A general block-based approach to joint source-channel coding for channels with feedback

Swaroop Appadwedula*, Douglas L. Jones, and Kannan Ramchandran

EDICS 1-SOUR: Source/Channel Coding

S. Appadwedula and D.L. Jones are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign 1308 West Main St, Urbana, IL 61801.

K. Ramchandran is with the Berkeley Audio-visual Signal Processing and Communication Systems (BASiCS) group at University of California at Berkeley 269 Cory Hall, Berkeley, CA 94720.

appadwed@uiuc.edu (217)244-0575, jones@ifp.uiuc.edu (217)244-6823, kannanr@eecs.berkeley.edu (510)642-2353, Fax:(217)244-1642

This research was supported by the National Science Foundation under contract no. MIP-9707742.

Abstract

Application of joint source-channel coding in heterogeneous multimedia environments demands general adaptive source-channel approaches suitable for a wide variety of source coding standards, channel coders, and variable channel conditions. We develop a general approach for joint source-channel matching based on parametric models for source and channel coders that capture the key characteristics of these coders with respect to source-channel matching. We exploit the flexibility of these coders through the models to obtain an adaptive approach that reconfigures the source-channel system to match time-varying channels. We formulate source-channel matching as an optimization problem by writing the end-to-end performance of the system in terms of the source-channel parameters and including the system resource constraints. We solve the optimization problem for various source-channel combinations to obtain algorithms that determine the optimal source-channel parameters. Our simulations indicate that it may be possible to obtain nearly all of the benefits of joint source-channel optimization by matching existing source and channel coding standards using the simple and general approach we propose.

Keywords

Image transmission, Source-channel coding, wireless systems

I. Introduction

Design of efficient image transmission systems requires the proper allocation of system resources to protect and transmit the image bitstream. Significant interest in joint source-channel systems has been generated by the dramatic performance gains of source-channel systems over traditional systems. Traditionally, communication systems have been designed by applying the source-channel separation principle. In [1], Shannon introduced the source-channel coding theorem, which says that under certain assumptions, source coding and channel coding can be treated separately without any loss of performance. A source coder is designed to remove redundancy in the source data to create independent symbols, and a channel coder is designed to introduce redundancy to protect the data from the noisy channel. Source-channel separation greatly simplifies the design of communications systems because the source coder is designed independently of the channel characteristics, and the channel coder is designed independently of the source characteristics. Unfortunately, infinite channel symbol length may be required to achieve optimal separate design of the source coder and the channel coder. In practice, such coders cannot be implemented due to the complexity and delay constraints of real systems.

By joint design, it may be possible to approach theoretical rate-distortion limits with finite complexity and finite delay. Dunham and Gray [2] show that a communication system using trellis encoding of a stationary and ergodic source over a discrete memoryless noisy channel can perform arbitrarily close to the source distortion-rate function evaluated at the channel capacity. They prove the existence of a joint source-channel system for a specific source and a specific channel. There has been significant work on joint source-channel coding both from an information-theoretic viewpoint and in terms of specific implementations. Demir and Sayood [3] do a good job of classifying and summarizing the work in the field.

Results in these papers indicate that exploiting the trade-off between data and redundancy improves performance. In order to approach the theoretical performance bounds with practical systems, a joint source-channel coding system can be used. The methods found in the literature are based on a specific source coder, a specific channel coder, and a particular assumed channel. In practice, a variety of image transmission systems are possible. Source coding standards include Joint Photographic Experts Group (JPEG), Set Partitioning in Hierarchical Trees (SPIHT), and the emerging JPEG 2000. Channel coding methods can be constructed from a variety of channel codes such as convolutional codes, block codes, and turbo coders combined with a variety of modulation techniques such as phase shift keying or quadrature amplitude modulation. In addition, image and video may travel through a heterogeneous network of which the wireless link is only one component. In practice, then, a broadly useful system must support many different standards for both source and channel coding. A general approach to source-channel matching that can support various types of source coding standards, channel coders, and channels has yet to be developed.

Another desirable feature of a source-channel matching technique is adaptivity. The most significant gains of source-channel matching arise due to proper matching to the time-varying channel conditions. For example, a transmission system designed for channels with little noise will fail miserably on noisy channels. Many of the current joint source-channel techniques match the source coder and channel coder for a specific channel, and any change in the channel condition requires significant effort to redesign the system. In this work, we seek a flexible matching technique that adaptively matches the source coder and the channel coder to the time-varying channel.

Several papers in source-channel coding have followed the trend towards generality. Bystrom and Modestino [4] examine source-channel coding over slow-fading Rician channels. Their approach is to use universal distortion-rate characteristics for the source coder, which determine the distortion as a function of source rate and probability of error. The optimal trade-off between source and channel rate is determined by combining the distortion-rate characteristics with bounds on the probability of error for trellis-coded modulation using a modified distance metric to account for fading effects. Their results show that for H.263 video streams, more powerful codes may be necessary for this type of source characterization due to loss of synchronization information. In [5], a systematic method for instantaneous rate allocation between a progressive source coder and a channel coder is considered. Closed-form expressions for the end-to-end distortion and rate allocation are derived for the probability of error function modeled as a log-affine function. Although both of these papers consider relatively general source-channel schemes, they concentrate primarily on progressive coders.

We develop a general adaptive approach to source-channel matching that can be applied to a large class of source coders and channel coders operating over variable channel conditions [6]. The approach is to consider the source coder and channel coder as parametric blocks with parameters that can be optimally chosen to satisfy the system constraints and match the time-varying channel. By using the end-to-end distortion of the source-channel system as the cost function to measure the true performance of the communication system, the optimal parameters for the source coder and channel coder are obtained. The source-channel parameters can then be updated to eliminate any mismatch with the channel.

The remainder of the paper is organized as follows. Section II describes the general source-channel matching approach and the end-to-end distortion criterion. Sections III and IV describe the general source and channel models respectively. Once the source coder and channel coder models are obtained, the end-to-end distortion criterion is simplified in Section V by considering the progressive and nonprogressive components of the source stream separately. Following the simplification, optimization algorithms for various channel coders are presented in Section VI. In Section VII, a progressive and a nonprogressive source coder are described as well as a technique for determining the rate-distortion function of the progressive coder. In Section VIII, the optimization techniques described in Section VI are applied to various source-channel combinations for the Lena image.

II. A General Approach

Application of source-channel coding in wireless image communication system design results in the basic problem of finding a general approach for matching a source coder to a channel coder. We seek a matching that optimizes the end-to-end system performance subject to constraints on the system resources. To solve this problem, we first develop models for the source encoder/decoder and the channel encoder/decoder combinations for a particular channel based on their basic definitions. In particular, we use a rate-distortion-based model for the source coder obtained from experimental data, and analytical expressions for the channel coder performance over a particular channel obtained from error analysis. Then, we combine these models by considering a possible performance criterion for the combined system and write several optimization problems that can be solved by numerical methods to provide a solution to the image transmission problem.

One possible optimization criterion desired by the user of an image transmission system is to maximize the end-to-end distortion subject to constraints on the system resources. We state the problem as follows

$$\begin{aligned} \min \quad & \text{end-to-end distortion} = \\ & f(\text{source coder \& channel coder parameters}, \text{channel response}) \\ s.t. \quad & \text{source rate} + \text{channel rate} \leq \text{total rate} \\ & \text{total energy transmitted} \leq \text{maximum available energy} \end{aligned} \tag{1}$$

There are many other criteria that may be desirable to the user. For example, the average error-free source rate could be maximized subject to a minimum quality constraint [7]. Most optimization criteria include end-to-end distortion and/or minimum quality-of-service metrics, which are all included in the optimization problem we consider.

III. Source Coder Model

To simplify the modeling of the source coder, the standard approach is to use an image-dependent rate-distortion model for the source coder, which describes the distortion d due to compression to a particular rate of r bits assuming that the source decoder receives all the source-encoded bits without error [8]. The rate-distortion model is an operational model since it describes the achievable rate-distortion configurations of a particular coder. The rate-distortion configurations are described as $(r_0, d_0), (r_1, d_1), \ldots, (r_{M-1}, d_{M-1})$ for the M specific source coder configurations. We wish to extend this model for source-channel matching purposes by considering error-resilient components of the source stream separately from components of the source stream that must be transmitted without errors. Instead of considering various rates at which the source coder can operate, we fix the rate to R_s symbols of the source-encoded stream and differentiate between the "vital" and "non-vital" components of the source-encoded stream.

Image coders typically produce bit streams with both progressive and nonprogressive components. Progressive components of the encoded stream contain separated coarse-level information and detail information that could potentially be removed or corrupted without entirely destroying the reconstruction at the source decoder. Progressive elements of the source-encoded stream arise naturally due to transform coding and quantization type of approaches that have a high degree of scalability. On the other hand, nonprogressive components of the encoded stream contain essential information that cannot be corrupted or removed in order to recover any reasonable form of the original image. Nonprogressive components of the encoded stream arise due to header, marker, and synchronization information, which are typically intolerant to errors.

Based on these characteristics of the source-encoded stream, we model the source coder as a rate-controlled system component with a mixed-progressive output stream. When the source data is compressed to a rate of R_s symbols, the distortion due to the loss of progressive source symbol k is

$$D_s(k) = d(k)\,(1 - Pr(\text{fail})) + d_{tot} Pr(\text{fail}) \tag{2}$$

where $d(k)$ is the distortion of the progressive component when the k^{th} source symbol $k \in 0, 1, \ldots, K-1$ is in error, and d_{tot} is the distortion due to failure in the nonprogressive component of the source-encoded stream. Since image compression introduces error even if all the source symbols are correctly received, we represent the distortion due to compression to R_s symbols as d_c. We will show that the d_c term does not affect the optimization problems we consider. We define a source symbol as a group of source-encoded bits whose length m is determined by the choice of channel coder.

The rate-distortion function of a particular image may be obtained during source encoding or decoding with some additional processing (see Section VII-A). Often the distortion function is measured for a small set of the source symbols. When the distortion function is known to be smooth, interpolation methods such as curve fitting can be used on the measured data to determine d for all necessary k [9]. For source coders that have a fixed set of unevenly-spaced feasible operating rates, the rate-distortion function is the set of (r_k, d_k) points where the source symbols have varying length of r_k bits.

The source model simplifies when either completely progressive or completely nonprogressive coders are used. For completely progressive coders, the source model is simply the distortion function $d(k)$, which can be reordered from most important source symbol to least important source symbol to yield a monotonically decreasing function. For completely nonprogressive coders, the distortion function is binary: when the entire stream is error-free, the distortion is d_c, and when any errors occur, the distortion is d_{tot}. For completely nonprogressive coders, the mixed-progressive model is not very useful, since a fixed rate R_s must be transmitted. An operational rate-distortion model where (r_k, d_k) represents different choices of compression rate r_k bits and associated distortion d_k when all the source-encoded bits are correctly received is more useful for these coders.

Based on the source-coder definition and its typical characteristics, a source coder model has been proposed in Eqn. (2) that can be applied to a wide variety of source coders.

IV. Channel-Coder Model

The channel coder is a functional block that introduces redundancy to discrete symbols and converts the redundant representation to an analog signal. Channel coders can generally be divided into two sections. The forward error correction (FEC) section converts the source-encoded stream into a redundant bit stream that can withstand bit errors. The modulation section converts the redundant bit stream to an analog signal for transmission.

A reasonable approach used in the communications community and found in many textbooks is to consider the channel (en)coder, channel, and channel decoder together as an inner system. Using this approach, we can model the inner system as providing a finite set of channel symbols with corresponding transition probabilities that are a function of the system resources such as total energy and total bitrate. A simpler approach is to consider the probability that each symbol is in error by summing the transition probabilities to any of the other symbols. We choose the simpler approach for the source model because error expressions are available for many different inner systems.

In the channel coder, we consider allocating the system resources of total energy and total bitrate over various blocks of data. Each source-channel block $i \in 0, 1, \ldots, N-1$ is described by a source rate of $r_s(i)$ symbols, a channel rate of $r_c(i)$ symbols and a block energy of $E(i)$, where there are r_{sc} symbols in each block. We denote by $\mathbf{r_s}$, $\mathbf{r_c}$, and $\mathbf{E}$ the set of source rates, channel rates, and energies for the blocks of an image. Note that the channel encoding need not be systematic so that $r_s(i)$ and $r_c(i)$ describe the ratio of data to redundancy instead of counting actual symbols in the source-coded stream. For a specific channel coder choice, an expression for the block-error probability $P_b(i)$ can be found that follows the general definition. A graphical representation of the general channel coder is shown in Figure 1.

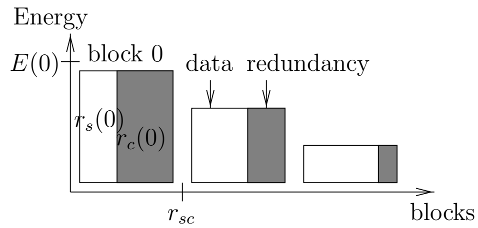

Fig. 1. General channel coder.

V. End-to-end Distortion Criterion

Using the source coder and inner system models, the image transmission problem can be reformulated in terms of the rate-distortion function, total energy, and bitrate as follows

$$\begin{aligned} \min \quad & E[D] = f(D_s, \mathbf{r_s}, \mathbf{r_c}, \mathbf{E}, \text{channel}) \qquad (3) \\ s.t. \quad & Pr(\text{fail}) \leq \delta \\ & \sum_{i=0}^{N-1} (r_s(i) + r_c(i)) \leq R_{tot} \\ & \sum_{i=0}^{N-1} E(i) \leq E_{tot} \end{aligned}$$

where $E[D]$ is the expected value of the end-to-end distortion, δ is the maximum probability of failure for the image, R_{tot} is the total number of symbols available for the image, and E_{tot} is the total energy available to transmit the image.

For completely non-progressive coders, this criterion translates to choosing the compression rate, R_s, and ensuring that all R_s symbols are transmitted without error with probability $1 - \delta$.

A. Progressive Coders

For the optimization criterion described, we can consider embedded progressive coders and determine the choice of parameters that maximize the end-to-end distortion. Once algorithms for finding the parameters for a progressive system are determined, the total rate and total energy constraints can be adjusted to meet the $Pr(\text{fail})$ constraint for the nonprogressive component of a mixed progressive system.

Due to the embedded property of the progressive stream, the end-to-end distortion depends on the location of the first block error, because all blocks after an erroneous block are corrupted due to their dependency on the erroneous block. The expected end-to-end distortion for the embedded progressive coder is thus

$$E[D] = \sum_{i=0}^{N-1} D_s \left(\sum_{l=0}^{i-1} r_s(l) \right) P_b(i) \prod_{j=0}^{i-1} (1 - P_b(i)) + d_c \prod_{i=0}^{N-1} (1 - P_b(i)) \tag{4}$$

where $P_b(i)$ is the probability that block i is in error, and $D_s\left(\sum_{l=0}^{i-1} r_s(l)\right)$ is the distortion due to error in the first symbol of the $r_s(i)$ symbols of block i. The first term is the distortion for block i weighted by the probability that block i is the *first* uncorrectable block. The expression can be simplified by subtracting d_c from each of the D_s's to yield

$$E[D] = d_c + \sum_{i=0}^{N-1} \left(D_s \left(\sum_{l=0}^{i-1} r_s(l) \right) - d_c \right) P_b(i) \prod_{j=0}^{i-1} (1 - P_b(j)) \tag{5}$$

For simplicity, we can ignore the d_c term since it does not affect the optimization problem

$$E[D] = \sum_{i=0}^{N-1} D_s \left(\sum_{l=0}^{i-1} r_s(l) \right) P_b(i) \prod_{j=0}^{i-1} (1 - P_b(j)) \tag{6}$$

B. Nonprogressive Coders

In nonprogressive coders, we would like to minimize the probability of losing the entire image. The problem is then to choose the largest compression rate R_s while ensuring that the probability of any error in the R_s reconstructed source symbols is less than δ. Since all the symbols of the nonprogressive stream have equal importance, all of the blocks have the same error probability P_b, number of source symbols $r_s(i) = r_s$, and number of channel symbols $r_c(i) = r_c$. The total number of blocks to be transmitted, M ($M < N$), determines the source-coding rate R_s. The optimization problem is restated as

$$\begin{aligned} \max \quad & R_s = M r_s \\ s.t. \quad & (1 - P_b(E, r_c))^M \geq 1 - \delta \\ & M(r_s + r_c) \leq R_{tot} \\ & ME \leq E_{tot} \end{aligned} \tag{7}$$

where the first constraint is the probability that no errors occur in the source-encoded stream. The optimal solution occurs at $M(r_s + r_c) = R_{tot}$ and $ME = E_{tot}$, since we could always do better by increasing the energy per symbol or the number of source symbols when there is any slack in the constraint. The problem simplifies considerably to finding the set of solutions (M, r_c) that satisfy

$$\left(1 - P_b\left(\frac{E_{tot}}{M}, r_c\right)\right)^M \geq 1 - \delta \tag{8}$$

From the set of solutions (M, r_c), we choose the one that maximizes the source rate.

Now that the characteristic of progressive and nonprogressive components have been considered for the end-to-end distortion criterion, we will describe various optimization algorithms for finding the optimal system parameters. Two approaches are described, one for the energy allocation scenario and one for the rate allocation scenario. Both approaches utilize gradient or slope information to iteratively update the optimal system parameters.

VI. Optimization Algorithms

Our general technique allows different combinations of source coders and channel coders to be considered. We choose a few combinations as reasonable design choices in a wireless communications system, determine the associated optimization problem, and propose an optimization method. Various optimization techniques such as Lagrange multiplier techniques, feasible directions techniques, or dynamic programming techniques can be used to solve the constrained nonlinear matching problems [10]. In this section we describe optimization techniques suited to each combination. In particular, gradient-projection algorithms are a good choice in most of the problems due to the linearity of the problem constraints. With good initialization, gradient-project algorithms provide fast convergence, whereas dynamic programming methods such as in [7] have exponential order.

Progressive coders provide the most flexibility in terms of choice of channel coders. Due to bitstream scalability and variable importance of bit errors, the benefits of source-channel matching are simple to realize. Many different kinds of channel coders provide feasible choices due to the inherent resiliency in the bitstream.

A. Progressive Variable-Energy Combination

In this example we consider minimizing the end-to-end distortion of a system with a progressive source coder and a variable-energy channel coder. In a variable-energy channel coder, each block of the source-encoded stream is modulated with a different amount of energy, and no redundancy is introduced. Simple expressions for bit-error probability are available for most modulation techniques over various types of channels. If we assume that a block is in error if any of the symbols in the block are in error, then we obtain the block-error probability as

$$P_b(i) = 1 - (1 - p(i))^{m\, r_{sc}} \tag{9}$$

where $p(i)$ is the bit-error probability in block i, and mr_{sc} is the number of bits in each block.

By substituting the expression for block-error probability of the channel coder into Eqn. (6), the end-to-end distortion of the system can be obtained as

$$E[D] = \sum_{i=0}^{N-1} D_s(ir_{sc}) P_b(i) \prod_{j=0}^{i-1} (1 - P_b(j)) \tag{10}$$

The overall system performance can be optimized by allocating the energy in each block subject to a total energy constraint.

A gradient-projection method can be used to find the optimal power allocation. The gradient $\mathbf{g} = dE[D]/d\mathbf{E}$ can be computed using the product rule as $\frac{dE[D]}{dP_b(i)} \frac{dP_b(i)}{dE(i)}$. The second term can be obtained by differentiating the block-error probability of the inner system $\frac{dP_b(i)}{dE(i)} = \frac{dP_b(i)}{dp(i)} \frac{dp(i)}{dE(i)}$, where $\frac{dP_b(i)}{dp(i)} = (1 - p(i))^{mr_{sc}-1}$.

The term $\frac{dE[D]}{dP_b(i)}$ can be obtained from Eqn. (6) as

$$\frac{dE[D]}{dP_b(i)} = D_s(ir_{sc}) \prod_{j=0}^{i-1} (1 - P_b(j)) - \sum_{l=0, l \neq i}^{N-1} D_s(lr_{sc}) P_b(l) \prod_{j=0, j \neq i}^{l-1} (1 - P_b(j)) \tag{11}$$

At each iteration of the gradient-projection algorithm, the gradient $\mathbf{g} = dE[D]/d\mathbf{E}$ determines the direction in which to change the energy in each block to provide a descent in $E[D]$. Then, to match the total energy constraint, the energy in each block is projected onto the surface of fixed total energy E_{tot}. Now, we derive a simple projection method for the simplex constraint $\sum_i E(i) = E_{tot}$. Note that the optimal solution $\mathbf{E}^*$ must occur at $\sum_i E(i) = E_{tot}$, since $\sum_i E(i) < E_{tot}$ would imply that the energy in any one of the blocks can be increased to obtain a lower end-to-end distortion.

A true projection operator f minimizes the distance between the starting point $\mathbf{E}$ and the constraint set. That is, we wish to find

$$\mathbf{E}_p^* = \underset{\mathbf{E_p} \in \mathcal{E}}{\operatorname{argmin}} f(\mathbf{E_p}) = \|\mathbf{E_p} - (\mathbf{E} - \eta \mathbf{g})\|^2 \tag{12}$$

where $\mathbf{E}_p^*$ is the projection of $\mathbf{E} - \eta\, \mathbf{g}$ on the constraint set $\mathcal{E} = \{\mathbf{E_p} : \sum_i E_p(i) \leq E_{tot}, E_p(i) \geq 0\}$. Taking the derivative of $f(\mathbf{E_p})$, we obtain

$$\frac{df}{dE_p(i)} = 2(E_p(i) - (E(i) - \eta g(i))) \tag{13}$$

By applying the first-order necessary condition for a local minimum over the simplex constraint, we obtain [10]

$$\frac{df(\mathbf{E}_p^*)}{dE_p(i)} = \lambda \ \forall i \text{ with } E_p^*(i) > 0 \tag{14}$$

$$\frac{df(\mathbf{E}_p^*)}{dE_p(i)} \geq \lambda \ \forall i \text{ with } E_p^*(i) = 0 \tag{15}$$

where λ is a Lagrange multiplier. Substituting Eqn. (13) into these equations, we obtain

$$E_p^*(i) = \begin{cases} 0 & \forall i \text{ with } E(i) - \eta g(i) \leq -\lambda \\ E(i) - \eta\, g(i) + \lambda & \text{otherwise} \end{cases}$$

which implies that $E(i) - \eta\, g(i)$ below some constant threshold $-\lambda$ are set to zero. In an attempt to determine λ, consider using the constraint relationships. We have

$$\begin{aligned} E_{tot} &= \sum_i^N E_p^*(i) \\ &= \sum_i^N E(i) - \eta \sum_i^N g(i) + N\lambda + \sum_j^M \varepsilon(j) \end{aligned} \tag{16}$$

where $\varepsilon(i)$ is the slack required to make $E_p^*(i)$ zero for M of the blocks. Solving for λ, we obtain

$$\lambda = \frac{1}{N}\left(\eta \sum_i^N g(i) - \sum_i^M \varepsilon(i)\right) \tag{17}$$

Because only one equation is available, it is not possible to solve for the variables M and λ analytically. However, we can determine λ and M by iteration. Because M is a finite integer and $0 < M < N$, one iterative projection technique is to increase M until the constraint set is met. Figure 2 describes the projection in detail for the assumption that the function D_s is monotonically decreasing. Note that D_s can always be reordered to satisfy this assumption.

Do until convergence

I. Calculate gradient $\mathbf{g}$

II. Update $\mathbf{E} = \mathbf{E} + \bar{\eta}\mathbf{g}$, where $\bar{\eta} = \eta \frac{E_{tot}}{\sum_i g(i)}$

III. Projection

1 Set $\mathbf{E}_p = \mathbf{E} + (E_{tot} - \mathbf{E})/N$

2. If $E_p(i) > 0 \; \forall \; i = \{1, \ldots N\}$ then goto step 3, else

i. Set $j = N$ and $\mathbf{E}_p = \mathbf{E}$

ii. Set $E_p(j) = 0$

iii. $\lambda = \frac{1}{j-1}(E_{tot} - \sum_i E_p(i))$

iv. If $\lambda + \min_{k=\{1,\ldots j\}} E_p(k) < 0$ then decrement j and goto step ii

v. Otherwise $E_p(k) = E_p(k) + \lambda$ for $k = \{1, \ldots j\}$

3. Set $\mathbf{E} = \mathbf{E}_p$

Fig. 2. Gradient-projection algorithm.

At each iteration of the gradient-projection algorithm, the cost $E[D]$ decreases and the algorithm is guaranteed to converge to a local minimum. When the function $E[D]$ is unimodal, the local minimum is also the global minimum. For the progressive coder case, the behavior of $\prod_{j=0}^{i-1}(1 - P_b(j))$ makes the function $E[D]$ have multiple extrema.

Note that the gradient step taken by $\mathbf{E}$ is normalized by $\bar{\eta}$ to a unit step. The choice of the parameter η is based on the trade-off between convergence speed and convergence error. The starting point of the algorithm is important for system stability. A good choice is to allocate all the total energy to the most important blocks so that these blocks are received without errors.

B. Progressive Total Rate Combination

Here, we consider minimizing the end-to-end distortion of a system with a progressive source coder and a channel coder with variable source/channel rate in each block. We are motivated here by linear block coders such as Reed-Solomon, or other algebraic codes. In many situations, designers must use off-the-shelf modems that have fixed modulation techniques and associated bit-error probability. In order to take advantage of source-channel matching, flexibility can be introduced in the channel coder using FEC. More important information can be transmitted at lower rates by introducing redundancy, and less important information can be transmitted without protection at higher rates.

Two major classes of FEC techniques are block coders and convolutional coders. In a block code, the information bits are divided into nonoverlapping blocks and are encoded separately. Each block of information bits is mapped to a code of longer length containing redundant information such as parity bits. For this channel model, the block-error probability is described in terms of the integer number of protection symbols $r_c(i)$ in each block, and the other symbols in each block are source symbols $r_s(i) = r_{sc} - r_c(i)$, where $\mathbf{r_s}$ and $\mathbf{r_c}$ are in terms of m-bit symbols instead of individual bits. The overall system performance can be optimized by allocating the protection symbols in each block subject to a total rate constraint.

For the discrete parameter $\mathbf{r_c}$, we use the slope-based technique shown in Figure VI-B. The technique involves successive passes over all the blocks in which we travel in the negative slope direction. In each pass, the protection symbols in each block are refined while holding all the other blocks fixed. The basic idea of the algorithm is to refine each block based on its effect on the end-to-end distortion. Experimental results with the SPIHT source coder and Reed-Solomon channel coder over an additive white Gaussian noise (AWGN) channel show that 2 to 3 iterations over all the blocks is sufficient for convergence. As a result of the $2Nlog_2(r_{sc})$ to $3Nlog_2(r_{sc})$ evaluations of the distortion function $E[D]$, the algorithm is relatively slow. The allocation of protection symbols $r_c(i)$ for a particular block takes $log_2(r_{sc})$ iterations and determines the global minimum when the function has a single value of $r_c(i) = r_c^*$ for which $E[D]|_{r_c^*-1} > E[D]|_{r_c^*} < E[D]|_{r_c^*+1}$. In this case, the function has at most one extremum at r_c^*, and the minimum must lie either at r_c^* or at one of the end points $r_{c,min} = 1$, or $r_{c,max} = r_{sc}$.

Due to the discrete nature of the problem and the existence of local minima, a gradient technique using a continuous approximation to the end-to-end distortion may not reach the global minimum, but could be used to find a good starting region for local-search algorithms.

C. Progressive Adjustable-Rate and Progressive Adjustable-Protection Combinations

For the adjustable-rate coder, each block has the same energy, so the end-to-end distortion in Eqn. (6) simplifies to

$$E[D] = \sum_{i=0}^{M-1} D_s(r_{sc}i) P_b (1 - P_b)^i \tag{18}$$

Do until convergence

For each $i = 0, ...N-1$ do

I. Hold all $r_c(j)$, $j \neq i$ fixed.

II. Find the optimal $r_c(i)$ using slope-based search

1. Start with $r_{c,low} = 1$ and $r_{c,high} = r_{sc}$

2. Do until $r_{c,low} = r_{c,high}$

i. Set $r_{c,mid} = \frac{r_{c,low}+r_{c,high}}{2}$

ii. If $E[D]|_{r_{c,mid}} - E[D]|_{r_{c,mid}+1} < 0$ then

a. If $r_{c,low} \neq r_{c,mid}$ then set $r_{c,low} = r_{c,mid}$

b. Otherwise set $r_{c,low} = r_{c,mid} + 1$

iv. Otherwise

a. If $r_{c,high} \neq r_{c,mid} + 1$ then set $r_{c,high} = r_{c,mid} + 1$

b. Otherwise set $r_{c,high} = r_{c,mid}$

3. Set $r_c(i) = r_{c,low}$

Fig. 3. Protection symbols allocation algorithm.

where M blocks have energy E_{tot}/M and $(N-M)$ blocks have zero energy. As in Figure VI-B, a slope-based search technique can be used to find M. The algorithm is described in Figure VI-C and converges when the function is unimodal in the discrete sense. When the discrete function changes slope from positive to negative only at a single value, the function is considered unimodal.

Typically, changing the protection level on a block-by-block basis with forward error correction is infeasible. In an adjustable-protection scenario, the fixed rate of $r_c(i) = r_c$ is used for all of the blocks. The end-to-end distortion in Eqn. (6) simplifies considerably to

$$E[D] = \sum_{i=0}^{N-1} D_s(ir_s)P_b(1-P_b)^i \tag{19}$$

where all of the blocks have the same probability of error $P_b(r_c)$. As in Figure VI-C, a slope-based search technique can be used to find r_c except that iteration over all the blocks is unnecessary.

1. Start with $M_{low} = 1$ and $M_{high} = N$
2. Do until $M_{low} = M_{high}$
   i. Set $M_{mid} = \frac{M_{low}+M_{high}}{2}$
   ii. If $E[D]|_{M_{mid}} - E[D]|_{M_{mid}+1} < 0$ then
      a. If $M_{low} \neq M_{mid}$ then set $M_{low} = M_{mid}$
      b. Otherwise set $M_{low} = M_{mid} + 1$
   iv. Otherwise
      a. If $M_{high} \neq M_{mid} + 1$ then set $M_{high} = M_{mid} + 1$
      b. Otherwise set $M_{high} = M_{mid}$

Fig. 4. Adjustable rate allocation algorithm.

D. Nonprogressive and Adjustable Protection Combination

For a nonprogressive coder, all the source symbols may be treated equally, since all symbols must be received to ensure proper decoding. By optimally choosing the number of source symbols to be transmitted (and correspondingly the source-encoding rate), a failure-probability constraint can be satisfied. For a mixed progressive source coder, the nonprogressive component has a fixed rate and no tradeoff between source rate and failure probability is possible.

Here, we consider using a nonprogressive source coder with an adjustable-rate channel coder. All the source-encoded symbols are treated equally and block codes are used to provide FEC to guarantee delivery of a particular source rate R_s. In the optimization problem given in Eqn. (8), $P_b(r_c)$ is the probability that a block is decoded correctly, and M is the total number of codeword blocks. The optimal values for M and r_c can be found by directly solving Eqn. (8) via iteration.

VII. Source Coders and Determining the Rate-Distortion Function

The variability in the input data is captured by the rate-distortion points. There are several ways of estimating these rate-distortion points. The straightforward approach is to apply the source encoder and decoder at various rates, calculate the distortion between the original and decoded images, and interpolate between these (r, d) points. In [9], Lin et al. use a cubic spline interpolation to get smooth rate-distortion curves for gradient-based optimization algorithms. Since multiple (r, d) points must be measured by encoding and decoding at these rates, the processing power required for this computation might make it infeasible in mobile applications.

For transform-based source coders such as SPIHT and JPEG, the error in the encoded image is primarily due to quantization, especially at high rates. The transform-domain quantization error energy at various rates provides an estimate of the operational rate-distortion points.

A. SPIHT Coder

For the progressive coder, we consider the bit-progressive set partitioning in hierarchical trees (SPIHT) coder, which is an extension by Said and Pearlman [11] of the embedded zerotree wavelet coding algorithm introduced originally by Shapiro [12]. The SPIHT coder is a well-known wavelet coder with good performance on natural images. A key property of the SPIHT coder is that it produces an embedded bit stream in which each bit depends on the previous bit. An error in a particular bit corrupts all future bits in the stream. Due to this property, there is a simple procedure for determining the rate-distortion points which can be performed during encoding or decoding.

In the SPIHT coder, wavelet transform coefficients are encoded hierarchically according to bit-planes, starting from the most significant bit-plane. The compressed bitstream has the embedded property of containing all the lower rates. Due to the embedded nature of this compressed bitstream, shortening the compressed bitstream is the same as compressing to the lower rate. In terms of estimating the (r, d) points, this implies that the encoder need only be run at the maximum rate.

We can compute the error energy at the encoder progressively from lower rates to higher rates as follows. Let $\{c_1^0, c_2^0, \ldots, c_D^0\}$ be the set of original wavelet coefficients, where D is the number of pixels in the image. Let n be the number of bits to encode the largest coefficient. At zero source rate, the distortion energy $E_0 = \sum_{i=1}^{D}(c_i^0)^2$ is the total energy in the wavelet coefficients. When the most significant bit-plane is encoded, the new distortion energy is $E_1 = \sum_{i=1}^{D}(c_i^1)^2$, where the coefficients are

$$c_i^1 = c_i^0 - I(c_i^0)2^{n-1} \tag{20}$$

$$I(c_i^0) = \left\{ \begin{array}{ll} 1 & \text{if } c_i^0 \geq 2^{n-1} \\ 0 & \text{otherwise} \end{array} \right\} \tag{21}$$

The indicator function $I(c)$ just determines whether the particular coefficient c has a 1 in the most-significant bit-plane. By expanding and collecting terms, we can write

$$E_1 = E_0 - 2^{n-1}(2\sum_{i=1}^{D} I(c_i^0)c_i^0 - M2^{n-1}), \tag{22}$$

where $M = \sum_{i=1}^{D} I(c_i^0)$ is the number of coefficients with a 1 in the most-significant bit-plane. For the coefficients not being updated, $I(c) = 0$, so the summation is over relatively few values. The summation can easily be computed by shifting the coefficients being updated, masking off the error, and accumulating (M does not have to be computed explicitly). For the next most significant bit-plane, we update E_1 based on the number of coefficients with 1 at this bit-plane and so on. The introduction of this calculation does not require any additional memory, since only the original coefficient values need to be accessible. To convert error energy to distortion, we simply divide by the number of total image pixels. The rate at which a particular distortion occurs is known as a consequence of the encoding (length of the output buffer). We also point out the update above can occur at finer steps than at each bit-plane. So, the complete rate-distortion curve can be found using this technique in real-time to progressively update the error energy as more significant bits are encoded.

B. JPEG Coder

The JPEG coder is a widely used source-coding standard for image transmission on the Internet. The JPEG coder is characterized by the Q factor, which determines the accuracy of the quantization and in turn the encoded rate. Experimental results obtained by measuring the distortion due to a single bit error in the JPEG-encoded bit stream show that it is highly sensitive to errors due to marker information in the bit stream.

Treating the JPEG coder as a completely nonprogressive coder, we can characterize it by a set of operational rate-distortion points (r_k, d_k), $k \in 0, 1, \ldots, K-1$, where d_k is the distortion due to compression to a rate of r_k bits.

VIII. Simulations

In order to demonstrate the utility and flexibility of our approach, we simulated various source-channel combinations. For the completely progressive SPIHT coder and the nonprogressive JPEG coder, we consider various channel coders that allocate energy and/or rate among the blocks. We show the adaptability of our approach by comparing our source-channel system, which adapts itself to the channel condition, with a fixed system designed for a particular channel condition.

We also compare these approaches with the theoretically best possible system, which attains the distortion function evaluated at the channel capacity.

Computer simulations were performed for various source-channel combinations with a 512×512, 8 bits per pixel (bpp) gray-scale Lena image over a channel with a total rate of $R_s = 1$ bpp (i.e., 512^2 bits). The simulations involved compressing the source image to a chosen rate, channel coding, and transmitting over a noisy channel that introduces bit errors. The end-to-end distortion of the system was measured by decoding the received bits and measuring the mean-squared error (MSE) relative to the original image. All simulation results are provided in terms of $\text{PSNR} = 10 \log_{10} (255^2/\text{MSE})$ dB averaged over 1000 or more iterations. In Table I, we summarize the combinations that have been considered and abbreviate their names.

For the distortion-based optimizations, we use the SPIHT coder to provide a progressive bit stream. Source distortion parameters for the SPIHT coder were determined for a compression rate of $R_s = 1$ bpp.

A. SPIHT variable-energy combination

Simulations of optimally matching the SPIHT source coder with the variable-energy BPSK channel coder (SPIHT-VE) were performed for various values of E_{tot} with $N_0 = 1$. In a variable-energy channel coder, each block of the source-encoded stream is modulated with a different amount of energy, and no redundancy is introduced. For binary phase shift keying (BPSK) over an additive white Gaussian noise (AWGN) channel with a fixed rate of $R_s = R_{tot}$ symbols, the bit-error probability for block i can be expressed as

$$p(i) = Q\left(\sqrt{\frac{2E(i)}{m\, r_{sc} N_0}}\right) \tag{23}$$

where $\frac{N_0}{2}$ is the channel noise variance, $E(i)$ is the energy in block i, $m\, r_{sc}$ is the number of bits in each block, and $Q(x)$ is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-x^2}{2}} dx \tag{24}$$

Alternatively, we can write the bit-error probability in terms of the signal-to-noise ratio (SNR) per bit in each block as $p(i) = Q(\sqrt{SNR(i)})$.

By using the bit-error probability expression in Eqn. (9), we can obtain the block-error probability $P_b(i)$ to be used in Eqn. (6). The optimization algorithm given in Figure 2 can be applied directly, since the gradient $dE[D]/d\mathbf{E}$ can be obtained by differentiating Eqn. (23)

$$\frac{dp(i)}{dE(i)} = -\frac{1}{\sqrt{mr_{sc}}} \frac{e^{\frac{-E(i)}{2mr_{sc}\sigma^2}}}{2\sigma\sqrt{2\pi E(i)}} \tag{25}$$

For comparison with the adaptive system, we simulated a nonadaptive system with fixed total rate $r_{sc} = 0.125$ bpp, 0.25 bpp, 0.50 bpp, or 0.75 bpp and associated fixed energy per bit for all the bits. Figure 5 shows that the adaptive system performance curve is the convex hull of the fixed-energy-per-bit systems. Whereas the adaptive system provides the optimal performance over the entire range of E_{tot}, the nonadaptive system is nearly optimal for a fixed value of E_{tot}. Although the fixed system is not exactly optimal for a fixed value of E_{tot} because it does not vary the energy in each block, the nearly optimal results show that the gains due to variable energy arising from variable priority of the blocks are not significant. On the operating curves for the non-adaptive systems, as E_{tot} decreases, the performance drops off rapidly, and as E_{tot} increases the performance plateaus at a sub-optimal level.

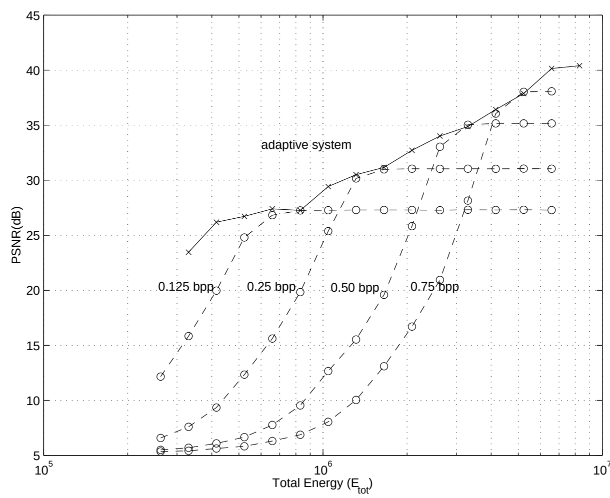

Fig. 5. Simulation results for Lena using SPIHT source coder and variable-energy channel coder.

B. SPIHT adjustable-protection combination

Simulations of optimally matching the SPIHT source coder with the adjustable-protection RS channel coder (SPIHT-RS) were performed for various BERs. The Reed-Solomon coder is a linear block coder defined by (n, k, t) or (n, k) that has length n and contains $k = n - 2t$ information symbols and $2t$ symbols of redundancy. The code can correct at most t symbol errors in the n symbol block, or $2t$ erasures when the location of the erasures is known [13]. For example, RS codes exist for various values of n, including $n = 2^m - 1$ corresponding to a Galois field GF(2^m) where m is the symbol length in bits.

When an (n, k, t) code has more than t symbol errors, the decoder will be unable to recover the original k data symbols and generally stops. When more than t symbol errors occur, analysis of the number of errors in the decoded codeword is difficult. To simplify the analysis, we consider the worst case that an entire code n is lost when more than t errors occur. With this assumption, the probability that a code is decoded incorrectly is determined by the binomial probability density function (pdf)

$$P_b(t) = \sum_{v=t+1}^{n} \binom{n}{v} p_s^v (1 - p_s)^{n-v} \tag{26}$$

where p_s is the symbol-error probability. For systematic encoders, the knowledge that an error has occurred can be used at the source decoder to perform better reconstruction.

When any of the channel symbols making up the code symbol are in error, the entire symbol is lost. The symbol error probability thus becomes $p_s = 1 - (1 - p)^m$ using the binomial pdf, where p is the channel bit-error probability. By using the expression for block-error probability from Eqn. (26) in Eqn. (6), the end-to-end distortion for the progressive adjustable protection system can be obtained. The optimization algorithm given in Figure VI-B can be used to find the optimal number of protection symbols $2t$ in each block. Based on simulations with the rate-distortion points of the Lena image, it was found that the PSNR improvement of allocating a different t to each block rather than a fixed level of protection t for each block was negligible. The protection symbols allocation algorithm was applied to the SPIHT source coder and an adjustable rate RS coder with fixed protection symbols $2t$ in each block.

Figure 6 shows the performance of the SPIHT-RS system and various fixed RS (n, k) systems. Whereas the fixed RS systems achieve the optimal PSNR at a single design BER, the adaptive SPIHT-RS system varies the coding rate according to the channel BER to achieve the optimal PSNR over the entire range of BERs. As in the fixed-energy systems, the fixed-rate systems have a performance drop-off as the BER is increased, and they reach a suboptimal plateau as the BER is decreased. The adaptive coder performs very well over a large range of BERs until the error-correction capacity of RS coding is exceeded. Comparing the adaptive system to the SPIHT distortion function evaluated at the capacity of the Binary Symmetric Channel (BSC), it is clear that better codes are needed at higher BERs to achieve capacity.

The performance of the SPIHT-RS system and the SPIHT-VE system can be compared by

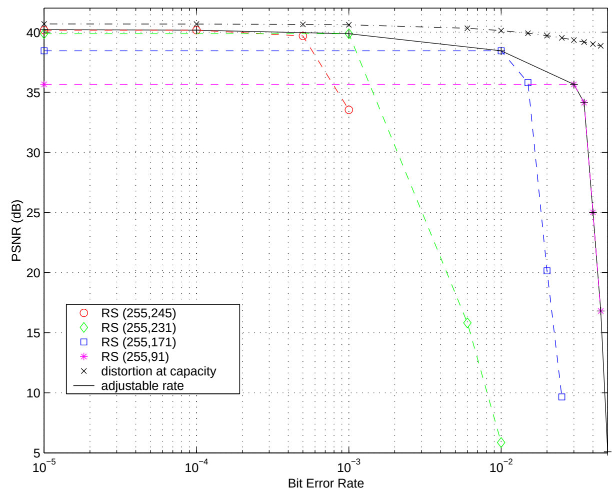

Fig. 6. Simulation results for Lena using SPIHT source coder and adjustable-protection RS channel coder.

converting the BER of the BSC to an equivalent E_{tot} for BPSK over an AWGN channel according to Eqn. (23). Figure 7 shows that the SPIHT-RS system performs better than the SPIHT-VE system when E_{tot} is larger than 5×10^5. According to the flexibility of the respective coders, the SPIHT-VE coder performs better in low-SNR channels where energy allocation is important, whereas the SPIHT-RS coder performs better in low-error channels where infrequent errors can be corrected using FEC. These results suggest that a combined variable-energy variable-rate channel coder would have better performance than either of these coders. Such a coder can also be optimized using the methods presented in this paper.

C. SPIHT RCPC combination

We consider minimizing the end-to-end distortion of a SPIHT source coder and an RCPC channel coder using BPSK symbols over an AWGN channel. For the convolutional coder, we consider the RCPC coders introduced by Hagenauer [14]. RCPC codes can be generated using a flexible encoding technique in which the encoding rate of a fixed-rate convolutional code can be varied by puncturing the generator matrix to yield a higher-rate code. RCPC codes are a natural fit for source-channel coding, because the basic structure of the encoder and the decoder remains the same as the coding rate changes.

RCPC codes are described by a mother code of rate $R = 1/N$, memory M, having an $M \times N$ generator tap matrix. The puncturing period P determines the range of code rates $R = \frac{P}{P+l}$

with $l = 1, \ldots, (N-1)P$. The performance of RCPC codes over various channels is given in terms of distance spectra $\{a_d\}$ and $\{c_d\}$, which depend on the code. The general expression for the first error event probability is

$$p_f \leq \frac{1}{P} \sum_{d=d_{free}}^{\infty} a_d P_d \tag{27}$$

A table of distance spectra for $M = 4$ and $P = 8$ is provided in [14]. The expression for P_d is based on the type of channel and amount of channel information. For an AWGN channel,

$$P_d = \frac{1}{2} Q\left(\sqrt{\frac{2dE_s}{N_0}}\right) \tag{28}$$

For a Rayleigh fading channel with soft decisions and full information of the received signal amplitude,

$$P_d \leq \frac{1}{2}\left(1 + \frac{E_s}{N_0}\right)^{-d} \tag{29}$$

The block-error probability for a block of B bits becomes

$$P_b = 1 - (1 - p_f)^B \tag{30}$$

where the second term is the probability that the first error event does not occur in any of the B bits in the block.

The end-to-end distortion of the SPIHT-RCPC source-channel coder can be expressed by substituting the expressions for P_b given in Eqn. (30) into Eqn. (19). To find the optimal rate r_s, we can use an algorithm similar to the one in Figure VI-C because this is just a rate-allocation problem. Once a value for r_s is found, we can choose the value of l that comes closest to the optimal rate. That is, we choose the smallest l such that $\frac{P}{P+l} \leq \frac{r_s}{r_{cs}}$. Another option is to exhaustively search the possible values of l, because there are only a limited number for a fixed choice of M and P.

For the SPIHT source coder, RCPC channel coder combination (SPIHT-RCPC), we obtained PSNR results from simulation at various BERs. Figure 8 shows a comparison of the SPIHT-RS and SPIHT-RCPC coders. The SPIHT-RS coder performs better at lower BERs, whereas the SPIHT-RCPC coder performs better at higher BERs. These results follow closely the block-probability-of-error characteristic as determined by the respective P_b expressions.

D. SPIHT adjustable-rate adjustable-protection combination

Motivated by the results of the SPIHT-VE, SPIHT-RS, and SPIHT-RCPC coders, we simulated the performance of a SPIHT adjustable-rate adjustable-protection combination (SPIHT-ARAP) coder that adjusts both the rate and the protection. For this source-channel combination, the end-to-end distortion given in Eqn. (6) simplifies to

$$E[D] = \sum_{i=0}^{M-1} D_s(ir_s) P_b (1 - P_b)^i \tag{31}$$

for a RS channel coder. The choice of the number of blocks, M, and number of protection symbols in each block, $2t$, determine the end-to-end system performance. We apply a combination of the algorithms given in Figures 2 and VI-B to select t and M. We apply a conjugate method where each parameter t or M is updated independently followed by the other parameter.

Figure 9 shows a comparison of the performance of the SPIHT-ARAP coder with Sherwood and Zeger's (SZ) coder introduced in [15]. These results show that the two systems are very similar in performance. The SZ coder performs better at lower E_{tot}, whereas the SPIHT-ARAP coder performs better at higher E_{tot}. In low-noise situations such as an average BER of 0.001, our results are 0.62 dB better than results for the SZ coder. In high noise situations such as an average BER of 0.1, our results are 0.86 dB lower than the results for the SZ coder. The primary reason for the difference in performance is the channel-coding scheme, because both source-channel combinations use the SPIHT source coder. While the SZ coder uses an RCPC coder designed for high BERs, the SPIHT-ARAP coder uses an RS coder which works better at low BERs.

The similarity in the performance of the SZ coder and the SPIHT-ARAP coder brings up a significant strength of our matching technique. The SZ coder is a image transmission system designed with a specific source coder and channel coder to take advantage of source-channel coding. The source-encoded stream is modified significantly in order to be matched to the chosen channel coder. On the other hand, the SPIHT-ARAP coder is obtained by applying the generic matching technique to a particular source coder and channel coder with only a basic model of the operation of the individual coders. These results suggest that it may be possible to achieve most of the benefits of fully joint source-channel optimizations using standard coders and the simple matching schemes proposed here.

E. Fading Channel

Many researchers have addressed the problem of finding accurate expressions for BER, symbol error probability, and block error probability over fading channels [16],[17]. Expressions for block-error rate in terms of coding rate and/or signal-to-noise ratio can be obtained from these papers and used directly in our work.

The extreme cases of fast fading and slow fading, where the received signal power varies quickly or slowly over time, respectively, are considered here. In these two cases, block-based error probabilities can be considered for the channel. Under the assumption that the symbol error probability stays the same over the feedback communication interval between the transmitter and receiver, slow fading can be combatted by updating the system parameters every interval. For fast fading, where the signal power is constant over each symbol, but varies from one symbol to the next, interleaving is a way to spread out the effect of the fade.

Consider the case of a perfect interleaver that interleaves the channel symbols so that the source coder sees the average BER of the channel. This is typically used over fast fading channels where it is difficult to design methods that take advantage of symbol correlation over time. The BER of block i for BPSK on a Rayleigh channel is found to be [13]

$$p(i) = \frac{1}{2}\left(1 - \sqrt{\frac{E_b}{2\sigma^2 + E_b}}\right) \tag{32}$$

where the energy per bit $E_b = E(i)/(mr_{sc})$. Using interleaving, the effect of the fade can be dispersed over many symbol blocks, and FEC can be overlayed to correct errors within individual blocks.

The only difference between the channel coder described here and the BPSK channel coder for an AWGN channel is the bit-error probability as a function of SNR. The results obtained in the previous sections apply to the fading channel described here by simply mapping the SNR of the fading channel to an equivalent SNR of the AWGN channel. Alternatively, to optimize a variable-energy channel coder for the fading channel, the optimization algorithm given in Figure 2 could be used with the following expression for the gradient. For BPSK over a fading channel with perfect interleaving, we obtain by differentiating Eqn. (32)

$$\frac{dp(i)}{dE(i)} = \frac{-1}{8\sigma^2\, m\, r_{sc}\left(\sqrt{\frac{E(i)}{2\sigma^2\, m\, r_{sc}}}(1 + \frac{E(i)}{2\sigma^2\, m\, r_{sc}})^{\frac{3}{2}}\right)} \tag{33}$$

F. JPEG adjustable protection combination

For the JPEG source coder and RS channel coder (JPEG-RS) combination, analytical simulations are performed for a fixed failure probability of 10^{-3} and various values of channel BER. For a particular BER, the optimization determines the optimal rate for the JPEG coder, which can be converted to an approximate Q factor for compression. To determine the performance of the system, the rate-distortion curve for the Lena image was evaluated at the chosen rate.

Results for the JPEG-RS coder, given in Figure 10, show that it performs reasonably well over a wide range of channel BERs. However, the JPEG-RS coder performs worse than the SPIHT-VE and SPIHT-RS coders over the entire range of BERs due to the superiority of the SPIHT coder.

IX. Conclusions

The general approach presented in this paper for matching a source coder with a channel coder can be applied to a wide variety of source-channel systems. The empirical rate-distortion model and the block-error channel model are sufficient to characterize most source and channel coders of interest, while allowing a general optimization framework to be developed. The simple and efficient algorithms that are obtained can be used in practical source-channel implementations.

Simulation results obtained for the Lena image demonstrated the effectiveness of the proposed methodology for a variety of source-channel combinations over a wide range of channel conditions. A significant strength of the technique is the ability to use the technique for real-time adaptation of the source-channel system to match the time-varying channel conditions. Using simulations, it was shown that our general approach works as well as more specific source-channel optimizations designed around specific coders. These results suggest that it may be possible to obtain most of the benefits of joint source-channel coding using the simple and general approach we propose. Simulation over a wide range of channel conditions shows the need for more flexible channel coders.

References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 623–656, 1948.

[2] J. G. Dunham and R. M. Gray, "Joint source and noisy channel trellis encoding," *IEEE Transactions on Information Theory*, vol. IT-27, no. 4, pp. 516–519, July 1981.

[3] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," in *1998 Data Compression Conference*, 1998, pp. 139–148.

[4] M. Bystrom and J. W. Modestino, "Combined source-channel coding for transmission of video over a slow-fading Rician channel," in *International Conference on Image Processing*, October 1998.

[5] J. Lu, Aria Nosratinia, and B. Aazhang, "Progressive source-channel coding of images over bursty error channels," in *International Conference on Image Processing*, October 1998.

[6] S. Appadwedula, D. Jones, K. Ramchandran, and L. Qian, "Joint source channel matching for wireless image transmission," in *International Conference on Image Processing*, Chicago, IL, October 1998.

[7] V. Chande and N. Farvardin, "Joint source-channel coding for progressive transmission of embedded source coders," in *1999 Data Compression Conference*, Snowbird, UT, 1999.

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

[9] L.-J. Lin, A. Ortega, and C.-C. Jay Kuo, "Cubic spline approximation of rate and distortion functions for MPEG video," in *Visual Communication and Image Processing*, Orlando, FL, March 1996.

[10] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[11] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[12] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[13] D. E. Blahut, *Digital Transmission of Information*, Addison-Wesley, Reading, MA, 1990.

[14] J. Hagenauer, "Rate-compatible punctured convolutional codes and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–399, Apr. 1988.

[15] G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," in *1997 Data Compression Conference*, Snowbird, UT, March 1997.

[16] M. K. Simon and M.-S. Alouini, "A unified approach to the performance analysis of digital communication over generalized fading channels," *Proceedings of the IEEE*, vol. 86, no. 9, pp. 1860–1877, September 1998.

[17] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1468–1477, November 1998.

Channel Coder	Source-Channel Combination	Description
Variable Energy (VE)	SPIHT-VE	Energy-constrained BPSK over AWGN
		Vary the energy in each block
Adjustable Rate (AR)	SPIHT-AR	Energy-constrained BPSK over AWGN
		Adjust the number of blocks
Adjustable Protection (AP)	SPIHT-RS	BSC with total rate constraint r_{sc}
		Adjust number of source (r_s)
		and channel (r_c) symbols
		in a Reed-Solomon codeword
		r_s and r_c same for all the blocks
	JPEG-RS	Same as SPIHT-RS
	SPIHT-RCPC	BSC with total rate constraint
		Adjust the source/(source+channel) rate of a
		Rate Compatible Punctured Convolutional Coder
		All blocks have the same source to
		total rate ratio of $P/(P+l)$
Adjustable Rate and	SPIHT-ARAP	Energy-limited BSPK with
Adjustable Protection		Reed-Solomon coding over an
(ARAP)		AWGN channel with rate constraint
		Adjust the number of blocks
		r_s and r_c same for all the blocks

TABLE I

SIMULATED SOURCE-CHANNEL CONFIGURATIONS.

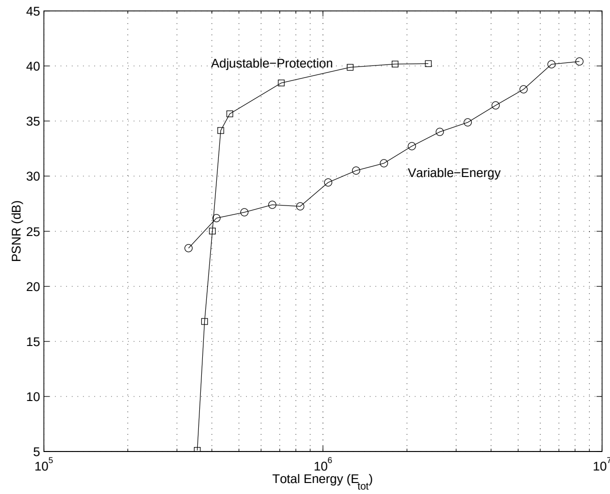

Fig. 7. Simulation results for Lena using SPIHT source coder and variable-energy and adjustable-protection channel coders.

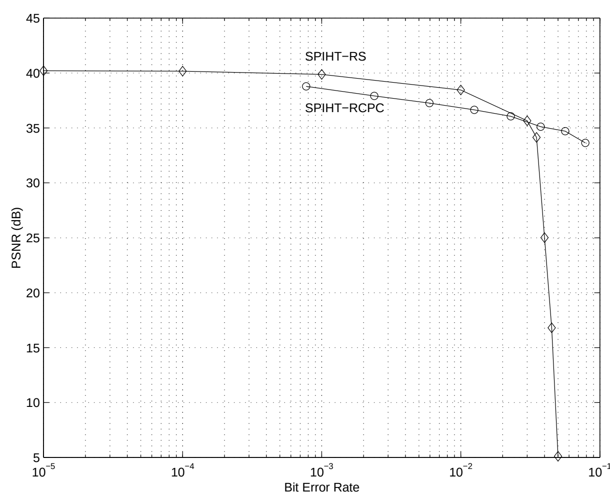

Fig. 8. Simulation results for Lena using SPIHT source coder and RS channel coder or RCPC channel coder.

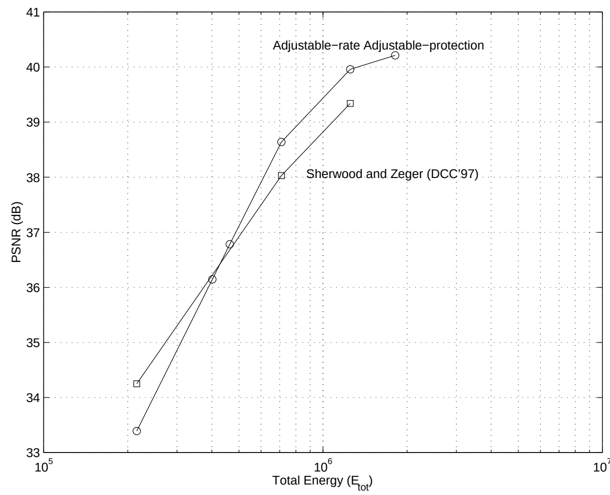

Fig. 9. Simulation results for Lena using SPIHT source coder and adjustable-rate adjustable-protection channel coder.

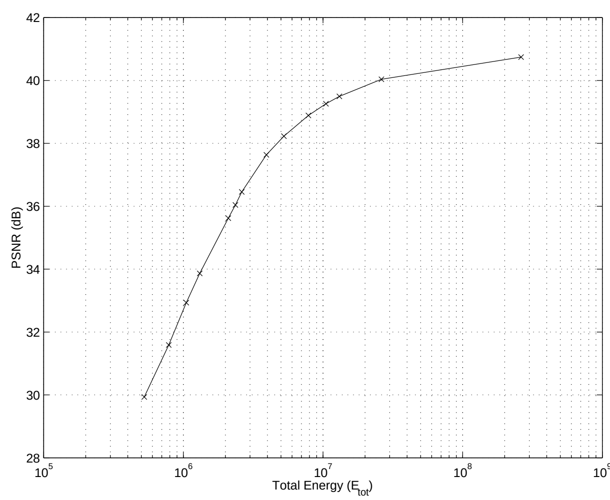

Fig. 10. Analytical simulation results for Lena using JPEG source coder and RS channel coder with a $Pr(\text{fail})$ of 10^{-3}.