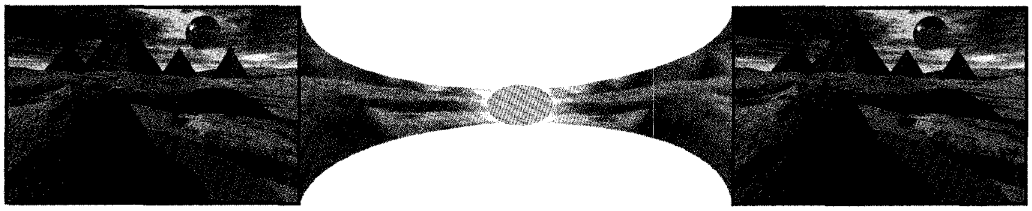
Rate-Distortion Methods for IMAGE AND VIDEO COMPRESSION



Antonio Ortega and Kannan Ramchandran

From Information Theory to Optimization of Compression Standards

In this article we provide an overview of rate-distortion (R-D) based optimization techniques and their practical application to image and video coding. We begin with a short discussion of classical rate-distortion theory and then we show how in many practical coding scenarios, such as in standards-compliant coding environments, resource allocation can be put in an R-D framework. We then introduce two popular techniques for resource allocation, namely, Lagrangian optimization and dynamic programming. After a discussion of these two techniques as well as some of their extensions, we conclude with a quick review of recent literature in these areas citing a number of applications related to image and video compression and transmission. We also provide a number of illustrative boxes to capture the salient points of the article.

From Shannon Theory to MPEG Coding

Recent years have seen significant research activity in the area of R-D optimized image and video coding (see [1,2] and many of the references in this article). In this section, we start by discussing classical rate-distortion theory and briefly outlining some of its fundamental contributions related to its establishment of the fundamental performance bounds on compression systems for specific sources. This will lead us to show the potential limitations of these theoretical results when it comes to dealing with complex sources such as images or video, thus setting the stage for the more restrictive R-D optimization frameworks to be discussed in later sections.

Classical R-D theory

The starting point of classical rate distortion (R-D) theory can be found in Shannon's seminal work [3, 4], whose 50th anniversary is being celebrated this year. Rate-distortion theory comes under the umbrella of source coding or compression, which is concerned with the task of maximally stripping redundancy from a source, subject to a fidelity criterion. In other words, rate-distortion theory is concerned with the task of representing a source with the fewest number of bits possible for a given reproduction quality.

1053-5888/98/$10.00©1998IEEE

Having flexibility at the encoder increases the robustness of compression against modeling mismatches.

Source representation is a rather vague proposition unless we first establish what a "source" is. For example we can consider a source to be one particular set of data (a text file, a segment of digitized audio, an image or a video clip). Alternatively we can consider a *class* of sources that are characterized by their statistical properties (text files containing C code, speech segments, natural images or videoconferencing sequences). When one considers a class of sources, it is clear that efficient source coding entails taking advantage of the "typical" behavior within that class. For example, this means that techniques that work well for speech may not get us too far if applied to video. However, even a narrowly defined "class" of inputs will likely show significant variations among inputs (e.g., different scenes in a video sequence), and thus techniques that allow an "input-by-input" parameter selection are likely to be superior to those that result in a "one size fits all" coding for all inputs in the class. In this article, we will present techniques that strive in some sense to attain the best of both worlds to be achieved: the coding scheme is designed based on typical features of a class of signals, but the coding parameters, within the selected coding framework, are chosen on an input-by-input basis to optimize the particular realization in the statistical class of interest.

Compression can be achieved with "lossless" techniques where the decoded or decompressed data is an exact copy of the original (as is the case in such staple software tools as *zip*, *gzip*, or *compress*). Lossless compression is important where one needs perfect reconstruction of the source. However, this requirement also makes compression performance somewhat limited, especially for applications where the amount of source information is voluminous, bandwidth or storage constraints are severe, and a perfect rendition of the source is overkill. As an example, consider terrestrial broadcast (at about 20 Mb/s) of HDTV (raw bit rate of over 1 Gb/sec) that would require a compression ratio exceeding 50:1, which is at least an order of magnitude in excess of the capacity of the best lossless image compression methods.

In such scenarios, "lossy" compression is called for. Higher compression ratios are possible at the cost of imperfect source representation. The trade-off between source fidelity and coding rate is exactly the rate-distortion trade-off. Lossy approaches are preferred for coding of images and video (and are used in popular compression algorithms such as JPEG [5]). Compression is lossy in that the decoded images are not exact copies of the originals but, if the properties of the human visual system are correctly exploited, original and decoded images will be almost indistinguishable. In the lossy case one can thus trade-off the number of bits in the representation (the rate) with the fidelity of the representation (the distortion). This, as noted by Shannon, is a fundamental trade-off as it states the question: how much fidelity in the representation are we willing to give up in order to reduce the storage (or the number of bits required to transmit the data)? The main purpose of this article is to survey and overview how these R-D trade-offs are taken into account in practical image and video coders, thus clarifying how these information-theoretic techniques have had an impact in everyday practice. Along the way we will discuss several coding problems that are typically solved using R-D techniques and will introduce the optimization techniques (such as Lagrangian optimization and dynamic programming) that are becoming an essential part of the coder designer toolbox.

Although R-D theory, as stated earlier, comes under the umbrella of source coding, it is important to note that the theory is applicable also in the more general context of data transmission over a noisy channel. This is due to Shannon's celebrated *separation principle* of digital communication, where he proved the optimality of dividing the problem of optimal transmission of information (optimal in the sense of most efficient use of available resources such as power, bandwidth, etc.) into that of (i) representing the information efficiently and then (ii) protecting the resulting representation so that it can be transmitted virtually loss-free to a receiver. We will see more about this a little later (see Box 5). The idea seems to be intuitively good, as signal representation issues appear to be inherently different from those involved in efficient digital communication. There are many practical situations in which separation holds (and even in situations where it does not, this "divide and conquer" approach provides a good way to tackle a problem!). The impact of this result is hard to overestimate as it has set the stage for the design of all current digital communications systems. Still, there are, as we will see in Box 5 and later in this article, important cases in which joint consideration of source and channel (i.e., ignoring separation) may in fact be useful.

Distortion Measures: The Elusive Problem

The issue of what distortion measures are more suitable for speech, audio, images, or video has been the object of continuing study for as long as digital representation of these signals has been considered. Clearly, since these sources are encoded and transmitted to be ultimately played back or displayed for a human listener/observer, a distortion measure should be consistent with what the subject can observe or hear. Thus, distortion measures that correlate well with the perceptual impact of the loss should be favored. Leaving aside obvious differences in perception between individuals, finding a general enough, not to mention easily computable, measure of perceptual quality has proven to be an elusive goal. Thus, in practice, simple and perceptually sound design rules are applied, wherever perceptual quality measures are unavailable or too complex. For example, known characteristics of human perception dictate that not all frequencies in an audio signal or an image have the same importance. With these design rules in mind, appropriate frequency weighting can be introduced at the encoder. After the perceptual weighting has been performed, an optimized encoder can still be used to minimize an objective distortion measure, such as for example the mean squared error (MSE).

It is worth noting that while it is typical to dismiss MSE as being poorly correlated to human perception, systems built on the above philosophy (i.e., based on a perceptually meaningful framework) can be optimized for MSE performance with excellent results not only in MSE (as one would hope should be the case) but also in terms of perceptual quality. An example of this observation can be found in the current JPEG 2000 image compression standardization process, which seeks to replace the current JPEG standard [6]. The comparisons made at the Sydney meeting in November 1997 showed that coders that incorporated techniques to minimize the MSE were ranked at the top in *both perceptual and objective tests* [7]!

However, it is also important to realize that significant gains in objective (e.g., average MSE) quality may not translate into comparably significant gains in perceptual quality. Since the success of a particular coding application ultimately does not depend on objective quality measures, it will be necessary to determine at the design stage if any applicable optimization approaches can be justified in terms of the trade-off between implementation cost and perceptual quality. This further emphasizes the need to incorporate perceptual criteria into the coder design so that any further optimizations of the encoding have to choose among "perceptually friendly" operating points.

In the remainder of the article we will assume that MSE, or some suitably weighted version of MSE, has been used. We refer to [8] and references therein for a review of perceptual-coding issues for audio, images, and video.

Optimality and R-D Bounds

Rate-distortion theory [9] has been actively studied in the information-theory community for the last 50 years. The focus of this study has been to a large extent the derivation of performance bounds; that is, determining the region of achievable points in the rate distortion (or bits-fidelity) trade-off for certain limited statistical source classes.

One can distinguish between two classes of bounds, those based on Shannon theory [3, 9, 10] and those derived from high rate approximations [11-13]. The former provides asymptotic results as the sources are coded using longer and longer blocks. The latter assumes fixed input block sizes but estimates the performance as the encoding rate becomes arbitrarily large. A comparison between these two approaches can be found in [14]. Bounds computed with either set of techniques will allow us to determine boundaries between achievable and nonachievable regions. However, the bounds may not be tight for situations of practical relevance (e.g., relatively low rate and small block sizes). Moreover, generally these bounds are not constructive. Still, if bounds could be computed they would provide useful information to benchmark specific applications.

Unfortunately, to derive bounds one needs to first characterize the sources and this can be problematic for complex sources such as video. Indeed, bounds are likely to be found only for the simpler statistical source models. For example, bounds have been known for independent identically distributed (i.i.d.) scalar sources with Gaussian, Laplacian, or generalized Gaussian distributions. The latter distribution is fairly general and can be used to model numerous real-life phenomena; it includes both the Gaussian and Laplacian distributions as special cases and provides a family of probability-density functions where, for a given variance, a shape parameter can be selected to match a range of statistical behaviors, from heavy tailed to fast-decay probability-density functions. The R-D function itself is known in closed form only for Gaussian sources, while for other distributions one would have to resort to numerical optimization methods; e.g., the Blahut-Arimoto algorithm [15].

An interesting and useful special case in R-D theory refers to the use of scalar quantizers, where the samples are quantized one at a time rather than as a collection or vector. The theory of optimal scalar quantization (how to design scalar quantizers with performance sufficiently close to the bounds) has been widely studied. For these simple sources, practical quantizers with various degrees of approximation to the optimal values are available [16]. The simplest kind of scalar quantizer is the uniform scalar quantizer, where the quantizer step sizes are uniform. More sophisticated extensions include fixed-rate nonuniform quantizers (where each quantization level is represented with the same number of bits), also known as Lloyd-Max quantizers, and the variable-length nonuniform quantizer dubbed as the entropy-constrained scalar quantizer, as well as its extensions to vector quantization [17].

While the above-mentioned techniques deal with optimal quantization strategies for a given source distribution, when dealing with complex sources such as images and video signals, the question of what the right source distribution should be involves accurate source modeling. It is therefore important to consider both issues, and optimizing image or video coding performance in fact consists of two steps:

1. Given a particular type of data, say an image, what is the appropriate probabilistic, or other, model for that source?

2. Given the selected model, and any applicable bounds, how close can a practical algorithm come to the optimal performance dictated by the bound?

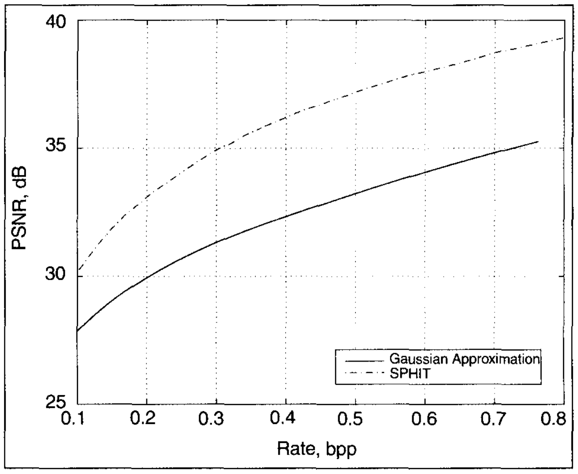
For image and video coding *both steps are equally important* because models that can adequately capture the statistical redundancies may not be available (or may be too

Box 1 - Use of Statistical Image Models for Compression

As mentioned in the "Optimality and R-D Bounds" section, devising a good image coding algorithm involves two important intellectual components: **(i)** selecting a sound operational model or framework, and **(ii)** striving to optimize coding performance in the selected framework. Although this article concentrates (as does most of the coding community) on the latter component, we would like to briefly emphasize the importance of the first one as well. To this end, let us embark on a statistical experimental adventure that we hope will help highlight the salient features of typically deployed models. Our goal is to enable one to "visualize" the effectiveness of these models by synthesizing "images" derived from typical realizations of these assumed models. The advantages of such an exercise are manyfold: it can not only expose the powers and shortcomings of different attributes of the test models, but it can additionally inspire the creation of new and improved frameworks that can embellish some of the drawbacks.

▲ *1. Original Lena image.*

While practical image coding algorithms have been founded on a variety of operational frameworks, simple statistical models have been very popular. For example, early state-of-the-art subband image coding frameworks were based on i.i.d. models for image subbands (based on Gaussian or Laplacian p.d.f.s) and optimal bit-allocation techniques to ensure that bits were optimally distributed among the subbands in proportion to their importance, as gleaned through the variance of their distributions [19]. A coding algorithm based on such a framework would no doubt be very efficient at incorporating the second component mentioned in the previous paragraph, but it raises the obvious question about how accurate an i.i.d. Laplacian or Gaussian subband model might be. Let us try to address this question by taking a subband decomposition of a typical image (such as the Lena image, (see Fig. 1)), measuring the empirical variances of the different subbands, and modeling these subbands as i.i.d. Gaussian distributions. Figure 2 shows the theoretically attainable rate-distortion performance (using water-pouring concepts from information theory [15]). Note that this is the optimal performance theoretically attainable using infinite Shannon-style complexity involving asymptotic random coding arguments based on infinitely long vectors of samples [15]. Yet, as seen in Fig. 2, this coder is handsomely outperformed by a low-complexity modern-day wavelet image coder such as Shapiro's EZW



▲ *2. Rate-distortion curves achieved with (i) the SPIHT coder [21]; and (ii) with the Shannon R-D bounds corresponding to an i.i.d. zero-mean Gaussian model for each wavelet subbands (with empirically measured variances): this results in a Gaussian vector source, and water-pouring arguments are used to find the theoretical R-D bounds [15].*

complex to even allow us to find a bound!). This point is illustrated by the example of Box 1. Additional discussions of approaches for practical modeling of complex data can be found in the article by Effros in this special issue [18].

R-D Meets R&D: Operational R-D in Practical Coder Design

As just discussed, R-D performance is the fundamental trade-off in the design of any lossy compression system. We have highlighted how fundamental theoretical research has resulted in the computation of performance bounds but also indicated two major concerns with these R-D theoretical benchmarks:

1. complexity (how much memory, delay or computation is required? Can we construct a practical algorithm to approach the bound?), and

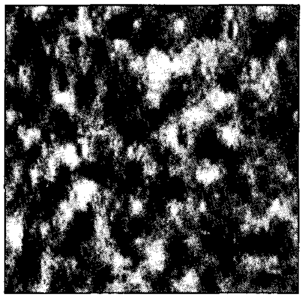
2. model mismatch (how good are our modeling assumptions? are they too simple to characterize the sources fully?).

These will be addressed in the following sections.

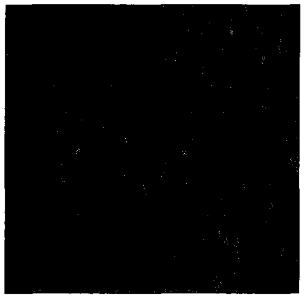
Choosing the Parameters of a Concrete System: Operational R-D

To guarantee that our design will be practical we can abandon our search for the best unconstrained R-D performance of *any* system. Instead let us start by choosing a specific coding scheme that efficiently captures the relevant statistical dependencies associated with the source, while also satisfying our system requirements of coding complexity, delay and memory. Then we can search for the best operating points for *that specific system*.

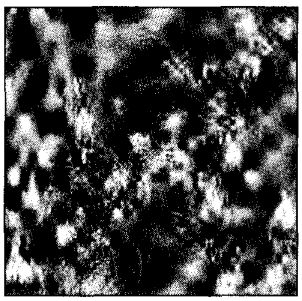
For example consider a scalar quantizer followed by an entropy coder. This quantizer is completely defined by its quantization bins, the reproduction level for each bin, and the associated codewords for each reproduction level. Well-known techniques are then available to find the best choice (in an R-D sense) of these parameters for a specific (statistical) source. Similar results are available for other compression schemes (e.g., fixed-rate scalar quantizer, vector quantizer of dimension N, etc.).



▲ 3. *Image synthesized from the statistics of the original image but without sign information and with a single variance assigned to each of the subbands (i.e., no spatially local information is available).*



▲ 4. *Here the image is synthesized again from the global variance measurements but the correct sign information is used.*



▲ 5. *Here we use local measurements of variance but no sign information.*

▲ 6. *Finally, here we synthesize an image with both correct local variance and sign information.*

coder [20] or its improved variant, the SPIHT coder [21]; for example, at a coding rate of 0.5 bit per pixel, the SPIHT coder outperforms the infinite-complexity i.i.d.-Gaussian based Shannon source coder by over 3 dB in SNR! One concludes that the low-complexity SPIHT coder, despite no claims to rate-distortion optimality, is based on a far more accurate image model than the hypothetical Shannon coder, and this makes all the difference in the world.

To drive the point further home, let us try to synthesize "images" derived from the statistical parameters for the assumed model. As before, we take a subband decomposition of the Lena image, measure the subband variances, then create a random realization using an i.i.d. Laplacian model for the image subbands, and then synthesize the image based on the subband description. If one assumes a random sign for the magnitudes of the coefficients (i.e., use a truly random two-sided Laplacian model), one realizes the "bizarre" image of Fig. 3. If the magnitude is assumed to be Laplacian distributed but the *sign* of the random variable is known, then one synthesizes the image of Fig 4., where some of the edge structure becomes faintly exposed. Let us now try a more "local" model that models the image subbands as Laplacian distributed with spatially varying variances, given by an i.i.d. model involving local (say 3×3 windowed) neighborhoods rather than the global subband window. Such a model captures the space and frequency characterization of the wavelet image decomposition (although with a larger number of parameters). Figures 5 and 6 show the image synthesized based on the statistical parameters derived from the Lena image, corresponding to no knowledge and knowledge of the sign information, respectively. From Fig. 6, one can clearly see that this leads to a much more natural looking image and shows the promise of this model. In fact, the EZW and SPIHT coder and other coders of this genre (such as the SFQ coder [22], the subband classification-based coder [23], the EQ coder [24], and other coders based on context-based backward adaptation [25, 26]) can be conceptualized as being far more spatially "localized" in their model. Thus, by interpreting the wavelet data as "space-frequency" sets of information, they derive significant performance gains over the early subband coders that treated the data only as "frequency" sets of information.

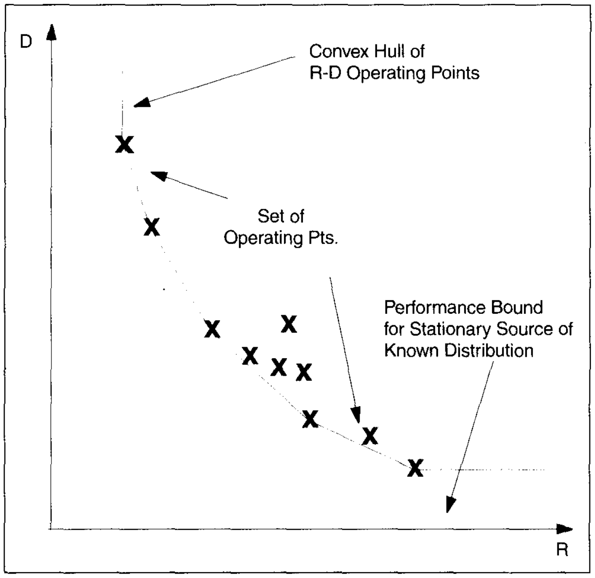
For a given system and source, if we consider all possible quantization choices, we can define an *operational rate-distortion curve*. This curve is obtained by plotting for each rate the distortion achieved by designing the best encoder/decoder pair for the rate. Note that these points are *operational* in that they are directly achievable with the chosen implementations and for the given set of test data. This bound will allow us to distinguish between the best achievable operating points and those that are suboptimal or unachievable. While the bound given by Shannon's theoretical R-D function gives no constructive procedure for attaining that optimal performance, in the operational R-D case, we always deal with achievable points.

A particular case of interest, which we will describe later, is one where the encoder can select among a fixed and discrete set of coding parameters, with each R-D point being obtained through the choice of a specific combination of coding parameters. In that scenario, as illustrated by Fig. 7, we can plot the individual admissible operating points. The boundary between achievable and nonachievable performance is then defined by the convex hull of the set of operating points.

From now on, we will consider optimality in the *operational* sense; i.e., the best achievable performance for a given source (as described by a training set or by a given statistical model) given our choice of compression framework.

Before we address the issue of modeling, let us first consider in more detail the problem of optimal encoder/decoder design. The basic design mechanism can be summarized as follows. First select a particular compression framework (for example, a scalar quantizer with entropy coding as above), then proceed by alternatively designing the encoder (i.e., the rule to map real-valued inputs to quantization indices) and the decoder (i.e., the rule to reproduce a particular quantizer index at the receiver).

First the encoder is optimized for the given decoder; i.e., given the reproduction levels at the decoder, an encoder is designed that produces a mapping having the minimum distortion for a given rate, for the given train-



▲ *7. The operational R-D characteristic is composed of all possible operating points obtained by applying admissible coding parameters to each of the elements in a particular set of data. This is different from the best achievable performance for a source having the same statistical characteristics of the data.*

ing source. Then, the decoder is optimized for the given encoder; i.e., once inputs have been assigned to indices, we choose the best reproduction for a particular set of indices. This design process iterates between these two steps until convergence.

Variations of this approach (known as the Lloyd Algorithm, which was initially proposed for scalar quantizers [27, 28]) have proved to be very popular, even though global optimality cannot be guaranteed. Examples of scenarios where variations of the Lloyd Algorithm have been applied include entropy-constrained quantizers, vector quantizers [29], tree-structured vector quantizers [30] and entropy-constrained vector quantizers [17], among other frameworks. Details can be found in [31] and a more detailed discussion of these types of algorithms and their application is presented in another article in this issue [18].

Choosing a Good Model: The Transform-Coding Paradigm

In our discussion so far, "best" has been defined not only for a particular framework, but also for the given source, as specified by a probabilistic model or by a training set of representative data. Since any applicable probabilistic models will have to be derived in the first place from a training set of data samples from the source, here, without loss of generality, we assume sources to be represented by appropriate training sets. Furthermore, in many cases, e.g., when dealing with multidimensional sources, practical closed-form models may not be available.

It would seem that models are inherent properties of the sources to be compressed and therefore the coder designer has little flexibility in the model selection. Nothing could be further from the truth. In fact a fundamental part of designing a coder is the selection of the underlying model and indeed many choices are typically available. For example, a source can be considered to be scalar, or treated as a set of vectors, or we can model the source after it has been decomposed into its frequency components, etc. (see Box 1). Each of these approaches can model the same original data but provides widely varying compression results.

To first-order approximation, good compression systems based on complex models tend to be more complex to implement (but may provide better performance) than systems based on simpler models. A simple illustration of this rule can be seen when comparing scalar and vector quantizers. Thus, the main difficulty in achieving good R-D performance for images and video is, as exemplified in Box 1, finding a model that is

▲ simple enough that a compression system matched to this model can achieve good performance with reasonable cost,

▲ but complex enough to capture the main characteristics of the source.

For example, as seen in Box 1, using a simple i.i.d. model for each subband as the basis for our coder design results in very poor coding performance as it does not exploit existing spatial redundancies (see Figs. 2 and 3). However, the (still simple) models that assume correct local variance and sign information can be seen to capture a great deal of image information (see Fig. 6) and indeed models of this kind underlie many state-of-the-art wavelet image coders.

There are many approaches to achieving this dual goal of "matching without excessive complexity," most of them based on a simple principle: *to replace a single complex model by a multitude of simple models*. This approach is described in detail in the article by Effros in this special issue [18]. Here we describe one particular instance of this method, namely, *transform coding*.

The transform-coding paradigm calls for decomposing the source into its frequency components using block transforms such as the discrete cosine transform (DCT) or subband coding using now-popular wavelet filters. It is only after decomposing the source into its frequency components or bands that we apply quantization. Thus, we consider the R-D trade-offs in the transform domain. From the standpoint of modeling this has the main advantage of allowing us to use simple models, as shown in Box 1.

That this approach is extremely useful can be verified by how widely it is used in recent image and video coding standards, from JPEG to MPEG 1/2/4 and on to JPEG 2000. We refer the reader to Boxes 2 and 3 for two examples of transform coding, based on block transforms such as the DCT and wavelet transforms, respectively. Our goal is twofold: to provide intuition as to why these approaches are a good idea, and secondly to give examples of resource allocation issues that arise in transform coding frameworks.

Box 2 - An Example of Transform Coding: JPEG

The term transform coding generically describes coding techniques where the source data is first decomposed using a linear transform and where each of the frequency components obtained from the decomposition are then quantized.

A typical transform-based image coder comprises the cascade of a front-end linear transform followed by a scalar quantization stage and then an entropy coder. The transform serves the dual roles of (i) energy compaction, so that the bulk of the signal energy is isolated in a small fraction of the transform coefficients and (ii) signal decorrelation, so that there is little loss in performance due to simple scalar quantization: this is possible because the set of all transform coefficients representing a given frequency can be, to first order, modeled as a memoryless source (e.g., i.i.d. Gaussian or Laplacian) for which efficient simple quantizers can be found. The scalar quantizer is the lossy part of the framework and confines the representation to a discrete set of indices corresponding to discrete quantization levels, while the last-stage entropy coder removes the redundancy in the quantization index stream.

Commercial image and video compression standards are based on the discrete cosine transform (DCT). Figure 8 provides an example of the most popular mode of operation, the so-called "baseline," within the JPEG compression standard [5]. A brief description of the JPEG coding algorithm follows.

The image is decomposed into 8×8 blocks for the purpose of transform, quantization, and entropy coding. Blocks are processed in a raster scan order and are transformed independently using a block DCT. After the DCT, each 8×8 block is quantized using uniform scalar quantization. Quantization step sizes are defined for each of the 64 frequency coefficients using an 8×8 quantization matrix. Typically, a single quantization table is used for each color component; however, up to four different tables may be used if needed. The values of the quantization tables are encoded in the header of the compressed file. Quantization is a lossy step; i.e., the information cannot be recovered perfectly at the decoder. However, it is the quantization operation that allows one to achieve a high compression rate at the price of some quality degradation.

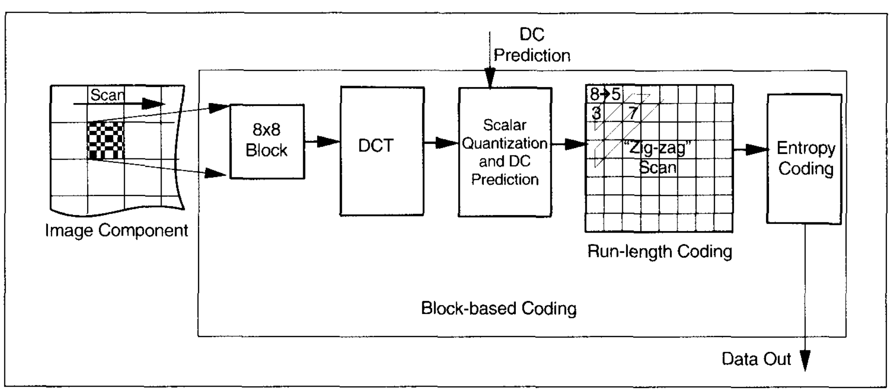
The first quantized frequency coefficient, called the DC coefficient, represents the average sample value in a block and is predicted from the previously encoded block to save bits. Only the difference from the previous DC coefficient is encoded, which typically is much smaller than the absolute value of the coefficient. The remaining 63 frequency coefficients (called AC coefficients) are encoded using only the data of the current block.

The entropy coder constitutes the second basic component in the R-D trade-off, as it determines the number of bits that will be used for a particular image and quantization setting. The entropy coder is lossless and it maps each of the various quantization indices to given codes. A simple way of compacting the quantization index stream would be to assume a memoryless model for the indices and compress it to the first-order entropy of the stream. However, the memoryless assumption is typically a bad one, and significant gains can be had by exploiting the memory in the quantized bitstream (e.g., zero values tend to cluster). A simple way to exploit this is through zero runlength coding. JPEG uses a two-dimensional entropy code based on the length of the zero-run and the magnitude of the nonzero coefficient breaking up the run.

The AC coefficients are processed in a "zig-zag" manner (see Fig. 8) that approximately orders coefficients from lowest to highest frequency. Run-length codes represent the sequence of quantized coefficients as (*run, value*) pairs, where "run" represents the number of zero-valued AC coefficients between the current nonzero coefficient and the previous nonzero coefficient, and "value" is the value (nonzero) of current coefficient. A special end-of-block (EOB) code signals the end of nonzero coefficients in the current block. For the example in Fig. 8, with three nonzero AC coefficients, the sequence after run-length encoding is (0,5)(0,3)(4,7)(EOB). The sequence of "runs" and "values" is compressed using Huffman or arithmetic codes.

Despite the apparent rigidity of the JPEG syntax, there is a surprising amount of room for gains attainable with clever encoder optimization [39, 40]. The syntax allows for the quantization matrix and the entropy coding table to be adapted on a per-image basis as well as for arbitrary compression ratios desired. A more subtle option available is for the encoder to "dupe" the decoder optimally in a rate-distortion sense while being fully syntax-compatible. As an example, small nonzero values that break up potentially long zero-runs are typically very expensive in bit-rate cost in comparison to their relative contribution to reducing quantization distortion. If the encoder can "lie" to the decoder about the magnitude of these coefficients; i.e. call these nonzero values zeroes, then the decoder is none the worse off, while the R-D performance is significantly increased.

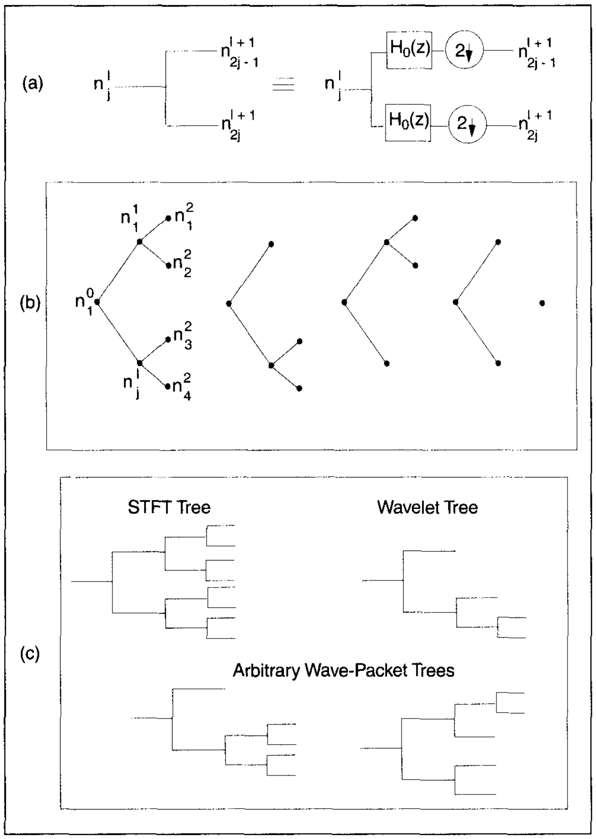
A systematic way of doing this optimally in the R-D sense, termed coefficient thresholding, has been described in [41]. The good news is that sizeable R-D performance gains, of the order of 25% in compression efficiency can be realized *while being completely faithful to the JPEG syntax*. Another article in this issue [2] will provide further evidence of the practical value of R-D techniques in improving the quality in standards-based video coding, where there is even more flexibility in choice of operating parameters.



▲ *8. Block diagram of a JPEG coder.*

Box 3 - Adaptive Transforms Based on Wavelet Expansions

Adaptivity in signal processing is one of the most powerful and desirable features, and the compression application is no exception. Here we outline the role of having adaptive transforms in wavelet image coding. While the DCT is the transform of choice in commercial compression standards such as JPEG, MPEG, etc., the discrete wavelet transform (DWT) has recently emerged as a superior alternative and looks set to replace the DCT as the next-generation transform in the newly emerging JPEG 2000 image compression standard. The wavelet or subband decomposition consists of an octave-band frequency decomposition. It is implemented by a tree-structured filter bank consisting of a cascade of lowpass filters ($H_1(z)$) and highpass filters ($H_0(z)$) followed by decimators (see Figure 9(a)). By recursively splitting the outputs of the lowpass branches, we realize an octave-band frequency decomposition associated with a logarithmic filter-bank tree structure (see Figure 9(c)). This logarithmic frequency decomposition property of the wavelet transform gives good frequency selectivity at lower frequencies and good time (or spatial) selectivity at higher frequencies. This trade-off is well suited to many "natural" images that exhibit long-duration, low-frequency events (e.g., background events in typical scenes) and short-duration high-frequency events (e.g. edges). Other choices of subtrees lead to other time-frequency selectivity trade-offs, as seen in Fig. 9(c). These trees, which represent generalizations of the wavelet tree, are dubbed in the literature as wavelet packets (see Fig. 9).



▲ *9. (a)Two-channel decomposition as a node and two branches in the decomposition tree. (b) All possible binary wavelet packet decompositions of depth 2. (c) Some typical depth-3 binary wavelet packet subtree decompositions.*

The general resource allocation problem for the adaptive-transform coding framework involves selecting the operating point of the *combination* of transform, quantizer, and entropy-coder in order to realize the best rate-distortion trade-off. Depending on the flexibility (and complexity concerns) of the framework, one or all of the above functional components can be jointly optimized. The traditional approach in compression is to use a fixed transform (like the DCT or the DWT) and then choose a quantization strategy matched to the properties of the input process and the fixed transform. The quantization strategy (bit-allocation policy) is typically based on a model for the probability-density functions characterizing the transform coefficients, or in the absence of an analytical model, from training over a large class of "typical" possible signals.

As a first step toward attaining an adaptive transform, it is clear that an improvement can be found if we search over the whole set of binary trees for a particular filter set, instead of using the fixed tree of the wavelet transform. (see Fig. 9). A fast algorithm, also known as the "single tree" algorithm, to find the best tree (dubbed the "best basis") jointly with the best quantization and entropy coding strategy has been described in [42]. The idea is to search for the best basis (for a rate-distortion cost function as appropriate for compression) for the signal from a library of wavelet packet bases. In order to achieve this, two entities are needed: a cost function for basis comparison and a fast search algorithm. See the section on basic components in image/video coding algorithms and Box 9 for a more detailed treatment of these issues.

When considering video sources, we will need additional tools to allow us to fully exploit the redundancy between consecutive frames in a video sequence. Motion compensation is the most popular approach to achieve this goal. The encoder computes the motion parameters (for example block-based motion vectors as in MPEG [32]) and the decoder uses those in the reconstruction framework. A particular framework will specify how the motion information is transmitted and how it is interpreted by the decoder. A more detailed description can be found in the article by Sullivan and Wiegand in this issue [2].

Lack of space prevents us from going into more depth, but we refer the interested reader to recent textbooks for a general description of the more popular methods and algorithms, and how the transform-coding paradigm is put to practice [5, 31-37]. We will revisit the various building blocks in transform-coding frameworks later in the article, where we will outline examples of R-D optimization applied to these algorithms.

Standards-Based Coding: Syntax-Constrained R-D Optimization

Thus far we have considered the complete design of a compression system in which for a given set of constraints, we find encoding and decoding algorithms to suit our needs. Let us assume now that the decoder has been selected; that is, we have a complete specification of the language that can be understood by the decoder, with an accompanying description of how an output (the decoded image or video) is to be produced given a specific input stream.

This scenario is precisely that assumed by most recent international compression standards (JPEG, MPEG, H.26x and so on). Motivated by the desire to maximize interoperability, these standards provide an agreed-upon *bitstream syntax* that any standard-compliant decoder will use to provide an output signal. Agreeing on such standards allows encoding/decoding products from different vendors to talk to one other and has become the preferred way to achieve affordable, widely available digital and video compression.

Still, it is not clear a priori how much flexibility the encoder can enjoy in selecting its modes of operations, if it is constrained by a particular decoder. It turns out that most of these standards are designed to endow the encoder with a lot of flexibility and creativity in its selection of the system parameters, and there is a big gap in performance between the best choice and the worst. In all standards-based applications the encoder can select parameters that will result in various levels of R-D performance. This leads to a situation where the number of operating points is discrete, and thus the operational R-D bound is determined by the convex hull of the set of all operating points (see Fig. 7). For example, as anybody who has used JPEG compression can verify, one can select different rate-quality targets for a particular image and still guarantee that these images can be decoded. Likewise, in video coding, each frame or scene requires a different rate to achieve a given perceptual quality and the encoder needs to control the coding parameter selection to enable proper transmission (see Box 4). In typical image and video coding optimization scenarios such as those involving these standards, the encoding task of selecting the best operating point from a discrete set of options agreed upon a priori by a fixed decoding rule is often referred to as *syntax-constrained* optimization. The selected operating choice is communicated by the encoder to the decoder as side-information, typically as part of the header.

Having flexibility at the encoder also increases the robustness of compression against modeling mismatches. That is, a system designed for a particular source model will perform poorly if the source model should differ. A more robust system can be designed if several compression modes are available and the encoder can select among them. It is important to note that this robustness and flexibility is not free: the amount of side information needed to configure the decoder appropriately will increase if more modes are to be accommodated, and, more importantly, the computational complexity will increase as well.

We have now set the stage to define the class of problems that will occupy the rest of this article. We consider that a general coding framework has been selected, which can accommodate different types of sources, and thus specific parameters can be selected for *each image* (and for each desired rate) and sent to the decoder as overhead. Our goal is then:

Formulation 1 - Discrete R-D Optimization: Parameter Selection for a Given Input

Given a specific encoding framework where the decoder is fully defined, optimize the encoding of a particular image, or video sequence in order to meet some rate/distortion objectives.

Note that here we are assuming deterministic knowledge of the input and our goal is to optimize the parameter selection for *that* input. We no longer seek optimality over an ensemble of inputs, but rather confine ourselves to doing our best for the given input, given the constraints imposed by the coding framework. This is a very realistic scenario, as it can be applied, for example, to most image and video compression standards defined to date (e.g., JPEG, [5], MPEG [32] or H.263 [38]) where the encoding mode selection can be optimized for each input. As is shown in the article by Sullivan and Wiegand in this issue, the potential gains when using these optimization techniques are significant [2].

However, the selection of the initial coding framework is key to the system performance. No matter how sophisticated the optimization techniques one is willing to utilize (and we will describe some fairly complex ones!), if the coding framework is inherently limited or flawed, not much improvement will be achievable. Recall that we are placing ourselves in an operational R-D framework and are thus limited to *only* those R-D operating points that the initial framework can achieve. Thus, a good coding framework without any form of optimization is likely to be superior to a subpar coding approach, no matter whether parameter selection has been R-D optimized in the latter scheme. We refer once more to Box 1 for an example of how the model-selection problem may often be more important than having an optimized algorithm.

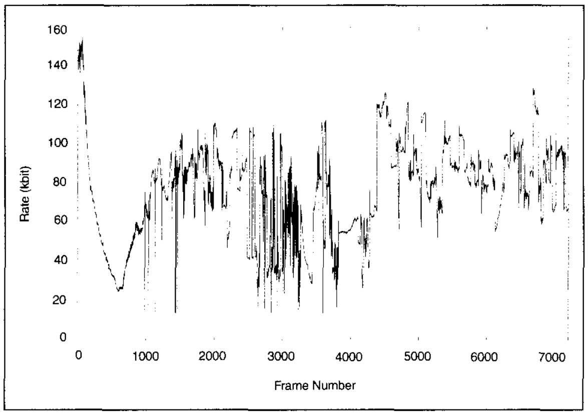
In the spirit of operational R-D, we define here the "optimal" solution as that achieving the best objective function among all possible operating points. Note then that we consider finite sets of coding choices at the encoder and therefore *there exists* one achievable optimal choice of parameters: if all else fails, one could do an exhaustive search comparison of all possible operating points and choose the best. Obviously, our goal will be to find those operating points without a brute-force search.

Typical Allocation Problems

The previous section introduced the framework of discrete R-D optimization and Boxes 2, 3, 4 and 5 gave us

Box 4 - Delay-Constrained Transmission and Buffer Control

Most practical video compression algorithms exploit spatial and temporal redundancy through transform coding and motion estimation, respectively. However, the degree of redundancy, and therefore the resulting rate for a given distortion, can fluctuate widely from scene to scene. For example, scenes with high motion content will require more bits than more stationary ones. This is illustrated by the sample rate trace [43] of Fig. 10. This trace is obtained from the movie *Star Wars*, and more specifically its opening 4 minutes (as true fans no doubt will have guessed). It shows rate changes of close to an order of magnitude when the quantization step size is kept constant (a JPEG coder was used to code each frame.)



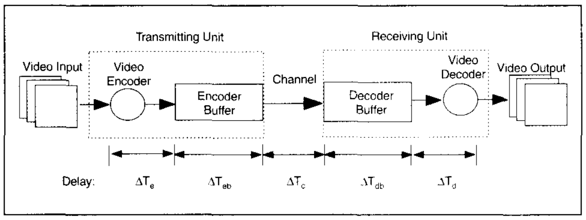
▲ *10. Bit rate per frame for initial 4 minutes in the "Star Wars" trace produced by Mark W. Garrett at Bellcore [43]. This trace was computed using Motion JPEG (i.e., each frame is encoded independently using JPEG, without motion estimation and with the same quantization parameters for every frame) and it clearly shows very significant variations in rate between different scenes. Similar variations can be observed when compressing using other algorithms such as MPEG.*

Let us now consider a typical real-time transmission as illustrated in Fig. 11. As just described, video frames require a variable bit rate and thus it will be necessary to have buffers at encoder and decoder to smooth the bit rate variations. Assuming the video input and video output devices capture and display frames at a constant rate, and no frames are dropped during transmission, it is easy to see that the end-to-end delay in the system will remain constant [44].

Let us call ΔT the end-to-end delay: a frame coded at time t has to be decoded at time $t+\Delta T$. This imposes a constraint on the rate that can be used for each frame (it has to be low enough that transmission can be guaranteed within the delay).

Consider the case when transmission takes place over a CBR channel. Of the delay components of Fig. 11 only ΔT_{eb} and ΔT_{ed} (the time spent in the encoder and the decoder buffer, respectively) will now be variable. Consider, for example ΔT_{eb}—this delay will be at most $B_{\max}/C$, where $B_{\max}$ is the physical buffer size at the encoder and C is the channel rate in bits per second. It is clear that $B_{\max}$ has to be smaller than $\Delta T \cdot C$ or otherwise we could store in the buffer frames, which will then experience too much delay.

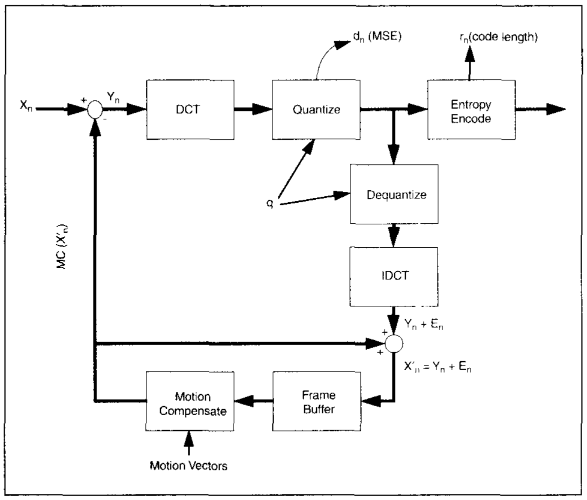
If we consider the transmission of a sequence such as that of Fig. 10, either (i) we will have to use very large buffers (and correspondingly long end-to-end delays), or (ii) we will have to adjust the source rate, and thus the delivered quality, to make it possible to use a smaller buffer (shorter delay) without losing any data. The delays required for the trace of Fig. 10 would be exceedingly long —therefore, in practical applications it is necessary to perform *rate control* to adjust the coding parameters and meet the delay constraints.



▲ *11. Delay components in a communication system.*

As shown in Fig. 12 it is possible to adjust the video rate (and the quality) by modifying the quantization step sizes used for each frame. It is therefore easy to see that rate-control problems can be cast as resource-allocation problems where the goal is to determine how many bits to use on different parts of the video sequence and to do so in such a way as to maximize the quality delivered to the end user. Of course, a natural way to approach these problems is to consider the R-D trade-offs in the allocation, and techniques such as those described in this article have been widely used for rate control [45-48].

Note that even in cases where transmission is performed over a VBR channel, or where the sequence is pre-encoded and stored (e.g., in a digital versatile disk, (DVD)), it is also necessary to perform rate allocation. For example, to store a full-length movie in a DVD it may be necessary to preanalyze the movie and then allocate appropriate target rates to the various parts of the movie. In this way, allocating more bits to the more challenging scenes and fewer to the easier ones will result in a globally uniform quality. R-D-based approaches for rate control for VBR channels have also been studied [49-52].



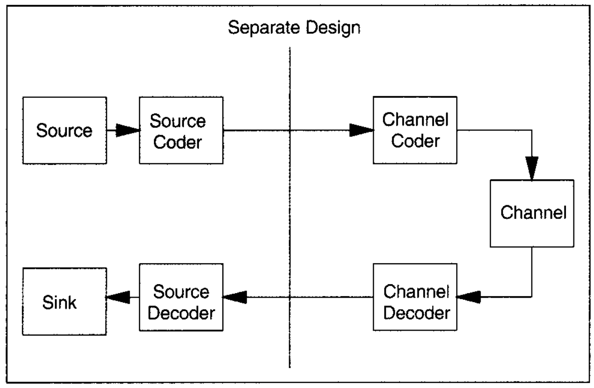
▲ *12. Block diagram of a typical MPEG coder. The quantization parameter can be adjusted to make the rate comply with the channel constraints.*

Box 5 - Rate-Distortion Techniques for Joint Source-Channel Coding

While rate-distortion-based techniques have had a major impact on image and video compression frameworks, their utility extends beyond compression to the bigger framework of image and video transmission systems. The problem of transmitting image and video signals naturally involves both source coding and channel coding. The image or video source has an associated rate-distortion characteristic that quantifies the optimal trade-off between compression efficiency and the resulting distortion. The classical goal of source coding is to operate as closely as possible to this rate-distortion bound. Then comes the task of reliably transmitting this source-coded bitstream over a noisy channel that is characterized by a channel capacity that quantifies the maximum rate at which information can be reliably transmitted over the channel. The classical goal of channel coding is to deliver information at a rate that is as close to the channel capacity as possible. For point-to-point communications with no delay constraints, one can theoretically separate the source and channel coding tasks with no loss in performance. This important information-theoretic result, as stated in the introduction, goes back to Shannon's celebrated *separation principle*, which allows the separate design of a source compression/decompression scheme and a channel coding/decoding scheme, as long as the source code produces a bit rate that can be carried by the channel code. The separation principle is illustrated in Fig. 13.

It is nevertheless important to recall that information theory relies on two important assumptions, namely (i) the use of arbitrarily long block lengths for both source and channel codes and (ii) the availability of arbitrarily high computational resources (and associated delays). It is obvious that such conditions are not met in practice, both because of delay constraints and practical limits on computational resources. For certain multiuser communication scenarios such as broadcast [53] and multicast, the separation theorem does not apply even theoretically, and there is a need for closer interaction between the source and channel coding components. A key issue in the design of efficient image and video transmission systems for these cases therefore involves the investigation of joint design of these source and channel coding components.

The "Role of R-D Techniques in Joint Source-Channel Coding" section will take a more detailed look at typical scenarios involving joint source-channel coding and will give pointers to some of the more recent work in the field. In this overview box, we will formulate the basic joint source-channel coding problem at a high level. It is insightful to note that there are many flavors to this problem depending on the number of jointly designed elements involved. If only the source coder and forward error-correction devices are involved, the resource to be allocated is the total bit rate between the source and channel coder. If the modulator-demodulator (modem) is included in the optimization box, then the transmission power or energy can become the constraint. The cost function is typically the end-to-end distortion of the delivered copy of the source: due to the probabilistic nature of the channel, one has to typically abandon the deterministic distortion metrics of the earlier formulations and instead consider *expected* distortion measures. The distortion is due to *both* source quantization, which is deterministic in nature for a fixed discrete set of quantizer choices, *and* channel noise, which is obviously of a probabilistic nature. This is in contrast to the use of R-D methods for pure source coding, where quantization is the only source of distortion.



▲ *13. Separation principle. Optimality is achieved by separate design of the source and channel codecs.*

Formulation 2 - Joint Source-Channel Coding Optimization

Given a specific operational transmission framework involving a source coder (characterized by its collection of possible R-D operating points), a channel coder (characterized by its collection of error-correcting strengths and code rates), and possibly a modem device (characterized by the modem parameters like constellation, transmission power, etc.), optimize the expected end-to-end delivered image or video quality subject to a constraint on the total bit rate (for source and channel coding) or the total transmission power or energy (if the modem is included in the optimization box) and possibly subject to other constraints like bandwidth and delay as well. Alternatively, the expected distortion can be fixed and the cost function could involve total bit rate or transmission energy.

The bit-allocation problem in the case of joint source-channel coding can be thus formulated as finding the optimal distribution of bits, R_{budget} between source bits, R_{source}, in order to reduce quantization distortion, and channel coding parity bits, $R_{channel}$, in order to minimize the expected distortion $E(D)$ due to both the source quantization and the noisy channel:

$$\min_{\{\text{source parameters},\ \text{channel parameters}\}} E(D) \ \textit{subject to}$$
$$R_{source} + R_{channel} \le R_{budget}.$$

concrete examples of scenarios where this optimization is called for; i.e., where the encoder has to make choices among a finite set of operating modes.

We now present a series of generic problem formulations that spell out some of the possible constraints the encoder will have to meet when performing this parameter selection. These problem descriptions are divided into two classes depending on whether compression is targeted for storage or transmission applications. Before presenting these formulations we discuss several practical issues and introduce the notation.

Selection of the basic coding unit. Until now, we have considered generic R-D trade-offs where a *coding unit*, be it a sample, an image block, or an image, is encoded with given distortion when a particular rate is selected. In a practical scenario it will be necessary to decide at what level, or levels, of granularity to optimize the encoding process. For example, it is possible to consider video frames as the basic coding units in a video coding environment, and thereby measure frame-wise rate and distortion for each frame in the sequence, and then decide frame-wise operating points. Alternatively one can operate at a finer level and consider coding choices for a single frame with the basic coding unit being, for example, the 8 by 8 pixel blocks used in JPEG.

Complex algorithms can be justified in scenarios where encoding is performed just once but decoding is done many times.

Complexity. Complexity is a determining factor in assessing the practicality of the various R-D optimization techniques we will describe. Two major sources of complexity can be identified. First, the R-D data itself may have to be measured from the images and thus several encode/decode operations may have to be performed to determine the R-D values. (Note that this is consistent with our assumption of an operational R-D framework, where the goal is to select the best among those operating points that are achievable.) In order to reduce the computations required to measure rate and distortion for each coding unit, one could resort to models (as, for example, in [48]) that would be employed in the optimization algorithm instead of the actual values. The second source of complexity comes from the search itself. Even if the R-D data is known, or has been adequately modeled, we will have to search for the best operating point and that in itself could be a complicated task.

In addition, complexity depends not only on the number of operations required but also on the delay in computing the optimal solution and, related to it, the storage required by the search algorithm. Obviously, more complex algorithms can be applied in off-line encoding applications whereas live or real-time encoding will limit the admissible delay and complexity. Complex algorithms can also be justified if the quality improvements are significant in scenarios where encoding is performed just once but decoding is done many times. Since standards such as MPEG provide a common decoding framework it is possible to develop encoders covering a range of scenarios; from high-quality, high-complexity professional encoding to low-cost, low-complexity consumer products.

Cost function. Both distortion and rate may be part of the objective functions to be optimized. The objective functions can easily be computed for each coding unit, but when our problem involves deciding on the allocation for a set of coding units, defining the overall cost function requires some additional thought. For example, assume distortion is our objective function; then there are several alternatives for defining the overall distortion measure given the individual distortion measures for each of the coding units. For example one can assume that minimizing the *average* distortion is a desirable objective. But consider now a long video sequence: is it really true that an average distortion measure is appropriate? Would the viewer find more objectionable a result where both average quality and peak distortion are higher, as compared to a scenario where the average quality is lower but so is the worst-case distortion? These are valid questions and they justify the need to consider alternatives to average MSE; for example, minimax approaches (where the worst-case distortion is minimized) [54] or approaches based on lexicographic optimization, which can be seen as a more general case of minimax [55, 56].

Perceptually weighted versions of these cost functions can also be accommodated. As in our earlier discussion of distortion measures, we should emphasize that large gains in a particular objective function (for example MSE) may not always result in comparably large improvements in perceptual quality, even if careful perceptual weighting has been introduced.

Notation. Let us consider N coding units where each coding unit has M different available operating points. For each coding unit i we have information about its rate r_{ij} and distortion d_{ij} when using quantizer j. We make no assumptions of any particular structure for the r_{ij} and d_{ij}; we simply use the convention that quantization indices are listed in order of increasing "coarseness"; i.e. $j = 1$ is the final quantizer (highest r_{i1} and lowest d_{i1}) and $j = M$ is the coarsest. There are no other assumptions made: for example, we do not take into account any possible correlation between the rate and distortion characteristics of consecutive coding units or assume any properties for r_{ij} and d_{ij}. We consider here that the R-D data is known; it is possible replace measured r_{ij} and d_{ij} by values that are estimated based on models but this would not affect the algorithms we propose. Examples of rate-allocation applications that utilize models instead of actual data can be found in [47, 48, 57-60].

It would be trivial to achieve minimal distortion if no constraints on the rate were imposed. More interesting issues arise when one tries to achieve the best performance given some constraints on the rate. We will formulate two classes of closely related problems where the rate constraints are driven by (i) total bit budget (e.g., for storage applications) and (ii) transmission delay (e.g., for video transmission).

Storage Constraints: Budget-Constrained Allocation

In the first class of problems we consider, the rate is constrained by some restriction on the maximum total number of bits that can be used. This total number of bits available, or budget R_T has to be distributed among the different coding units with the goal of minimizing some overall distortion metric. For example we may want to use JPEG as in Box 2 to compress the images within an image database so that they all fit in a computer disk (where now we may be concerned with the aggregate quality over all the images). This problem can be re-stated as follows:

Formula 3 - Budget Constrained Allocation

Find the optimal quantizer, or operating point, $x(i)$ for each coding unit i such that

$$\sum_{i=1}^{N} r_{ix(i)} \leq R_T \tag{1}$$

and some metric $f(d_{1x(1)}, d_{2x(2)}, \cdots, d_{Nx(N)})$ is minimized.

For example, if we are interested in a minimum average distortion (MMSE) problem we have that

$$f(d_{1x(1)}, d_{2x(2)}, \cdots, d_{Nx(N)}) = \sum_{i=1}^{N} d_{ix(i)}.$$

Alternatively, a minimax (MMAX) approach [54, 61] would be such that

$$f(d_{1x(1)}, d_{2x(2)}, \cdots, d_{Nx(N)}) = \max_{i=1}^{N} d_{ix(i)}.$$

Finally, lexicographically optimal (MLEX) approaches [55] have been recently proposed as extensions of the minimax solution. The MLEX approach compares two solutions by sorting their distortions or, as in [55], their quantization indices. For simplicity, assume the quantization indices are used in the comparison, with $j = 1$ being the finest quantizer. Then, to compare two solutions, we sort the quantization indices of all the coding units from largest to smallest: we then compare the resulting sorted lists and we say that the one represented by the smallest number is the best in the MLEX sense. For example, consider four coding units that receive the following two allocations (1,3,4,4) and (3,2,3,2). After sorting, we obtain (4,4,3,1) and (3,3,2,2) and given that we have 3322 < 4431, the second allocation is the better one in the MLEX sense. Allocations derived under the MLEX constraint have the interesting property of tending to equalize the distortion or the quantization scale across all coding units.

In the remainder of the article we will concentrate on the MMSE since it is by far the most widely used. Examples of schemes based on MMAX and MLEX can be found in [54, 61] and [55, 56], respectively.

Allocation Under Multiple Partial Budget Constraints

A more general version of the problem of Formulation 3 may arise in situations where there are not only limitations on total rate but also on the rate available for subsets of coding units. Assume, for example, that a set of images has to be placed in a storage device that is physically partitioned (e.g., a disk array) and that it is impossible (or undesirable for performance reasons) to split images across one or more devices. In this case, we will have to deal with partial constraints on the set of images assigned to each particular device, in addition to the overall budget constraint. An optimal allocation that considers only the aggregate storage constraint may result in an invalid distribution between the storage devices.

Consider the case where two storage devices, each one of size $R_T/2$, are used. We will have then the following constraint, in addition to the budget constraint of Eq. (1):

$$\sum_{i=1}^{N_1} r_{ix(i)} \leq R_T/2,$$

where N_1 is the number of coding units that are stored in the first storage device. N_1 itself may not be given and may have to be determined.

Delay-Constrained Allocation and Buffering

A simple storage-constrained allocation such as that in Formulation 3 cannot encompass situations where the coding units (for example, a series of video frames) are streamed across a link or a network to a receiver. In this situation, as outlined in Box 4, each coding unit is subject to a delay constraint; i.e., it has to be available at the decoder by a certain time in order to be played back.

For example, let a coding unit be coded at time t and assume that it will have to be available at the decoder at time $t + \Delta T$, where ΔT is the end-to-end delay of the system. If each coding unit lasts t_u seconds, then the end-to-end delay can be expressed as $\Delta N = \Delta T / t_u$ in coding units. For example if a video encoder compresses 30 frames per second and the system operates with an end-to-end delay of $\Delta T = 2$ seconds, then the decoder will wait 2 seconds to decompress and display the first frame (assuming no channel transmission delay) and at any given time there will be $\Delta N = 2/(1/30) = 60$ video frames in the system (stored in the encoder or decoder buffers or being transmitted). The video encoder will have to ensure that the rate selection for each frame is such that no frames arrive too late at the decoder.

Given the delay constraints for each coding unit, our problem becomes:

Formulation 4 - Delay-Constrained Allocation

Find the optimal set of quantizers $x(i)$ such that (i) each coding unit i encoded at time t_i is received at the decoder before its "deadline" $t_i + \delta_i$ and, (ii) a given distortion metric, for example one of those used in Formulation 3, is minimized.

This would be an easy problem if there were no constraints on the transmission bandwidth (e.g. when reading video data from a DVD, where peak read-out bandwidth from the disk exceeds the maximum coding rate for any frame). Note, however, that even if users have access to broadband channels, by the universal maxim that expenditures shall always rise to meet the incomes, we may assume that limited bandwidth will be the dominant scenario for the foreseeable future. (We hope readers will forgive two video compression researchers for not claiming otherwise. One of our most respected and senior colleagues in the source-coding community reassures us that he has heard for over 30 years how bandwidths are exploding and there is no more need for compression!)

The complexity of this allocation problem depends on the channel characteristics. Specifically we will need to know if the channel provides a constant bit rate (CBR) or a variable bit rate (VBR), if the channel delay is constant, if the channel is reliable, etc. For simplicity, in what follows let us assume that we have $\delta_i = \Delta T$ for all i.

In both CBR and VBR cases, as shown in Box 4, data will be stored in buffers at encoder and decoder. Assume a variable channel rate of $C(i)$ during the i-th coding unit interval. Then, we will have that the encoder buffer state at time i is

$$B(i) = \max(B(i-1) + r_{ix(i)} - C(i), 0),$$

with $B(0) = 0$ being the initial state of the buffer.

Let us now consider what constraints need to be applied to the encoder buffer state (it can be shown that controlling the encoder buffer suffices to guarantee that the delay constraints are met [44, 49]). First, the buffer state $B(i)$ cannot grow indefinitely because of the finite physical buffer. If $B_{\max}$ is the physical memory available then we will need to guarantee that $B(i) \le B_{\max}$ at all times. In addition, in order for the delay constraint of Formulation 4 not to be violated, we need to guarantee that the data corresponding to coding unit i is transmitted before $t_i + \Delta T$; that is, transmission has to be completed during the next ΔN coding unit intervals. Intuitively, in order for this constraint to be met, all we need to ensure is that the future channel rates, over the next ΔN units, are sufficient to transmit all the data in the buffer.

Let us define the *effective buffer size* $B_{eff}(i)$ as

$$B_{eff}(i) = \sum_{k=i+1}^{i+\Delta N} C(k),$$

i.e., the sum of future channel rates over the next ΔN intervals. Then it is easy to see [44, 49] that correct transmission is guaranteed if

$$B(i) \le B_{eff}(i), \quad \forall i.$$

As an example, consider the case where $C(i) = \overline{C} = R_T / N$ is constant. Then, if the system operates with an end-to-end delay ΔN the buffer can store no more than $\Delta N \cdot \overline{C}$ bits at time i. For a detailed analysis of the relationship between buffering and delay constraints, we refer to [44, 49].

We call this the *effective* size because it defines an imposed constraint regardless of the physical buffer size. In general, the applicable constraint will be imposed by the smallest of $B_{eff}(i)$ and $B_{\max}$. Assuming that sufficient physical buffer storage is available (i.e., $B_{\max}$ is always larger than $B_{eff}(i)$, certainly reasonable with the constantly decreasing price of memory) our problem becomes:

Formulation 5 - Buffer-Constrained Allocation

Find the optimal set of quantizers $x(i)$ for each i such that the buffer occupancy

$$B(i) = \max(B(i-1) + r_{ix(i)} - C(i), 0),$$

is such that

$$B(i) \le B_{eff}(i)$$

and some metric $f(d_{1x(1)}, d_{2x(2)}, \cdots, d_{Nx(N)})$ is minimized.

It is worth noting that the problems of Formulations 3 and 5 are very much related. For example, consider the case where $C(i)$ is constant and equal to $\overline{C}$. In this situation the overall allocation is in both cases constrained by the same total budget. Therefore, if the solution to Formulation 3 meets the constraints of Formulation 5 it is also the optimal solution to the latter problem. This fact can be used to find approximate solutions to the problem of Formulation 5 as shown in [46].

It is also interesting to note that the constraints depend on the channel rates. When the channel rates can be chosen by the user (e.g., transmission over a network), this leads to interesting questions on which is the best combination of source and channel rates given constraints on the channel rates [49-51]. In scenarios where the channel is unreliable, we cannot deterministically know what the future rates will be, but it is possible, if channel models are available, to replace channel rates by their estimated values in the above formulation [52, 62, 63].

The R-D Optimization Toolbox

In this section we describe in more detail some of the basic techniques that can be applied to the problems that we have just described. Our goal here is to explain these tools in generic terms. Later sections will provide pointers to specific work where modified versions of these methods have been successfully been applied to a varierty of compression scenarios.

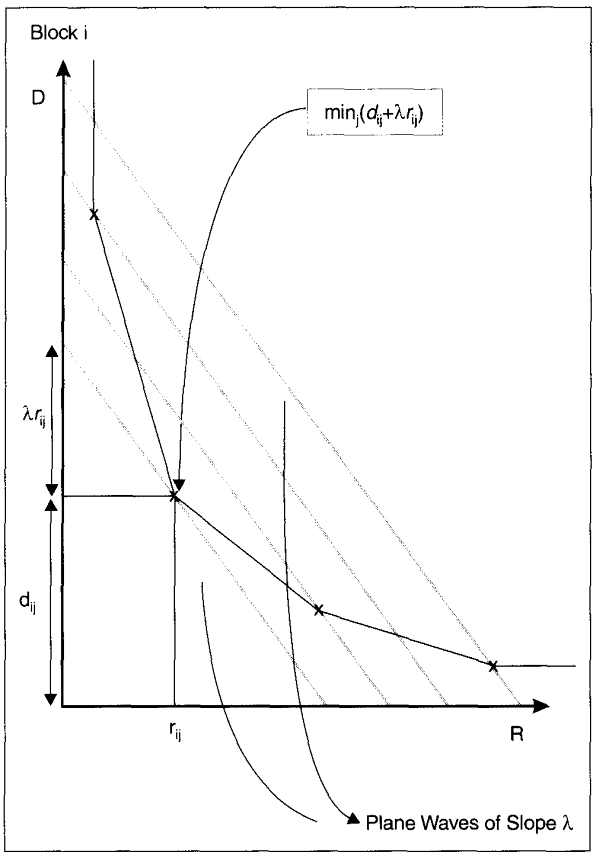
Independent Problems

We consider first the case where the rate r_{ij} and distortion d_{ij} can be measured independently for each coding unit; i.e., the R-D data for coding unit i can be computed without requiring that other coding units be encoded as well. One example of this scenario is the allocation of bits to different blocks in a DCT image coder such as that of Box 2 where blocks are individually quantized and entropy coded (i.e., inter-block prediction is used). Another example would be the allocation of bits to video frames encoded by MPEG in INTRA-only mode, or using motion JPEG.

It is also useful to note that scenarios that involve any prediction or context-based coding are by nature "dependent" but can sometimes be approximated using "independent" allocation strategies with little performance loss in practice [49] (see the "Dependency Problems" section for a more detailed description of dependent allocation problems.) Even if making the independence approximation results in performance loss, the dependency effects are often ignored to speed up the computation. For example, it is common to consider the allocation of bits to frames in a video sequence as if these could be treated independently; however, due to the motion estimation loop, the bit allocation for one frame has the potential to affect subsequent frames.

Lagrangian Optimization

The classical solution for the problem of Formulation 3 is based on the discrete version of Lagrangian optimization



▲ 14. *For each coding unit, minimizing $d_{ix(i)} + \lambda r_{ix(i)}$ for a given λ is equivalent to finding the point in the R-D characteristic that is "hit" first by a "plane wave" of slope λ.*

first introduced by Everett [64]. This approach was first used in a source coding application, following the framework we describe here, by Shoham and Gersho [65] and by Chou, Lookabaugh and Gray [17, 30] in tree pruning and entropy-constrained allocation problems. Since then this approach has been used by numerous authors [42, 45, 46, 50, 66-68].

The basic idea of this technique is as follows. Introduce a Lagrange multiplier $\lambda \geq 0$, a non-negative real number, and let us consider the Lagrangian cost $J_{ij}(\lambda) = d_{ij} + \lambda \cdot r_{ij}$. Refer to Fig. 14 for a graphical interpretation of the Lagrangian cost. As the quantization index j increases (i.e., the rate decreases and the distortion increases) we have a trade-off between rate and distortion. The Lagrange multiplier allows us to select specific trade-off points. Minimizing the Lagrangian cost $d_{ij} + \lambda \cdot r_{ij}$ when $\lambda = 0$, is equivalent to minimizing the distortion; i.e., it selects the point closer to the y-axis in Fig. 14. Conversely, minimizing the Lagrangian cost when λ becomes arbitrarily large is equivalent to minimizing the rate, and thus finding the point closest to the x-axis in Fig. 14. Intermediate values of λ determine intermediate operating points.

Then the main result states that

Theorem 1 [64, 65] *If the mapping $x^*(i)$ for $i = 1,2,\ldots, N$, minimizes:*

$$\sum_{i=1}^{N} d_{ix(i)} + \lambda \cdot r_{ix(i)}, \tag{2}$$

then it is also the optimal solution to the budget-constrained problem of Formulation 3, for the particular case where the total budget is:

$$R_T = R(\lambda) = \sum_{i=1}^{N} r_{ix^*(i)}, \tag{3}$$

so that:

$$D(\lambda) = \sum_{i=1}^{N} d_{ix^*(i)} \leq \sum_{i=1}^{N} d_{ix(i)}, \tag{4}$$

for any x satisfying Eq. (1) with R given by Eq. (3).

Since we have removed the budget constraint of Eq. (1), for a given operating "quality" λ, Eq. (2) can be rewritten as:

$$\min\left(\sum_{i=1}^{N} d_{ix(i)} + \lambda r_{ix(i)}\right) = \sum_{i=1}^{N} \min(d_{ix(i)} + \lambda r_{ix(i)}), \tag{5}$$

so that the minimum can be computed independently for each coding unit. Note also that for each coding unit i, the point on the R-D characteristic that minimizes $d_{ix(i)} + \lambda r_{ix(i)}$ is that point at which the line of absolute slope λ is tangent to the convex hull of the R-D characteristic (see Fig. 14). For this reason we normally refer to λ as the slope, and since λ is the same for every coding unit on the sequence, we can refer to this algorithm as a "*constant slope optimization.*"

The intuitive explanation of the algorithm is simple. By considering operating points at constant slope we are making all the coding units operate at the same marginal return for an extra bit in the rate-distortion trade-off. Thus, the MSE reduction in using one extra bit for a given coding unit would be equal to the MSE increase incurred in using one less bit for another unit (since we need to maintain the same overall budget). For this reason, there is no allocation that is more efficient *for that particular budget*. Box 6 illustrates why this approach is intuitively sound. This technique was well known in optimization problems where the cost and objective functions were continuous and differentiable. Everett's contribution [64] was to demonstrate that the Lagrangian technique could also be used for discrete optimization problems, with no loss of optimality if a solution exists with the required budget; i.e., as long as there exists a point in the convex hull that meets the required budget.

The properties of the Lagrange multiplier method are very appealing in terms of computation. Finding the best quantizer for a given λ is easy and can be done independently for each coding unit. (Note that here we are considering that the R-D data has already been computed and we are discussing the search complexity. Finding the R-D data may in itself require substantial complexity.) Still, one has to find the "right" λ in order to achieve the optimal solution at the required rate; i.e., find λ such that $R(\lambda)$ as defined above is close or equal to the prespecified budget. Finding the correct λ can be done using the bisection search [42, 65] or alternative approaches such as those proposed in [69]. Note that the number of iterations required in searching for λ can be kept low as long as we do not seek to have an *exact* match of the budget rate. Moreover, in scenarios such as video coding, where we may be performing allocations on successive frames having similar characteristics, it is possible to initialize the Lagrange multiplier for a frame with the values at which convergence was achieved for previous frames, which will again reduce the number of required iterations; i.e. providing a good initial guess of λ leads to reduced complexity.

Generalized Lagrangian Optimization

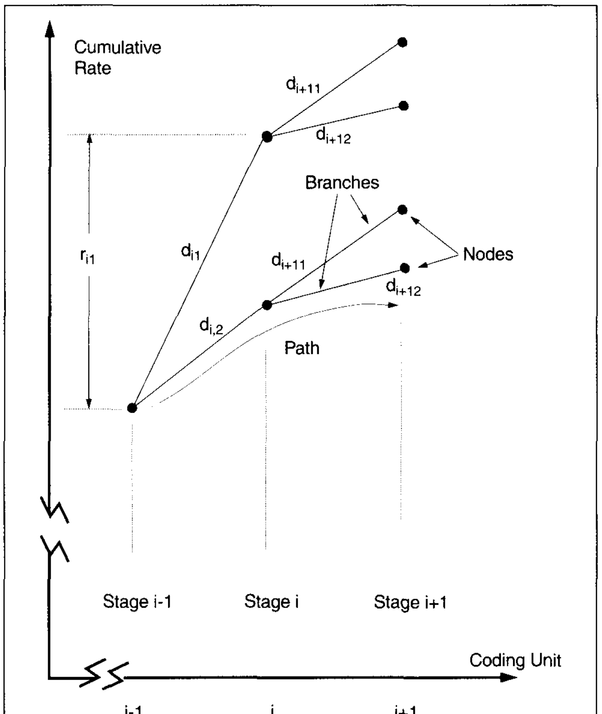
Allocation problems with multiple constraints, such as those mentioned in Formulation 4 earlier, can also be solved using Lagrangian techniques. These approaches are based on generalized Lagrangian relaxation methods [70]. The basic idea is to introduce a Lagrange multiplier for each of the constraints, which can thus be relaxed. The problem now is that the solution can be found only for the right *vector* Lagrange multiplier $\Lambda = \{\lambda_1, \ldots, \lambda_c\}$, and the search in a multidimensional space is not as straightforward as it is when a single Lagrange multiplier is used.

Typically the problems we consider involve significant structure in the constraints, and that can guide the search for the vector Lagrange multiplier [50, 67, 71-73]. For example, in some cases, these constraints are embedded—if there are N coding units, we have a series of c constraints where constraint k is a budget constraint affecting coding units 1 through n_k; i.e., constraint k limits the total rate allocation for units 1 through n_k. The other constraints likewise affect blocks 1 through $n_1, n_2, \ldots, n_c = N$, respectively. In those and similar cases, a search strategy can be derived to find the optimal vector Λ in an iterative fashion [72].

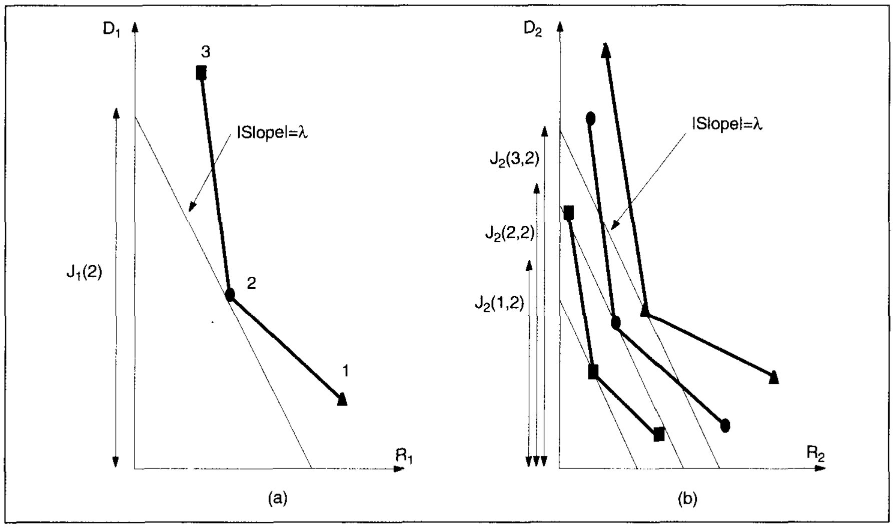
Dynamic Programming

As mentioned above, and discussed in more detail in Box 8, Lagrangian techniques have the shortcoming of not being able to reach points that do not reside on the convex hull of the R-D characteristic. An alternative formulation is then to formulate the allocation as a deterministic dynamic programming problem.

In this case, we create a tree that will represent all possible solutions. Each stage of the tree corresponds to one of the coding units j and each node of the tree at a given stage represents a possible cumulative rate usage. For example, as seen in Fig. 15, to the accumulated rate at block



▲ *15. Trellis diagram to be used for the Viterbi algorithm solution. Each branch corresponds to a quantizer choice for a given coding unit and has an associated cost, while its length along the vertical axis is proportional to the rate. For instance, quantizer 1 at stage i produces a distortion $d_{i,1}$ and requires rate $r_{i,1}$. A path will correspond to a quantizer assignment to all the coding units in the sequence.*



▲ *16. Operational R-D characteristics of two frames in a dependent coding framework, where frame 2 depends on frame 1. (a) Independent frame's R-D curve. (b) Dependent frame's R-D curves. Note how each quantizer choice for frame 1 leads to a different $R_{2\text{-}D}$ curve. The Lagrangian costs shown are $J=D+\lambda R$ for each frame.*

$i-1$ we add the rate corresponding to each possible quantization choice, thus generating new nodes with the appropriate accumulated rate. Each branch has as a cost the distortion corresponding to the particular quantizer, and therefore as we traverse the tree from the root to the leaves we can compute the accumulated distortion for each of the solutions. It should be clear that this is indeed a way of representing all possible solutions, since by traversing the tree we get successive allocations for each of the coding units.

Let us now consider what happens if two paths converge into a single node; i.e., two alternative solutions provide the same cumulative rate. It seems intuitive that the solution having higher distortion up to that point should be removed (i.e., pruned from the tree), since from that stage on both solutions have the same remaining bits to use. Those paths that are losers so far will be losers overall. This is the gist of the Optimality Principle introduced by Bellman [74-76], as it applies to this particular problem. See also Box 7 for a simple example of its applicability. This particular brand of dynamic programming (DP), which handles deterministic cost functions and helps us find the shortest (in the sense of the branch cost) path in a graph, is also known as the Viterbi algorithm [76] or Dykstra's shortest-path algorithm. In compression applications, dynamic programming is used in the encoder in the trellis coded quantizer (TCQ) [77, 78], as well as in the scalar vector quantizer (SVQ)[79]. It is also used, as will be explained in Box 9, to optimally prune trees in applications such as wavelet-packet optimization [42], or tree-structured vector quantization [30].

It will be easy to incorporate additional constraints to the tree growth so that the problems of Formulations 3 and 5 can be solved. For example, to introduce an overall budget constraint, it suffices to prune the branches that exceed the desired total rate allocation (the tree cannot grow above the "ceiling" specified by the budget constraint). Similarly, if a buffering constraint such as that of Formulation 5 is introduced, then we will need to prune out branches that exceed the maximum buffer size at a given stage [46].

The algorithm can be informally summarized as follows. At stage i, for all the surviving nodes, add the branches corresponding to all quantization choices available at that stage (the rate r_{ij} determines the end node and the distortion d_{ij} is added to the path distortion.) Prune branches that exceed the rate constraints, then, for each remaining node at stage $i+1$, keep only the lowest-cost branch.

Given the above discussion one might conclude that Lagrangian optimization is to be generally preferred given its complexity advantages. However, the Lagrangian approach does have one drawback in that only points in the convex hull of the global operational R-D characteristic can be reached. This is not a problem if the convex hull is "sufficiently dense"; however, in some applications it may result in significant suboptimality. See also Box 8 for an example of this scenario.

Dependency Problems

So far we have assumed that selection of the coding mode can be made independently for each coding unit without affecting the other units. There exist, however, scenarios where this assumption is no longer a valid one.

This is typically the case in coding schemes based on prediction [57, 80]. For example, assume that each coding unit i is predicted from the preceding coding unit $i-1$. The predictor is constructed using the past quantized data, and thus we code $X_i - P(\hat{X}_{i-1})$; i.e., the prediction error. As we use quantized data, the prediction error and thus the admissible R-D operating points for i depend on our choice of quantizer for $i-1$. Each choice $x(i-1)$ results in a different characteristic.

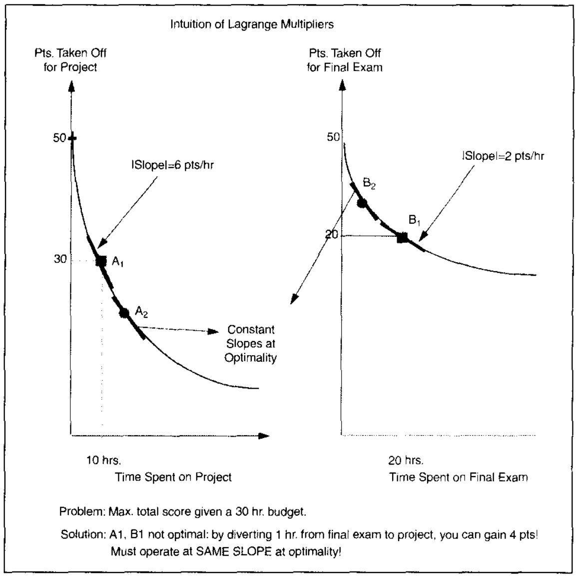
One example of this scenario is illustrated by Fig. 16, where we depict all the available R-D choices for two video frames where each frame can be coded using three different quantization settings, and where frame 2 is predicted from frame 1 (note that there are nine possible choices for frame 2, since the choices for frame 1 affect the resulting R-D values for frame 2). It should be noted that an algorithm that considers the two frames independently would select (for the given slope λ) quantizer 2 for both

Box 6 - Example of Resource Allocation Using Lagrangian Optimization

We show the generality and intuition of Lagrangian optimization by showing an example that is well outside the scope of image coding or even engineering. This will hopefully highlight the general applicability of this class of techniques even to general problems involving resource allocation that we are faced with in our day-to-day lives.

Let us take the example of Bob Smart, a physics freshman at a U.S. four-year college. It is three weeks before Finals Week, and Bob, like all motivated freshmen, is taking five demanding courses, one of which is Physics 101. Bob wakes up to the realization that his grade for the course will be based on (i) a term project that he has not started on yet, and (ii) a final exam. Each of these two components is worth 50% of the grade for the course. As Bob has spent a few more hours on his extra-curricular activities than he probably should have, and he has to devote a considerable amount of his time to the other courses as well, he realizes that he has to budget his study time for Physics very carefully. Suppose he is able to project his expected performance on both the project and the final exam based on how much time he devotes to them, and further he can quantify them using the curves shown in Fig. 17, which measure his expected deviation from perfection (50 points for each component) versus the amount of time spent on the component. After carefully surveying his situation, Bob realizes that he can spare a maximum of 30 hours total between both the project and the final exam. The question is: how does he allocate his time optimally in order to maximize his score in the course?

One option would be for him to devote 10 hours to the project (Bob was never big on projects!) and 20 hours to studying for the final exam. This would amount to operating on Points A_1 and B_1 in Fig. 17. Based on the trade-off that models Bob's propensity with respect to both the project and the exam, this would result in an expected score of 20 (or a deviation of 30 from 50) on the project (point A_1) and a score of 30 on the final exam (point B_1) for a net of 50 points. This does not bode well for Bob, but can he make better use of his time? The answer lies in the slopes of the trade-off curves that characterize both the project and the final exam. Operating point A_1 has an absolute slope of 6 points/hour, while operating point B_1 has a slope of only 2 points/hour. Clearly, Bob could help his cause by diverting one hour from the final exam to the project: this would increase his performance by 4 points! It is clear that he should keep stealing time from the final-exam study time and spending it on the term project until he derives the same marginal return for the next hour, minute, or second that he spends on either activity. This is exactly the operating points A_2 and B_2 on the curves that live on the same slope of the trade-off characteristics.



▲ *17. Illustration of Lagrangian optimization.*

This is exactly the constant-slope paradigm alluded to in the body of the text. In a compression application, the trade-offs involve rate and distortion rather than scores on exams and studying time, but the principles are the same. An important point to be emphasized in this example is that the constant-slope condition holds only under the constraint that the rate-distortion (or equivalent trade-off) curves are *independent*. In our example above, this means that we assume that the final-exam curve is independent of the project curve—something that may or may not be true in reality. If the amount of time spent on the project influences Bob Smart's preparedness for the final exam, then we have a case of "dependent coding" for the compression analogue (see the "Dependency Problems" section).

frames; i.e., it would incur a cost $J_1(2)$ for frame 1 and then, given that quantizer 2 was selected for frame 1, would choose the minimum among all $J_2(2,x)$, which turns out to be $J_2(2,2)$. However, in this particular example, the greedy approach, allocating first for frame 1 and then for frame 2, can be outperformed. The better overall performance can be achieved when quantizer 1 is used for the first frame and quantizer 2 is used for the second. Even though $J_1(2) < J_1(1)$ we have that $J_1(2) + J_2(2,2) > J_1(1) + J_2(1,2)$.

Several types of dependency scenarios can be identified. Rather than attempt to provide a complete taxonomy of all these schemes, let us consider two concrete examples within the MPEG coding framework that illustrate different forms of dependency.

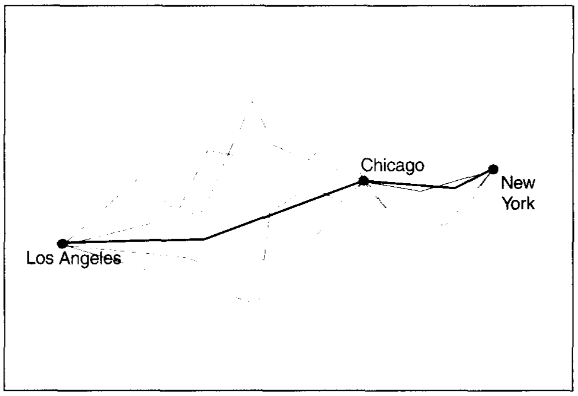
Trellis-based Dependency. The selection of macroblock-level quantization in an MPEG video stream is a dependent problem because the rate r_{ij} for macroblock i and quantizer j depends on the quantizer chosen for macroblock $i-1$. This is because predictive entropy coding of the quantization indices is used to increase the coding efficiency. In this situation it is possible to represent all the possible selections as a trellis where each state represents one quantizer selection for a given macroblock, with each stage of the trellis corresponding to one macroblock. Dynamic programming can then be

Box 7 - Example of Bellman's Optimality Principle of Dynamic Programming

The idea of dynamic programming, or Bellman's optimality principle of dynamic programming, can be captured very easily with a very simple example that illustrates the basic idea.

Suppose we are interested in finding the shortest auto route between Los Angeles and New York City (see Fig. 18). Further suppose that you know that the shortest route between L.A. and New York goes through Chicago. Then Bellman's optimality principle states the obvious fact that in this case, the Chicago to New York leg of the shortest journey from L.A. to New York will be identical to the shortest auto route between Chicago and New York; i.e. to the shortest route on a trip that starts at Chicago and ends in New York. Why is this obvious observation useful? Because it can result in a lot of computational savings in finding the best path from L.A. to New York: if we find the best path from L.A. to Chicago, then we only need to add on the shortest auto distance between Chicago and New York, if we already know the answer to that.

Sophisticated applications of this basic principle can lead to fast optimal algorithms for a variety of problems of interest. A popular incarnation of the above principle in signal processing and communications involves the omniscient Viterbi Algorithm, which uses the dynamic programming principle illustrated above in finding the "best path" through a trellis induced by a finite-state machine. The cities in the example above are analogous to the "states" in the Viterbi Algorithm at various time stages of the trellis. Recall that a state decouples the future from the past; that is, given the state, future decisions are not influenced by past ones that led to that state. Thus, if two paths merge at a system state, then the costlier of the two paths can be pruned. The analogy to the above example can be captured as follows: suppose there are two separate paths from L.A. to Chicago, one through St. Louis and the other through Denver. If the L.A.-Denver-Chicago route is longer than the L.A.-St. Louis-Chicago route, then the former can be thrown out because the best route from L.A. to New York passing through Chicago can never go through Denver. This principle is true for every state in the system and is the guiding principle behind the popular Viterbi algorithm. As described in the main body, the principle of dynamic programming finds application in a variety of scenarios in image and video coding based on rate-distortion considerations like buffer-constrained video compression (see also Box 4) and adaptive wavelet-packet based transform coding (see Box 9).



▲ *18. Illustration of dynamic programming.*

used to find the minimal cost path in this trellis, where the branch cost is typically defined as the Lagrangian cost introduced above [81-83]. As in the example of Fig. 16, taking the dependency into account avoids "greedy" selection of coding parameters, where the quantizer assignment is optimized for the current coding unit alone.

In general, trellis-based dependencies arise in cases where the underlying structure is such that the memory in the system is finite (i.e., coding choices for i depend only on a finite set of previous coding units) and the number of possible cases is also finite. In other words, in this case, the available coding parameters for a given coding unit depend on the "state" of the system—the finite set of parameters that completely determine achievable values. As in [81-83], for these types of dependencies one can use a dynamic programming approach, where the state corresponds to the state of the system, and branches (each corresponding to a choice of quantization) have associated a Lagrangian cost that combines the rate and distortion for the given parameter choices.

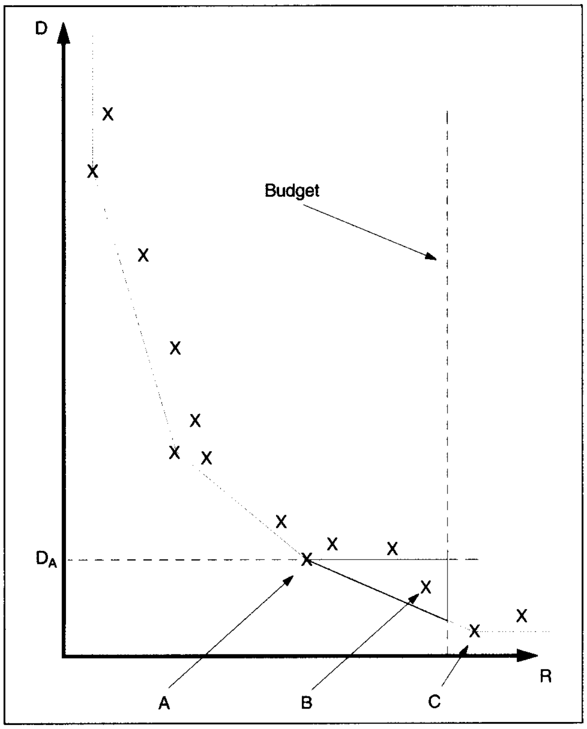
Tree-based dependency. A second example of dependency can be seen when we analyze the effect of motion compensation in an MPEG framework. After motion compensation, the encoder transmits the difference between the previously decoded frame and the current frame. This difference frame is in turn compressed and used to reconstruct the decoded version of the current frame. It is easy to see that we have a recursive prediction loop, and thus the residue frame will depend on the selection of quantization parameters for all previous frames since the last INTRA-frame [80]. In this case we can observe that all possible combinations generated by successive quantizer choices can be represented as a tree with the number of branches growing exponentially with the number of levels of dependency (i.e., number of frames since the last INTRA-frame).

The problem of dependent coding with an application to an MPEG framework is studied in [80]. The main conclusion is that exponential growth in the number of combinations makes the exact solution too complex. However it is possible to make approximations that simplify the search for the optimal solution. Good heuristics include the use of so-called monotonicity assumptions (a more finely quantized predictor typically results in smaller prediction error) [80], or greedy approaches where, for example, only a few quantization choices are kept at any given stage. The problem can also be alleviated by resorting to models of the dependent R-D characteristics so that not all the operating points in the tree need to

Box 8 - Comparison of Lagrangian and Dynamic Programming Approaches

The basic difference between the Lagrangian and dynamic programming (DP) approaches is that the Lagrangian approach is limited to select only operating points on the convex hull of the *overall* R-D characteristic. No such constraint affects the DP approach.

This can be easily seen with the example of Fig. 19, which represents the combined R-D characteristic for a set of coding units. The figure shows an instance of the problem of Formulation 3 where rate has to be allocated to coding units in order to meet a budget constraint. In Fig. 19 the operating point C exceeds the budget and thus is not an admissible solution. The nearest convex hull point (A) has higher distortion than B. Therefore B would be the optimal solution for the problem at hand. In fact, any points in the shaded area would be better than A. However, none of them is reachable using Lagrangian techniques, as these points are located "over" the convex hull. However, these points would be reachable through dynamic programming.



▲ *19. Comparison between Lagrangian optimization and dynamic programming.*

Note that this situation comes about because we are dealing with a discrete allocation and therefore the set of achievable points on the convex hull is also discrete (the example of Box 6 assumes a continuous range of choices and thus the Lagrangian approach is indeed optimal in that case.) In many instances, in particular when the convex hull is densely populated, this situation is less likely to arise or in any case the gap between the best Lagrangian solution and the optimal solution may be small. Exceptions include scenarios where the number of coding units is small and the convex hull is sparse, as for example in the coding for scalar vector quantization (SVQ) [79]. When that is the case, the performance gap may be larger and using DP all the more important. However it may also be possible to use a Lagrangian solution to initialize the DP search (see, for example, [84]).

In terms of complexity, the Lagrangian approach is preferable, since it can be run independently in each coding unit, whereas DP requires a tree to be grown. The complexity of the DP approaches can grow exponentially with the number of coding units considered, while the Lagrangian approach's complexity will only grow linearly. Thus, in many situations the Lagrangian approach is a sufficiently good approximation once computation complexity has been taken into account.

We also refer the reader to examples given in the webpage at http://sipi.usc.edu/~ortega/RD_Examples/, which demonstrates both the Lagrangian and DP techniques at work. One of the examples shows how for different values of λ, different points are achieved for each of the coding units and in the overall allocation. We demonstrate how the operating points can be searched with the bisection algorithm until the desired operating point is reached [42, 65]. A second example demonstrates the operation of the DP algorithm where a tree is grown until the minimum cost path that meets the constraint is found.

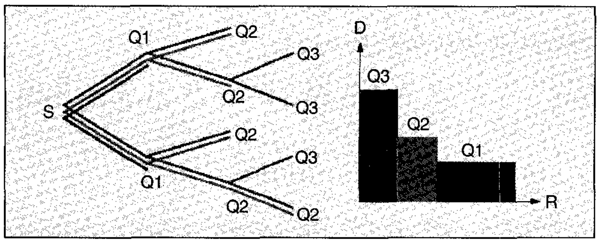
be explicitly computed [48, 57], or by considering models of the rate [47, 60] and assuming that the quantization scale provides a good estimate of quality.

Application to Basic Components in Image/Video Coding Algorithms

In what follows we briefly outline scenarios where variations of the generic formulations described in the previous section have been found to be useful. While the formulations and algorithms are similar, there are key differences that would make the discussion too long; we therefore limit ourselves to providing an overview of the applications with pointers to relevant work in the various areas. We structure our discussion along the lines of the two classes of problems we introduced, namely, budget and delay constrained allocations, and refer the reader to Boxes 2, 3, 4.

Budget Constraint Problems

Let us now visit a few applications involving budget-constrained optimization in the context of image and video coding. Due to lack of space, we will omit comprehensive coverage of the image and video coding frameworks and algorithms, referring the reader instead to a number of both old and recent textbooks on the subject [1, 5, 31-37]. We will dwell only and briefly on transform coding, which was introduced earlier in the context of DCT and wavelet-based coding in Boxes 2 and 3, respectively, and refer the reader to textbooks that treat this topic [85-88].



▲ *20. Illustration of joint source-channel coding for multicast scenario.*

The budget-constrained optimization problem in source coding is to minimize the quantization distortion subject to a bit-rate constraint. The most general formulation in the context of transform coding involves selecting the operating point involving the *combination* of transform, quantizer, and entropy coder in order to realize the best rate-distortion trade-off. Depending on the flexibility (and complexity concerns) of the framework, one or all of the above functional components can be jointly optimized. Typically, the transform is held fixed (e.g. based on DCT or a discrete wavelet transform) and the quantization and entropy coder are jointly optimized. The quantization modes can vary from simple scalar quantizers to rather sophisticated vector quantizers, but by abstracting these quantization modes as constituting a *discrete set of quantizers*, a host of different frameworks can be considered under a common conceptual umbrella.

Fixed-Transform-Based R-D Optimization

A good example of a fixed-transform-based application involves syntax-constrained optimization of image and video coding standards like JPEG, where the quantizer choice (8×8 quantizer matrix for the image) and the entropy coding choice (Huffman table) can be optimized on a per-image and per-compression-ratio basis (see Box 2). The spectrum of general applications for optimizing the quantizer and entropy coding choice can range from the selection of completely different quantizers (or codebooks in a vector quantization scheme) to simply scaling a quantizer by a scaling factor, as is usually done by users of JPEG.

Another recently popular haven for R-D-based techniques is wavelet-based image coding, where several state-of-the-art coding algorithms realize their high performance gains by using a variety of sophisticated rate-distortion based optimization techniques. A partial list of coding algorithms that derive their gains from rate-distortion optimization includes forward-adaptive frameworks involving subband classification criteria [23, 89], space-frequency quantization (SFQ) that R-D optimizes the popular wavelet zerotree framework [22], as well as backward-adaptive frameworks such as the estimation-quantization (EQ) framework [24] and backward adaptive quantization [25].

Similarly, optimization techniques can be applied with impressive performance gains for video coding frameworks such as MPEG and H.263. There is a considerable amount of spatio-temporal redundancy in typical video data, and it is particularly important to pay attention to the temporal dimension, which has the bulk of the inherent redundancy for typical video sequences.

R-D based techniques for variable-size motion compensation can be found in [68], while variable bit-rate motion-vector encoding based on DPCM can be found in [69, 90-92]. Other examples of applications of R-D techniques to video coding can be found in [51, 93-95]. A detailed account of the impact of these techniques on state-of-the-art video coders, concentrating specifically on the motion-related issues, can be found in another article in this issue [2].

Further, rate-distortion optimization techniques can be applied to shape coding, where trade-offs between the fidelity of the shape representation versus the bit rate needed to represent the shape can be optimized [1, 96, 97]. These techniques are likely to be used in newer standards, such as MPEG-4, which introduce support for video objects, rather use the video frame as their basic video unit.

Adaptive Transform-Based R-D Optimization

While coding algorithms that use a fixed transformation can be useful if the class of signals is well suited in some sense (e.g. in time-frequency characterization) to the fixed transform, this may not be adequate for dealing with arbitrary classes of signals with either unknown or time-varying characteristics. For example, for images or image segments having high-frequency stationary components, the wavelet transform is a bad fit. This motivates us to consider a more powerful adaptive framework that can be robust when dealing with a large class of signals of either unknown or more typically, time (or space-)varying characteristics. In this approach, the goal is to make the transformation signal-adaptive. See Boxes 3 and 9 for applications involving wavelet packets, which represent generalizations of the wavelet transform.

The idea behind adaptive-transform frameworks is to replace the fixed transform with a large library of transforms that can additionally be searched efficiently. The library of transforms can be fairly general and can include for example, the family of quadtree spatial segmentations [68, 83, 98] or variable block-size DCTs (e.g. 4×4, 8×8, 16×16 blocks). Similarly, one can take the standard wavelet decomposition and customize its various parameters (filters, tree-structure, including the number of levels of decomposition) to a particular image [42] or to parts of an image [99]. These techniques have been shown to provide substantial gains over nonadaptive decomposition techniques [100, 101, 103]. Examples such as the family of wavelet packet transforms of Box 9 are illustrations of the joint optimization of the transform, quantizer, and entropy-coding choice in an image-adaptive rate-distortion sense. Many of the sophisticated wavelet-based frameworks described in the previous subsection can be extended to their adap-

Box 9 - Adaptive Transforms Based on Wavelet Expansions

We continue the thread from Box 3 in our quest for designing adaptive transforms based on wavelet expansions that are R-D optimized.

Let us first address the cost function, namely the R-D function. Returning to the problem of jointly finding the best combination of wavelet packet (WP) transform (or basis) and the quantization and entropy-coding choices, we assume that arbitrary (finite) quantization choices are assumed available to quantize the WP coefficients in each tree node (see Figs. 9 and 21), with both rate (R) and distortion (D) being assumed to be additive cost metrics over the WP tree: i.e., $R(tree) = \sum R(leaf\,nodes)$; and $D(tree) = \sum D(leaf\,nodes)$. As an example, the commonly used first-order entropy and MSE measures for R and D satisfy this additivity condition.

Turning now to the fast-search problem, one possible approach to finding the best tree is the "greedy tree growing" algorithm, which starts at the root and divides each signal in two if it is profitable to do so (if the cost of the subsignals generated is less than the cost of the signal they come from). It terminates when no more profitable splits remain. It is easy to determine that this, however, does not find the globally optimal tree, which is found by starting at the deepest level of the tree and pruning pairs of branches having higher total cost than that of their parent.

We now describe the details using a 1-D case for simplicity (see Fig. 21). The idea is to first grow the full (STFT-like) tree (see Fig. 21(a)) to full depth (or some maximum fixed depth in practice) for the whole signal. Note that due to the tree-structure of the bases, we now have available the WP coefficients corresponding to all the bases on our search list. That is, if we grow the coefficients of a depth-5 tree, we know the coefficients associated with all subtrees grown to depth-5 *or less.*

The next step is to populate each WP tree node with the minimum Lagrangian cost over all quantization choices for that tree node. This minimum cost at each node is associated with the quantizer that minimizes the rate-distortion trade-off (for a fixed "quality factor" λ):

$$J(node) = \min_{quantizer} [D(node) + \lambda R(node)].$$

Note the implication of this step—we do not yet know if an arbitrary tree node will be part of our desired optimal subtree choice, but we *do* know what quantization choice to use for that node *if* it is part of the best-basis subtree. This is particularly satisfying because it has enabled us to decouple the best quantizer/basis choice without sacrificing optimality.

We now have remaining only the unfinished business of finding the best basis. The special tree structure of the basis can be exploited in formulating a fast tree-based search strategy. The idea is to use a bottom-up recursive "split-merge" decision at each node, corresponding to whether it is costlier, in the Lagrangian sense, to keep the parent node or its children nodes. This fast dynamic programming (DP) based pruning method is also optimal because the signal subspace spanned by the parent node is the direct sum of the signal subspaces spanned by its children nodes thanks to the orthogonality of the filter bank. We now describe the details. Assume known the optimal subtree from a tree node n "onwards" to the full tree-depth $\log N$. Then, by Bellman's optimality principle of DP [74] (see also Box 7), all surviving paths passing through node n must invoke this same optimal "finishing" path. There are only two contenders for the "surviving path" at every node of the tree, the parent and its children, with the winner having the lower Lagrangian cost. That is, starting from the full tree, the leaf nodes are recursively subjected to an optimal split-merge decision, following a policy of:

Prune if: $J(parentnode) \leq [J(child1) + J(child2)]$,

tive-transform counterparts such as those based on wavelet packets or adaptive wavelet packets [102].

Delay-Constrained Allocation Problems

Box 7 introduces the problem of delay-constrained allocation. This class of problems, as also outlined in Formulation 4, is typically encountered in video transmission under delay constraints. The more traditional view of the problem is as a *buffer control* problem but, as described above and in [44, 49], the delay constraint is more general. Rate-distortion techniques have been applied to the rate control under CBR transmission conditions. For example, [46] provides an overall optimal solution using dynamic programming as well as Lagrangian based approximations. An alternative formulation is to consider the buffering constraints as a set of budget constraints as in [67]. The traditional direct-feedback mechanism used in buffer-control algorithms [104] where quantization scale is controlled by buffer fullness is replaced in [105] by a feedback mechanism that controls instead the value of the Lagrange multiplier to be used in the selection of optimal points. Both these approaches concentrated on the independent allocation case, the dependent case was considered by [80] with applications provided to MPEG coding scenarios. More recent work has also considered R-D-optimized MPEG coding using models of the R-D characteristics to reduce the complexity [48]. The rate-control problem can be formulated not only in terms of selection of quantization parameters as in the above reference but also in terms of selection of the best types of frames (I, P or B) as in [106].

A second area in rate-control research is that of control for VBR channels. Here we can consider two classes of problems. First, in some cases it is possible for the encoder to select both the source and the channel rate; where the selection of channel rate may be subject to constraints, such as, for example, the policing constraints in an ATM network. Examples of this type of optimization include [49], which employs dynamic programming techniques, and [50], which utilizes multiple budget constraints and a Lagrangian approach. Other approaches include [51, 107].

where $J(childnode)$ corresponds to the cost of the cheapest path that "goes through" the child node. Using this, we begin at the complete tree-depth $n = \log N$ and work our way toward the root of the tree, using the above split/merge criterion at each node, making sure that we record the optimal decisions along the way, until we arrive at the tree root. At this point, the best basis is known by simply backtracking our way down the tree using our recorded decisions at each tree node. In fact, both the best basis *and* the best quantization choice are now known!

Of course, this corresponds to a particular choice of λ, which was fixed during this tree-pruning operation. Unfortunately, this λ may not be the correct one: we want the one that corresponds to the target bit budget R. However, due to the convexity of the rate-distortion curve, the optimal slope λ^* matched to the desired R can be easily obtained using standard convex search techniques; e.g. the bisection method or Newton's method or other standard root-solving methods. An important point of note is that the Lagrangian method can only obtain solutions that reside on the convex-hull of the rate-distortion curve, and, thus, a target rate whose optimal operating point is not on the convex hull will be approximated by the nearest convex-hull rate. In practice, for most practical coding applications, the convex hull of the R-D curve is dense enough that this approximation is almost exact.

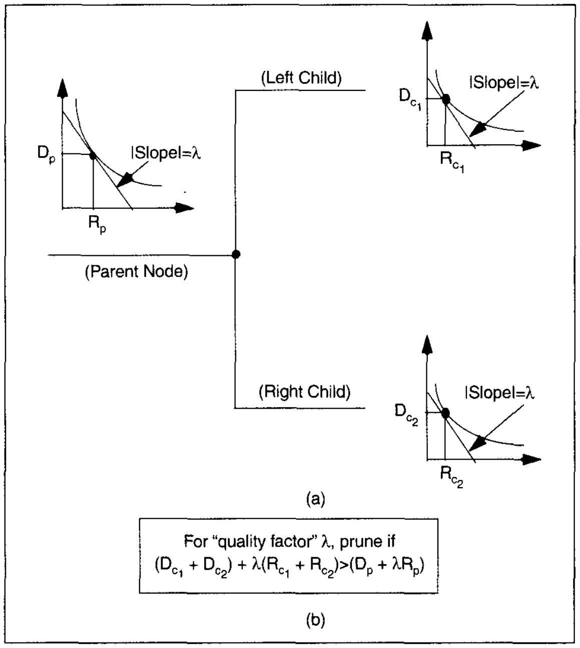
We will now summarize the single-tree algorithm:

▲ Grow a full-balanced (STFT-like) tree to some desired fixed depth (i.e., find all the WP coefficients associated with all bases in the library);

▲ For a fixed λ, populate each node of the full tree with the best Lagrangian cost $D + \lambda R$ over all quantizer choices (i.e., find the best quantizer choice for each node);

▲ Prune the full tree recursively, starting from the leaf nodes (i.e., find the best-basis subtree);

▲ Iterate over λ using a convex search method to meet the target bit rate (i.e., match the best subtree/quantizer choice to the desired bit budget).



▲ *21. The single-tree algorithm finds the best tree-structured wavelet packet basis for a given signal. (a) The algorithm starts from the full STFT-like tree and prunes back from the leaf nodes to the root node until the best pruned subtree is obtained. (b) At each node, the split-merge decision is made according to the criterion: prune if* $J(parentnode) \le [J(child1) + J(child2)]$

The second class of problems is that where the channel rate is subject to random variations or can vary from link to link in a networked transmission. In [107] R-D methods are provided to reduce the bit rate of encoded data without requiring that it be decoded and recompressed. In [52, 63, 108, 109] approaches based on dynamic programming and Lagrangian optimization are presented to address the problems of transmission in burst-error channels such as those encountered in a wireless-transmission environment [110].

The Role of R-D Techniques in Joint Source-Channel Coding

We have thus far focused (with the exception of the previous paragraph) on rate-distortion methods for source coding when dealing with image and video sources. We now briefly address the problem of the applicability of such methods for the bigger problem of image and video transmission, specifically in the context of joint source-channel coding. Box 5 highlights the essence of the problem. Here, we take a look at some of the applications driving these techniques and provide some pointers to recent activities in the field.

Background and Problem Formulation

With the explosion in applications involving image and video communication, such as those afforded by the boom in multimedia- and Internet-driven applications, as well as those afforded by emerging applications like cable modems and wireless services, the image communication problem has recently assumed heightened interest and importance, as visual data represents by far the largest percentage of multimedia traffic.

A natural question to ask is: why do we need to re-invent data communications just because of the current multimedia explosion? There are several reasons to revisit the existing paradigms and systems. The primary one is that current communication link designs are primarily mismatched for image and video sources as they fail to account for important source considerations such as **(i)** highly time-varying source and channel characteristics, **(ii)** high source tolerance to channel loss, and **(iii)** unequal importance of transmitted bits. This comes from a long history of data communications, where loss of bits is disastrous (e.g. data files), and where every bit is equally sacred. Some relevant important attributes of the image/video source are summarized below:

▲ The performance metric is the delivered visual quality (e.g. mean-squared-error or more correctly, the perceptual distortion) of the source due to both source quantization *and* channel distortion under constraints of fixed-system resources such as bandwidth and transmission energy. This contrasts with commonly used performance criteria, such as bit-error rates, that are appropriate for traditional data communications.

▲ The unequal error sensitivities of a typical video bitstream (e.g., bits representing motion vectors or synchronization/header information versus bits representing high-frequency motion-compensated error residue or detail in textured image areas) emphasizes the desirability of a layered approach to both source and channel coding and calls for a rehauling of conventional "single-resolution" digital transmission frameworks with systems that have a multiresolution character to them.

▲ Due to the stringent delay requirements of synchronous video applications, there is a need to include finite buffer constraints (efficient rate-control strategies). These requirements will also influence the choice of error control coding strategies such as forward error correction (FEC) vs. automatic repeat request (ARQ) techniques, as well as more powerful hybrid FEC/ARQ choices [111].

In Box 5 we motivated the need for joint source-channel coding due to the practical shortcomings of the separation principle (see Fig. 13) as well as its theoretical inapplicability to a number of multiuser communication scenarios of interest such as broadcast [53] and multicast. There is thus the potential for performance gains if there is closer interaction between the source- and channel-coding functions. The understanding of the superiority of a joint approach to source and channel coding in such cases has recently initiated numerous research activities in this area, a partial list of which can be found among [112-114]. Examples of successful deployment of joint source-channel coding principles for multiuser communications frameworks like broadcast and multicast can be found in [115-116] and [117] respectively.

Overview of Applications and Methodologies

Image and video transmission problems of the kind formulated above come in various application-driven flavors. Of particular interest are image and video delivery over heterogeneous packet networks such as the Internet and ATM, as well as wireless video for point-to-point as well as broadcast/multicast scenarios. A key challenge involving video sources involves the stringent synchronous-delay requirements. For networking applications there are constraints imposed by the network related to both the average and peak burst rates. The available channel capacity can fluctuate quite a bit also, such as due to network congestion in networking applications, or fading in wireless-communication applications.

Joint source-channel coding schemes studied in the literature have been historically driven by two high-level ideologies. Very crudely, these ideologies may be classified as being inspired by "digital" versus "analog" transmission methods.

The digital class of techniques is based on optimally allocating bits between digital source and channel codes. Source-coding bits correspond to a digitally compressed and entropy-coded stream. Channel-coding bits correspond to the parity information of a digital error-correction code. A popular rate-distortion-based approach to this digitally inspired source-channel coding paradigm consists of minimizing the expected end-to-end source distortion (due to both source quantization *and* channel transmission impairments) subject to a total rate on both source coding *and* channel coding. This boils down to an allocation problem not only among source coding elements but also *between source coding and channel coding elements*. Extensions of the Lagrangian method described earlier can be invoked here, with the twist that the trade-offs involve expected distortion versus total rate, due to the presence of the channel coder. Due to the typically unequally important nature of source bit layers when dealing with image and video sources, as pointed out earlier, these layers are matched with unequal levels of channel protection. This comes under the category of unequal error protection (UEP) channel codes. One of the most popular classes of deployed UEP channel codes is the family of rate-compatible punctured convolutional (RCPC) codes [118] that is promising for a number of recent applications involving layered video coding and streaming for Internet and other applications. The joint source-channel coding problem becomes one of optimally matching the resolution "trees" for both the source and the channel coders in a rate-distortion sense. A number of researchers have contributed significantly to this class of algorithms: a summary of this is provided in reference [119]. An example of a modulation-domain-based UEP scheme is described in [116], which has been recently considered for European digital audio and video broadcast [120]. Each layer of different error protection corresponds to a specific type of the receiving monitor (typically, there are three layers or resolutions) and has different bit-error-rate requirements. Thus, the quality of

Even if making the independence approximation results in performance loss, the dependency effects are often ignored to speed up the computation.

the received video varies gracefully with the receiver type as well as its distance from the transmitter.

It should be noted that this "digital" ideology, while allowing higher source compression because of entropy coding, can also lead to increased risk of error propagation. The popular solution is to insert periodic resynchronization capabilities using packetization. The resulting synchronization and packetization overheads that are needed to increase error-resilience obviously eat into the compression efficiency. The problem becomes one of optimizing this balance.

The other ideology has been inspired essentially by the "graceful degradation" philosophy reminiscent of analog transmission. Thus, while the single-resolution digital philosophy adopts an "all or nothing" approach (within the packetization or layering operation) resulting in the well-known "cliff effect," the analog-inspired approach carries a "bend but do not break" motto. The idea is to do intelligent mappings of source codewords into channel constellation points, so as to have a similarity mapping between "distances" in the source-coding domain and "distances" in the channel-modulation domain [113, 121-124]. Thus, large source distortions are effectively mapped to high noise immunity (i.e., to low probability error events, and vice versa) with intelligently chosen index assignments. The advantages of such an approach are increased robustness and graceful degradation. The disadvantage is the lack of a guaranteed quality of service (there is no notion of "perfect" noise immunity).

It is interesting to note also that hybrid versions of these two philosophies that are aimed at exploiting the "best of both worlds" have been advocated recently [125] with significant performance gains.

If the modem is included in the optimization box, then the standard rate-distortion problem becomes transformed into a power-distortion trade-off problem (where the constraint now becomes the transmission power or energy rather than the bit rate). This leads to interesting extensions of well-known rate-distortion-based optimization algorithms to their power-distortion counterparts [113, 123]. The reader is referred to [119] for a more detailed historical perspective of joint source-channel coding of images and video sources.

An example of an area where joint source-channel coding ideas have had an impact is in communicating over heterogeneous networks. In particular, the case of multicast in a heterogeneous environment is well suited for multiresolution source and channel coding. The idea is very simple: give each user the best possible quality by deploying a flexible networking infrastructure that will reach each user at its target bit rate. More precisely, a multicast transmission can be conceptualized as living on a multiresolution tree, which is a set of trees carrying various resolutions. Each user then reaches as many levels of the multiresolution tree as is possible given its access capabilities. Such a scheme was proposed in [117] for a heterogeneous packet environment, as for example the Internet. Figure 20 succinctly captures the basic idea.

While currently mostly wired links are involved, it is clear that mobile components are becoming more and more important as well. Such a scheme would be suitable for such an environment as well, possibly with bridges between wired and wireless components.

Summary

In this article we have provided an overview of rate-distortion optimization techniques as they are used in practical image and video applications. We started by establishing the link between these techniques and the rate-distortion theory developed in the field of information theory. We motivate that standard-based image/video coding can benefit from optimization techniques as it allows the encoder to optimize the selection of its coding parameters, while preserving decoder compatibility. We then defined a generic resource allocation problem and gave two concrete examples: namely, budget-constrained allocation and delay-constrained allocation. We explained in detail the Lagrangian optimization and dynamic programming techniques, which have become essential tools to solve these allocation problems. This allowed us to give an overview of applications where rate-distortion optimization has proven to be useful. We ended by describing how these techniques can also be found to be useful within joint source-channel coding frameworks.

Acknowledgments

We thank the anonymous reviewers, in particular reviewer 1, and Dr. Jong-Won Kim of USC, for their careful reading of the manuscript and their constructive comments. We also would like to thank Raghavendra Singh and Arif Karu from USC for their implementation of the example webpage referred in Box 8.

Ortega's work was supported in part by the National Science Foundation under grant MIP-9502227 (CAREER), and by the Integrated Media Systems Center (a National Science Foundation Engineering Research Center), the Annenberg Center for Communication at the University of Southern California, and the California Trade and Commerce Agency. Ramchandran's work was supported in part by the National Science Foundation under grant MIP 97-03181 (CAREER), the Army Research Office under award DAAH04-96-1-0342, and the Office of Naval Research under award N00014-97-1-0864.

Antonio Ortega is an Assistant Professor with the University of Southern California's Integrated Media Systems Center in the Department of Electrical Eng.-Systems in Los Angeles, California, USA (e-mail:ortega@sipi.usc.edu). Kannan Ramchandran is an Assistant Professor with the Beckman Institute of the Department of Electrical and Computer Engineering at the University of Illinois, Urbana-Champaign, Illinois, USA (e-mail:kannan@ifp.uiuc.edu).

References

1. G.M. Schuster and A.K. Katsaggelos, *Rate-Distortion Based Video Compression, Optimal Video Frame Compression and Object Boundary Encoding*. Kluwer Academic Publishers, 1997.

2. G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," this issue, pp. 74-90.

3. C.E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. Journal*, vol. 27, pp. 379- 423, 1948.

4. C.E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE National Convention Record, Part 4*, pp. 142-163, 1959.

5. W. Pennebaker and J. Mitchell, *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, 1994.

6. D. Lee, "New work item proposal: JPEG 2000 image coding system." ISO/IEC JTC1/SC29/WG1 N390, 1996.

7. Testing Ad Hoc Group, "JPEG2000 testing results." ISO/IEC JTC1J/SC29/WG1/N705, Nov. 1997.

8. N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proc. of the IEEE*, Oct. 1993.

9. T. Berger, *Rate-Distortion Theory. A Mathematical Theory for Data Compression*. Prentice-Hall, 1971.

10. R.M. Gray, *Source Coding Theory*. Kluwer Academic Publishers, 1990.

11. W. Bennett, "Spectra of quantized signals," *Bell Sys.Tech.J.*, vol. 27, pp. 446-472, Jul. 1948.

12. P. Zador, *Development and Evaluation of Procedures for Quantizing Multivariate Distributions*. PhD thesis, Stanford University, Stanford, CA, 1964.

13. A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. on Info. Th.*, vol. IT-23, pp. 373-380, Jul. 1979.

14. D.L. Neuhoff, "The other asymptotic theory of source coding," in *Joint IEEE/DIMACS Workshop on Coding and Quantization*, (Rutgers University, Piscataway, NJ), Oct 1992. Also available as ftp://ftp.eecs.umich.edu/people/neuhoff/OthAsymptThy.ps.

15. T.M. Cover and J.A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

16. N. Farvardin and J.W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources, *IEEE Trans. on Info. Th.*, vol. IT-30, pp. 485-497, May 1984.

17. P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. ASSP*, vol. 37, pp. 31-42, Jan. 1989.

18. M. Effros, "Optimal modeling for complex system design," this issue, pp. 51-73.

19. P.H. Westerink, J. Biemond, and D.E. Boekee, "An optimal bit allocation algorithm for sub-band coding," *Proc. of ICASSP*, pp. 757-760, 1988.

20. J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Proc.*, vol. 41, pp. 3445-3462, Dec. 1993.

21. A. Said and W.A. Pearlman, "A new fast and efficient image coder based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, pp. 243-250, June 1996.

22. Z. Xiong, K. Ramchandran, and M.T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. on Image Proc., vol. 6, pp. 677-693, May 1997.*

23. R.L. Joshi, H. Jafarkhani, J.H. Kasner, T.R. Fischer, N. Farvardin, M.W. Marcellin, and R.H. Bamberger, "Comparison of different methods of classification in subband coding of images," *IEEE Trans. on Image Processing*, vol. 6, pp. 1473-1486, Nov. 1997.

24. S. LoPresto, K. Ramchandran, and M.T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Data Compression Conference '97*, (Snowbird, Utah), pp. 221-230, 1997.

25. Y. Yoo, A. Ortega, and B. Yu, "Adaptive quantization of image subbands with efficient overhead rate selection," in *Proc. of the Intl. Conf. on Image Proc., ICIP'96*, vol. 2, (Lausanne, Switzerland), pp. 361-364, Sept. 1996.

26. C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," in *Data Compression Conference '97*, (Snowbird, Utah), pp. 241-250, 1997.

27. S.P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Info. Th.*, vol. IT-28, pp. 127-135, Mar. 1982. Also, Unpublished Bell Labs Technical Report, 1957.

28. J. Max, "Quantizing for minimum distortion," *IEEE Trans. on Info. Th.*, pp. 7-12, Mar. 1960.

29. Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Comm.*, vol. COM-28, pp. 84-95, Jan. 1980.

30. P.A. Chou, T. Lookabaugh, and R.M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. on Inform. Theory*, vol. 35, pp. 299-315, Mar. 1989.

31. A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

32. J. Mitchell, W. Pennebaker, C.E. Fogg, and D.J. LeGall, *MPEG Video Compression Standard*. New York: Chapman and Hall, 1997.

33. A.N. Netravali and B.G. Haskell, *Digital Pictures: Representation, Compression and Standards,* New York: Plenum Press, 2nd ed., 1995.

34. R.J. Clarke, *Digital Compression of Still Images and Video*. Academic Press, 1995.

35. V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer Academic Publishers, 1996.

36. K.R. Rao and J.J. Hwang, *Technique & Standards for Image, Video & Audio Coding*. Prentice Hall, 1996.

37. B.G. Haskell, A. Puri, and A.N. Netravali, *Digital Video: An Introduction to MPEG-2*. Chapman and Hall, 1997.

38. ITU-T (formerly CCITT), "Video coding for low bitrate communication," ITU-T Recommendation H.263; version 1, Nov. 1995, version 2, Jan. 1998.

39. S. Wu and A. Gersho, "Rate-constrained picture-adaptive quantization for JPEG baseline coders" in *Proc. IEEE International Conference Acoustics, Speech and Signal Processing, ICASSP'93*, vol. 5, (Minneapolis, MN), pp. 389-392, April 1993.

40. K. Ramchandran and M. Vetterli, "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Trans. on Image Proc.*, vol. 3, pp. 700-704, Sept. 1994.

41. M. Crouse and K. Ramchandran, "Joint thresholding and quantizer selection for transform image-coding: Entropy-constrained analysis and applications to baseline JPEG," *IEEE Transactions on Image Processing*, vol. 6, pp. 285-297, February 1997.

42. K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. on Image Proc.*, vol. 2, pp. 160-175, Apr. 1993.

43. M.W. Garrett, "Statistical analysis of a long trace of VBR coded video." Ph.D. Thesis, Chapter IV, Columbia University, 1993.

44. A.R. Reibman and B.G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. on CAS for video tech.*, vol. 2, pp. 361-372, Dec. 1992.

45. S.-W. Wu and A. Gersho, "Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV," *IEEE Trans. on Circuits and Sys. for Video Tech.*, vol. 1, pp. 100-112, Mar. 1991.

46. A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximation," *IEEE Trans. on Image Proc.*, vol. 3, pp. 26-40, Jan. 1994.

47. W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 6, pp. 12-20, Feb 1996.

48. L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 8, pp. 446-459, Aug. 1998.

49. C.-Y. Hsu, A. Ortega, and A. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 1016-1028, Aug. 1997.

50. J.-J. Chen and D.W. Lin, "Optimal bit allocation for coding of video signals over ATM networks" *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 1002-1015, Aug. 1997.

51. W. Ding, "Joint encoder and channel rate control of VBR video over ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 266-278, April 1997.

52. C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE J. on Sel. Areas in Comm.*, 1998. to appear.

53. T. Cover, "Broadcast channels," *IEEE Trans. on Inform. Theory*, vol. IT-18, pp. 2-14, Jan. 1972.

54. G.M. Schuster and A.K. Katsaggelos, "The minimun-average and minimum-maximum criteria in lossy compression," *Vistas in Astronomy*, vol. 41, no.3, pp. 427-437, 1997.

55. D.T. Hoang, *Fast and Efficient Algorithms for Text and Video Compression*. PhD thesis, Brown University, 1997.

56. D.T. Hoang, E.L. Linzer, and J.S. Vitter, "Lexicographic bit allocation for MPEG video," *Journal of Visual Communication and Image Representation*, vol. 8, Dec. 1997.

57. K.M. Uz, J.M. Shapiro, and M. Czigler, "Optimal bit allocation in the presence of quantizer feedback," in *Proc. of ICASSP'93*, vol. V, (Minneapolis, MN), pp. 385-388, Apr. 1993.

58. E. Frimout, J. Biemond, and R.L. Lagendijk, "Forward rate control for MPEG recording," in Proc. of SPIE Visual Communications and Image Processing '93, (Cambridge, MA), pp. 184-194, Nov. 1993.

59. J. Katto and M. Ohta, "Mathematical analysis of MPEG compression capability and its application to rate control," in *Proc. of ICIP'95*, vol. II, (Washington, D.C.), pp. 555-559, 1995.

60. H.-M. Hang and J.-J. Chen, "Source model for transform video coder and its application," *IEEE Trans. on CAS for Video Tech.*, vol. 7, pp. 287-311, Apr. 1997.

61. G.M. Schuster and A.K. Katsaggelos, "The min-max approach in video coding," in *Proc. of 1997 Intl. Conf. Acoust., Speech and Signal Processing, ICASSP'97*, vol. IV, (Munich, Germany), pp. 3105-3108, Apr. 1997.

62. A. Ortega and M. Khansari, "Rate control for video coding over variable bit rate channels with applications to wireless transmission," in *Proc. of the 2nd. Intl. Conf on Image Proc., ICIP'95* (Washington, D.C.), Oct. 1995.

63. C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over wireless channels," in *Proc. of Visual Comm. and Image Proc., VCIP'97* (San Jose, CA), Feb. 1997.

64. H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399-417, 1963.

65. Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. ASSP*, vol. 36, pp. 1445-1453, Sep. 1988.

66. S.-Z. Kiang, R.L. Baker, G.J. Sullivan, and C.-Y. Chiu, "Recursive optimal pruning with applications to tree structured vector quantizers," *IEEE Trans. on Image Proc.,* vol. 1, pp. 162-169, Apr. 1992.

67. D.W. Lin, M.-H. Wang, and J.-J. Chen, "Optimal delayed-coding of video sequences subject to a buffer-size constraint," in *Proc. of SPIE Visual Communications and Image Processing '93*, (Cambridge, MA), Nov. 1993.

68. G.J. Sullivan and B.L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. on Image Proc., vol. 3, pp. 327-331, May 1994.*

69. G.M. Schuster and A.K. Katsaggelos, "An optimal quadtree-based motion estimation and motion based interpolation scheme for video compression," *IEEE Trans. on Image Proc.*, vol. 7, no. 11, pp. 1505-1523, Nov. 1998.

70. M.L. Fisher, "The Lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 27, pp. 1-18, Jan. 1981.

71. J.-J. Chen and D.W. Lin, "Optimal coding of video sequences over ATM networks," in *Proc. of the 2nd Intl. Conf. on Image Processing*, ICIP-95, vol. I, (Washington, D.C.), pp. 21-24, Oct. 1995.

72. A. Ortega, "Optimal rate allocation under multiple rate constraints," in *Proc. of the Data Compression Conference, DCC'96* (Snowbird, UT), pp. 349-358, Apr. 1996.

73. J.-J. Chen and D.W. Lin, "Optimal bit allocation for video coding under multiple constraints," in *Proc. IEEE Intl. Conf. on Image Proc.*, ICIP'96, 1996.

74. B. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

75. D.P. Bertsekas, *Dynamic Programming*. Prentice-Hall, 1987.

76. G.D. Forney, "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, pp. 268-278, March 1973.

77. M.W. Marcellin and T.B. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. on Comm.*, vol. 38, pp. 82-93, Jan. 1990.

78. T.B. Fischer and M. Wang, "Entropy-constrained trellis-coded quantization," *IEEE Trans. on Info. Th.*, vol. 38, pp. 415-426, Mar. 1992.

79. R. Laroia and N. Farvardin, "A structured fixed-rate vector quantizer derived from a variable length scalar quantizer part I: memoryless sources," *IEEE Trans. on Inform. Theory*, vol. IT-39, pp. 85 1-867, May 1993.

80. K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. on Image Proc.*, vol. 3, pp. 533-545, Sept. 1994.

81. A. Ortega and K. Ramchandran, "Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to MPEG video coders," in *Proc. of SPIE, Digital Video Compression: Algorithms & Technologies 95*, (San Jose, CA), Feb. 1995.

82. T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S.K. Mitra, "Rate-distortion optimized mode selection for very low bit-rate video coding and the emerging H.263 standard," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 6, pp. 182-190, Apr. 1996.

83. G.M. Schuster and A.K. Katsaggelos, "A video compression scheme with optimal bit allocation between displacement vector field and displaced frame difference," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 1739-1751, Dec. 1997.

84. Y. Yoo, A. Ortega, and K. Ramchandran, "A novel hybrid technique for discrete rate-distortion optimization with applications to fast codebook search for scalar vector quantization," in *Proc. of ICASSP'96* (Atlanta, GA), Apr. 1996.

85. M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice Hall, 1995.

86. G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

87. N.S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

88. A.K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

89. R.L. Joshi, T.R. Fischer, and R.H. Bamberger, "Optimum classification in subband coding of images," in *Proc. of ICIP'94* (Austin, TX), pp. 883-887, Nov. 1994.

90. M.C. Chen and A.N. Willson Jr., "Rate-distortion optimal motion estimation algorithms for motion-compensated transform video coding," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 8, pp. 147-158, Apr. 1998.

91. D. Tzovaras and M. Strintzis, "Motion and disparity field estimation using rate-distortion optimization," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 8, pp. 171-180, Apr. 1998.

92. D.T. Hoang, P.M. Long, and J.V. Vitter, "Efficient cost measures for motion estimation at low bit rates," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 8, pp. 488-500, Aug. 1998.

93. J. Ribas-Corbera and D.L. Neuhoff, "Optimal bit allocations for lossless video coders: Motion vectors vs difference frames," in *Proc. of ICIP'95*, vol. 3, pp. 180-183, 1995.

94. J. Ribas-Corbera and D.L. Neuhoff, "Optimizing block size in motion-compensated video coding," *Journal of Electronic Imaging*, vol. 7, pp. 155-165, Jan. 1998.

95. W.C. Chung, F. Kossentini, and M.J.T. Smith, "A new approach to scalable video coding," in *Proc. of DCC'95*, (Snowbird, UT), pp. 381-390, Apr 1995.

96. G.M. Schuster, G. Melnikov, and A. Katsaggelos, "Operationally optimal vertex-based shape coding," this issue, pp. 91-108.

97. G.M. Schuster and A.K. Katsaggelos, "An optimal boundary encoding scheme in the rate-distortion sense," *IEEE Trans. on Image Proc.*, vol. 7, pp. 13-26, Jan. 1998.

98. G.M. Schuster and A.K. Katsaggelos, "A video compression scheme with optimal bit allocation among segmentation, displacement vector field, and displaced frame difference," *IEEE Trans. on Image Proc.*, vol. 6, pp. 1487-1502, Nov 1997.

99. C. Herley, J. Kovacevic, K. Ramchandran, and M. Vetterli, "Tilings of the time-frequency plane: construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Trans. on Signal Proc., Special Issue on Wavelets*, vol. 41, pp. 3341-59, Dec.1993.

100. Z. Xiong, K. Ramchandran, C. Herley, and M.T. Orchard, "Flexible time segmentations for time-varying wavelet packets," *IEEE Trans. on Signal Proc.*, vol. 45, pp. 333-345, Feb. 1997.

101. Z. Xiong, M.T. Orchard, and K. Ramchandran, "Wavelet packet image coding using joint space-frequency quantization," *IEEE Trans. on Image Proc.*, pp. 892-898, June 1998.

102 K. Ramchandran, M. Vetterli, and C. Herley, "Wavelets, subband coding and best bases," *Proceedings of the IEEE*, pp. 541-560, April 1996. Special Issue on Wavelets: Invited Paper.

103. R. Coifman and V. Wickhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform Theory,* vol. IT-38, pp. 713-718, Mar. 1992.

104. C.-T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudoconstant bit rate video coding," *IEEE Trans. on Image Proc.*, vol. 2, pp. 50-59, Jan. 1993.

105. J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled Lagrange multiplier method," *IEEE Trans. on Image Proc.*, vol. 3, pp. 546-558, Sept. 1994.

106. J. Lee and B.W. Dickinson, "Rate distortion optimized frame-type selection for MPEG coding," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 7, pp. 501-510, June 1997.

107. A. Eleftheriadis and D. Anastassiou, "Constrained and general dynamic rate shaping of compression digital video," in *Proc. of ICIP'95*, vol. III (Washington, D.C.), pp. 396-399, 1995.

108. A. Ortega and M. Khansari, "Rate control for video coding over variable bit rate channels with applications to wireless transmission," in *Proc. of ICIP'95*, (Washington, D.C.), Nov 1995.

109. C.-Y. Hsu and A. Ortega, "A Lagrangian optimization approach to rate control for delay-constrained video transmission over burst-error channels," in *Proc. of ICASSP'98*, (Seattle, WA), May 1998.

110. M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit-rate video transmission over fading channels for wireless microcellular systems," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 6, pp. 1-11, Feb. 1995.

111. S. Lin and J.D.J. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1984.

112. J. Modestino, D.G. Daut, and A. Vickers, "Combined source-channel coding of images using the block cosine transform," *IEEE Trans. on Commun.*, vol. COM-29, pp. 1261-1274, Sept. 1981.

113. N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. on Inform. Theory*, vol. IT-37, Jan. 1991.

114. F.H. Liu, P. Ho, and V. Cuperman, "Joint source and channel coding using a non-linear receiver," in *Proc. of ICC'93*, June 1993.

115. W.F. Schreiber, "Considerations in the design of HDTV systems for terrestrial broadcasting," *SMPTE Journal*, pp. 668-677, Sept. 1991.

116. K. Ramchandran, A. Ortega, K.M. Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source-channel coding," *IEEE J. on Sel. Areas in Comm.*, vol. 11, pp. 6-23, Jan. 1993.

117. S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 983-1001, Aug.1997.

118. J. Hagenauer, "Rate-compatible punctured convolutional codes (rcpc codes) and their applications," *IEEE Trans. on Comm.*, vol. COM 36, pp. 389-400, Apr. 1988.

119. K. Ramchandran and M. Vetterli, "Multiresolution Joint Source-Channel Coding for Wireless Channels," in *Wireless Communications: A Signal Processing Perspective*, Editors: V. Poor and G. Wornell. Prentice-Hall, 1998.

120. B. Schafer, "Terrestrial transmission of DTVB signals - the European specification," in *International Broadcasting Convention*, no. 413, September 1995.

121. H. Kumazawa, M. Kasahara, and T. Namekawa, "A construction of vector quantizers for noisy channels," *Electron. Eng. Japan*, vol. 67-B, pp. 39-47, 1984.

122. K.A. Zeger and A. Gersho, "Zero redundancy channel coding in vector quantization," *IEEE Electron. Letters*, pp. 654-655, June 1987.

123. I. Kozintsev and K. Ramchandran, "Robust image transmission over energy-constrained time-varying channels using multiresolution joint source-channel coding," *IEEE Trans. on Signal Processing, Special Issue on Wavelets and Filter Banks*, vol. 46, pp. 1012-1026, April 1998.

124. M.W. Marcellin and T.R. Fischer, "Joint trellis coded quantization and modulation," *IEEE Trans. on Comm.*, Jan. 1993.

125. I. Kozintsev and K. Ramchandran, "A hybrid compressed-uncompressed framework for wireless image transmission," in *Proc. of ICIP'97*, (Santa Barbara, California), Oct. 1997.