \;=\; H(y_1, y_2) \;=\; H(y_1) + H(y_2),$$

where the left hand side is a lower bound on the rate without the transform. We cannot normally expect to make y_1 and y_2 independent, but we can approximately achieve this condition by choosing $\widehat{T}$ to be an approximation to

[28] We are concerned here with high rates; at low rates it is hard to predict the best coordinates for scalar quantization.

a KLT for x. Because the construction of the discrete transform introduces only $O(\Delta)$ error (see Appendix 5.A), in the high rate limit y_1 and y_2 are independent.

This was experimentally confirmed with a two-dimensional Gaussian source with correlation matrix

$$R_x = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}.$$

The KLT is a $\pi/4$ radian rotation. A comparison between using no transform, using the KLT (before quantization), and using a discrete approximation to the KLT is shown in Figure 5.33. If the entropy coding operates on vectors there is virtually no difference between the three transform choices.[29] Removing the correlation in the source is important with scalar entropy coding. The discrete transform performs almost as well as the KLT; of course, it cannot perform better because the KLT makes the transform coefficients independent.

5.B.3 Complexity Reduction

The previous section demonstrated that a discrete transform can do about as well as a continuous transform when scalar entropy coding is to be used. It was implicit that each scalar entropy code was optimized to its corresponding transform coefficient. Having N separate entropy codes increases the memory requirements and is thus undesirable. In the previous example, this could be seen as an argument for using scalar entropy coding in the original coordinates, despite the higher rate.

For simplicity, consider a source with independent components: $R_x = \mathrm{diag}(\sigma_1^2, \sigma_2^2)$, $\sigma_1 \geq \sigma_2$. There is little flexibility in the choice of continuous transforms since they must be orthogonal to maintain cubic partition cells. For this source, only reflections and trivial rotations of $k\pi/2$ radians do not increase the rate; thus, there is no hope to equalize the p.d.f.'s of the transform coefficients without hurting the rate–distortion performance. The family of discrete transforms used here gives more flexibility. We need not start with an orthogonal transform; any transform with determinant 1 will suffice. Discrete transforms derived from initial transforms of the form

$$\widehat{T} = \begin{bmatrix} \alpha & \pm\sigma_2^{-1}\alpha\sigma_1 \\ \mp(2\alpha\sigma_1)^{-1}\sigma_2 & (2\alpha)^{-1} \end{bmatrix}$$

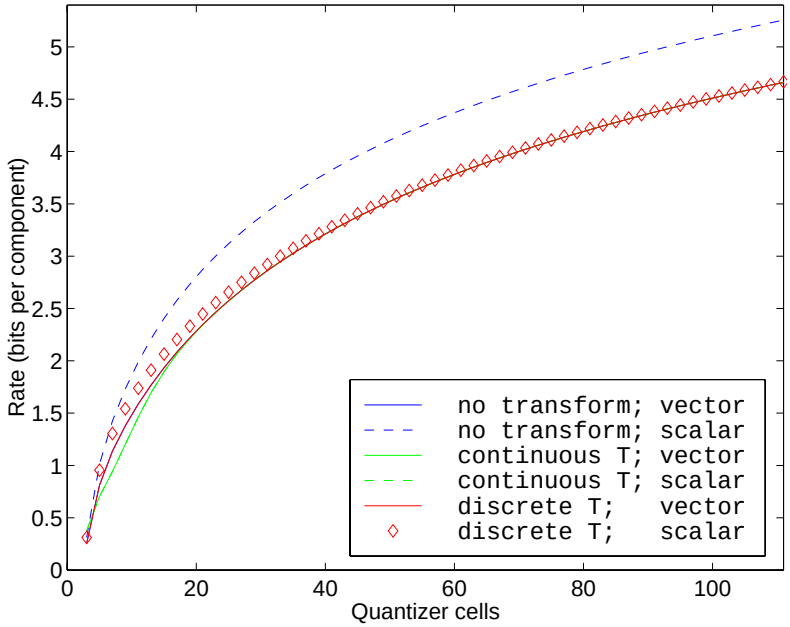
all give the optimal coding gain. In particular,

$$\widehat{T} = \begin{bmatrix} \alpha & (2\alpha)^{-1} \\ -\alpha & (2\alpha)^{-1} \end{bmatrix}, \qquad \text{with} \quad \alpha = \sqrt{\frac{\sigma_2}{2\sigma_1}}, \tag{5.75}$$

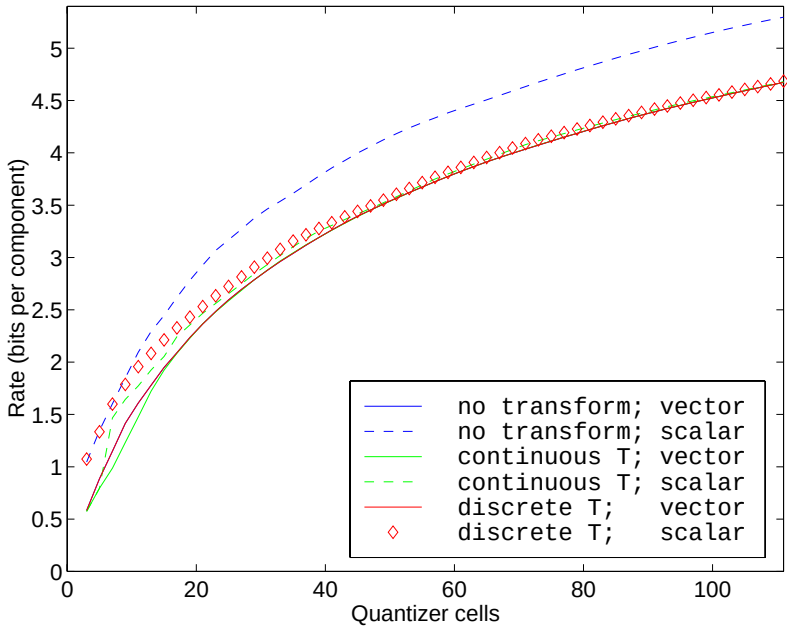
gives optimal coding gain and produces transform coefficients with identical distributions. This is a special case of the general analysis of Section 5.3.2.2.

An experimental confirmation is shown in Figure 5.34. The source was chosen to have the same power and eigenvalue spread as in the previous example, so $\sigma_1^2 = 1.9$ and $\sigma_2^2 = 0.1$. Since the source components are independent, the best case performance is to quantize and apply separately optimized entropy codes to the two variables. However, when a discrete transform based on (5.75) is used, the best performance is almost matched even with a single entropy code applied to both transform coefficients.

[29] The "no transform; vector" and "discrete T; vector" cases give precisely the same rates, as do "continuous T; vector" and "continuous T; scalar."

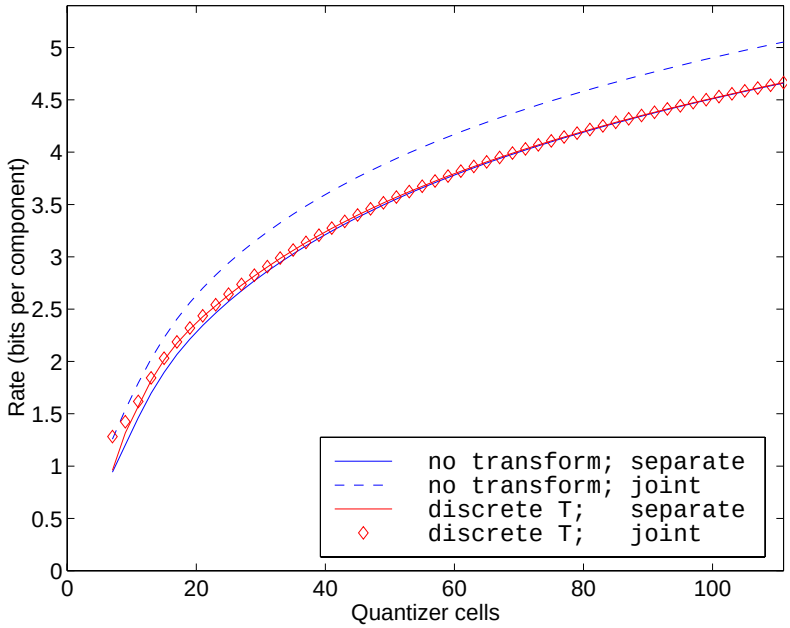


(a)

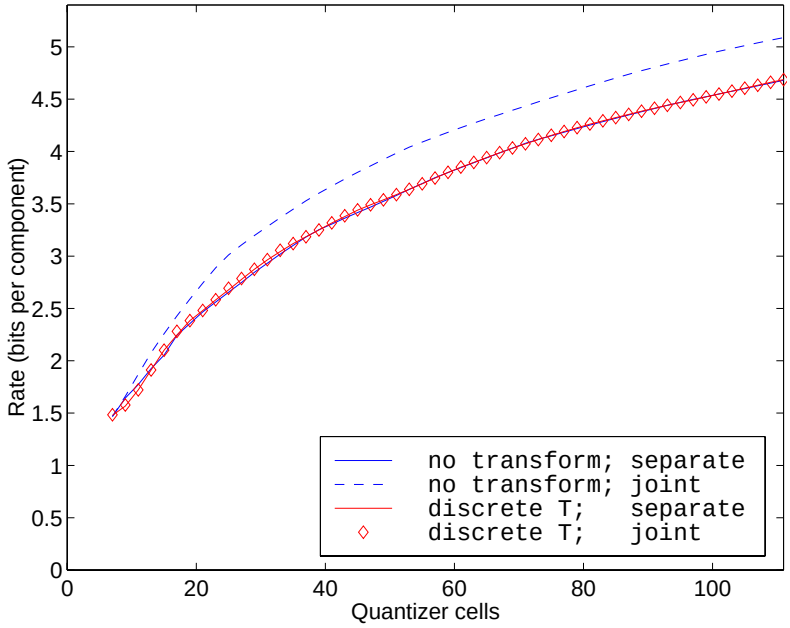


(b)

Figure 5.33: Experiment showing that the coding gain of the (continuous) KLT is almost matched by a discrete transform which approximates the KLT. The horizontal axis gives the number of cells covering $[-6, 6]$ for each component quantizer. Legend entries indicate whether no transform, a KLT, or a discrete approximation of the KLT was used; and whether the entropy coding is based on scalars or vectors. (a) Rates based on empirical entropies; (b) Rates based on explicit Huffman codes.



(a)



(b)

Figure 5.34: Experiment showing that a discrete transform makes it possible to simultaneously achieve optimal coding gain and use the same entropy code for each transform coefficient. The horizontal axis gives the number of cells covering $[-6, 6]$ for each component quantizer. Legend entries indicate whether a transform is used and whether the separate entropy codes are used for each transform coefficient. (a) Rates based on empirical entropies; (b) Rates based on explicit Huffman codes.

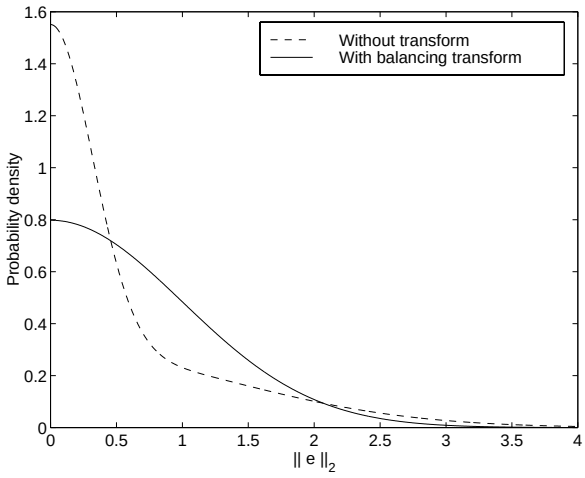


Figure 5.35: Probability densities of the norm of the reconstruction error with and without the transform (5.75). The source has independent components with variances $\sigma_1^2 = 1.9$ and $\sigma_2^2 = 0.1$. The transform does not change $E[\|e\|^2]$, but it reduces $E[\|e\|^4]$. Thus the MSE distortion is unchanged but the variance of the squared-error is lowered.

Other manipulations of the transform coefficient variances may be useful. For example, the probability of a zero coefficient affects both the efficacy of run-length coding and decoding optimizations in the spirit of [119, 120].

5.B.4 Erasure Resilience

The set of transforms described by (5.75) is familiar from Section 5.3.2.2 as optimal transforms for multiple description transform coding when both channels are equally likely to fail. In addition, these transforms produce two channels with equal rates. The choice of α in (5.75) is the extreme case where no redundancy is added and the average MSE when there is an erasure is not reduced. However, even in this case there is a change in the *distribution* of the distortion. This effect is described in this section.

As in the previous section, consider the coding of a zero-mean Gaussian source described by $R_x = \text{diag}(\sigma_1^2, \sigma_2^2)$, $\sigma_1 \geq \sigma_2$. Suppose that in a multiple description system, finely-quantized versions of the two components are sent independently over two channels. Since the components are independent, a reconstruction from only x_1 would simply be $\hat{x} = [[x_1]_\Delta,\ 0]^T$. Neglecting the quantization error, the reconstruction error magnitude $\|x - \hat{x}\|$ is the absolute value of a Gaussian random variable with variance σ_2^2. Similarly, reconstructing from x_2 gives an error magnitude which is the absolute value of a Gaussian random variable with variance σ_1^2. Assume the channels are equally likely to fail and denote the reconstruction error with one channel failure by $e = x - \hat{x}$. Then the norm of e is the absolute value of a Gaussian mixture. For $\sigma_1^2 = 1.9$ and $\sigma_2^2 = 0.1$, the probability density of $\|e\|$ is shown in Figure 5.35 with the dashed curve.

Using the transform (5.75) with the specified α does not change the total rate or the average distortion when one of the two channels is lost. However, it does change the distribution of $\|e\|$. Evaluating (5.26) for either channel failure shows that $\|e\|$ is the absolute value of a Gaussian random variable with variance $(\sigma_1^2 + \sigma_2^2)/2$. The probability density is shown in Figure 5.35 with a solid curve.

The transform has not changed the mean-squared error $E[\|e\|^2]$. This is consistent with Section 5.3, where reductions in the MSE with channel failures came only at the expense of increased rate. However, the probability of large errors has been reduced, which may be considered an increase in robustness despite the unchanged MSE. This can easily be characterized by computing $E[\|e\|^4]$. Without the transform

$$E[\|e\|^4] = \frac{3}{2}(\sigma_1^4 + \sigma_2^4),$$

but with the transform

$$E[\|e\|^4] \;=\; \frac{3}{4}(\sigma_1^2 + \sigma_2^2)^2 \;\leq\; \frac{3}{2}(\sigma_1^4 + \sigma_2^4).$$

Thus the variance of $\|e\|^2$ is unchanged or reduced by the transform.

5.C Proofs

5.C.1 Proof of Theorem 5.6

The proofs of Theorems 5.6 and 5.7 utilize the following lemma [157]:

Lemma 5.11 *Let A, X, and Y be symmetric, real, positive definite matrices and let $Q = Y^{1/2}AY^{1/2}$. Suppose X and Q each have distinct eigenvalues. Denote orthogonal eigendecompositions of X and Q by $Q = V_Q\Lambda_Q V_Q^T$ and $X = V_X\Lambda_X V_X^T$, respectively, where Λ_Q and Λ_X have decreasing diagonals. Then the optimization problem*

$$\textit{minimize} \quad J(U) = \operatorname{tr} AU^TU \quad \textit{subject to} \quad U^TXU = Y$$

is solved by

$$U_0 = V_X\Lambda_X^{-1/2}V_Q^TY^{1/2}, \tag{5.76}$$

yielding

$$J(U_0) = \operatorname{tr}\Lambda_Q\Lambda_X^{-1}. \tag{5.77}$$

This solution is unique up to the sign choices in defining V_X and V_Q.

In the proof of Lemma 5.11, the following elementary fact is used:

Lemma 5.12 *Suppose* $\operatorname{tr} BS = 0$ *for all skew-symmetric matrices S. Then B is symmetric.*

Proof (Lemma 5.12): For any $i \neq j$, let S be the matrix with $+1$ in the (i,j) position, -1 in the (j,i) position, and remaining elements equal to zero. Since $0 = \operatorname{tr} BS = B_{ji} - B_{ij}$, $B_{ij} = B_{ji}$. □

Proof (Lemma 5.11): We may first convert the constraint $U^TXU = Y$ to a simpler form. Left and right multiplying by $Y^{-1/2}$ and splitting $X = X^{1/2}\cdot X^{1/2}$ gives $Y^{-1/2}U^TX^{1/2}\cdot X^{1/2}UY^{-1/2} = I$. With the definition $\widetilde{U} = X^{1/2}UY^{-1/2}$, we have the constraint $\widetilde{U}^T\widetilde{U} = I$.

The objective function is now

$$\begin{aligned}
\widetilde{J}(\widetilde{U}) &= J(X^{-1/2}\widetilde{U}Y^{1/2}), && (5.78)\\
&= \operatorname{tr} AY^{1/2}\widetilde{U}^TX^{-1}\widetilde{U}Y^{1/2},\\
&= \operatorname{tr} Y^{1/2}AY^{1/2}\widetilde{U}^TX^{-1}\widetilde{U}, && (5.79)\\
&= \operatorname{tr} Q\widetilde{U}^TX^{-1}\widetilde{U}, && (5.80)
\end{aligned}$$

where (5.79) uses the fact that cyclic permutation of factors does not affect the trace. We are left with minimizing (5.80) over orthogonal transforms $\widetilde{U}$.

A differential analysis will reveal a single critical point, up to sign choices. Since A is positive definite, this critical point must be a minimum. Consider a small change to $\widetilde{U}$, $\widetilde{U}_\delta = \widetilde{U} + \delta$. To obey the orthogonality constraint, we must have $\widetilde{U}_\delta^T \widetilde{U}_\delta = I$. Expanding $\widetilde{U}_\delta^T \widetilde{U}_\delta$ and neglecting the δ^2 term gives the constraint

$$\widetilde{U}^T \delta + \delta^T \widetilde{U} = 0. \tag{5.81}$$

The perturbation has the following affect on the objective function:

$$\begin{aligned} \widetilde{J}(\widetilde{U}_\delta) &= \operatorname{tr} Q(\widetilde{U}+\delta)^T X^{-1}(\widetilde{U}+\delta), \\ &= \operatorname{tr} Q\widetilde{U}^T X^{-1}\widetilde{U} + \operatorname{tr} Q\widetilde{U}^T X^{-1}\delta + \operatorname{tr} Q\delta^T X^{-1}\widetilde{U} + \operatorname{tr} Q\delta^T X^{-1}\delta, \\ &\approx \operatorname{tr} Q\widetilde{U}^T X^{-1}\widetilde{U} + 2\operatorname{tr} Q\widetilde{U}^T X^{-1}\delta, \qquad (5.82) \\ &= \widetilde{J}(\widetilde{U}) + 2\operatorname{tr} Q\widetilde{U}^T X^{-1}\delta, \end{aligned}$$

where the approximation (5.82) results from discarding the $O(\|\delta\|^2)$ term, and using $\operatorname{tr} M = \operatorname{tr} M^T$. Thus a critical point of $\widetilde{J}(\,\cdot\,)$ is a transform $\widetilde{U}$ that satisfies

$$\operatorname{tr} Q\widetilde{U}^T X^{-1}\delta = 0 \tag{5.83}$$

for all small δ satisfying (5.81).

The solutions of (5.81) are simple; they are $\delta = \widetilde{U}S$, where S is an arbitrary skew-symmetric matrix. Thus the solutions of (5.83) are $\widetilde{U}$ such that $\operatorname{tr} Q\widetilde{U}^T X^{-1}\widetilde{U}S = 0$ for all skew-symmetric S. By Lemma 5.12, $Q\widetilde{U}^T X^{-1}\widetilde{U}$ must be symmetric. Notice the effect of transposing this matrix: Since Q and X are symmetric, the transpose is $\widetilde{U}^T X^{-1}\widetilde{U}Q$; thus, Q and $\widetilde{U}^T X^{-1}\widetilde{U}$ commute.

Diagonalizable matrices commute if and only if they are simultaneously diagonalizable [99]. Furthermore, for a matrix with distinct eigenvalues the orthogonal transform that diagonalizes and leaves the diagonal in decreasing order is unique up to sign choices. On one hand, Q is diagonalized as

$$\underbrace{V_Q^T}_{\text{diagonalizing transform}} \cdot Q \cdot \underbrace{V_Q}_{\text{transposed transform}} = \Lambda_Q;$$

on the other hand, $\widetilde{U}^T X^{-1}\widetilde{U}$ is diagonalized as

$$\underbrace{V_X^T\widetilde{U}}_{\text{diagonalizing transform}} \cdot \widetilde{U}^T \underbrace{V_X\Lambda_X^{-1}V_X^T}_{X^{-1}}\widetilde{U} \cdot \underbrace{\widetilde{U}^T V_X}_{\text{transposed transform}} = \Lambda_X^{-1}.$$

Ignoring sign choices, we may equate the diagonalizing transforms using a permutation matrix P:

$$PV_Q^T = V_X^T\widetilde{U}.$$

The permutation will be chosen after its effect is made clear. A simple sequence of substitutions yields the optimal transform U_0:

$$U_0 = X^{-1/2}\widetilde{U}Y^{1/2} = V_X\Lambda_X^{-1/2}V_X^T \cdot V_X P V_Q^T \cdot Y^{1/2} = V_X\Lambda_X^{-1/2}PV_Q^T Y^{1/2}. \tag{5.84}$$

Evaluating $J(U_0)$ gives

$$\begin{aligned} J(U_0) &= \operatorname{tr} AU^T U, \\ &= \operatorname{tr} \underbrace{Y^{-1/2} Q Y^{-1/2} \cdot Y^{1/2} V_Q P^T \Lambda_X^{-1/2} V_X^T \cdot V_X \Lambda_X^{-1/2} P} \underbrace{V_Q^T Y^{1/2}}, \qquad (5.85) \\ &= \operatorname{tr} V_Q^T Q V_Q \cdot P^T \Lambda_X^{-1} P, \\ &= \operatorname{tr} \Lambda_Q P^T \Lambda_X^{-1} P, \qquad (5.86) \end{aligned}$$

where terms are separated to emphasize substitutions, and underbraces in (5.85) mark terms that are subsequently commuted. Since Λ_Q and Λ_X are already sorted in the same order, (5.86) is minimized by choosing the identity permutation $P = I$. This finally yields (5.76) and (5.77). □

We are now prepared to prove the theorem. The overall strategy is as follows: Starting with any transform T, we can find a transform V such that VT results in identical side distortion and at most the same redundancy as T. At the same time, VT yields a correlation R_y with a particular, simple form. This simple form in turn leads to a simple expression for the side distortion D_1. Lemma 5.11 is then used to show that the transform that yields minimum D_1 among transforms with correlation R_y has the desired form (5.40). Since the performance with any transform can at least be matched with a transform of the form (5.40), the proof is complete. Each step is now detailed.

Recall that T is an arbitrary transform. Let $R_y = TR_xT^T$, the correlation matrix of the transform coefficients when T is used. Since y_2 and y_3 are sent on the same channel, an invertible transform applied to the two will not change D_1. However, if y_2 and y_3 are correlated, the rate can be reduced by applying a decorrelating transform. Denote a KLT for

$$\begin{bmatrix} (R_y)_{22} & (R_y)_{23} \\ (R_y)_{32} & (R_y)_{33} \end{bmatrix}$$

by $\widetilde{V}_1$, and let

$$V_1 = \begin{bmatrix} 1 & 0_{1\times 2} \\ 0_{2\times 1} & \widetilde{V}_1 \end{bmatrix}.$$

Then using V_1T in place of T does not change the side distortion D_1, and does not increase the redundancy ρ.

After the application of V_1, the correlation can be written in the following form:

$$V_1 T R_x T^T V_1^T = \begin{bmatrix} \varsigma_1^2 & a_1 & a_2 \\ a_1 & \varsigma_2^2 & 0 \\ a_2 & 0 & \varsigma_3^2 \end{bmatrix}.$$

This type of correlation structure cannot be produced by a transform of the desired form (5.40) unless $a_2 = 0$, so we simplify the correlation structure further. Let

$$\widetilde{V}_2 = \frac{1}{\sqrt{a_1^2\varsigma_3^2 + a_2^2\varsigma_2^2}} \begin{bmatrix} \sigma_2^{-1} a_1 \varsigma_3^2 & \sigma_2^{-1} a_2 \varsigma_2^2 \\ -a_2\sigma_2 & a_1\sigma_2 \end{bmatrix} \quad \text{and} \quad V_2 = \begin{bmatrix} 1 & 0_{1\times 2} \\ 0_{2\times 1} & \widetilde{V}_2 \end{bmatrix}.$$

Then

$$V_2 V_1 T R_x T^T V_1^T V_2^T = \begin{bmatrix} \varsigma_1^2 & \sigma_2^{-1}\sqrt{a_1^2\varsigma_3^2 + a_2^2\varsigma_2^2} & 0 \\ \sigma_2^{-1}\sqrt{a_1^2\varsigma_3^2 + a_2^2\varsigma_2^2} & \sigma_2^{-2}\varsigma_2^2\varsigma_3^2 & 0 \\ 0 & 0 & \sigma_2^2 \end{bmatrix}. \qquad (5.87)$$

(The reason for selecting V_2 with deference to σ_2 is revealed below.) With reference to (5.24), recall that the redundancy depends only on the product variances of the transform coefficients. Since the product of the diagonal elements of $V_2V_1TR_xT^TV_1^TV_2^T$ and $V_1TR_xT^TV_1^T$ are equal, using V_2V_1T in place of V_1T does not change the redundancy. Furthermore, since V_2 merely alters the second and third components in an invertible manner, D_1 is also unchanged. With $V = V_2V_1$, we have found a transform such that VT is at least as good for multiple description coding as T, and $VTR_xT^TV^T$ has a simple form; the first step of the proof is complete.

We now wish to show that for R_y of the form (5.87), the optimal transform has the desired form (5.40). In light of the uniqueness of the solution in Lemma 5.11, this lemma could be used to directly compute the best transform for the given R_y. This is not done for several reasons.[30] Most importantly, our ultimate goal is to minimize the side distortion for a given redundancy, not a given R_y. Even if two transform coefficient correlation matrices yield the same redundancy, their corresponding minimum distortions may not be the same; after finding the minimum distortion as a function of R_y we would have to minimize over all R_y with the a particular redundancy. Finding only the form of the optimal transform also simplifies the following computations significantly.

To simplify notation, let

$$R_y = \begin{bmatrix} \gamma_1 & a & 0 \\ a & \gamma_2 & 0 \\ 0 & 0 & \sigma_2^2 \end{bmatrix}.$$

Inverting $TR_xT^T = R_y$ gives $U^TR_x^{-1}U = R_y^{-1}$ where, as in Section 5.3.2.1, $U = T^{-1}$. When y_1 (Channel 1) is lost, A in (5.27) is given by

$$A_1 = \gamma_1 - \begin{bmatrix} a & 0 \end{bmatrix} \begin{bmatrix} \gamma_2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} a \\ 0 \end{bmatrix} = \gamma_1 - \frac{a^2}{\gamma_2}.$$

When (y_2, y_3) (Channel 2) is lost, the corresponding quantity is

$$A_2 = \begin{bmatrix} \gamma_2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} - \begin{bmatrix} a \\ 0 \end{bmatrix} {\gamma_1}^{-1} \begin{bmatrix} a & 0 \end{bmatrix} = \begin{bmatrix} \gamma_2 - \gamma_1^{-1}a^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}.$$

The average side distortion per component when the channels are equally likely to be lost is given by

$$D_1 = \frac{1}{6} \operatorname{tr} AU^TU \quad \text{where} \quad A = \begin{bmatrix} A_1 & 0_{1\times 2} \\ 0_{2\times 1} & A_2 \end{bmatrix}.$$

It is now clear that we have an optimization that can be solved with Lemma 5.11. Identify $X = R_x^{-1} = \operatorname{diag}(\sigma_1^{-2}, \sigma_2^{-2}, \sigma_3^{-2})$ and

$$Y = R_y^{-1} = \begin{bmatrix} \widetilde{Y} & 0_{2\times 1} \\ 0_{1\times 2} & \sigma_2^{-2} \end{bmatrix},$$

where we need not specify $\widetilde{Y}$ because we are primarily interested in sparsity. Other quantities that appear in Lemma 5.11 can easily be computed.[31] Since X is already diagonal, $V_X = I$ and $\Lambda_X^{-1/2} = \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3)$. The sparsity of Y gives

$$Q = Y^{1/2}AY^{1/2} = \begin{bmatrix} \widetilde{Y}^{1/2} & 0_{2\times 1} \\ 0_{1\times 2} & \sigma_2^{-1} \end{bmatrix} A \begin{bmatrix} \widetilde{Y}^{1/2} & 0_{2\times 1} \\ 0_{1\times 2} & \sigma_2^{-1} \end{bmatrix} = \begin{bmatrix} \widetilde{Y}^{1/2}\widetilde{A}\widetilde{Y}^{1/2} & 0_{2\times 1} \\ 0_{1\times 2} & 1 \end{bmatrix}, \tag{5.88}$$

[30] The plain reason is that we need not do more than satisfy the statement of the theorem.

[31] Arbitrary sign and permutation choices are made in diagonalizing transforms; sorting of diagonal elements is handled later.

so a diagonalizing transform of Q will have the form

$$V_Q = \begin{bmatrix} V_{\widetilde{Q}} & 0_{2\times 1} \\ 0_{1\times 2} & 1 \end{bmatrix}.$$

Now substituting in (5.84) gives

$$U_0 = \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3) P \begin{bmatrix} V_{\widetilde{Q}} & 0_{2\times 1} \\ 0_{1\times 2} & 1 \end{bmatrix} \begin{bmatrix} \widetilde{Y}^{1/2} & 0_{2\times 1} \\ 0_{1\times 2} & \sigma_2^{-1} \end{bmatrix}.$$

The optimal transform is the inverse of U_0:

$$T = \begin{bmatrix} \widetilde{Y}^{-1/2} V_{\widetilde{Q}}^T & 0_{2\times 1} \\ 0_{1\times 2} & \sigma_2 \end{bmatrix} P^T \operatorname{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \sigma_3^{-1}). \tag{5.89}$$

It remains now to determine the permutation P in (5.89). This depends on how Λ_Q must be permuted to match the ordering of Λ_X. First note that $\Lambda_X = X$ is sorted in decreasing order. It is not necessary to precisely determine the eigenvalues of Q to find the required permutation. The sum of the eigenvalues of Q is given by

$$\operatorname{tr} Q \;=\; \operatorname{tr} Y^{1/2} A Y^{1/2} \;=\; \operatorname{tr} AY \;=\; \operatorname{tr} \begin{bmatrix} 1 & -a/\gamma_2 & 0 \\ -a/\gamma_1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \;=\; 3.$$

From the form of (5.88) one of the eigenvalues is 1. In the generic case, the eigenvalues are distinct and so the remaining eigenvalues sandwich the eigenvalue 1. To counteract the different sorting of Λ_X and Λ_Q, the permutation must move the third element to the middle. With such a permutation, (5.89) simplifies to

$$T = \begin{bmatrix} \widetilde{Y}^{-1/2} V_{\widetilde{Q}}^T \operatorname{diag}(\sigma_1^{-1}, \sigma_3^{-1}) \widetilde{P} & 0_{2\times 1} \\ 0_{1\times 2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

where $\widetilde{P}$ is a 2×2 permutation that depends on whether $V_{\widetilde{Q}}$ is a clockwise or counterclockwise rotation. (The cancellation of σ_2 now retrospectively explains why this standard deviation was singled out.) The final transform is in the form (5.40) and thus the proof is complete.

This analysis could be pushed further to determine the optimal transform. However, this would merely be the optimal transform given R_y, not the optimal transform for given redundancy ρ. Applying this theorem, an optimal transform with an upper bound on ρ is easy to compute using the results from Section 5.3.2.2 (see Section 5.3.2.3).

5.C.2 Proof of Theorem 5.7

This theorem is similar to Theorem 5.6 so an abbreviated proof is given. The strategy of the proof is to start with an arbitrary transform T. The corresponding correlation matrix TR_xT^T may be fully dense, but there is a transform V with determinant 1 such that $VTR_xT^TV^T$ has a simple desired form and VT is no worse than T for use in the system. An application of Lemma 5.11 then shows that the optimal transform is of the desired form (5.42). The simplification of the correlation matrix is detailed below, but the application of Lemma 5.11 is omitted.

As in Appendix 5.C.1, components sent over the same channel can be made uncorrelated without affecting the side distortion, while not increasing the redundancy. Using Karhunen–Loève transforms to decorrelate in such a manner gives

$$V_1 T R_x T^T V_1^T = \begin{bmatrix} \varsigma_1^2 & a_{11} & 0 & a_{12} \\ a_{11} & \varsigma_2^2 & a_{21} & 0 \\ 0 & a_{21} & \varsigma_3^2 & a_{22} \\ a_{12} & 0 & a_{22} & \varsigma_4^2 \end{bmatrix}.$$

Now we would like to find $\widetilde{W}_1$ and $\widetilde{W}_2$ with $\det \widetilde{W}_i = 1$, $i = 1, 2$, such that

$$V_2 = \begin{bmatrix} \widetilde{W}_1 & 0_{2\times 2} \\ 0_{2\times 2} & \widetilde{W}_2 \end{bmatrix} P, \quad \text{with} \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

gives the desired correlation structure. A solution is obtained with

$$\begin{aligned} \widetilde{W}_1 &= \begin{bmatrix} \alpha & \beta \\ -(\alpha^2\varsigma_1^2 + \beta^2\varsigma_3)^{-1}\beta\varsigma_3^2 & (\alpha^2\varsigma_1^2 + \beta^2\varsigma_3)^{-1}\alpha\varsigma_1^2 \end{bmatrix} \\ \widetilde{W}_2 &= \begin{bmatrix} \gamma & \delta \\ -(\gamma^2\varsigma_2^2 + \delta^2\varsigma_4)^{-1}\delta\varsigma_4^2 & (\gamma^2\varsigma_2^2 + \delta^2\varsigma_4)^{-1}\gamma\varsigma_3^2 \end{bmatrix} \end{aligned}$$

where α is a root of

$$\alpha^2 - \frac{\beta(\varsigma_3^2\varsigma_4^2 a_{11}^2 + \varsigma_2^2\varsigma_3^2 a_{12}^2 - \varsigma_1^2\varsigma_4^2 a_{21}^2 + \varsigma_1^2\varsigma_2^2 a_{22}^2)}{\varsigma_1^2(\varsigma_4^2 a_{11}a21 + \varsigma_2^2 a_{12}a_{22})}\alpha - \frac{\beta^2\varsigma_1^2}{\varsigma_3^2} = 0 \tag{5.90}$$

and

$$\gamma = \frac{\delta\varsigma_4^2(\alpha a_{11} + \beta a_{21})}{\varsigma_2^2(\alpha a_{12} + \beta a_{22})}. \tag{5.91}$$

The validity of this solution places no constraint on δ. The choice of β should ensure that (5.90) has real roots and (5.91) does not involve a division by zero. These requirements are easily satisfied by choosing β to have the same sign as $\varsigma_3^2\varsigma_4^2 a_{11}^2 + \varsigma_2^2\varsigma_3^2 a_{12}^2 - \varsigma_1^2\varsigma_4^2 a_{21}^2 + \varsigma_1^2\varsigma_2^2 a_{22}^2$ and eliminating a few isolated points.[32]

With $V = V_2 V_1$, we have a transform such that VT is at least as good as T; *i.e.*, VT gives the same average side distortion and at most the same redundancy as T. Also, $VTR_xT^TV^T$ has a simple block diagonal form. This block diagonal form permits an application of Lemma 5.11 which shows that the optimal transform is of the desired form (5.42). The application of Lemma 5.11 parallels its use in Appendix 5.C.1 and is thus omitted. This completes the proof.

Note that the complicated dependence of V_2 on the various parameters hinders extending this method of proof to Conjecture 5.9.

5.C.3 Proof of Theorem 5.8

This appendix gives a proof of Theorem 5.8. First note that for high ρ, the distortion given by (5.55) is dominated by the first term, so the ς_{2i}'s must be the K smallest variances. Since we are interested in the

[32] The case $\varsigma_4^2 a_{11}a_{21} + \varsigma_2^2 a_{12}a_{22} = 0$ should be handled separately. In this situation one can achieve the desired correlation structure with $\widetilde{W}_1 = I$.

pairing but not the order of the pairs, we may assign

$$\varsigma_{2i} = \sigma_{2K+1-i}, \quad i = 1, 2, \ldots, K,$$

without loss of generality. Now it remains to show that

$$\varsigma_{2i-1} = \sigma_i, \quad i = 1, 2, \ldots, K, \tag{5.92}$$

minimizes (5.55). The proof is completed by showing that any permutation other than (5.92) can be improved, or is already equivalent to (5.92) because of nondistinct σ_i's.

Suppose some permutation other (5.92) is used and let $i^\star$ be the smallest i for which (5.92) is violated. Say $\varsigma_{2i^\star-1} = \sigma_j$ (instead of $\sigma_{i^\star}$). Then $j > i^\star$ because $j \le i^\star$ would contradict the definition of $i^\star$. Similarly, if $\sigma_{i^\star}$ is paired with σ_k, then $k < 2K - i^\star + 1$, for if not (5.92) would be violated at $i = 2K - k + 1 < i^\star$. ($k = 2K - i^\star + 1$ has been eliminated because this would imply that (5.92) is *not* violated at $i^\star$.)

We assert that the distortion is reduced by swapping $\sigma_{i^\star}$ and σ_j (unless $\sigma_{i^\star} = \sigma_j$ or $\sigma_k = \sigma_{2K-i^\star+1}$, in which case the distortion is unchanged and we may proceed by looking for larger $i^\star$). The first term in (5.55) is unaffected by the swap, but the second term is multiplied by

$$\left[\frac{(\sigma_{i^\star}^2 - \sigma_{2K-i^\star+1}^2)(\sigma_j^2 - \sigma_k^2)}{(\sigma_{i^\star}^2 - \sigma_k^2)(\sigma_j^2 - \sigma_{2K-i^\star+1}^2)}\right]^{1/K}.$$

When the σ's in question are distinct, this factor is less than one because of calculations for the two pair case. Since only (5.92) and equivalent permutations are not improved by this process, the theorem is proven.

5.C.4 Proof of Theorem 5.10

This appendix gives a proof of Theorem 5.10. A packing of lines as required by the theorem is called an optimal M packing in the $G(N, 1)$ Grassmannian space [31]. This packing is equivalent to the packing of antipodal spherical caps. The proof given here closely mimics computations by Wyner [212] on the optimal packing of (non-antipodal) spherical caps.

For a given N and angle θ, suppose a maximum size frame has been constructed which has minimum angle θ between frame elements. This frame has $M(N, \theta)$ elements. For each frame vector, construct a pair of antipodal spherical caps cut from the surface of the unit sphere with half angle θ, situated such that their axes align with the vector. The set of all such caps must cover the entire surface of the sphere; if not, a vector from the origin to the uncovered point could be added to the frame. Since the surface area of the sphere is

$$S_N = \Gamma\Big(\frac{N+2}{2}\Big)^{-1} N\pi^{N/2}$$

and the surface area of a single cap of half angle θ is

$$A_N(\theta) = \Gamma\Big(\frac{N+1}{2}\Big)^{-1} (N-1)\pi^{(N-1)/2} \int_0^\theta \sin^{N-2}\phi \, d\phi,$$

the covering property gives the following bound:

$$M(N,\theta) \ge \frac{S_N}{2A_N(\theta)} = \frac{N\sqrt{\pi}}{2(N-1)} \Gamma\Big(\frac{N+2}{2}\Big)^{-1} \Gamma\Big(\frac{N+1}{2}\Big) \Big(\int_0^\theta \sin^{N-2}\phi \, d\phi\Big)^{-1}. \tag{5.93}$$

The remainder of the proof uses asymptotic estimates as $N \to \infty$. Showing that $M(N,\theta)$ must grow exponentially with N for any $\theta < \pi/2$ will complete the proof.

Taking logarithms of both sides of (5.93) and dividing by N gives

$$\frac{1}{N}\ln M(N,\theta) \geq \frac{1}{N}\ln\frac{N\sqrt{\pi}}{2(N-1)} + \frac{1}{N}\ln\Gamma\Big(\frac{N+2}{2}\Big)^{-1}\Gamma\Big(\frac{N+1}{2}\Big) - \frac{1}{N}\ln\bigg(\int_0^\theta \sin^{N-2}\phi\, d\phi\bigg). \tag{5.94}$$

It is shown in [212, App. G] that

$$\lim_{N\to\infty}\frac{1}{N}\ln\bigg(\int_0^\theta \sin^{N-2}\phi\, d\phi\bigg) = \ln\sin\theta. \tag{5.95}$$

Taking the limit of (5.94) and using (5.95) gives

$$\lim_{N\to\infty}\frac{1}{N}\ln M(N,\theta) \geq -\ln\sin\theta. \tag{5.96}$$

Since M grows linearly with N, the left side of (5.96) is zero. This shows that for the construction specified in the theorem, the asymptotic value of θ must be $\pi/2$. This completes the proof.

Chapter 6

Computation-Optimized Source Coding

THE MOST well-known aspects of information theory are bounds on the performance of communication systems. These theoretical bounds depend on the way that sources and channels are modeled, but not on the technology of encoding and decoding. The practice of communication is, however, limited by the technology used in realizing the encoder and the decoder—both the algorithms and the hardware. In addition, performance requirements may introduce a constraint inconsistent with the theory, for example, a maximum delay. Thus, even with the simplest of sources and channels, there will generally be a gap between the "optimal" performance and the performance of a constructed system.

Imagine that you are faced with the problem of designing a source coder for a memoryless source with a known distribution. Guided only by a random coding proof of the achievability of the rate–distortion function [34], you would take the following steps: choose a block length N; for rate R, generate a codebook with 2^{NR} sequences of length N by randomly drawing samples from the source; encode by nearest-neighbor rule. This is highly impractical and would not perform well. It is impractical because nearest-neighbor encoding has high complexity, unless the block length is very low (see Table 1.1). The performance would be improved by using the samples drawn from the source in an iterative codebook design algorithm. This demonstrates that an analytical framework for optimal rate–distortion performance does not directly lead to techniques with good operational performance.

Taking a different point of view, we may ask: What is the best compression performance possible with C instructions per source sample on microprocessor X? The question suggests that an arbitrarily long sequence of samples will be processed and that the constraint applies to the limiting average. In this form, the problem is certainly unsolvable because the number of possible execution sequences grows without bound.

This chapter takes a "doubly operational" approach to computation-optimized source coding, since optimization is performed over a precisely defined set of algorithms and the performance criterion is operational rate–distortion. Applications to transform coding are analyzed in detail, along with a few others.

This chapter includes research conducted jointly with Martin Vetterli [72, 73].

6.1 Introduction

It is clear to all source coding practitioners that computational complexity is of importance, and there is no question that a lot of effort has gone into optimizing certain calculations which are common in source coding; *e.g.*, matrix multiplications, filtering operations, and discrete cosine transforms. But, at least amongst theoreticians, there seems to be a gap between the design of coding algorithms and their computational optimization. One occasionally finds discussions of computational trade-offs in papers on coding algorithms, but rarely finds precise numerical comparisons or justifications. Comparisons between algorithms should ideally be done by comparing their performances with a fixed computational budget. Since many coding methods are at least partially computation-scalable (for example, by changing the block size in block transform coding or vector quantization), this is a sensible objective.

This chapter describes a flexible framework for systematically studying the trade-off between computational complexity and coding performance. When specialized to a communication scenario, it yields *operational computation–rate–distortion*, to replace operational rate–distortion. The value of the framework itself is to provide a common vocabulary; it does not intrinsically aid in the analysis.

The bulk of the chapter is devoted to analyses of transform coding. Section 6.3.1 addresses the efficacy of the Karhunen–Loève transform (KLT) and the discrete cosine transform (DCT) for block transform coding of a Gauss–Markov source. Stochastic simulations are avoided through the use of reasonable approximations and explicit calculations. Section 6.3.2 considers the even more practical problem of JPEG encoding and decoding. Through the analysis of a set of simplified encoding and decoding algorithms, a precise characterization of an achievable set of rate–PSNR–complexity triples is found. ("Simplified" means that certain calculations that would be performed in a standard JPEG encoder/decoder are omitted even though this will generally impair the rate–distortion performance.) In particular, it is shown that for a fixed rate, the computation vs. distortion trade-off exhibits a diminishing returns characteristic, so the computation can be reduced somewhat with little effect on image quality. An application outside of source coding is briefly outlined in Appendix 6.A.

Before the introduction of the framework in Section 6.2, the following subsection reviews some related results from the information theory literature, and Section 6.1.2 gives a brief overview of complexity measures.

6.1.1 Performance versus Complexity in Information Theory

The extraordinary importance of channel capacity stems from the fact that it provides not just a bound on the rate of reliable communication, but, through the channel coding theorem [34, Thm. 8.7.1], a tight bound. The rate–distortion function is similarly tight; it is a lower bound on the achievable rate, and all higher rates can be achieved [13]. Thus, in his original paper [170], Shannon "solved" the communication problem, at least for memoryless sources. The problem is that the random coding arguments used in proofs of these results suggest codes that are very difficult to implement. Specifically, each suggests a sequence of codes, each of which is an unstructured mapping from a sequence of source symbols to a sequence of channel symbols. Attaining distortion close to the distortion–rate function evaluated at the channel capacity requires the processing of long sequences. Unfortunately, the encoding and decoding operations require searches over sets with numbers of elements exponential in the sequence length, so performance near the theoretical bound comes with unrealistically high complexity.

The relationship between performance and complexity has been been a central research topic in channel coding.[1] Consider a system that uses a block code of length n and maximum likelihood (ML) decoding for communication over a memoryless channel. Gallager [58] showed that, when the code is chosen optimally, the probability of error decreases exponentially with the block length as

$$P(\text{error}) \approx \exp(-nE(R)).$$

$E(R)$, called the *error exponent*, is a function of the rate R. It is positive for rates less than the capacity; thus, the probability of error can be made arbitrarily small at rates less than the capacity. Suppose the complexity is measured by the number of operations in the ML decoding. The decoding complexity increases exponentially with N and the probability of error decreases only subexponentially with the complexity. This subexponential rate inspired Forney's study of concatenated codes [55]. He showed that with the appropriate choice of inner and outer codes it is possible to have probability of error that decreases exponentially with complexity. The explosive popularity of turbo codes [16] is due to the efficacy of iterative decoding of certain concatenated codes. In particular, it is not just that the codes themselves are good for a given constraint length, but that the complexity versus performance trade-off is good.

Source encoding with an unstructured codebook is dual to maximum likelihood decoding of an unstructured channel code. Both involve searching over a codebook for a minimum distance element. In source encoding, the distance measure is specified by a fidelity (distortion) criterion, while in channel decoding the distance measure incorporates the channel model. Recalling that an "unstructured source code" could also be called an unstructured vector quantizer (UVQ), the duality reinforces that the encoding complexity of UVQ is exponential in the vector length, as shown in Table 1.1.

Much of the activity in VQ research is aimed at replacing UVQ with methods having lower encoding complexity. The lower complexity is accompanied by at least one of the following: a constraint on codebook structure that may preclude optimality, suboptimal encoding for a given codebook, and increased storage. Many such methods have been proposed and all but the most recent of the important variants are described in [60].

Tree-structured and *tree-searched* VQ are important and illustrative examples. In tree-structured VQ (TSVQ) [23], exhaustive search over the full codebook is replaced by a sequential encoding operation. For convenience, neglect the possibility of entropy coding and assume a balanced, binary tree, though any tree could be used.[2] Encoding an n-dimensional source at rate R bits per component involves the use of a binary decision tree of depth nR. Each of the nodes is labeled with an n-tuple and the 2^{nR} leaves correspond to the codewords. The mapping of a source vector to a codeword is completed by traversing the tree from the root to a leaf, making nR binary decisions. The decision at each node is to choose the child node whose label is closer to the source vector. Comparing to Table 1.1, the space complexity is doubled because each intermediate node is labeled with a vector, but the number of distance calculations is reduced from 2^{nR} to $2nR$.

While TSVQ does not constrain the locations of the codewords, it suffers from suboptimal encoding: source vectors are not necessarily mapped to the nearest codeword. Still, TSVQ is popular in practice because the performance loss is offset by the lower encoding complexity. Designing an optimal decision tree is an NP-complete problem [101], so TSVQ design generally follows one of two heuristic strategies: tree-growing [130, 162] or tree-pruning [29]. Tree-pruning will be discussed further in Section 6.3.3, where joint optimization with respect

[1] The summary here follows [56]. See [56] and references therein for more details.
[2] Decision trees are described fully in [19]. Only cursory familiarity is needed here.

to rate, distortion, and encoding complexity will be considered. Though UVQ is the subject of many analytical studies, [143] and [122] are notable as two of few papers to analytically predict or bound the performance of TSVQ.

In tree-searched VQ (TS-UVQ),[3] full-search over the codebook is replaced without sacrificing optimality of encoding. The encoding process combines the use of a binary decision tree with a final search. First the tree is followed from the root to a leaf. As in TSVQ, at each intermediate (non-leaf) node, the branch traversed depends on which side of a hyperplane the source vector lies. Each leaf has associated with it not just a single codeword but a set of codewords, called a *bucket*. The encoding is completed by a full search over the codewords in the bucket. Viewing the decision tree as a TSVQ, the bucket associated with a leaf contains all the codewords whose Voronoi cells intersect the cell of the TSVQ.

The design of a TS-UVQ encoder is subtle. If one insists that each bucket contain a single codeword, eliminating the search step, then the average depth of the tree will be high—perhaps much higher than the logarithm of the number of codewords.[4] On the other hand, little is gained from the decision tree if its depth is too low. An effective design method and many earlier methods are described in [155].

Tree-searching was used in a very interesting study by Moayeri and Neuhoff [135]. They showed that in TS-UVQ and tree-based fine–coarse VQ (which will not be described here), time- and space-complexity can be traded off while approximately maintaining a particular operational rate–distortion performance point. Performance versus memory trade-offs appear also in universal lossless coding [213, 232].

Before turning to a discussion of complexity measures, a final note is in order. The closest analogy in source coding to the error exponent result for channel coding would be to look at the best possible performance of a source coder that acts independently on blocks of length n. This is a standard concept in rate–distortion theory; the bounding functions are denoted $R_n(D)$ and $D_n(R)$ in [13]. Results on the convergence of $R_n(D)$ to $R(D)$ are summarized in [13, Section 6.2.1].

6.1.2 Complexity Measures

Without the benefit of any formal definitions, the previous section made declarations regarding the complexities of various algorithms. Complexity was measured with the intuitive notion of counting "elementary" operations (arithmetic operations, comparisons, branches) in a sequential computation. This form of complexity will be used routinely in this chapter, but many other formal concepts of complexity have been developed.

Multiplicative complexity The field of multiplicative complexity is based on minimizing the number of multiplications and divisions in computing a function for an arbitrary input. During the peak of this field in the 1970's and '80's, counting multiplications and divisions while ignoring additions and subtractions was justified by the difference in computation times for these operations on the hardware of the day. In addition, the framework was chosen to facilitate the advance of the theory. Many tight lower bounds have been proven, and, of course, any working algorithm yields an upper bound on complexity. Several of the key results in this field are due to Winograd [207].

In multiplicative or arithmetic complexity theory, it is typical to consider the computation of quantities

[3] This abbreviation, following [135], emphasizes that the result of the encoding is equivalent to full-search UVQ.
[4] If it is not clear that it is possible for each bucket to contain a single codeword, see [60, Figures 10.10–11].

of the form[5]

$$\psi_k = \sum_{j=1}^{s} \sum_{i=1}^{r} a_{ijk} x_i y_j, \qquad \text{for} \quad k = 1, 2, \ldots, t, \tag{6.1}$$

where the x_i's and y_j's denote arbitrary real input data and the a_{ijk}'s are real constants independent of the data. In general this is called a *bilinear form*; if the y_j's all equal one, it is a *linear form*. The problem is to compute (6.1) with a minimum number of general multiplications and divisions (m/d steps). An m/d step is any field operation (+, -, ×, /; denoted ∘) other than:

- $b \pm c$; or
- $b \circ c$ with b and c both independent of the indeterminates $\{x_i\} \cup \{y_j\}$; or
- $b \times c$ with $b \in \mathbb{Q}$ (the rational numbers).

In essence, only multiplications and divisions where both operands are general (*i.e.*, not necessarily rational) are counted, except those that could be precomputed independently of the input data. The justification for ignoring rational multiplications is that they can be computed as a sequence of additions and a final rescaling of the problem.

A sense of the style of multiplicative complexity theory is revealed by looking at the multiplication of two 2×2 matrices. The product

$$\begin{bmatrix} \psi_{11} & \psi_{12} \\ \psi_{21} & \psi_{22} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}$$

can be written as

$$\begin{aligned} \psi_{11} &= x_{11}y_{11} + x_{12}y_{21}, \\ \psi_{12} &= x_{11}y_{12} + x_{12}y_{22}, \\ \psi_{21} &= x_{21}y_{11} + x_{22}y_{21}, \\ \psi_{22} &= x_{21}y_{12} + x_{22}y_{22}, \end{aligned}$$

showing that it can be computed with eight multiplications and four additions. Strassen's algorithm [181] is a clever rearrangement, with some reuse of intermediate answers, that computes the same result with only seven multiplications:

$$\begin{aligned} h_1 &= (x_{12} - x_{22})(y_{21} + y_{22}), \\ h_2 &= (x_{11} + x_{22})(y_{11} + y_{22}), \\ h_3 &= (x_{11} - x_{21})(y_{11} + y_{12}), \\ h_4 &= (x_{11} + x_{12})y_{22}, \\ h_5 &= x_{11}(y_{12} + y_{22}), \\ h_6 &= x_{22}(y_{21} + y_{11}), \\ h_7 &= (x_{21} + x_{22})y_{11}, \end{aligned}$$

[5] Notation is taken from [207].

Microprocessor	MIPS R3010	Weitek 3364	TI 8847
Cycles/add	2	2	2
Cycles/mult	5	2	3
Cycles/divide	19	17	11
Cycles/square root	-	30	14

Table 6.1: Summary of relative execution times for arithmetic operations on three floating-point microprocessors. Reproduced from [95].

$$\begin{aligned}
\psi_{11} &= h_1 + h_2 - h_4 + h_6, \\
\psi_{12} &= h_4 + h_5, \\
\psi_{21} &= h_6 + h_7, \\
\psi_{22} &= h_2 - h_3 + h_5 - h_7.
\end{aligned}$$

It can be shown that the number of multiplications cannot be reduced below seven.

The following two results are used in Section 6.3.1.1.

Proposition 6.1 *If $n = 2^k$, $k \in \mathbb{Z}$, the multiplicative complexity of multiplying two $n \times n$ matrices is at most $7^k = n^{\log_2 7}$.*

Proof: Strassen's algorithm as described above does not require commutativity, and hence can be applied to multiplication of block matrices. Recursively halving the size of the $n \times n$ matrix multiplication problem and using Strassen's algorithm gives a method with 7^k multiplications. The number of additions also grows only as $n^{\log_2 7}$ [18]. □

Proposition 6.2 *If $n = 2^k$, $k \in \mathbb{Z}$, the multiplicative complexity of computing a length n DCT is $2^{k+1} - k - 2$.*

Proof: No elementary proof is known; see [93]. □

Note that Proposition 6.1 is not tight for large n. Denoting the minimum complexity of multiplying $n \times n$ matrices by $O(n^\omega)$, ω is called the *exponent of matrix multiplication*, and determining upper bounds for the exponent has received considerable attention. The current "world record" of $\omega < 2.38$ was obtained by Coppersmith and Winograd; see [22] for details. Determination of the minimum exponent is a formidable research area, tangential to this chapter. Strassen's algorithm was exhibited as a concrete example of an algorithm with exponent less than three.

On current hardware, the execution time difference between an addition and a multiplication is not very large (see Table 6.1). Moreover, actual execution time depends on memory access times, cache hit rate, instruction and data pipelining, and many other factors [95]. A further shortcoming of multiplicative complexity theory is that it concerns only exact answers computed from exact data in exact arithmetic.

Kolmogorov and Chaitin complexities Multiplicative complexity is intended to reflect the minimum time needed to compute a function on a serial computer. An alternative is to gauge the length of the shortest program

that computes a desired result. Complexity metrics of this type were developed by Kolmogorov [110, 111] and Chaitin [25, 26].

We need not look any further than the matrix multiplication example above to see that this can give completely different relative complexities. Although we would have to agree on a syntax for programs, it is clear that Strassen's algorithm would require a longer program than the straightforward calculation with eight multiplications. Similarly, one would expect TS-UVQ and TSVQ to require longer programs than full-searched UVQ, even though the programs may terminate more quickly.

Complexity based on program length also allows one to define a quantity like Shannon's entropy for deterministic strings. This type of complexity measure will not be pursued further in this chapter, in part because it is not suitable for operational optimization. The reader is referred to [121, 150, 235], in addition to the references above, for details.

Other computational models Let us note again that multiplicative complexity is reasonably well-suited to computations that are completed through a sequence of calculations on current serial computing hardware. Another concrete computational model that has been studied extensively is VLSI [197]. It has been shown that the fundamental limits to the computing power of a VLSI circuit depend on the product of the area of the circuit and the square of the time the circuit is allowed to compute. The computational model matters, as evidenced by comparing the sorting of N elements and a length N DFT. Both are typically considered to have $O(N \log N)$ complexity,[6] but their VLSI complexities are $\Omega(N^2 \log N)$ and $\Omega(N^2 \log^2 N)$, respectively [186, 187].[7]

It is completely possible that new computational models—like quantum, optical, or biochemical computing—will turn upside down our view of easy and hard computations. Thus the best we can hope for is to optimize within the constraints of a particular computational model. This is the approach taken in the sequel.

6.2 An Abstract Framework

With a few ideas of how to measure computational complexity, we are now ready to establish a formal framework for optimization. Let $\mathcal{P}$ be a set of computational problems which are posed according to some underlying probability distribution and let ρ be a distortion measure on approximate solutions to problems in $\mathcal{P}$. Suppose also there is a computational cost function on algorithms for (approximately) solving $P \in \mathcal{P}$, $c : \mathcal{A} \times \mathcal{P} \to \mathbb{R}^+$, where $\mathcal{A}$ is a set of such algorithms. Then define the *distortion–computation function* of algorithms $\mathcal{A}$ for problems $\mathcal{P}$ by

$$D(C) = \inf_{\{A \in \mathcal{A} \,:\, Ec(A,P) \le C\}} E\rho(P, A(P)). \tag{6.2}$$

In the context of source coding, we can specialize the definition. Consider the problem of finding a variable-length, approximate representation of a source with expected length bounded above by R. Denote the source and the reproduction by x and $\hat{x}$, respectively. Then, define the *distortion–computation function at rate* R by

$$D_R(C) = \inf_{\{A \in \mathcal{A} \,:\, Ec(A,x) \le C,\, E\ell(\hat{x}) \le R\}} E\rho(x, \hat{x}), \tag{6.3}$$

[6] See [108] for the complexity of sorting.
[7] The $\Omega(\cdot)$ notation means "grows at least as fast as."

where $\ell(\hat{x})$ is the length of the representation of $\hat{x}$ in bits. Notice that in contrast to the definition of a rate distortion function [34], we do not use the mutual information between x and $\hat{x}$. Doing so would implicitly assume that the entropy coding of $\hat{x}$ is ideal; instead, we would like to leave open the possibility that the entropy coding is included in the computational cost. Varying the parameter R yields a *computation–rate–distortion surface*.

A few properties of these functions are obvious: $D(C)$ must be nonincreasing and $D_R(C)$ must be nonincreasing with respect to both R and C. If the set of algorithms produces a discrete set of complexities, as is the case in Section 6.3.2, then $D(C)$ may be only piecewise continuous. If we allow time- or probabilistic-multiplexing, then operational $D(C)$ and $D_R(C)$ can always be made convex. The optimization of a system with independent units that each contribute additively to the computation and distortion is also obvious; if all the needed slopes are available, a Lagrangian solution can be used.

6.3 Applications to Source Coding

This section presents three applications of distortion–computation analysis and optimization. The first application is a detailed comparison between the KLT and DCT for transform coding of a Gauss–Markov source. In this same context, Gormish and Gill [65] made earlier mention of the concept of a computation–rate–distortion surface. Their analysis is similar to the one presented here, but less detailed. The other applications are to JPEG image encoding and pruned tree-structured vector quantization encoding.

6.3.1 Autoregressive Sources: KLT vs. DCT

The usual justification of using the DCT rather than the KLT in transform coding is that the DCT is a fixed transform which can be implemented with a fast algorithm; thus, even if the KLT would give better rate–distortion performance than the DCT for a fixed block size n, the DCT may be preferable because it makes larger values of n feasible. The distortion–computation function approach allows a precise characterization of this trade-off.

As an example, we will consider the transform coding (with scalar quantization) of a Gaussian first-order autoregressive source X with correlation coefficient α, *i.e.*, a source with autocorrelation sequence $r_X(m) = \alpha^{|m|}$. Distortion is measured by MSE per sample. First, algorithms based on an exactly computed transform followed by quantization are considered, and complexity is measured by the minimum number of general multiplications.[8] This case allows for many precise statements but is of limited practical consequence. A more relevant comparison uses the numbers of multiplications in typical implementations. A scenario with variable-precision KLT computation is also considered.

6.3.1.1 Coding with exact computations

Computing distortion As a preliminary to finding operational distortion–computation functions for DCT and KLT coding, we first study the $D(R)$ performance of these methods as the block size n is varied. This step relies on certain assumptions about the coding process, namely in relation to the bit allocation and design of the

[8] See Section 6.1.2.

scalar quantizers. No assumptions about the computational model are needed. For comparison, the performance attained without any transform and the optimal performance attainable for any method that utilizes correlation only within n-tuples are exhibited.[9]

Denote the KLT for block size n by T_n, *i.e.* $T_n R_X T_n^T = \Lambda$, where Λ is a diagonal matrix with nonincreasing entries. Let U denote a DCT matrix given elementwise by[10]

$$u_{ij} = \begin{cases} \sqrt{\frac{1}{n}} \cos(\frac{\pi}{n}(i-1)(j-\frac{1}{2})), & i = 1, \\ \sqrt{\frac{2}{n}} \cos(\frac{\pi}{n}(i-1)(j-\frac{1}{2})), & i = 2, 3, \ldots, n. \end{cases}$$

Since the quantization is scalar, the performance depends only on the transform coefficient variances, which are given (without regard to ordering) by $(\lambda_1, \lambda_2, \ldots, \lambda_n) = \mathrm{diag}(T_n R_X T_n^T)$ and $(\mu_1, \mu_2, \ldots, \mu_n) = \mathrm{diag}(U R_X U^T)$ for KLT and DCT coding, respectively.

For large n, the approximation

$$\lambda_k \approx \mu_k \approx S_X(\tfrac{2\pi k}{n}) \tag{6.4}$$

holds, where

$$S_X(\omega) = \frac{1-\alpha^2}{1 - 2\alpha\cos\omega + \alpha^2}$$

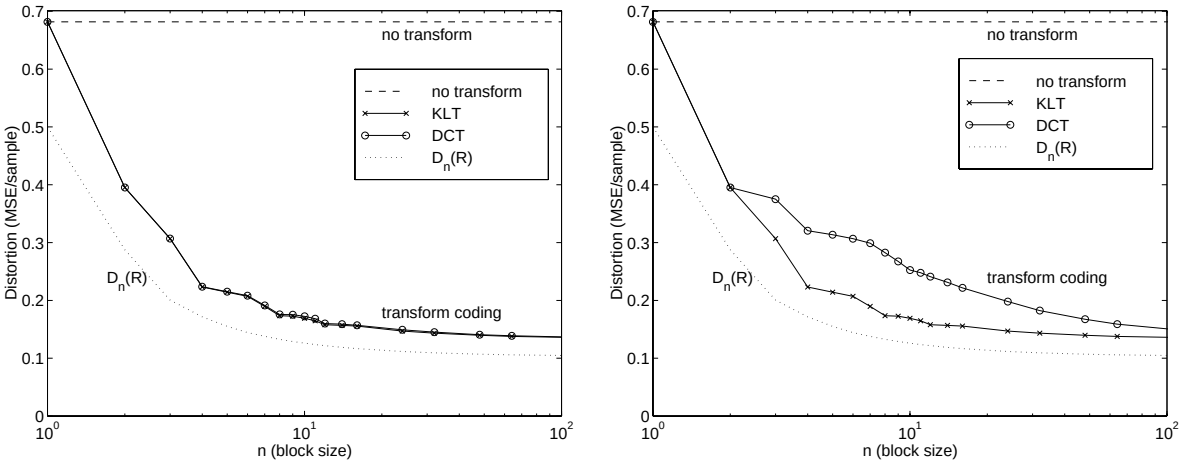
is the power spectral density of X [87, 83]. This approximation, however, dismisses the coding gain difference between the KLT and the DCT and obscures the dependence on n, so in the remainder of the chapter the exact values of the λ_k's and μ_k's are used.

Using high rate approximations and assuming optimal scalar quantization leads to the bit allocation (1.9), but by abandoning high rate approximations one can obtain more precise and realistic results. In particular, instead of using approximate expressions for optimal companding, the distortions given in this section follow from using nonnegative integer bit allocation obtained with a greedy algorithm [60, §8.4] and uniform quantization with optimal loading. The results are shown in Figure 6.1, along with the performance obtained with no transform and $D_n(R)$. Note that for $\alpha = 0.9$, the performance of KLT coding is virtually indistinguishable from that of DCT coding. On the other hand, the performance gap is significant for $\alpha = -0.9$, although as in the previous case, the DCT is asymptotically equivalent to the KLT. Note also that the $D_n(R)$ curve is based on using a transform of length n and then coding each of n coefficient streams *at the rate–distortion bound*. This explains the gap between $D_n(R)$ and the other curves even for $n = 1$.

Estimating computational load Consider first the computational complexity measure given by the number of multiplications between arbitrary real numbers per input sample. This model is familiar because of its connection to Winograd convolution algorithms and is described in Section 6.1.2.

The computation of any square linear transform of size n can be viewed as a multiplication between an $n \times n$ matrix and an $n \times 1$ vector. Since the KLT does not in general have a structure conducive to a fast algorithm, the KLT algorithms that minimize the number of multiplications are simply those that use the most efficient matrix multiplication techniques. For convenience, assume that $n = 2^k$, $k \in \mathbb{Z}^+$. To code n vectors at a time would entail multiplying pairs of $n \times n$ matrices, which can be done with $n^{\log_2 7}$ multiplications

[9] The latter quantities are $R_n(D)$ points as defined in [13].
[10] This is the "original" DCT first reported in [3] and classified as DCT-II in [159].



(a) Source correlation coefficient $\alpha = 0.9$ (b) Source correlation coefficient $\alpha = -0.9$

Figure 6.1: Operational $D(n)$ for DCT and KLT coding of first-order autoregressive sources at rate 0.5 bits/sample. This is based on the use of greedy nonnegative integer bit allocation and optimally loaded uniform scalar quantizers. Performance with no transform and a theoretical bound are also shown. Note that the block size is given on a logarithmic scale.

using Strassen's method (see Proposition 6.1). Normalizing by n^2, the number of samples transformed in each multiplication, gives a multiplicative complexity of

$$C_{KLT} = n^{(\log_2 7)-2} \text{ multiplies/sample.} \tag{6.5}$$

On the other hand, the special structure of the DCT allows calculations to be done much more efficiently than as a general matrix multiplication, especially when n is a power of two. The minimum number of multiplications to compute a length-n DCT is $2n - \log_2 n - 2$ (see Proposition 6.2). Normalizing gives

$$C_{DCT} = 2 - \tfrac{1}{n}(\log_2 n - 2) \text{ multiplies/sample.} \tag{6.6}$$

When using moderate block sizes and typical computer architectures, algorithms that minimize the number of multiplications are generally not efficient. For example, while Strassen's algorithm for multiplying a pair of 2×2 matrices uses only 7 multiplications (instead of the usual 8), it increases the number of additions from 4 to 18. Similarly, DCT algorithms that have very low numbers of multiplications tend to have more additions and more complicated data flow. Therefore we would like to also compare multiplicative complexity for typical implementations of the KLT and DCT.

Computing the KLT using typical matrix–vector multiplication requires n^2 multiplications. If $m < n$ of the transformed components are allocated bits in the coding, the computational load can be reduced to mn multiplications (or m multiplications per sample) by computing only the components that will be coded. For fixed n, m is determined explicitly by the bit allocation algorithm. One approximation of m is the number of components allocated at least one bit in (1.9). For large n, combining this with (6.4) yields

$$\frac{m}{n} \approx \frac{\omega^*}{\pi},$$

where $\omega^* \in [0, \pi)$ is a solution of $(1-\alpha)^2 2^{2R} = 1 - 2\alpha\cos\omega + \alpha^2$. Since this approximation masks the difference in energy compaction between the KLT and the DCT, explicitly computed greedy nonnegative integer bit allocations are instead used in the following.

When n is a power of two, one possible implementation of the DCT (which does not have an inordinate number of additions) has $\frac{1}{2} n \log_2 n$ multiplications [159]. To maintain an analogy with the multiplication count for the KLT, we should consider pruned computations that determine only the DCT coefficients with positive bit allocations. The complexities of pruned DCT algorithms are not easily captured in a single expression, so we use the following simple bound:

$$C_{DCT} = \min\{m, \tfrac{1}{2}\log_2 n\} \text{ multiplies/sample}, \tag{6.7}$$

where as above m is the number of coefficients allocated at least one bit. This reflects the strategy of using a matrix multiplication when it is more efficient than a full DCT.

Operational distortion–computation Combining the $D(n)$ and $C(n)$ expressions derived above gives parametric descriptions of the operational $D(C)$ for KLT and DCT coding. Figure 6.2(a) shows operational $D(C)$ curves for coding at 0.5 bits/sample when computation is estimated using (6.5)–(6.6). This graph shows $D_{DCT}(C) < D_{KLT}(C)$ for all computational budgets C. The graph also shows the distortion that is obtained when no transform is used, and the signal is simply subjected to uniform scalar quantization. This is indicated with an arrow because zero multiplications is off the left edge of the plot.

The operational $D(C)$ curves look somewhat different when the computational complexity is measured using (6.7) and the corresponding expression for the KLT. These operational curves are shown in Figure 6.2(b). For $\alpha = 0.9$, the DCT is superior for all computational budgets as before. On the other hand, for $\alpha = -0.9$, the DCT is superior only when the block size is larger than about 64.

Limitation of the computational model The calculations made thus far pertain only to *operational* $D(C)$ for various implementations of two particular transform methods. The true distortion–computation function over the class of algorithms that follows a linear transform with scalar quantization—where complexity is measured by the number of general multiplications—has a very simple form. For any block size n, one can approximate the KLT of the source by a rational matrix. Since multiplication by rational numbers has no cost, the computational complexity of using this transform is zero. Making n arbitrarily large and using an arbitrarily good approximation of the KLT gives that $D(C)$ is a constant for all C, with distortion given by the infimum of distortions over all transform methods.

The infeasibility of using the coding strategy described above highlights the importance of having a good computational complexity metric. A good metric reflects the actual cost in some application environment; *e.g.*, execution time with particular hardware, or hardware costs to meet certain performance specifications. Yet at the same time, when the set of algorithms is limited to practical schemes as in Figure 6.2(b), this framework provides a reasonable comparison between algorithms.

6.3.1.2 Coding with finite precision computations

The second set of computational complexity calculations in Section 6.3.1.1 used the fact that transform coefficients that are not allocated any bits need not be calculated. This can be viewed as a special case of the

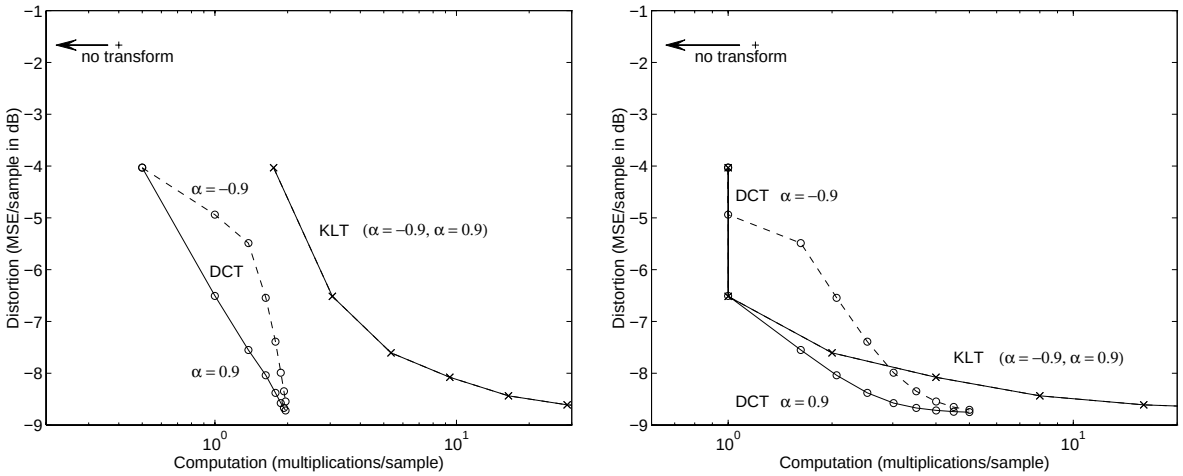


Figure 6.2: Operational $D(C)$ for DCT and KLT coding of first-order autoregressive sources with correlation coefficients $\alpha = -0.9$ and $\alpha = 0.9$ at rate 0.5 bits/sample. Block sizes are limited to powers of two and are given on a logarithmic scale. (a) The complexity is based on implementations that minimize the number of multiplications. (b) The complexity is based on typical implementations without inordinate numbers of additions.

principle that transform coefficient calculations need only be accurate enough to determine proper *quantized* values.[11] More generally, it may not be important to compute a transform coefficient with high accuracy if it will be quantized coarsely. This section presents an analysis of the relative benefits of spending computational resources on accurate transform coefficient calculation and on increased block lengths.

Consider a KLT-based coding system which uses nonnegative integer bit allocation. As before, denote by m the number of transform coefficients that have positive bit allocations. If the ith transform coefficient is computed with a B_i-bit mantissa, a reasonable cost function for a hardware implementation is

$$C = \sum_{i=1}^{m} B_i^2 + B_i(1 - \tfrac{1}{n}).$$

This is based on costs of B^2 and B for each B-bit multiplication and addition, respectively. Modeling each roundoff as the addition of uniformly distributed noise and using the central limit theorem leads to the approximation $\hat{y}_i = y_i(1 + \delta_i)$ for each computed transform coefficient, where y_i is the exact transform coefficient and $\delta_i \sim \mathcal{N}(0, 2^{-2B_i}/3n)$. An optimization was performed based on the additional assumption that the quantization error is independent of the errors from finite precision computations. The resulting operational $D(C)$ is as shown in Figure 6.3.

6.3.2 JPEG Encoding with Approximate DCT

In a typical transform image coder, there is no explicit bit allocation. Instead, zigzag scanning and run length entropy coding cause the average bit rate attributable to high-frequency coefficients to be low, without

[11] Obviously, determining which of a finite set of intervals a transform coefficient lies in cannot be harder than calculating the coefficient precisely.

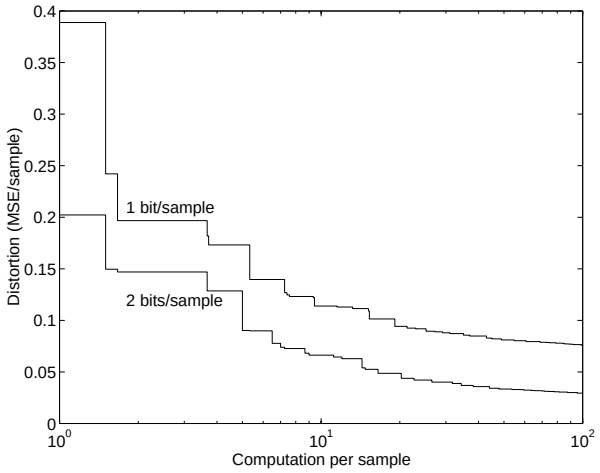


Figure 6.3: Operational $D(C)$ for transform coding of a first-order autoregressive source with correlation coefficient $\alpha = 0.9$ in which the block length and computational precisions of each transform coefficient are jointly optimized.

forcing the allocation of no bits. This precludes the computational optimizations of the previous section. Is there any point in spending computational resources to compute a transform coefficient that will likely be quantized to zero, even if occasionally it would have a nonzero quantized value? This question is explored in this section in the context of JPEG encoding.

6.3.2.1 Analysis

Of the major steps in JPEG coding (DCT, scalar quantization, zigzag scanning, run length coding, and entropy coding), only the computation of the DCT seems to be computation scalable; thus, let us focus on this step. $\mathcal{P}$, $\mathcal{A}$, ρ, and c must be explicitly defined in order to use the computation–rate–distortion framework:

$\mathcal{P}$ = JPEG-compatible encoding at R bits/pixel.

$\mathcal{A}$ = Approximate DCT followed by standard JPEG quantization and entropy coding. The approximate DCT is described below.

ρ = MSE

c = Number of multiplications per block in the approximate DCT computation.

The set of approximate DCT algorithms are algorithms that compute a subset of the DCT coefficients and assume that the remaining coefficients are zero. The image blocks in JPEG are 8×8, so there are 2^{64} approximate algorithms. Because of the frequency domain characteristics of natural images, it is clear that a small number of the possible algorithms are of interest. In this investigation, the set of calculated coefficients is limited to be a triangle or rectangle of low frequency coefficients, as shown in Figure 6.4. The use of the rectangular regions is motivated by the trade-off between the number of coefficients calculated and the computational complexity. Triangular regions are motivated by the zigzag scanning of AC coefficients.

The DCT calculations are done separably using *output-pruned* decimation-in-frequency 1-D DCT algorithms. These are decimation-in-frequency length-8 DCT algorithms [159] which are simplified because only

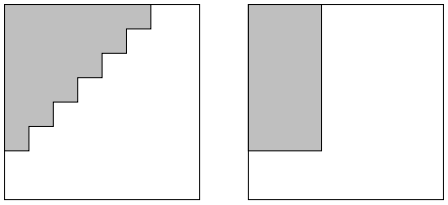


Figure 6.4: The DCT coefficients computed form either a triangle or rectangle. The lowest frequencies are in the upper left.

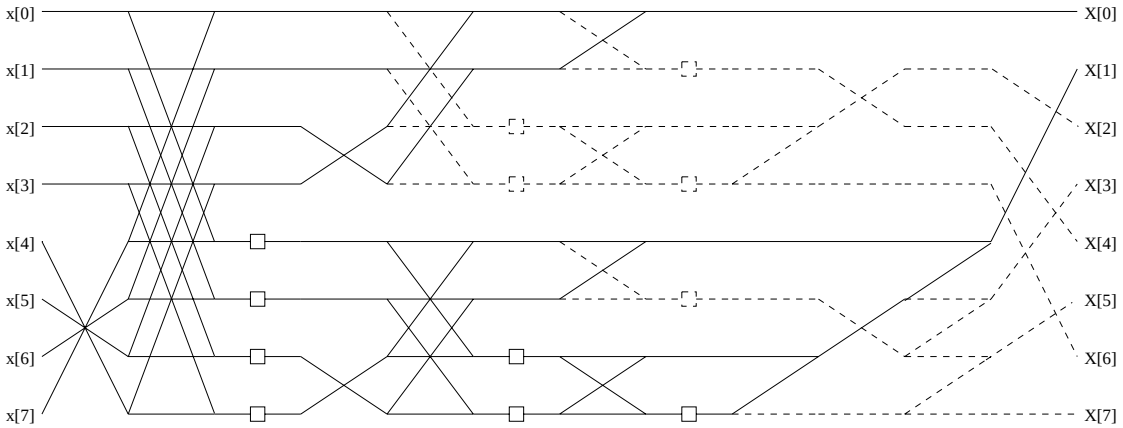


Figure 6.5: Output-pruned signal flow graph for computing the first two coefficients of a length-8 DCT. The full signal flow graph is from [159, p. 61] and represents a decimation-in-frequency algorithm. (Boxes represent multiplications other than by ± 1 and subtractions are not distinguished from additions.) The dotted curves represent calculations that can be eliminated because we desire only $X[0]$ and $X[1]$. The computational complexity is reduced from 12 multiplications and 29 additions for the full calculation to 7 multiplications and 20 additions.

a subset of the eight coefficients are desired. Figure 6.5 shows an output-pruned DCT where only the first two coefficients ($X[0]$ and $X[1]$) are desired. Because of the shapes of the regions considered (see Figure 6.4), we always require the k lowest frequency DCT coefficients, $1 \leq k \leq 8$. The computational complexities for these output-pruned 1-D DCTs are given in Table 6.2.

From Table 6.2 it might seem that significant computational savings are achieved only if, say, less than three of eight DCT coefficients are calculated; but, a large fraction of the savings comes from the 2-D separable nature of the computation. For example, computing the 4×4 block of lowest frequency coefficients requires eight horizontal DCTs from which the first four coefficients are desired (8 times 11 multiplications) and *four* vertical DCTs from which the first four coefficients are desired (4 times 11 multiplications) for a total of 132 multiplications. This is roughly two-thirds of the 192 multiplications for a full 8×8 DCT. The situation is very similar for decoding. If we assume that the DCT coefficients outside of a certain region are zero, we can use input-pruned 1-D inverse DCT algorithms. A signal flow graph for one such algorithm is shown in Figure 6.6.

6.3.2.2 Results

A large number of computation–rate–distortion points were determined for the standard *Lena* image [117] using approximate DCT calculations as described. The remaining components (quantization, zigzag

k	multiplications	additions
1	0	7
2	7	20
3	10	25
4	11	27
5	12	29
6	12	29
7	12	29
8	12	29

Table 6.2: Number of multiplications and additions to compute the first k coefficients of a length-8 DCT using algorithms designed through output-pruning.

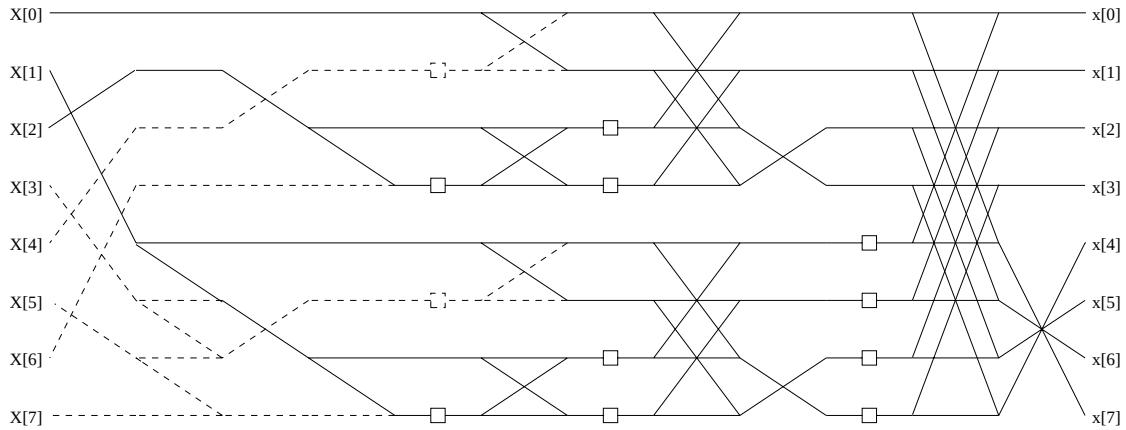


Figure 6.6: Input-pruned signal flow graph for a length-8 inverse DCT where the last five input coefficients are zero. The full signal flow graph is the inverse of Figure 6.5. Dotted curves represent calculations that can be eliminated because the result is known to be zero. The computational complexity is reduced from 12 multiplications and 29 additions for the full calculation to 10 multiplications and 25 additions.

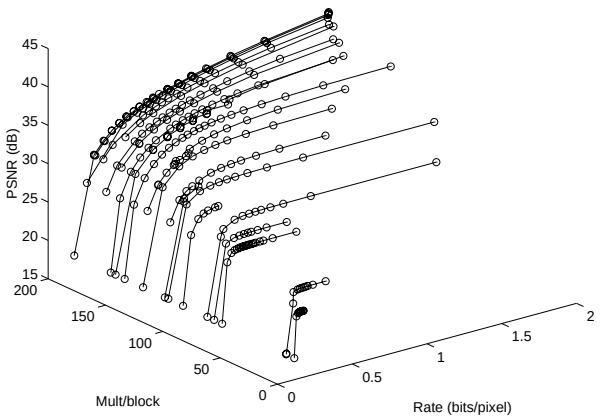


Figure 6.7: Operational $C - R - D$ for JPEG encoding of *Lena* with approximate DCT algorithms designed through output pruning.

scanning, and entropy coding) were implemented as in an Independent JPEG Group encoder[12] with "quality factors" 5, 10, ..., 95, and 99. The points with rates less than 2 bits per pixel which are presumably on the computation–rate–distortion surface are shown in Figure 6.7, where the connected points are at the same complexity.[13] Figure 6.8(a) shows the computation–distortion for several rates. For low- to moderate-rate coding, the distortion as a function of computation becomes very flat. For example, at 0.5 bits/pixel, one can have a 19% complexity reduction while lowering the PSNR by less than 0.1 dB, or have a 30% complexity reduction while lowering the PSNR by less than 0.4 dB. Similar results are shown in Figure 6.8(b) for the *baboon* image.

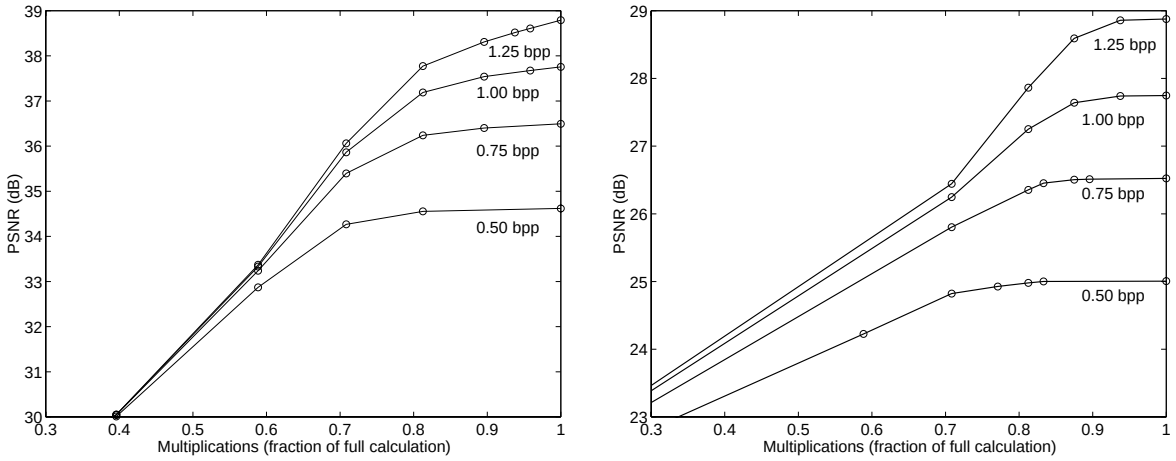
6.3.2.3 Variations and related methods

Not computing certain coefficients and setting them to zero seems a very rudimentary way to produce an approximate DCT algorithm, but it works reasonably well for this application. Because of the run length and entropy coding used in JPEG, even when a high-frequency DCT coefficient has a nonzero quantized value, coding that coefficient (as opposed to rounding it to zero) may not be wise in a rate–distortion sense [156]. The approximate DCT considered here forces longer runs of zeros and hence gets good coding efficiency.

Output pruning is not the optimal way to produce the desired 1-D DCT algorithms, but was done for conceptual transparency and so that the set of algorithms $\mathcal{A}$ could be precisely defined. Lengwehasatit [118] has provided the operation counts for hand-optimized algorithms which assume integer valued inputs and use some bit shifting. These operation counts are given in Table 6.3 and yield the $D(C)$ curves in Figure 6.9. The complexity reductions in Figure 6.9 are even more dramatic than those in Figure 6.8. At 0.5 bits/pixel, 0.1 dB and 0.4 dB PSNR losses occur with 38% and 46% reductions in complexity, respectively.

One approach to improving on the results presented here is to use a *variable complexity algorithm* (VCA). The algorithms we have discussed process each block identically, regardless of how many DCT coefficients of the block are zero. For decoding, blocks with many zero coefficients are clearly easiest to handle, assuming that they can be identified efficiently. The design of such input-dependent algorithms is generally done in an ad hoc fashion. A notable exception is the work of Lengwehasatit and Ortega [119, 120]. They have developed

[12] To download software, follow links from http://www.ijg.org/.
[13] These are "presumably" the best operating points because only a subset of the 2^{64} algorithms in $\mathcal{A}$ were tested.



(a) *Lena* (b) *baboon*

Figure 6.8: Operational $D(C)$ at various bit rates for JPEG encoding with approximate DCT algorithms designed through output pruning.

k	multiplications	additions
1	0	7
2	4	17
3	6	21
4	8	25
5	8	26
6	9	27
7	10	28
8	11	29

Table 6.3: Operation counts to compute the first k coefficients of a length-8 DCT [118].

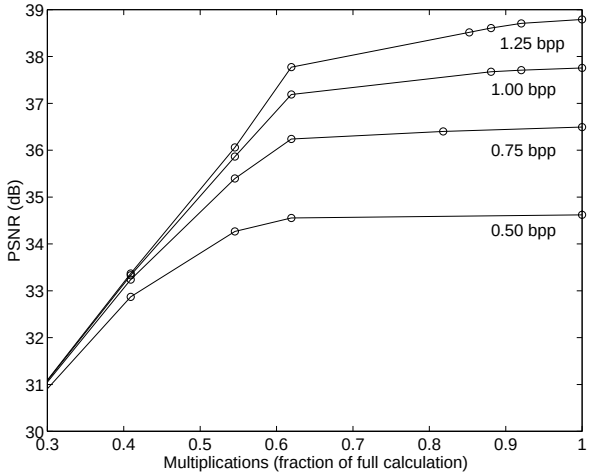


Figure 6.9: Operational $D(C)$ at various bit rates for JPEG encoding of *Lena* with computation counts from Table 6.3.

a decoder which classifies blocks based on their patterns of zero and nonzero coefficients and uses inverse DCT algorithms optimized for each class. The definitions of the classes themselves have been computationally optimized, including the cost of classifying. In a sense, they have developed a compiler which optimizes the implementation of the inverse DCT given the distribution of quantized DCT coefficients. The VCA work has a distinct advantage over the present work: the complexity is reduced with no degradation of rate–distortion performance. On the other hand, the degree of complexity reduction with this approach is limited.

6.3.2.4 Concluding comments

Traditionally, image coding techniques have been compared almost exclusively on the basis of their rate vs. distortion performances, with consideration of computational complexity, if any, reduced to binary determinations: "too complex" or "not too complex." Through a simple example, we have gotten a glimpse at what happens when a measure of computational complexity is thrown into the mix.

A precise framework was used to define the optimal trade-off between complexity and distortion. However, it should be noted that the ability to draw precise conclusions is dependent on the set of algorithms $\mathcal{A}$ and the computational complexity metric c. With $\mathcal{A}$ a relatively small discrete set, we were able to (essentially) exactly characterize $D_R(C)$ by using an (effectively) exhaustive search. If we were to enlarge $\mathcal{A}$, then we would have only an upper bound for $D_R(C)$. In principle, to find the best JPEG-compatible representation of an image it may be necessary to exhaustively search all syntactically valid bit streams of a given length.

6.3.3 Pruned Tree-Structured Vector Quantization

The final application of computation–rate–distortion optimization is to entropy pruned tree-structured vector quantization (EPTSVQ) [29]. For VQ with unconstrained codebook design and full-search encoding, the rate determines the codebook size and hence determines the computational complexity of encoding. Therefore to have variable computational load requires varying the vector dimension.[14] In contrast to full-search VQ, the complexity of EPTSVQ is not determinable from the output rate. Thus we can attempt to optimize simultaneously for rate, distortion, and computation.

In tree-structured VQ (TSVQ) [23], a binary tree is constructed with a codeword at each node. In the encoding process, one starts at the root of the tree and iteratively traverses the branch to the child node whose codeword is closest to the source vector. Coding terminates when a leaf node is reached. In the simplest form of TSVQ, the route from root to leaf is the channel codeword (say, each left child selected gives a zero and each right child gives a one). The rate can be lowered by using an entropy code on the leaf nodes. This is called entropy coded TSVQ. EPTSVQ is a design method for entropy coded TSVQ. The idea is to start with a deep TSVQ tree and prune it to minimize $J = R + \lambda D$, where R is the entropy coded rate, D is the distortion, and λ controls the trade-off between R and D. (Optimality is within the set of all subtrees of the original tree.) The pruning uses the greedy algorithm of Breiman, Freidman, Olshen, and Stone (BFOS) [19] which in general is not optimal, but is optimal in this case because the objective functions are monotonic, affine tree functionals [29].

Assuming a pair of distance determinations and a comparison takes one unit of computation, the average computational complexity of TSVQ encoding is the weighted average of the depths of the leaf nodes.

[14] The situation is slightly more complicated with entropy coding, but the fact remains that for a fixed output entropy it is difficult to meaningfully vary the size of the codebook.

Because the average tree depth is a monotonic, affine tree functional, $J' = R + \lambda D + \mu C$ can also be minimized with the BFOS algorithm. Thus, we have an efficient method to find the lower convex hull of computation–rate–distortion points in entropy coded TSVQ. However, because of the close coupling between rate and computation, the optimal pruning does not depend much on the relative weighting of rate and computation. Experiments for image coding at 1 bit per pixel show that computation can be reduced by about 5% while incurring an increase in distortion of about 5%.

6.4 Conclusions

An abstract framework for joint optimization of computational complexity and performance was introduced in this chapter. The bulk of the chapter was devoted to two analyses of transform coding. The first addressed the relative merits of the KLT and DCT in coding autoregressive sources. The second explored the possibility of using simple approximate DCT algorithms in JPEG encoding.

The results in this chapter are all operational, meaning that an operating point is known to be achievable when an explicit algorithm is exhibited. This is a significant limitation. Nevertheless, it has been demonstrated that, in certain cases, analytical techniques can be used to find algorithms that provide a good trade-off between performance and complexity.

Appendix

6.A Application to Rootfinding

The computation–distortion framework is not limited to source coding. An application to rootfinding is summarized in this appendix. Because of the tangential relation with source coding, details are omitted.

Newton's method is an iterative technique for finding a solution to an equation of the form $f(x) = 0$. From a given iterate x_n, the next iterate is computed by finding the root of Taylor's linear approximation to f at x_n, yielding

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{6.8}$$

The convergence properties, including the set of initial guesses x_0 for which the iteration converges, depends on the function f. A common use of Newton's method is for computing square roots. To compute the square root of $a > 0$, use $f(x) = x^2 - a$. Then (6.8) becomes

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right), \tag{6.9}$$

and the iteration converges for all $x_0 \neq 0$ [8].

Newton's method is used in the implementation of hardware square root instructions [95]. In this case, one can use (6.9) directly or use an alternative form which avoids division:

$$x_{n+1} = x_n\left(1 + \frac{1}{2}\left(1 - \frac{1}{a}x_n^2\right)\right).$$

Note that adding one and dividing by two can be simply implemented with bit shifts. The division by a becomes a multiplication if $1/\sqrt{\cdot}$ is the function to be calculated.

Consider the computation of (6.9) in binary arithmetic. The division by two is a trivial change in the exponent, which does not affect the accuracy of the computation. If the division and addition have relative errors of δ_1 and δ_2, respectively, then the iteration becomes

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}(1+\delta_1)\right)(1+\delta_2). \tag{6.10}$$

Using a computational model with a b_1-bit mantissa for the division and a b_2-bit mantissa for the addition, it is reasonable to model δ_1 and δ_2 as independent random variables, with δ_i uniformly distributed on $[-2^{-b_i}, 2^{-b_i}]$, $i = 1, 2$. Now b_1 and b_2 determine in some fashion the cost of one step of the iteration, and the computation–distortion framework can be used to optimize their selection.

The optimization was undertaken for the very simple cost metric $b_1 + b_2$. The result is qualitatively simple. With a reasonable, not too small, upper bound on $b_1 + b_2$, it is optimal to have $b_1 \approx b_2$. For small values of the maximum complexity, the addition is favored over the division ($b_1 \ll b_2$). This is because its relative error multiplies a larger factor. A similar result follows from optimizing the split of computational resources between two iterations: With a small limit on the total computation, one high-precision iteration can be more accurate than two low-precision iterations, but it depends on the accuracy of the initial guess. In practice, initial guesses come from low-resolution look up tables.

Chapter 7

Conclusions

THIS THESIS has touched on a range of issues in source and source–channel coding with transforms. The key results are summarized here, followed by a brief discussion of open issues.

Frames for signal representation At all stages, most source coding systems use a basis representation of the information signal. If not a basis, they use less than a basis; *i.e.*, the signal is approximated by its projection on to a proper subspace of the original space. In Chapter 2, signal representations with respect to *frames*, overcomplete sets of vectors, are analyzed. A linear representation with respect to a frame is obtained by forming inner products between each frame element and the signal vector. This representation has greater robustness than a basis representation. For a frame with redundancy r, meaning the number of elements in the frame is r times the dimension of the space, the frame representation leads to $O(1/r)$ mean-squared error (MSE) when the expansion coefficients are subjected to a general, random, white additive noise (see Proposition 2.2). When the noise is bounded, as in the case of quantization, the representation leads to $O(1/r^2)$ MSE (see Proposition 2.5, Conjecture 2.6, and Theorem 2.11). It is important to note that this improved performance with respect to increasing r is obtained only when the reconstruction procedure utilizes the hard bounds on the noise. A reconstruction satisfying all these hard bounds is called *consistent*. The order improvement of consistent reconstruction suggests that in this case "hard" information is much more valuable than "soft" information.

Regardless of improvements resulting from consistent reconstruction, a frame expansion as described above does not generally give good performance in source coding. The reason is simply that the bits used in representing the extra transform coefficients have more impact if they are used to more finely quantize an expansion in a basis subset of the frame. An alternative use of a frame is to represent a signal by a linear combination of a small number of frame elements. While the previous method took advantage of the redundancy of the frame to get extra measurements of the data, this method relies on the signal being close to one of the many subspaces spanned by some small subset of the frame. Finding an optimal representation with a bound on the size of the subset of the frame is not computationally feasible. Thus, a suboptimal, greedy technique called matching pursuit is often used. It is shown in Chapter 2 that when the coefficients in matching pursuit are quantized—as would be the case in any compression application—a linear reconstruction does not meet the hard bounds on the quantization error. Nonlinear consistent reconstruction algorithms are given, and the potentially large reduction in distortion from this improved reconstruction is demonstrated experimentally. The

compression performance of quantized matching pursuit is evaluated with random frames and certain structured frames. With consistent reconstruction and effective lossless coding, the low-rate performance is very good.

Basis adaptation for source coding In a variety of situations, it is useful to use a Karhunen–Loève basis, in which the expansion coefficients of a random signal are uncorrelated. In transform coding of a Gaussian signal, a KL basis is optimal either when the quantizer is optimal (Lloyd–Max) or the rate is high. The KL basis is problematic, however, in that it depends on the source probability density, which is generally unknown and may be time-varying. For this reason, it is necessary to use an approximation of the KL basis; this approximation may be adaptive. Using an adaptive basis for transform coding leads immediately to two questions: How should the transform be varied, and how does this affect performance?

The transform adaptation could follow either of two limiting strategies or any number of intermediate strategies. One extreme is to form a block of source vectors, compute a transform to use for this block, communicate the choice of transform (side information) to the decoder, and then use the selected transform to compress the block. This is a *forward adaptive* strategy. The other extreme case, a purely *backward adaptive* strategy, is proposed and analyzed in Chapter 3. In this approach, the encoder computes a transform based only on the data already available at the decoder, eliminating the transmission of side information. The adaptation is then limited to be strictly causal and driven by quantized data. For a source that produces independent, identically distributed vectors, there is a well-defined optimal transform. It is proven in Chapter 3 that the performance gap between using the optimal method and a backward adaptive method is asymptotically negligible (see Theorem 3.2). Thus, a *universal* transform coding system is obtained.

The results of Chapter 3 are for an idealized situation where an exact eigendecomposition is periodically computed. The computational difficulty of eigendecomposition motivates a search for alternative methods for computing transform updates. As detailed in Chapter 4, an analogy can be drawn between filter adaptation in FIR Wiener filtering and transform adaptation in transform coding. This suggests the transplantation of adaptive filtering techniques to adaptive transform coding. Random search and stochastic gradient descent are applied to transform adaptation in Chapter 4. In gradient descent, any number of cost functions can lead to convergent techniques with nontrivially different behavior. Two cost functions are proposed, and detailed convergence analyses are given for both (see Theorems 4.2–4.4).

Source–channel coding for erasure channels The dominant paradigm of communication includes the separation of source and channel coding. This works well when the decoded information is independent of the particular stochastic realization of the channel. In such cases, the channel coding—possibly including a retransmission protocol in addition to forward error- or erasure-correction—should provide a noiseless channel. For an ergodic channel, as block sizes grow without bound, restricting the source and channel coding to be separate is not a significant limitation because the channel impairment is predictable through the law of large numbers. However, for short block sizes, restricting to this framework may greatly impact performance. Block size limits are particularly important for interactive communication and transmission of small data sets.

A block channel code for an erasure channel is designed to deliver a certain number of bits as long as the number of erasures does not exceed a fixed maximum, namely the minimum distance of the code. In that case, the transmission is considered successful. No attention is given to the number of correctly decoded bits in an unsuccessful transmission. These conditions lead naturally to a "cliff effect" whereby the number

of correctly delivered bits is constant when the number of erasures is small and drops precipitously when the number of erasures exceeds the minimum distance of the code. The cliff effect is not a problem if the probability of exceeding the minimum distance of the code is very small.[1] However, with small block sizes, the relative deviation from the mean number of erasures is large, so the minimum distance of the code is often exceeded. Explicit consideration of each possible number of erasures gives more control over the performance of the system and may improve the average performance. This gives a direct analogy to (generalized) multiple description coding. In multiple description coding, a set of descriptions are produced and reconstructions are computed from each nonempty subset of descriptions. Such a reconstruction can arise in a multichannel communication system where a receiver gets a subset of the channels (see Figure 5.1) or at the end of an erasure channel where a subset of symbols have been erased. With a fixed total rate, to minimize average distortion or satisfy other criteria one can trade off the various distortions.

Two methods for multiple description coding (MDC) are introduced in Chapter 5. The redundancy properties produced by each method differ from those of block channel coding. The outputs of a block channel code have a strict deterministic relationship which enables decoding when there are a few erasures, but extra received symbols contain no additional information. The first MDC method is a *statistical* block channel code. It gives descriptions that are statistically correlated; thus, when a subset is lost, its elements can be estimated from the remaining descriptions. This provides a mechanism for adding a small amount of redundancy without having large block lengths. The second MDC method uses a quantized frame expansion to produce descriptions. These descriptions have an approximate linear dependence, yet each provides independent information about the source, regardless of how many other descriptions are received. These multiple description techniques give promising results when applied to audio and image coding.

Complexity reduction and computational optimization The first MDC method of Chapter 5 uses transforms from a simple discrete, quasilinear class. These transforms can approximately match the coding gain of the Karhunen–Loève transform (KLT), even though they operate on a discretized version of the source. In addition, they can be used to manipulate probability densities of transform coefficients in ways that continuous orthogonal transforms can not. One application of such manipulation is to reduce the complexity of the entropy coding stage of a transform coding system. Other examples of complexity reduction are also discussed in this thesis.

A framework for computational optimization is introduced in Chapter 6. The framework provides a vocabulary for joint optimization of computational complexity and compression performance. For example, a comparison between the KLT and discrete cosine transform (DCT) using this formalism shows that the DCT should be used for coding a first-order Gauss–Markov source with high correlation. Another example is to compare a set of approximate JPEG encoding algorithms. It is demonstrated that the complexity of the transform computation can be reduced by up to 30% without much performance degradation.

➢ ➢ ➣ ➣

An attempt to list the open problems left in the wake of this thesis would be futile; there are obviously many. A few of the subjects not covered are notable because of their proximity to the main results.

[1] The minimum distance must be balanced with the choice of code rate.

The properties of linear representations with respect to frames (see Chapters 2 and 5) depend on the frame and on the suitability of the frame for the signal. Nevertheless, rather than confronting the problem of designing application-specific frames, general methods for frame design have been employed. To some extent, this approach impedes specific results, but it also leads to some results that are independent of the frame. In particular, several distortion computations depend only on the frame being a normalized tight frame. Also, dictionary designs for matching pursuit have not been optimized. The experimental results are for ad hoc or random dictionaries, and they would be improved by dictionary optimization.

The techniques introduced and analyzed in this thesis have a decided bias, not just to transform coding, but to the transform itself. This was partially the result of a conscious effort to maintain focus. Also, the use of unbounded uniform scalar quantization simplified both notation and analysis, but this can generally be improved upon. In particular, universal coding and multiple description coding can be addressed by reconsidering the quantization and entropy coding blocks in a transform coding system, or by abandoning the transform altogether.

Publications and Patents

Journal papers and manuscripts

1. V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantized Overcomplete Expansions in $\mathbb{R}^N$: Analysis, Synthesis and Algorithms," *IEEE Trans. Information Th.*, vol. 44, no. 1, pp. 16–31, Jan. 1998.

2. V. K. Goyal, J. Zhuang, and M. Vetterli, "On-line Algorithms for Universal Transform Coding," submitted to *IEEE Trans. Information Th.*, April 1998.

3. S. Rangan and V. K. Goyal, "Recursive Consistent Estimation with Bounded Noise," submitted to *IEEE Trans. Information Th.*, July 1998.

4. R. Arean, J. Kovačević, and V. K. Goyal, "Multiple Description Source-Channel Coding of Audio," submitted to *IEEE Trans. Speech Audio Proc.*, Aug. 1998.

Conference, symposium, and workshop papers

1. V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantization of Overcomplete Expansions," Proc. IEEE Data Compression Conf. 1995 (Snowbird, UT, March 28–30), pp. 13–22.

2. V. K. Goyal and M. Vetterli, "Consistency in Quantized Matching Pursuit," Proc. IEEE Int. Conf. Acoustics, Speech, & Sig. Proc. 1996 (Atlanta, GA, May 7–10), vol. 3, pp. 1787–1790.

3. V. K. Goyal, M. Vetterli, and N. T. Thao, "Efficient Representations with Quantized Matching Pursuit," Proc. Int. Conf. Analysis & Opt. of Systems 1996 (Paris, France, June 26–28), pp. 305–311.

4. V. K. Goyal, J. Zhuang, M. Vetterli, and C. Chan, "Transform Coding Using Adaptive Bases and Quantization," Proc. IEEE Int. Conf. Image Proc. 1996 (Lausanne, Switzerland, Sept. 16–19), vol. II, pp. 365–368.

5. V. K. Goyal and M. Vetterli, "Dependent Coding in Quantized Matching Pursuit," Proc. IS&T/SPIE Visual Comm. & Image Proc. 1997 (San Jose, CA, Feb. 12–14), vol. 3024, pt. 1, pp. 2–12.

6. V. K. Goyal, J. Zhuang, and M. Vetterli, "Universal Transform Coding Based On Backward Adaptation," Proc. IEEE Data Compression Conf. 1997 (Snowbird, UT, March 25–27), pp. 231–240.

7. V. K. Goyal and M. Vetterli, "Computation–Distortion Characteristics of Block Transform Coding," Proc. IEEE Int. Conf. Acoustics, Speech, & Sig. Proc. 1997 (Munich, Germany, April 21–24), vol. 4, pp. 2729–2732.

8. V. K. Goyal and M. Vetterli, "Computation–Distortion Characteristics of JPEG Encoding and Decoding," Proc. 31st Asilomar Conf. on Signals, Systems, & Computers 1997 (Pacific Grove, CA, Nov. 2–5), vol. 1, pp. 229–233.

9. V. K. Goyal and J. Kovačević, "Optimal Multiple Description Transform Coding of Gaussian Vectors," Proc. IEEE Data Compression Conf. 1998 (Snowbird, UT, March 30–April 1), pp. 388–397.

10. V. K. Goyal and M. Vetterli, "Block Transform Adaptation By Stochastic Gradient Descent," Proc. IEEE Digital Signal Proc. Workshop 1998 (Bryce Canyon, UT, Aug. 9–12).

11. V. K. Goyal, J. Kovačević, and M. Vetterli, "Multiple Description Transform Coding: Robustness to Erasures using Tight Frame Expansions," Proc. IEEE Int. Symp. Inform. Th. 1998 (Cambridge, MA, Aug. 16–21), p. 408.

12. J. Kovačević and V. K. Goyal, "Multiple Descriptions: Source–Channel Coding Methods for Communications," Proc. 10th Tyrrhenian Int. Workshop on Dig. Comm.: Multimedia Comm. (Ischia, Italy, Sep. 16–18, 1998).

13. V. K. Goyal, J. Kovačević, and M. Vetterli, "Quantized Frame Expansions as Source–Channel Codes for Erasure Channels," Proc. Wavelets & Applications Workshop 1998 (Ascona, Switzerland, Sep. 28–Oct. 2).

14. V. K. Goyal, J. Kovačević, R. Arean, and M. Vetterli, "Multiple Description Transform Coding of Images," Proc. IEEE Int. Conf. Image Proc. 1998 (Chicago, IL, Oct. 4–7).

15. V. K. Goyal and M. Vetterli, "Manipulating Rates, Complexity, and Error-Resilience with Discrete Transforms," Proc. 32nd Asilomar Conf. on Signals, Systems, & Computers 1998 (Pacific Grove, CA, Nov. 1–4).

16. V. K. Goyal, J. Kovačević and M. Vetterli, "Quantized Frame Expansions as Source–Channel Codes for Erasure Channels," Proc. IEEE Data Compression Conf. 1999 (Snowbird, UT, March 29–31).

Technical memoranda, solution manual, and other presented work

1. J. Gifford, V. K. Goyal, M. R. Grenier, L. K. Gross, and S. J. Henley, "The Use of Bootstrapping in Principal Component Analysis." Presented at the Special Session on Undergraduate Research at the 1991 Annual Meeting of the Mathematical Association of America.

2. V. K. Goyal, "Quantized Overcomplete Expansions: Analysis, Synthesis and Algorithms," Univ. of California, Berkeley, EECS Dept. Tech. Memo. No. UCB/ERL M95/57, July 1995.

3. S. G. Chang, M. M. Goodwin, V. K. Goyal, and T. Kalker, "Solution Manual for *Wavelets and Subband Coding* by Martin Vetterli and Jelena Kovačević," Prentice-Hall, Englewood Cliffs, NJ, 1995.

4. V. K. Goyal, J. Kovačević, and M. Vetterli, "Multiple Description Transform Coding: Robustness to Erasures Using Tight Frame Expansions," Bell Labs Tech. Memo. No. BL0112170-971124-24TM, Nov. 1997.

5. V. K. Goyal and J. Kovačević, "Optimal Multiple-Description Transform Coding of Gaussian Vectors," Bell Labs Tech. Memo. No. BL0112170-971124-25TM, Nov. 1997.

6. J. Kovačević, V. K. Goyal, and R. Arean, "Multiple Description Source–Channel Coding of Audio," Bell Labs Tech. Memo. No. BL0112170-980818-19TM, Aug. 1998.

7. J. Kovačević and V. K. Goyal, "Multiple Descriptions: Source–Channel Coding Methods for Communications," Bell Labs Tech. Memo. No. BL0112170-980918-30TM, Sep. 1998.

Patents

1. V. K. Goyal and J. Kovačević, "Multiple Description Transform Coding Using Optimal Transforms of Arbitrary Dimension," U.S. Patent filed February 25, 1998.

2. V. K. Goyal, J. Kovačević and M. Vetterli, "Multiple Description Transform Coding of Images Using Optimal Transforms of Arbitrary Dimension," U.S. Patent filed September 30, 1998.

3. R. Arean, J. Kovačević, V. K. Goyal and M. Vetterli, "Multiple Description Transform Coding of Audio Using Optimal Transforms of Arbitrary Dimension," U.S. Patent filed November 12, 1998.

4. V. K. Goyal, "Method and Apparatus for Reduced Complexity Entropy Coding," U.S. Patent filed January 20, 1999.

Bibliography

[1] J. Adler, B. D. Rao, and K. Kreutz-Delgado. Comparison of basis selection methods. In *Proc. Asilomar Conf. on Sig., Sys. & Computers*, volume 1, pages 252–257, Pacific Grove, CA, November 1996. 76

[2] R. Ahlswede. The rate distortion region for multiple descriptions without excess rate. *IEEE Trans. Inform. Th.*, IT-31(6):721–726, November 1985. 103, 108

[3] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Trans. Comp.*, C-23(1):90–93, January 1974. 191

[4] T. W. Anderson, I. Olkin, and L. G. Underhill. Generation of random orthogonal matrices. *SIAM J. Sci. Stat. Comput.*, 8(4):625–629, July 1987. 79

[5] T. M. Apostol. *Mathematical Analysis*. Addison-Wesley, second edition, 1974. 64

[6] R. Arean. Multiple description transform coding: Application to the PAC audio coder. Professional Thesis Report filed with Institut Eurécom, Sophia Antipolis, France, July 1998. 99, 147, 147, 148

[7] R. Arean, J. Kovačević, and V. K Goyal. Multiple description source-channel coding of audio. *IEEE Trans. Speech Audio Proc.*, August 1998. Submitted. 99, 102, 148

[8] K. E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons, New York, second edition, 1989. 202

[9] J.-C. Batllo and V. A. Vaishampayan. Asymptotic performance of multiple description transform codes. *IEEE Trans. Inform. Th.*, 43(2):703–707, March 1997. 114

[10] K. L. Bell, Y. Steinberg, Y. Ephraim, and H. L. Van Trees. Extended Ziv-Zakai lower bound for vector parameter estimation. *IEEE Trans. Inform. Th.*, 43(2):624–637, March 1997. 49

[11] W. R. Bennett. Spectra of quantized signals. *Bell Syst. Tech. J.*, 27(3):446–472, July 1948. 8, 10

[12] F. Bergeaud and S. Mallat. Matching pursuit of images. In *Proc. IEEE Int. Conf. Image Proc.*, volume I, pages 53–56, Washington, DC, October 1995. 20, 41

[13] T. Berger. *Rate Distortion Theory*. Prentice-Hall, Englewood Cliffs, NJ, 1971. 4, 5, 14, 110, 184, 186, 186, 191

[14] T. Berger. Optimum quantizers and permutation codes. *IEEE Trans. Inform. Th.*, IT-18(6):759–765, November 1972. 6

[15] T. Berger and Z. Zhang. Minimum breakdown degradation in binary source encoding. *IEEE Trans. Inform. Th.*, IT-29(6):807–814, November 1983. 105, 107

[16] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Trans. Comm.*, 44(10):1261–1271, October 1996. 185

[17] H. S. Black and E. O. Edson. PCM equipment. *Elec. Eng.*, 66:1123–1125, November 1947. 5

[18] R. E. Blahut. *Fast Algorithms for Digital Signal Processing.* Addison-Wesley, 1985. Reprinted with corrections 1987. 188

[19] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, CA, 1984. 185, 200

[20] P. Brodatz. *Textures: A Photographic Album for Artists and Designers.* Dover, New York, 1966. 68, 69

[21] G. Buch, F. Burkert, J. Hagenauer, and B. Kukla. To compress or not to compress? In *IEEE Globecom*, pages 196–203, London, November 1996. 17, 108

[22] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften.* Springer-Verlag, Berlin, Germany, 1997. 188

[23] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel. Speech coding based upon vector quantization. *IEEE Trans. Acoust. Speech Signal Proc.*, ASSP-28:562–574, October 1980. 32, 185, 200

[24] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Computational Harmonic Anal.*, 5(3):332–369, July 1998. 167

[25] G. J. Chaitin. On the difficulty of computations. *IEEE Trans. Inform. Th.*, IT-16(1):5–9, January 1970. 189

[26] G. J. Chaitin. Information-theoretic computational complexity. *IEEE Trans. Inform. Th.*, IT-20(1):10–15, January 1974. 189

[27] L. Cheded and P. A. Payne. The exact impact of amplitude quantization on multi-dimensional, high-order moments estimation. *Signal Proc.*, 39(3):293–315, September 1994. 10, 59, 62, 71, 72

[28] P. A. Chou, M. Effros, and R. M. Gray. A vector quantization approach to universal noiseless coding and quantizaton. *IEEE Trans. Inform. Th.*, 42(4):1109–1138, July 1996. 54

[29] P. A. Chou, T. Lookabaugh, and R. M. Gray. Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Trans. Inform. Th.*, 35(2):299–315, March 1989. 7, 185, 200, 200

[30] P. M. Clarkson. *Optimal and Adaptive Signal Processing.* CRC Press, Boca Raton, FL, 1993. 84, 87, 96

[31] J. H. Conway, R. H. Hardin, and N. J. A. Sloane. Packing lines, planes, etc.: Packings in Grassmannian spaces. *Experimental Mathematics*, 5(2):139–159, 1996. 181

[32] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften.* Springer-Verlag, New York, 1988. 118, 150

[33] T. M. Cover. Personal communication, August 1998. 103

[34] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, 1991. 2, 5, 14, 14, 58, 61, 102, 103, 120, 170, 183, 184, 190

[35] Z. Cvetković. *Overcomplete Expansions for Digital Signal Processing.* PhD thesis, Univ. California, Berkeley, 1995. Available as Univ. California, Berkeley, Electron. Res. Lab. Memo. No. UCB/ERL M95/114, December 1995. 20

[36] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Inform. Th.*, 36:961–1005, September 1990. 20

[37] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992. 20, 20, 23, 23, 24, 24, 49

[38] G. Davis. *Adaptive Nonlinear Approximations*. PhD thesis, New York Univ., September 1994. 29, 29, 29, 30, 30

[39] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency approximations with matching pursuits. Technical Report 657, New York Univ., March 1994. 30

[40] L. D. Davisson. Universal noiseless coding. *IEEE Trans. Inform. Th.*, IT-19(6):783–795, November 1973. 55

[41] S. Deering and R. Hinden. Internet Protocol, version 6 (IPv6) specification. Network Working Group Request for Comments 1883, December 1995. Available on-line at `ftp://ftp.isi.edu/in-notes/rfc1883.txt`. 103

[42] R. D. Dony and S. Haykin. Optimally adaptive transform coding. *IEEE Trans. Image Proc.*, 4(10):1358–1370, October 1995. 55

[43] H. W. Dudley. The vocoder. *Bell Lab. Rec.*, 18:122–126, December 1939. 2

[44] R. J. Duffin and A. C. Schaeffer. A class of nonharmonic Fourier series. *Trans. Amer. Math. Soc.*, 72:341–366, 1952. 20

[45] M. Effros. Fast weighted universal transform coding: Toward optimal, low complexity bases for image compression. In J. A. Storer and M. Cohn, editors, *Proc. IEEE Data Compression Conf.*, pages 211–220, Snowbird, Utah, March 1997. IEEE Comp. Soc. Press. 76

[46] M. Effros and P. A. Chou. Weighted universal transform coding: Universal image compression with the Karhunen-Loève transform. In *Proc. IEEE Int. Conf. Image Proc.*, volume II, pages 61–64, Washington, DC, October 1995. 55, 55, 55, 66, 68, 69, 76

[47] M. Effros and A. Goldsmith. Capacity definitions and coding strategies for general channels with receiver side information. In *Proc. IEEE Int. Symp. Inform. Th.*, page 39, Cambridge, MA, August 1998. 166

[48] A. A. El Gamal and T. M. Cover. Achievable rates for multiple descriptions. *IEEE Trans. Inform. Th.*, IT-28(6):851–857, November 1982. 100, 103, 104, 104

[49] P. Elias. Two famous papers. *IRE Trans. Inform. Th.*, IT-4(3):99, September 1958. 2

[50] W. H. R. Equitz and T. M. Cover. Successive refinement of information. *IEEE Trans. Inform. Th.*, 37(2):269–275, March 1991. 107, 108

[51] European Broadcast Union. CD-sound quality assessment material: Recording for subjective tests. PolyGram Germany, 1997. 141, 142, 142, 142, 142, 142

[52] N. Farvardin. A study of vector quantization for noisy channels. *IEEE Trans. Inform. Th.*, 36(4):799–809, July 1990. 108

[53] N. Farvardin and J. W. Modestino. Optimum quantizer performance for a class of non-Gaussian memoryless sources. *IEEE Trans. Inform. Th.*, IT-30(3):485–497, May 1984. 9

[54] T. J. Ferguson and J. H. Rabinowitz. Self-synchronizing Huffman codes. *IEEE Trans. Inform. Th.*, IT-30(4):687–693, July 1984. 108

[55] G. D. Forney, Jr. *Concatenated Codes*. MIT Press, Cambridge, MA, 1966. 185

[56] G. D. Forney, Jr. Performance and complexity. *IEEE Inform. Th. Soc. Newsletter*, 46(1):3–4+, March 1996. From a 1995 Shannon Lecture. 185, 185

[57] J. Fourier. *Théorie Analytique de la Chaleur*. Didot, Paris, France, 1822. 11

[58] R. G. Gallager. A simple derivation of the coding theorem and some applications. *IEEE Trans. Inform. Th.*, IT-11(1):3–18, January 1965. 185

[59] A. M. Gerrish and P. M. Schultheiss. Information rates of non-Gaussian processes. *IEEE Trans. Inform. Th.*, IT-10(4):265–271, October 1964. 110

[60] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Acad. Pub., Boston, MA, 1992. 7, 8, 76, 185, 186, 191

[61] H. Gish and J. P. Pierce. Asymptotically efficient quantizing. *IEEE Trans. Inform. Th.*, IT-14(5):676–683, September 1968. 8, 9, 10

[62] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, Baltimore, MD, second edition, 1989. 29, 72, 76, 76, 76, 78, 85

[63] M. Goodwin. Matching pursuit with damped sinusoids. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, pages 2037–2040, Munich, Germany, April 1997. 36

[64] M. M. Goodwin. *Adaptive Signal Models: Theory, Algorithms, and Audio Applications*. PhD thesis, Univ. California, Berkeley, 1997. Available as Univ. California, Berkeley, Electron. Res. Lab. Memo. No. UCB/ERL M97/91, December 1997. 36

[65] M. J. Gormish and J. T. Gill. Computation-rate-distortion in transform coders for image compression. In *Proc. SPIE Conf. Image and Video Proc.*, volume 1903, pages 146–152, San Jose, CA, February 1993. 190

[66] V. K Goyal. Quantized overcomplete expansions: Analysis, synthesis and algorithms. Master's thesis, Univ. California, Berkeley, 1995. Available as Univ. California, Berkeley, Electron. Res. Lab. Memo. No. UCB/ERL M95/57, July 1995. 34, 36, 36, 36, 42

[67] V. K Goyal and J. Kovačević. Optimal multiple description transform coding of Gaussian vectors. In J. A. Storer and M. Cohn, editors, *Proc. IEEE Data Compression Conf.*, pages 388–397, Snowbird, Utah, March 1998. IEEE Comp. Soc. Press. 99, 102

[68] V. K Goyal, J. Kovačević, R. Arean, and M. Vetterli. Multiple description transform coding of images. In *Proc. IEEE Int. Conf. Image Proc.*, Chicago, October 1998. 99, 102

[69] V. K Goyal, J. Kovačević, and M. Vetterli. Multiple description transform coding: Robustness to erasures using tight frame expansions. In *Proc. IEEE Int. Symp. Inform. Th.*, page 408, Cambridge, MA, August 1998. 99, 102

[70] V. K Goyal, J. Kovačević, and M. Vetterli. Quantized frame expansions as source-channel codes for erasure channels. In *Proc. Wavelets & Appl. Workshop*, Ascona, Switzerland, September–October 1998. 99, 102

[71] V. K Goyal and M. Vetterli. Consistency in quantized matching pursuit. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 3, pages 1787–1790, Atlanta, GA, May 1996. 19

[72] V. K Goyal and M. Vetterli. Computation-distortion characteristics of block transform coding. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 4, pages 2729–2732, Munich, Germany, April 1997. 183

[73] V. K Goyal and M. Vetterli. Computation-distortion characteristics of JPEG encoding and decoding. In *Proc. 31st Asilomar Conf. Sig., Sys., & Computers*, volume 1, pages 229–233, Pacific Grove, CA, November 1997. 183

[74] V. K Goyal and M. Vetterli. Dependent coding in quantized matching pursuit. In *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, volume 3024, pages 2–14, San Jose, CA, February 1997. 19, 20, 41

[75] V. K Goyal and M. Vetterli. Block transform adaptation by stochastic gradient descent. In *IEEE Dig. Sig. Proc. Workshop*, Bryce Canyon, UT, August 1998. 75

[76] V. K Goyal and M. Vetterli. Manipulating rates, complexity, and error-resilience with discrete transforms. In *Proc. 32nd Asilomar Conf. Sig., Sys., & Computers*, volume 1, Pacific Grove, CA, November 1998. 99

[77] V. K Goyal, M. Vetterli, and N. T. Thao. Quantization of overcomplete expansions. In J. A. Storer and M. Cohn, editors, *Proc. IEEE Data Compression Conf.*, pages 13–22, Snowbird, Utah, March 1995. IEEE Comp. Soc. Press. 19, 42

[78] V. K Goyal, M. Vetterli, and N. T. Thao. Efficient representations with quantized matching pursuit. In *Proc. 12th Int. Conf. Anal. & Opt. of Sys.: Images, Wavelets & PDE's*, pages 305–311, Paris, France, June 1996. Springer-Verlag. 19

[79] V. K Goyal, M. Vetterli, and N. T. Thao. Quantized overcomplete expansions in $\mathbb{R}^N$: Analysis, synthesis, and algorithms. *IEEE Trans. Inform. Th.*, 44(1):16–31, January 1998. 19, 27

[80] V. K Goyal, J. Zhuang, and M. Vetterli. Universal transform coding based on backward adaptation. In J. A. Storer and M. Cohn, editors, *Proc. IEEE Data Compression Conf.*, pages 231–240, Snowbird, Utah, March 1997. IEEE Comp. Soc. Press. 54, 66

[81] V. K Goyal, J. Zhuang, and M. Vetterli. On-line algorithms for universal transform coding. *IEEE Trans. Inform. Th.*, April 1998. Submitted. 54, 59

[82] V. K Goyal, J. Zhuang, M. Vetterli, and C. Chan. Transform coding using adaptive bases and quantization. In *Proc. IEEE Int. Conf. Image Proc.*, volume II, pages 365–368, Lausanne, Switzerland, September 1996. 54, 54, 67, 71

[83] R. M. Gray. On the asymptotic eigenvalue distribution of Toeplitz matrices. *IEEE Trans. Inform. Th.*, IT-18(6):725–730, November 1972. 17, 191

[84] R. M. Gray. Quantization noise spectra. *IEEE Trans. Inform. Th.*, 36(6):1220–1244, November 1990. 10, 48

[85] R. M. Gray and T. G. Stockham, Jr. Dithered quantizers. *IEEE Trans. Inform. Th.*, 39(3):805–812, May 1993. 11, 48, 60

[86] R. M. Gray and A. D. Wyner. Source coding for a simple network. *Bell Syst. Tech. J.*, 53(9):1681–1721, November 1974. vii, 107, 108

[87] U. Grenander and G. Szegö. *Toeplitz Forms and Their Applications*. Univ. California Press, Berkeley, CA, 1958. 17, 191

[88] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford Univ. Press, second edition, 1992. 73, 102

[89] J. Hagenauer. Source-controlled channel decoding. *IEEE Trans. Comm.*, 43(9):2449–2457, September 1995. 17, 108

[90] R. H. Hardin, N. J. A. Sloane, and W. D. Smith. Library of best ways known to us to pack n points on sphere so that minimum separation is maximized. URL: `http://www.research.att.com/~njas/packings`. 27, 36, 38, 39

[91] R. V. L. Hartley. Transmission of information. *Bell Syst. Tech. J.*, 7:535–563, July 1928. 3

[92] S. S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Upper Saddle River, NJ, third edition, 1996. 49, 50, 51

[93] M. T. Heideman. *Multiplicative Complexity, Convolution, and the DFT*. Springer-Verlag, New York, 1988. 188

[94] C. Heil and D. Walnut. Continuous and discrete wavelet transforms. *SIAM Rev.*, 31:628–666, 1989. 20

[95] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufman, San Mateo, CA, 1990. Appendix A contributed by D. Goldberg. 188, 188, 202

[96] B. Hochwald. Tradeoff between source and channel coding on a Gaussian channel. *IEEE Trans. Inform. Th.*, 44(7):3044–3055, November 1998. 108

[97] B. Hochwald and K. Zeger. Tradeoff between source and channel coding. *IEEE Trans. Inform. Th.*, 43(5):1412–1424, September 1997. 108

[98] J. Hong. *Discrete Fourier, Hartley, and Cosine Transforms in Signal Processing*. PhD thesis, Columbia Univ., 1993. 167

[99] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge Univ. Press, 1985. Reprinted with corrections 1987. 76, 78, 120, 176

[100] J. J. Y. Huang and P. M. Schultheiss. Block quantization of correlated Gaussian random variables. *IEEE Trans. Comm.*, 11:289–296, September 1963. 2, 12, 12

[101] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is NP-complete. *Inform. Proc. Letters*, 5(1):15–17, May 1976. 185

[102] A. Ingle and V. A. Vaishampayan. DPCM system design for diversity systems with applications to packetized speech. *IEEE Trans. Speech Audio Proc.*, 3(1):48–57, January 1995. 102, 114

[103] F. Jelinek and K. S. Scheider. On variable-length-to-block coding. *IEEE Trans. Inform. Th.*, IT-18(6):765–774, November 1972. 108

[104] C. R. Johnson, Jr. *Lectures on Adaptive Parameter Estimation*. Prentice-Hall, Englewood Cliffs, NJ, 1988. 89

[105] J. D. Johnston and A. J. Ferreira. Sum-difference stereo transform coding. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 2, pages 569–572, San Francisco, March 1992. 15

[106] D. Jonsson. Some limit theorems for the eigenvalues of a sample covariance matrix. *J. Multivariate Anal.*, 12(1):1–38, March 1982. 163

[107] T. Kalker and M. Vetterli. Projection methods in motion estimation and compensation. In *Proc. IS & T/SPIE*, volume 2419, pages 164–175, San Jose, CA, February 1995. 20, 30, 31

[108] D. E. Knuth. *The Art of Computer Programming. Volume 3: Sorting and Searching*. Addison-Wesley, second edition, 1998. 189

[109] A. N. Kolmogorov. Evaluation of minimal number of elements of ϵ-nets in different functional classes and their application to the problem of representation of functions of several variables by superposition of functions of a smaller number of variables. *Usp. Mat. Nauk*, 10(1):192–194, 1955. In Russian. 3

[110] A. N. Kolmogorov. Three approaches for defining the concept of information quantity. *Information Transmission*, 1:3–11, 1965. 189

[111] A. N. Kolmogorov. Logical basis for information theory and probability theory. *IEEE Trans. Inform. Th.*, IT-14(5):662–664, September 1968. 189

[112] T. W. Körner. *Fourier Analysis*. Cambridge Univ. Press, New York, 1988. Paperback edition with corrections, 1989. 11

[113] J. Kovačević and V. K Goyal. Multiple descriptions: Source-channel coding methods for communications. In *Proc. Int. Workshop on Multimedia Comm.*, Ischia, Italy, September 1998. 99

[114] H. P. Kramer and M. V. Mathews. A linear coding for transmitting a set of correlated signals. *IRE Trans. Inform. Th.*, 23(3):41–46, September 1956. 1, 11

[115] E. A. Lee and D. G. Messerschmitt. *Digital Communication*. Kluwer Acad. Pub., Boston, MA, second edition, 1994. 5

[116] D. LeGall. MPEG: A video compression standard for multimedia applications. *Comm. ACM*, 34(4):46–58, April 1991. 16, 166

[117] *Lena*. Photograph of Lena Sjööblom from Playboy Magazine, November 1972, scanned and modestly cropped at the University of Southern California [138]. Used here simply because it is familiar to image processing researchers. 68, 136, 196

[118] K. Lengwehasatit. Personal communication, October 1997. viii, 198, 199

[119] K. Lengwehasatit and A. Ortega. DCT computation with minimal average number of operations. In *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, volume 3024, pages 71–82, San Jose, CA, February 1997. 174, 198

[120] K. Lengwehasatit and A. Ortega. Distortion/decoding time tradeoffs in software DCT-based image coding. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 4, pages 2725–2728, Munich, Germany, April 1997. 174, 198

[121] S. K. Leung-Yan-Cheong and T. M. Cover. Some equivalences between Shannon entropy and Kolmogorov complexity. *IEEE Trans. Inform. Th.*, IT-24(3):331–338, May 1978. 189

[122] J. Lin and J. A. Storer. Improving search for tree-structured vector quantization. In J. A. Storer and M. Cohn, editors, *Proc. IEEE Data Compression Conf.*, pages 339–348, Snowbird, Utah, March 1992. IEEE Comp. Soc. Press. 186

[123] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1983. 5

[124] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Comm.*, COM-28(1):84–95, January 1980. 7

[125] T. Linder and R. Zamir. On the asymptotic tightness of the shannon lower bound. *IEEE Trans. Inform. Th.*, 40(6):2026–2031, November 1994. 112

[126] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy. Quantization and dither: A theoretical survey. *J. Audio Eng. Soc.*, 40(5):355–375, May 1992. 11, 48, 60

[127] L. Ljung. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, MA, 1983. 49, 50

[128] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Th.*, IT-28(2):129–137, March 1982. Originally an unpublished Bell Telephone Laboratories tech. memo., 1957. 6

[129] T. Luczak and W. Szpankowski. A suboptimal lossy data compression based on approximate string matching. *IEEE Trans. Inform. Th.*, 43(5):1439–1451, September 1997. 55

[130] S. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. *Proc. IEEE*, 73(11):1551–1588, November 1985. 185

[131] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Proc.*, 41(12):3397–3415, December 1993. 18, 20, 28, 30, 30, 30

[132] V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Math. USSR–Sbornik*, 1(4):457–483, 1967. 163

[133] J. Max. Quantizing for minimum distortion. *IRE Trans. Inform. Th.*, IT-6(1):7–12, March 1960. 6

[134] J. C. Maxted and J. P. Robinson. Error recovery for variable length codes. *IEEE Trans. Inform. Th.*, IT-31(6):794–801, November 1985. See also [136]. 108

[135] N. Moayeri and D. L. Neuhoff. Time-memory tradeoffs in vector quantizer codebook searching based on decision trees. *IEEE Trans. Speech Audio Proc.*, 2(4):490–506, October 1994. 186, 186

[136] M. E. Monaco and J. M. Lawler. Corrections and additions to "Error recovery for variable length codes". *IEEE Trans. Inform. Th.*, IT-33(3):454–456, May 1987. 217

[137] N. J. Munch. Noise reduction in tight Weyl-Heisenberg frames. *IEEE Trans. Inform. Th.*, 38(2):608–616, March 1992. 20, 20

[138] D. C. Munson, Jr. A note on Lena. *IEEE Trans. Image Proc.*, 5(1):3, January 1996. 216

[139] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Computing*, 24(2):227–234, April 1995. 29, 29

[140] R. Neff and A. Zakhor. Very low bit-rate video coding based on matching pursuits. *IEEE Trans. Circuits Syst. Video Technol.*, 7(1):158–171, February 1997. 20, 31

[141] R. Neff, A. Zakhor, and M. Vetterli. Very low bit rate video coding using matching pursuits. In *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, volume 2308, pages 47–60, Chicago, September 1994. 20, 31

[142] D. L. Neuhoff, R. M. Gray, and L. D. Davisson. Fixed rate universal block source coding with a fidelity criterion. *IEEE Trans. Inform. Th.*, IT-21(5):511–523, September 1975. 54

[143] D. L. Neuhoff and D. H. Lee. On the performance of tree-structured vector quantization. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 4, pages 2277–2280, Toronto, Canada, April 1991. 186

[144] P. G. Neumann. Efficient error-limiting variable-length codes. *IEEE Trans. Inform. Th.*, IT-8(4):292–304, July 1962. 108

[145] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989. 11

[146] M. T. Orchard, Y. Wang, V. Vaishampayan, and A. R. Reibman. Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms. In *Proc. IEEE Int. Conf. Image Proc.*, volume I, pages 608–611, Santa Barbara, CA, October 1997. 102, 116, 117, 117, 118, 119, 122, 124, 124, 134, 135, 136, 164

[147] A. Ortega. *Optimization Techniques for Adaptive Quantization of Image and Video Under Delay Constraints*. PhD thesis, Columbia Univ., 1994. 91

[148] A. Ortega and M. Vetterli. Adaptive scalar quantization without side information. *IEEE Trans. Image Proc.*, 6(5):665–676, May 1997. 91

[149] L. Ozarow. On a source-coding problem with two channels and three receivers. *Bell Syst. Tech. J.*, 59(10):1909–1921, December 1980. 103, 108

[150] E. W. Packel, J. F. Traub, and H. Woźniakowski. Measures of uncertainty and information in computation. *Informational Sciences*, 65(3):253–273, November 1992. 189

[151] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1965. 60

[152] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, third edition, 1994. 44, 44

[153] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993. 15

[154] J. R. Pierce. The early days of information theory. *IEEE Trans. Inform. Th.*, IT-19(1):3–8, January 1973. 5

[155] V. Ramasubramanian and K. K. Paliwal. Fast K-dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding. *IEEE Trans. Signal Proc.*, 40(3):518–531, March 1992. 186

[156] K. Ramchandran and M. Vetterli. Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility. *IEEE Trans. Image Proc.*, 3(5):700–704, September 1994. 198

[157] S. Rangan. Personal communication, December 1998. 175

[158] S. Rangan and V. K Goyal. Recursive consistent estimation with bounded noise. *IEEE Trans. Inform. Th.*, July 1998. Submitted. 19, 27, 48, 49, 50, 50, 51

[159] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990. 15, 17, 191, 193, 195, 196

[160] A. H. Reeves. French Patent 852 183, October 23, 1939; U.S. Patent 2 272 070, February 3, 1942. 5

[161] B. Rimoldi. Successive refinement of information: Characterization of the achievable rates. *IEEE Trans. Inform. Th.*, 40(1):253–259, January 1994. 108

[162] E. A. Riskin and R. M. Gray. A greedy tree growing algorithm for the design of variable rate vector quantizers. *IEEE Trans. Signal Proc.*, 39(11):2500–2507, November 1991. 185

[163] J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Trans. Inform. Th.*, IT-30(4):629–636, July 1984. 56

[164] L. G. Roberts. Picture coding using pseudo-random noise. *IRE Trans. Inform. Th.*, IT-8(2):145–154, February 1962. 11

[165] K. Sayood and J. C. Borkenhagen. Use of residual redundancy in the design of joint source/channel coders. *IEEE Trans. Comm.*, 39(6):838–846, June 1991. 17

[166] D. W. E. Schobben, R. A. Beuker, and W. Oomen. Dither and data compression. *IEEE Trans. Signal Proc.*, 45(8):2097–2101, August 1997. 58

[167] A. Segall. Bit allocation and encoding for vector sources. *IEEE Trans. Inform. Th.*, IT-22(2):162–169, March 1976. 12

[168] S. M. Selby, editor. *Standard Mathematical Tables*. CRC Press, eighteenth edition, 1970. 45

[169] S. D. Servetto, K. Ramchandran, V. Vaishampayan, and K. Nahrstedt. Multiple-description wavelet based image coding. In *Proc. IEEE Int. Conf. Image Proc.*, Chicago, October 1998. 102, 114

[170] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, July 1948. Continued 27:623–656, October 1948. vi, 2, 3, 3, 4, 4, 184

[171] C. E. Shannon. The bandwagon. *IRE Trans. Inform. Th.*, IT-2(1):3, March 1956. 2

[172] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Int. Conv. Rec., part 4*, 7:142–163, 1959. Reprinted with changes in *Information and Decision Processes*, ed. R. E. Machol, McGraw-Hill, New York, 1960, pp. 93–126. 4, 7, 110

[173] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. Univ. Illinois Press, Urbana, IL, 1963. 5

[174] J. M. Shapiro. Embedded image coding using zero trees of wavelet coefficients. *IEEE Trans. Signal Proc.*, 41(12):3445–3462, December 1993. 168

[175] P. G. Sherwood and K. Zeger. Error protection of wavelet coded images using residual source redundancy. In *Proc. 31st Asilomar Conf. Sig., Sys., & Computers*, Pacific Grove, CA, November 1997. 17, 108

[176] D. Sinha and J. D. Johnston. Audio compression at low bit rates using a signal adaptive switched filterbank. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 2, pages 1053–1056, Atlanta, GA, May 1996. 15

[177] D. Sinha, J. D. Johnston, S. Dorward, and S. Quakenbush. The perceptual audio coder (PAC). In *The Digital Signal Processing Handbook*, chapter 42, pages 42.1–42.18. CRC and IEEE Press, 1998. 138, 138

[178] Y. Steinberg and M. Gutman. An algorithm for source-coding subject to a fidelity-criterion, based on string-matching. *IEEE Trans. Inform. Th.*, 39(3):877–886, May 1993. 55

[179] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986. 26, 26

[180] G. Strang. *Linear Algebra and Its Applications, Third Edition*. Harcourt Brace Jovanovich, San Diego, CA, 1988. 168

[181] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969. 187

[182] N. T. Thao. *Deterministic Analysis of Oversampled A/D Conversion and Sigma-Delta Modulation, and Decoding Improvements using Consistent Estimates*. PhD thesis, Columbia Univ., 1993. 20

[183] N. T. Thao and M. Vetterli. Deterministic analysis of oversampled A/D conversion and decoding improvement based on consistent estimates. *IEEE Trans. Signal Proc.*, 42(3):519–531, March 1994. 20

[184] N. T. Thao and M. Vetterli. Reduction of the MSE in R-times oversampled A/D conversion from $O(1/R)$ to $O(1/R^2)$. *IEEE Trans. Signal Proc.*, 42(1):200–203, January 1994. 19, 20, 22, 23, 25, 46

[185] N. T. Thao and M. Vetterli. Lower bound on the mean-squared error in oversampled quantization of periodic signals using vector quantization analysis. *IEEE Trans. Inform. Th.*, 42(2):469–479, March 1996. 20, 47, 49

[186] C. D. Thompson. Fourier transforms in VLSI. *IEEE Trans. Comp.*, C-32(11):1047–1057, November 1983. 189

[187] C. D. Thompson. The VLSI complexity of sorting. *IEEE Trans. Comp.*, C-32(12):1171–1184, December 1983. 189

[188] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Trans. Inform. Th.*, 39(3):821–834, May 1993. 112, 114, 115

[189] V. A. Vaishampayan. Application of multiple description codes to image and video transmission over lossy networks. In *Proc. Int. Workshop on Packet Video*, March 1996. 102, 114

[190] V. A. Vaishampayan and J.-C. Batllo. Asymptotic analysis of multiple description quantizers. *IEEE Trans. Inform. Th.*, March 1997. 114

[191] V. A. Vaishampayan, J.-C. Batllo, and A. R. Calderbank. On reducing granular distortion in multiple description quantization. In *Proc. IEEE Int. Symp. Inform. Th.*, page 98, Cambridge, MA, August 1998. 114

[192] V. A. Vaishampayan and J. Domaszewicz. Design of entropy-constrained multiple-description scalar quantizers. *IEEE Trans. Inform. Th.*, 40(1):245–250, January 1994. 114

[193] S. Vembu, S. Verdú, and Y. Steinberg. The source-channel separation theorem revisited. *IEEE Trans. Inform. Th.*, 41(1):44–54, January 1995. 4

[194] M. Vetterli and T. Kalker. Matching pursuit for compression and application to motion compensated video coding. In *Proc. IEEE Int. Conf. Image Proc.*, volume 1, pages 725–729, Austin, TX, November 1994. 20, 30, 31

[195] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, 1995. 11, 15

[196] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1978. 82

[197] J. Vuillemin. A combinatorial limit to the computing power of VLSI circuits. *IEEE Trans. Comp.*, 32(3):294–300, March 1983. 189

[198] G. K. Wallace. The JPEG still picture compression standard. *Comm. ACM*, 34(4):30–44, April 1991. 15

[199] G. K. Wallace. The JPEG still picture compression standard. *IEEE Trans. Consum. Electron.*, 38(1):xviii–xxxiv, February 1992. 15

[200] J. Walrand. *Communication Networks: A First Course*. McGraw-Hill, Boston, MA, second edition, 1998. 102

[201] Y. Wang, M. T. Orchard, and A. R. Reibman. Multiple description image coding for noisy channels by pairing transform coefficients. In *Proc. IEEE Workshop on Multimedia Sig. Proc.*, pages 419–424, Princeton, NJ, June 1997. 102, 114, 114, 116, 136, 136

[202] Y. Wang, M. T. Orchard, and A. R. Reibman. Optimal pairwise correlating transforms for multiple description coding. In *Proc. IEEE Int. Conf. Image Proc.*, Chicago, October 1998. 102, 164

[203] B. Widrow, I. Kollár, and M.-C. Liu. Statistical theory of quantization. *IEEE Trans. Intrum. Meas.*, 45(2):353–361, April 1996. 10

[204] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr. Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proc. IEEE*, 64(8):1151–1162, August 1976. 75, 97

[205] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Upper Saddle River, NJ, 1985. 75, 77, 79, 89

[206] N. Wiener. What is information theory? *IRE Trans. Inform. Th.*, IT-2(2):48, June 1956. 3

[207] S. Winograd. *Arithmetic Complexity of Computations*. Soc. for Industrial and Applied Mathematics, Philadelphia, PA, 1980. 186, 187

[208] H. S. Witsenhausen. On source networks with minimal breakdown degradation. *Bell Syst. Tech. J.*, 59(6):1083–1087, July–August 1980. 101, 107

[209] H. S. Witsenhausen and A. D. Wyner. Source coding for multiple descriptions II: A binary source. *Bell Syst. Tech. J.*, 60(10):2281–2292, December 1981. 107

[210] J. K. Wolf, A. D. Wyner, and J. Ziv. Source coding for multiple descriptions. *Bell Syst. Tech. J.*, 59(8):1417–1426, October 1980. 105

[211] R. C. Wood. On optimum quantization. *IEEE Trans. Inform. Th.*, IT-15(2):248–252, March 1969. 6

[212] A. D. Wyner. Capabilities of bounded discrepancy decoding. *Bell Syst. Tech. J.*, 44:1061–1122, July–August 1965. 181, 182

[213] A. D. Wyner and J. Ziv. Classification with finite memory. *IEEE Trans. Inform. Th.*, 42(2):337–347, March 1996. 186

[214] S.-M. Yang and V. A. Vaishampayan. Low-delay communication for Rayleigh fading channels: An application of the multiple description quantizer. *IEEE Trans. Comm.*, 43(11):2771–2783, November 1995. 114

[215] D. C. Youla. Mathematical theory of image restoration by the method of convex projections. In H. Stark, editor, *Image Recovery: Theory and Application*. Academic Press, 1987. 25, 34

[216] B. Yu. A statistical analysis of adaptive scalar quantization based on quantized past data. *IEEE Trans. Inform. Th.*, 1996. Submitted. 71

[217] G. Yu, M. M.-K. Liu, and M. W. Marcellin. POCS based error concealment for packet video using multi-frame overlap information. *IEEE Trans. Circuits Syst. Video Technol.*, August 1996. Submitted. 166

[218] G. Yu, M. W. Marcellin, and M. M.-K. Liu. Recovery of video in the presence of packet loss using interleaving and spatial redundancy. In *Proc. IEEE Int. Conf. Image Proc.*, volume II, pages 105–108, Lausanne, Switzerland, September 1996. 166

[219] R. Zamir. Gaussian codes and Shannon bounds for multiple descriptions. *IEEE Trans. Inform. Th.*, June 1998. Submitted. 110, 112

[220] R. Zamir and M. Feder. On universal quantization by randomized uniform/lattice quantization. *IEEE Trans. Inform. Th.*, 38(2):428–436, March 1992. 51

[221] R. Zamir and M. Feder. Rate-distortion performance in coding bandlimited sources by sampling and dithered quantization. *IEEE Trans. Inform. Th.*, 41(1):141–154, January 1995. 51, 52, 150

[222] R. Zamir and M. Feder. Information rates of pre/post-filtered dithered quantizers. *IEEE Trans. Inform. Th.*, 42(5):1340–1353, September 1996. 51, 52, 150

[223] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek. CREW: Compression with reversible embedded wavelets. In J. A. Storer and M. Cohn, editors, *Proc. IEEE Data Compression Conf.*, pages 212–221, Snowbird, Utah, March 1995. IEEE Comp. Soc. Press. 167

[224] Z. Zhang. *Matching Pursuit*. PhD thesis, New York Univ., 1993. 30

[225] Z. Zhang and T. Berger. New results in binary multiple descriptions. *IEEE Trans. Inform. Th.*, IT-33(4):502–521, July 1987. 101, 103, 104, 107

[226] Z. Zhang and T. Berger. Multiple description source coding with no excess marginal rate. *IEEE Trans. Inform. Th.*, 41(2):349–357, March 1995. 103, 107

[227] Z. Zhang and V. K. Wei. An on-line universal lossy data compression algorithm via continuous codebook refinement—Part I: Basic results. *IEEE Trans. Inform. Th.*, 42(3):803–821, May 1996. 54, 55

[228] J. Zhuang. Adaptive transforms and quantization. Master's thesis, Univ. California, Berkeley, 1997. Available as Univ. California, Berkeley, Electron. Res. Lab. Memo. No. UCB/ERL M97/54, August 1997. 59, 62, 67

[229] J. Ziv. Coding for sources with unknown statistics: Part I. Probability of error. *IEEE Trans. Inform. Th.*, IT-18(3):384–389, May 1972. 54

[230] J. Ziv. Coding for sources with unknown statistics: Part II. Distortion relative to a fidelity criterion. *IEEE Trans. Inform. Th.*, IT-18(3):389–394, May 1972. 54

[231] J. Ziv. On universal quantization. *IEEE Trans. Inform. Th.*, IT-31(3):344–347, May 1985. 51, 56

[232] J. Ziv. Back from infinity: A constrained resources approach to information theory. *IEEE Inform. Th. Soc. Newsletter*, 48(1):1+, March 1998. From a 1997 Shannon Lecture. 186

[233] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Inform. Th.*, IT-23(3):337–343, May 1977. 54

[234] J. Ziv and M. Zakai. Some lower bounds on signal parameter estimation. *IEEE Trans. Inform. Th.*, IT-15(3):386–391, May 1969. 49

[235] W. H. Zurek. Thermodynamic cost of computation, algorithmic complexity and the information metric. *Nature*, 341(6238):119–124, September 1989. 189