Concatenated Block Codes for Unequal Error Protection of Embedded Bit Streams

Suayb S. Arslan, *Student Member, IEEE*, Pamela C. Cosman, *Fellow, IEEE*, and Laurence B. Milstein, *Fellow, IEEE*

***Abstract*—A state-of-the-art progressive source encoder is combined with a concatenated block coding mechanism to produce a robust source transmission system for embedded bit streams. The proposed scheme efficiently trades off the available total bit budget between information bits and parity bits through efficient information block size adjustment, concatenated block coding, and random block interleavers. The objective is to create embedded codewords such that, for a particular information block, the necessary protection is obtained via multiple channel encodings, contrary to the conventional methods that use a single code rate per information block. This way, a more flexible protection scheme is obtained. The information block size and concatenated coding rates are judiciously chosen to maximize system performance, subject to a total bit budget. The set of codes is usually created by puncturing a low-rate mother code so that a single encoder–decoder pair is used. The proposed scheme is shown to effectively enlarge this code set by providing more protection levels than is possible using the code rate set directly. At the expense of complexity, average system performance is shown to be significantly better than that of several known comparison systems, particularly at higher channel bit error rates.**

***Index Terms*—Concatenated block coding, embedded bit streams, fine-grain scalability, H.264, joint source–channel coding (JSCC), JPEG2000, packetization, rate allocation, set partitioning in hierarchical trees (SPIHT), unequal error protection (UEP).**

I. Introduction

EMBEDDED multimedia encoders produce bit streams that are progressive in nature, i.e., every successive bit contributes a certain amount of refinement after decoding. This partial decoding property comes with the cost that the usefulness of correctly received bits depends on the reliable reception of the previous bits [1]. This makes progressive bit streams vulnerable to noisy channel effects, i.e., any error renders the remaining information bits useless even if they are reliably received. Therefore, an efficient error-protection scheme is needed for the transmission of embedded bit streams over noisy channels.

Manuscript received August 27, 2010; revised April 24, 2011 and August 05, 2011; accepted August 29, 2011. Date of publication September 08, 2011; date of current version February 17, 2012. This work was supported in part by LG Electronics, Inc., Intel, Inc., the National Science Foundation under Grant CCF-0915727, the Center for Wireless Communications, University of California, San Diego, and the University of California Discovery Grant Program of the State of California. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Mary Comer.

The authors are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: sarslan@ucsd.edu; pcosman@ucsd.edu; lmilstein@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2167342

Error protection for progressive sources is usually achieved by joint source–channel coding (JSCC), in which an appropriate channel code is used to protect the bit stream to optimize some criterion, such as minimization of distortion or maximization of the useful source rate. Unequal error protection (UEP) using forward error-correction (FEC) coding of progressive sources is extensively studied in the literature. Early studies include [2], where cyclic redundancy check (CRC) codes are cascaded with FEC coding to protect set partitioning in hierarchical trees (SPIHT) progressively coded images. This was shown to be more robust than previous image-encoding results for binary symmetric channels (BSCs). Error propagation is avoided by stopping the decoding when the first CRC failure is detected. Later, [3] used a similar idea at the packet level to provide UEP by using nonuniform channel coding throughout the bit stream. This way, a more flexible coding scheme is achieved. Rate-compatible punctured convolutional (RCPC) codes are also considered by applying different channel code rates for different kinds of bits (e.g., sign and significance bits) of SPIHT encoded data [4]. More powerful codes have been also used for the transmission of embedded sources. In particular, turbo codes are considered in [5] and [6], irregular repeat–accumulate (IRA) codes in [7], and low-density parity-check (LDPC) codes in [8] and [9]. Some of the gains reported in [5]–[9] compared with [2]–[4] can be attributed to the superiority of capacity-achieving codes over conventional coding schemes rather than strictly attributing the gains to the manner in which FEC is deployed. In addition, those studies use larger block sizes to exploit the "asymptotically good" performance of LDPC or IRA codes compared with studies that use convolutional codes. In other studies, Reed–Solomon (RS) codes are utilized for graceful degradation of image quality in packet erasure channels [10]. One of the constraints of these studies is that they use a discrete channel code set usually in the form of a mother code and an associated puncturing pattern. Therefore, only a predetermined number of protection levels are possible with this finite set. Concatenated coding can loosen this constraint significantly and provide a more flexible system.

Concatenated coding was introduced in [11], where the total bit stream is first encoded with an inner code and then the coded bit stream is further encoded with an outer code. This constructs a code that has an exponentially decreasing error probability with increasing block length [11]. It is also shown to be particularly effective in bursty environments. A typical use of concatenated codes can be found in space communications and usually involves a convolutional inner code and an RS outer code. Concatenated coding can also be used for UEP of progressive sources. In particular, product codes were used across the transmitted packets as a 2-D code structure to provide UEP

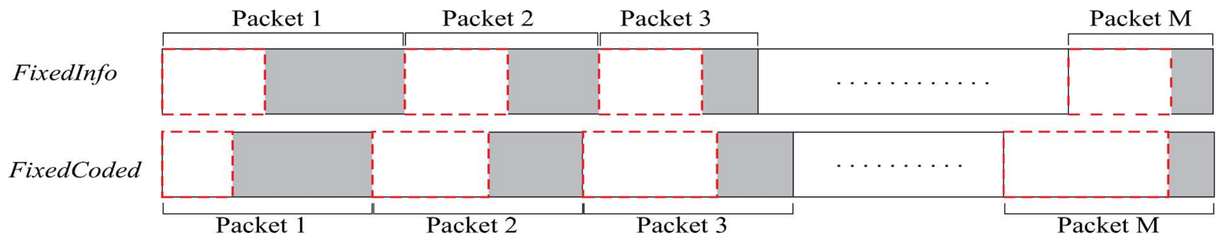


Fig. 1. Two methods of packet formatting for progressive source transmission.

for progressive sources in fading channels [12]. Later, product codes were shown to be very effective in the transmission of JPEG2000 [13] coded bit streams, particularly in correlated-Rayleigh fading using turbo and RS codes [14], [15]. The main objective of those studies is to use concatenated coding with RS coding to help improve system performance against packet erasures due to channel fading.

In addition to the choice of channel coding, packet formatting or packetization is another important parameter in a FEC-based overall system design for progressive source transmission. Packetization is typically done either by fixing the size of the information block [2], [16] or by fixing the total number of coded bits [5], [6]. These packet formats are shown in Fig. 1 and named *FixedInfo* and *FixedCoded* formats hereafter. For both methods, rate-maximization and distortion-minimization problems for various channel-coding choices are solved, and fast algorithms are proposed for the former [17]. The performance of the overall system design can vary based on the choice of the channel code set and the associated packetization format.

In this paper, we propose a novel packet formatting method along with a flexible system design for the transmission of embedded bit streams. By properly adjusting the channel codes, a flexible protection assignment strategy is achieved through the use of serially concatenated coding and random block interleavers. As is common, interleaving is added between concatenated codes to spread out the burst errors. Subject to a total bit budget constraint, we obtained notable performance improvement over some of the conventional methods. This paper focuses on BSCs similar to [2], [3], [5], [16], and [17]. In some situations, soft channel information is not available, and the only access to the data occurs after some form of hard-decision decoding takes place. In this case, a BSC is a reasonable model for any memoryless channel with a binary input such as binary input Gaussian and flat Rayleigh/Rician fading channels. Note that the proposed serial encoding scheme can be combined with RS codes similar to [12], [14], and [15] to create a 2-D product code to help mitigate the correlation among the random channel coefficients in correlated fading channels.

The remainder of this paper is organized as follows: In Section II, the proposed formatting is explained, and the problem formulation is introduced. In Section III, performance, as well as the algorithm design, is addressed, and the optimization framework is discussed. A low-complexity descent search algorithm is given as an alternative to the overall optimization problem. We also compare the decoding complexity of the proposed design to previous coding architectures. Section IV presents some of the numerical results using different sets of channel codes. We derive two conditions that restrict the search space of possible channel code rates. These two conditions lead to significant complexity reductions of the optimization process. Finally, a brief summary and conclusion follow in Section V.

II. Proposed Formatting and System Model

Similar to previous studies [2], [3], error detection is achieved by appending a CRC to every information block $\mathcal{I}_i, 1 \le i \le M$. Once an error is detected, all additional bits are excluded from the reconstruction process.

Consider the proposed M-codeword scheme shown in Fig. 2 and assume that a discrete convolutional code set $\mathcal{C}$ is chosen. First, we take b_1 source information bits (source block $\mathcal{I}_1$) and derive two bytes of CRC[1] $(N_c = 16 \text{ bits})$ based on b_1 bits to be concatenated with $\mathcal{I}_1$ for bit error detection. In addition, z zero tail bits are appended to end the trellis in the all-zero state. These bits together constitute payload $\mathcal{P}_1$ of the first codeword. The number of bits in the payload is $|\mathcal{P}_1| = b_1 + N_c + z$. These bits are encoded using channel code rate $r_1 \in \mathcal{C}$ to produce codeword c_1. In the next encoding stage, c_1 is concatenated with the second information block $\mathcal{I}_2$, N_c CRC bits, and z parity bits to produce the second payload $\mathcal{P}_2$, where $|\mathcal{P}_2| = [(b_1 + N_c + z)/r_1] + b_2 + N_c + z$ and $|\mathcal{I}_2| = b_2$. In the second encoding stage, N_c CRC bits are derived based on those b_2 bits. Since the errors out of a maximum-likelihood sequence estimator are bursty and convolutional codes show poor performance when channel errors are not independent and identically distributed (i.i.d.) [18], we use random block interleavers to break up the long burst noise sequences. We use $\boldsymbol{\pi}(x)$ to denote the random block interleaving function that chooses a permutation table randomly, with a uniform distribution, and permutes the entries of x bitwise based on this table. We choose the size of the random permutation table to be equal to the length of the payload size in each encoding stage except the first. After interleaving, the bits in $\boldsymbol{\pi}(\mathcal{P}_2)$ are encoded using code rate $r_2 \in \mathcal{C}$ to produce codeword c_2. This recursive process continues until we encode the last codeword c_M. Codeword c_M is sent over the BSC.

The receiver obtains the noisy version of codeword c_M, i.e., $\widehat{c}_M$. At the decoder, the sequential encoding operations of the encoding stage are performed in reverse order. In other words, the noisy received version $\widehat{c}_M$ is decoded first using the ordinary Viterbi decoder. Then, the deinterleaver is invoked to ob-

[1]Here, a CRC polynomial is chosen that gives a sufficiently small undetected error probability [19], and the same CRC polynomial is used for all information blocks. The selected CRC polynomial is $X^{16} + X^{12} + X^5 + X$.

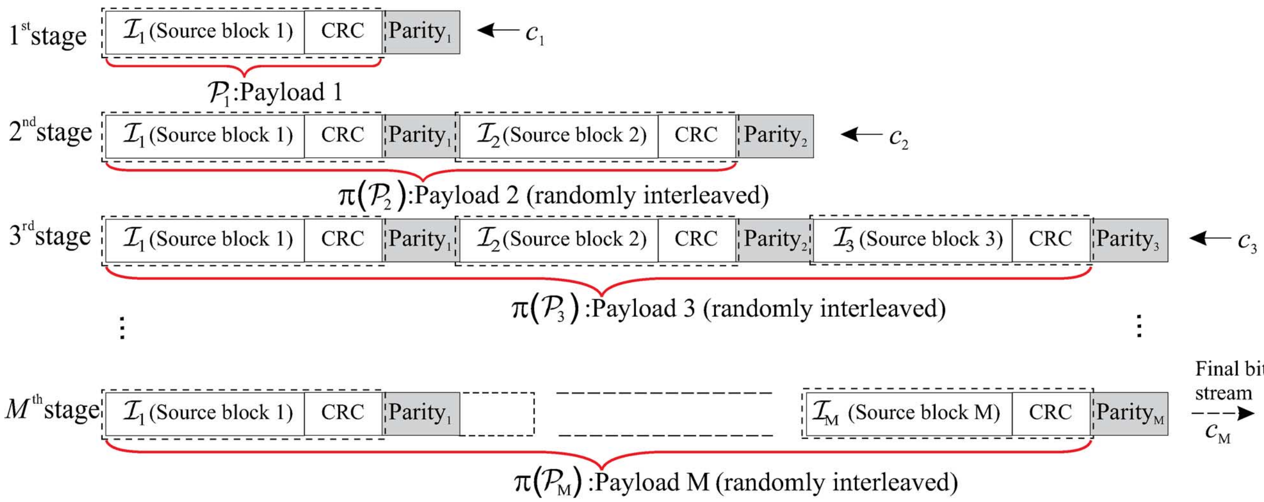


Fig. 2. Proposed formatting consisting of M stages of encoding.

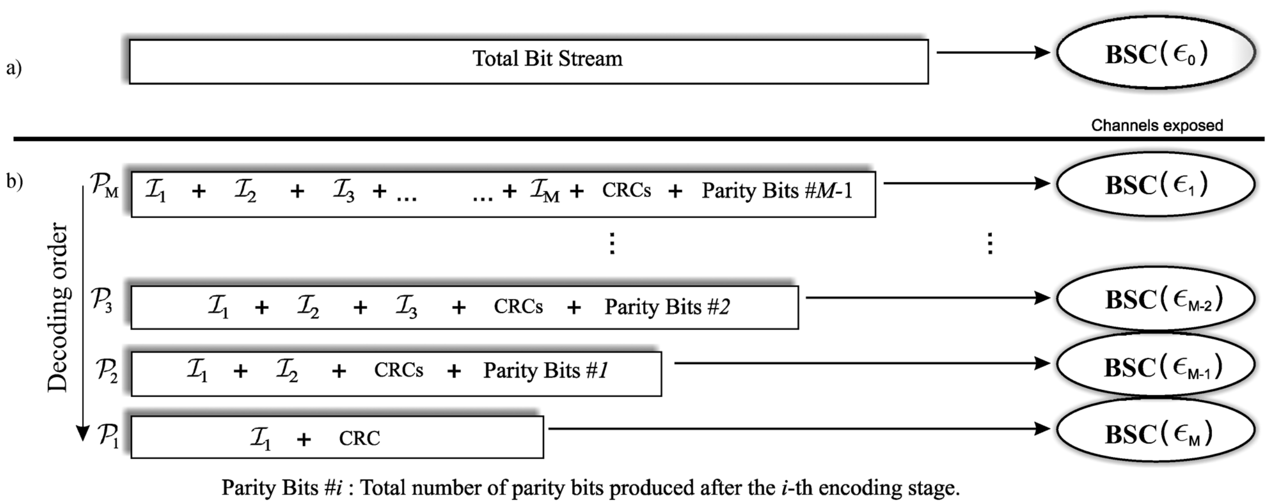


Fig. 3. Each payload in the embedded bit stream is exposed to a BSC with different crossover probabilities.

tain $\boldsymbol{\pi}^{-1}(\boldsymbol{\pi}(\widehat{\mathcal{P}}_M)) = \widehat{\mathcal{P}}_M$. The CRC of the Mth information block, i.e., $\mathcal{I}_M$, is performed to label $\mathcal{I}_M$ as useful or not for the reconstruction process. Thus, $\mathcal{I}_M$ is associated with a label and peeled off from $\widehat{\mathcal{P}}_M$. In the next decoding stage, $\widehat{c}_{M-1}$ is decoded and deinterleaved in the same manner to get $\widehat{\mathcal{P}}_{M-1}$. Based on the CRC, $\mathcal{I}_{M-1}$ is determined to be useful or not in the reconstruction. The decoding operation is finalized after decoding codeword c_1 and assigning a label to $\mathcal{I}_1$. Assuming that the first label with a CRC failure is associated with $\mathcal{I}_l$, then only the information blocks up to, but not including, block l are used to reconstruct the source.

III. Algorithm Design and Optimization

We define two sets $\mathcal{R} = \{r_1, \ldots, r_M\}$ and $\mathcal{B} = \{b_1, \ldots, b_M\}$, where $r_i \in \mathcal{C}$ is the code rate used to protect the ith payload $\mathcal{P}_i$. Those sets are optimized to minimize distortion, as will be explained in Section III-A. The length of codeword n ($|c_n|$), $1 \le n \le M$, can be found as

$$|c_n| = \left(\cdots\left(\left((b_1 + N_c + z)\frac{1}{r_1} + b_2 + N_c + z\right)\frac{1}{r_2} + b_3 + N_c + z\right)\cdots\right)\frac{1}{r_n}$$
$$= \sum_{i=1}^{n}\left(\prod_{j=i}^{n} r_j\right)^{-1}(b_i + N_c + z). \quad (1)$$

The length of codeword c_M is equal to the total allowed bit budget B, i.e., $|c_M| = B$. We send the total bit stream over a BSC with crossover error probability ϵ_0, as shown in Fig. 3(a). Because of the embedded nature of M convolutional codes, the various test statistics are dependent; thus, a closed-form analysis does not appear to be feasible. Therefore, for the purpose of algorithm design, we will ignore the dependence and set up the algorithm assuming statistical independence. The net result of this is that we approximate the sequence of data bit decisions at the output of each decoding stage as arising from a BSC, whereby the crossover probability of each BSC is functionally related to the crossover probability of the previous BSC, as well as the rate of the currently used channel encoder, as shown in Fig. 3(b). Note that, since our final results on performance are obtained by simulations, the dependence that was ignored in the algorithm design will be automatically present in the performance results.

After decoding the noisy codeword $\widehat{c}_M$ and deinterleaving, we assume that the decoded bit stream will have a bit error rate (BER) $\epsilon_1 < \epsilon_0$. In light of the previous paragraph, the decoded error rate ϵ_1 is a function of ϵ_0 and channel code rate r_M and is denoted by $\epsilon_1(\epsilon_0, r_M)$ hereafter. Next, the noisy codeword $\widehat{c}_{M-1}$ is decoded, etc. In general, the channel that $\mathcal{P}_i$ $\{1 \leq i \leq M\}$ experiences can be approximated as a BSC with crossover probability $\epsilon_{M-i+1}(\epsilon_{M-i}, r_i)$. Since $\mathcal{P}_i \supseteq \mathcal{I}_i$, the BER for $\mathcal{I}_i$ is $\epsilon_{M-i+1}(\epsilon_{M-i}, r_i)$. In this paper, $\{\epsilon_{M-i+1}\}_{i=1}^{M}$ are found by simulating the system for a given channel raw error rate ϵ_0 and code rates $r_i, r_{i+1}, \ldots, r_M$. As shown, the packet error rate (PER) appears in the expression for the expected distortion of the system. In principle, it is possible to determine the PERs directly by simulation. In practice, however, an information block can have any integer number of bits (up to the total length of the stream), and each such block size b_i corresponds to a different PER (PER_i). Therefore, in the optimization process, it would have been computationally intractable to find all possible PERs corresponding to all possible b_i values. For this reason, we instead use the simulation to find the BER and use that and the various b_i values to compute the PERs. Having approximated each channel as a BSC with some crossover probability, we approximate the PER of the ith information block, i.e., $\mathcal{I}_i$, by [20]

$$\text{PER}_i \approx 1 - (1 - \epsilon_{M-i+1})^{b_i}. \tag{2}$$

The described channel modeling is illustrated in Fig. 3(b). Using (2), the expected distortion of the progressive bit stream using the proposed formatting and the independence assumption is approximated by [21]

$$\mathbb{E}[\mathcal{D}] = \sum_{l=0}^{M} \text{PER}_{l+1} \prod_{i=0}^{l} (1 - \text{PER}_i) d_l \tag{3}$$

$$= d_0 - \sum_{l=1}^{M} \prod_{i=1}^{l} (1 - \text{PER}_i) \Delta_l \tag{4}$$

$$\approx d_0 - \sum_{l=1}^{M} \Delta_l \prod_{i=1}^{l} (1 - \epsilon_{M-i+1})^{b_i}$$

$$\triangleq d_0 - \overline{D}_M(\mathcal{B}, \mathcal{R}) \geq 0 \tag{5}$$

where d_0 is the distortion when the source decoder reconstructs the source without using any of the transmitted information, d_l is the distortion when the decoder decodes only $\sum_{j=1}^{l} b_j$ information bits, and $\Delta_l = d_{l-1} - d_l \geq 0$. We also assume $\text{PER}_0 = 0$ and $\text{PER}_{M+1} = 1$, as the zeroth and $(M+1)$th information blocks do not exist. Minimization of $\mathbb{E}[\mathcal{D}]$ is equivalent to maximizing the average distortion reduction $\overline{D}_M(\mathcal{B}, \mathcal{R})$. Note that the above expression corresponds to the expected distortion when all errors are assumed to have been caused by statistically independent events, as explained above. To help ensure that the errors are as uncorrelated as possible, we have introduced interleavers at the transmitter and corresponding deinterleavers at the receiver before each decoding stage.

A. Optimization

For the proposed scheme, the design objective is to find M, rate allocation $\mathcal{R}$, and information block size set $\mathcal{B}$ such that $\overline{D}_M(\mathcal{B}, \mathcal{R})$ is maximized. In other words, we intend to find set $\{M^*, \mathcal{R}^*, \mathcal{B}^*\} = \arg\max_{\{M,\mathcal{R},\mathcal{B}\}} \overline{D}_M(\mathcal{B}, \mathcal{R})$ subject to bit budget constraint B, where $*$ denotes the optimal values. For a given M, we need to optimize the overall parameter set $\{\mathcal{B}, \mathcal{R}\} = \{b_1, b_2, \ldots, b_M, r_1, r_2, \ldots, r_M\}$, which is subject to the total bit budget B. The problem can be stated as

$$\max_{\mathcal{B},\mathcal{R}} \overline{D}_M(\mathcal{B}, \mathcal{R}) \quad \text{subject to} \quad \sum_{i=1}^{M} \frac{1}{\prod_{j=i}^{M} r_j} (b_i + N_c + z) = B. \tag{6}$$

There are only either 9 RC-LDPC or 13 RCPC code rates in our set $\mathcal{C}$; therefore, including the uncoded case, we have either 10 or 14 possible values for r_i, whereas b_i has a vastly larger set of possible values (in theory, b_i can take on any value from 1 up to the total number of source bits). First, we assume a fixed channel code rate set $\mathcal{R}$ and optimize information block size set $\mathcal{B}$ using a descent search algorithm [22] by initializing $\mathcal{B}^* = \{b, b, \ldots, b\}$ for some b satisfying $Mb < B$.

This constrained optimization problem is reduced to solving an unconstrained optimization problem by using a line-search method.[2] We employ gradient vector $\nabla \overline{D}_M(\mathcal{B}, \mathcal{R})$ as the descent direction for fixed M and $\mathcal{R}$. Finally, we note that the elements of set $\mathcal{B}$ can only take discrete values (the unit is in *bits*). Thus, the components of the gradient vector are not analytically available. We use the "gradients by forward finite differences" as our approximation, as given in [22, Sec. 2.3.1.5]. Then, we carry out this optimization procedure for all possible channel code rates. Finally, for a given M, we choose set $(\mathcal{B}^*, \mathcal{R}^*) = \arg\{\max_{\mathcal{R}}\{\max_{\mathcal{B}} \overline{D}_M(\mathcal{B}, \mathcal{R})\}\}$ to be the optimal point. For ease of reference, we present a functional flow diagram of the optimization procedure in Fig. 4 to determine the optimum parameters $(M^*, \mathcal{B}^*, \mathcal{R}^*)$.

Let us fix code rate set $\mathcal{R} = \{r_i\}_{i=1}^{M} \in \mathcal{C}$. First, we realize that $\Delta_1 = d_0 - d_1$ depends on b_1 because d_0 is a fixed real number. Similarly, $\Delta_2 = d_1 - d_2$ depends on both b_1 and b_2. Thus, $\Delta_l = d_{l-1} - d_l$ depends on $b_1, b_2, \ldots, b_l$. To signify the functional dependence, they will be denoted by $\Delta_1(b_1)$, $\Delta_2(b_1, b_2)$, and $\Delta_l(b_1, b_2, \ldots, b_l)$ (see Fig. 5). For this general case, our constraint equation given by (6) becomes

$$\sum_{i=1}^{M} \frac{b_i}{\prod_{j=i}^{M} r_j} = B - (N_c + z) \sum_{i=1}^{M} \left(\prod_{j=i}^{M} r_j \right)^{-1}. \tag{7}$$

Letting $\widehat{B} \triangleq B - (N_c + z) \sum_{i=1}^{M} (\prod_{j=i}^{M} r_j)^{-1}$, we have for the lth information block

$$b_l = \left(\widehat{B} - \sum_{i=1,\, i \neq l}^{M} \frac{b_i}{\prod_{j=i}^{M} r_j} \right) \prod_{j=l}^{M} r_j. \tag{8}$$

[2]The line-search algorithms are summarized in [22, Sec. 2.1.1]. We used the golden section method (explained in [22, Sec. 2.2.1]) as our 1-D line-search approach. We found that a sufficient number of iterations for convergence is almost linear in M. The complexity of the optimization problem will be also discussed in the simulations section.

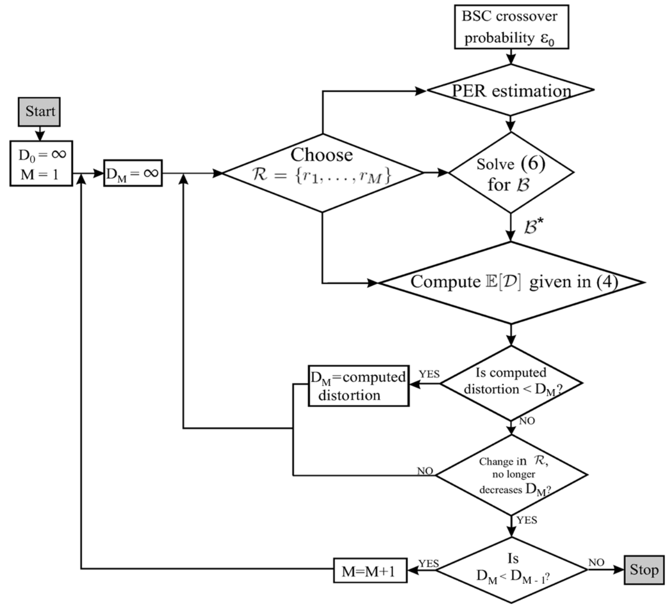


Fig. 4. Functional flow diagram of the proposed optimization and transmission scheme.

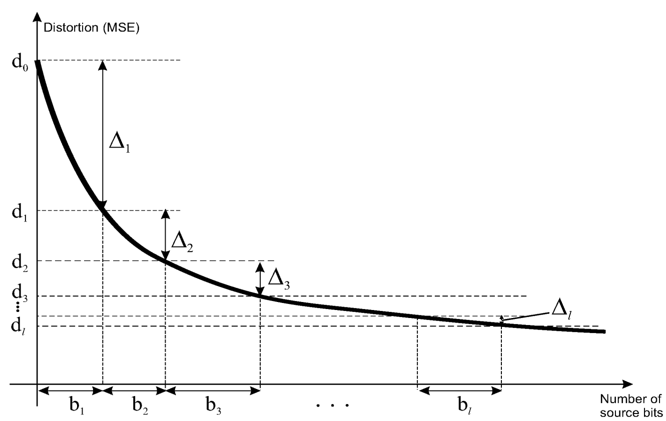


Fig. 5. Distortion versus number of source bits illustrating $\Delta_1, \Delta_2, \ldots, \Delta_l$ as functions of $b_1, b_2, \ldots, b_l$.

This means that one of the parameters, i.e., b_l, can be expressed in terms of the other parameters. For example, using (8), $b_M = r_M\widehat{B} - r_M\sum_{i=1}^{M-1}(b_i/\prod_{j=i}^{M} r_i)$, and

$$\Delta_M\left(b_1, b_2, \ldots, b_{M-1}, r_M\left(\widehat{B} - \sum_{i=1}^{M-1}\frac{b_i}{\prod_{j=i}^{M} r_j}\right)\right) \tag{9}$$

is now only a function of $b_1, \ldots, b_{M-1}$. Thus, using (5) and (9), we obtain

$$\begin{aligned}
&\overline{D}_M\left(\left\{b_1, \ldots, b_{M-1}, r_M\widehat{B} - r_M\sum_{i=1}^{M-1}\frac{b_i}{\prod_{j=i}^{M} r_i}\right\}, \mathcal{R}\right)\\
&= \sum_{l=1}^{M-1}\Delta_l(b_1, \ldots, b_l)\prod_{i=1}^{l}(1-\epsilon_{M-i+1})^{b_i}\\
&\quad + \Delta_M\left(b_1, b_2, \ldots, b_{M-1}, r_M\left(\widehat{B} - \sum_{i=1}^{M-1}\frac{b_i}{\prod_{j=i}^{M} r_j}\right)\right)\\
&\quad \times (1-\epsilon_1)^{r_M\left(\widehat{B}-\sum_{i=1}^{M-1}\frac{b_i}{\prod_{j=i}^{M} r_i}\right)}\prod_{i=1}^{M-1}(1-\epsilon_{M-i+1})^{b_i}
\end{aligned} \tag{10}$$

which is only a function of $\{b_1, \ldots, b_{M-1}\}$. The constrained optimization problem in (6) becomes unconstrained and has now $M-1$ parameters subject to optimization. We use the gradient descent algorithm to find a maximum $\{b_1^*, \ldots, b_{M-1}^*\}$ for this problem [22]. Finally, we calculate $b_M^* = r_M\widehat{B} - r_M\sum_{i=1}^{M-1}(b_i^*/\prod_{j=i}^{M} r_i)$.

A simple example that is easy to visualize (the concavity of $\overline{D}_M(\mathcal{B}, \mathcal{R})$ for a given $\mathcal{R}$) occurs when $M = 2$. In this particular case, $\Delta_2(b_1, b_2) = \Delta_2(b_1, r_2\widehat{B} - (b_1/r_1))$ is now only a nonincreasing function of b_1. After absorbing the constraint equality into the cost function, we will have an unconstrained optimization problem with the following cost function to maximize:

$$\begin{aligned}
&\overline{D}_2\left(\left\{b_1, r_2\widehat{B} - \frac{b_1}{r_1}\right\}, \mathcal{R}\right)\\
&\quad = (1-\epsilon_2(\epsilon_1(\epsilon_0, r_2), r_1))^{b_1}\,\Delta_1(b_1)\\
&\qquad + (1-\epsilon_2(\epsilon_1(\epsilon_0, r_2), r_1))^{b_1}\,(1-\epsilon_1(\epsilon_0, r_2))^{r_2\widehat{B}-b_1/r_1}\\
&\qquad \times \Delta_2\left(b_1, r_2\widehat{B} - \frac{b_1}{r_1}\right).
\end{aligned} \tag{11}$$

Several examples are shown in Fig. 6 for different values of ϵ_0 and r_{tr}, where we take $B = 0.3 \times 512 \times 512$ bits for a 512×512 *Lena* image using the RCPC code set with memory 6 from [23].

B. Concatenated Coding With LDPC Codes

The concatenated coding mechanism presented in this paper can be used with any type of error-correction code, i.e., we can employ a class of "asymptotically good" codes such as LDPC codes. However, there might be some minor changes in the proposed design when it is used with LDPC codes. For example, we would not need to consider the CRC codes in our design because of the inherent error-detection property of LDPC codes [26]. On the other hand, using LDPC codes increases both the computational complexity and memory requirements of the encoder. In the proposed design, the length of the codewords increases after each encoding stage. To enable linear encoding complexity, we resort to the semirandom LDPC encoding structure given in [27]. We optimize the system using the cost function, in which the approximate PER values are found using the analysis of LDPC codes given in [28] and the algorithm presented in [29], for finding the decoding thresholds of the semirandom rate-compatible LDPC (RC-LDPC) code family.

C. Complexity of the Proposed Scheme

As far as encoding of the convolutional codes is concerned, the encoders use the same amount of memory elements both in the proposed system and in the comparison systems of [2], [3], and [16]. Although convolutional encoding has a minor effect on the overall complexity, Viterbi decoding is a more complex operation. We focus on the decoding complexity at the receiver.

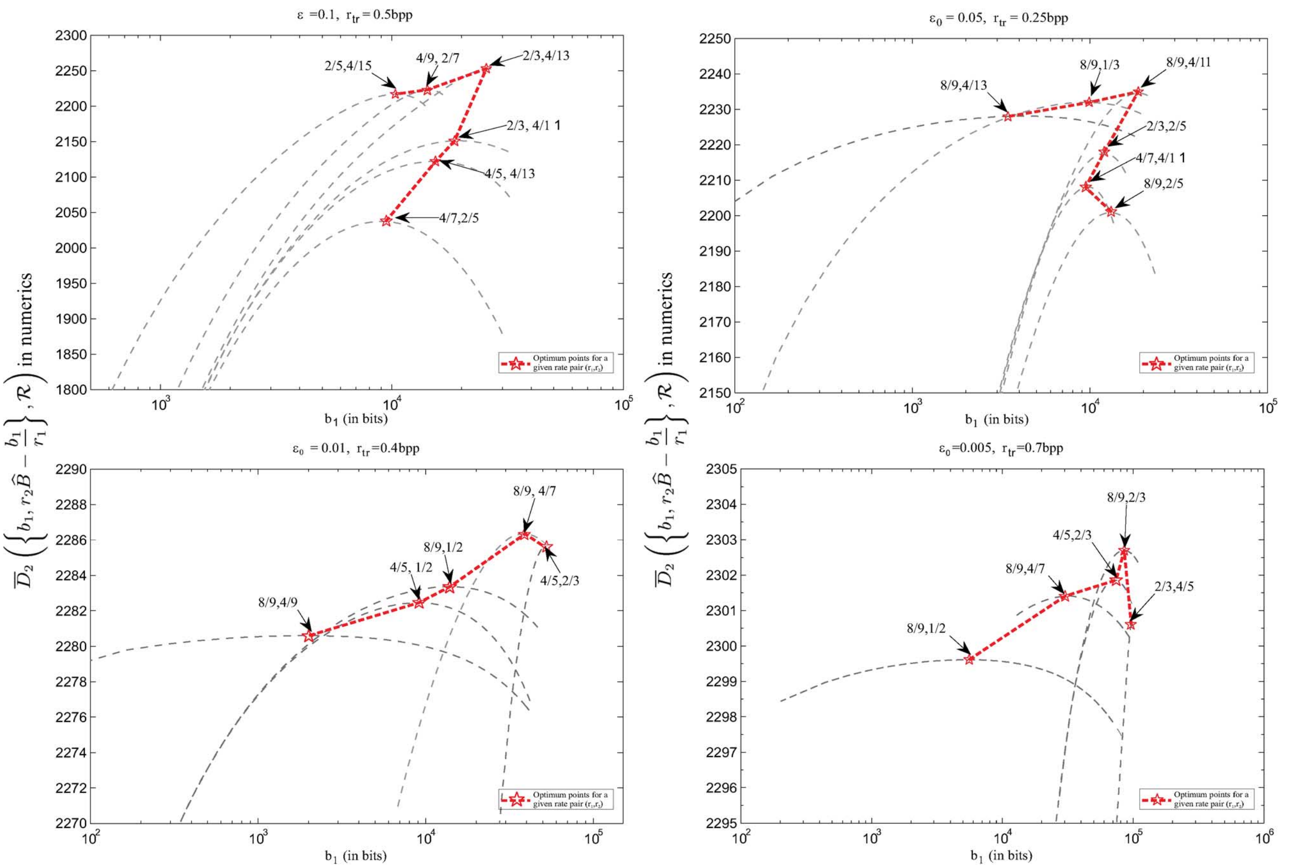


Fig. 6. For several different sets $\mathcal{R}$, ϵ_0, and r_{tr}, $\overline{D}_2(\{b_1, r_2\widehat{B} - (b_1/r_1)\}, \mathcal{R})$ is plotted for a SPIHT source encoder. As shown, it is a concave function of b_1 and has a single maximum point.

For a fixed-constraint length, the decoding complexity of punctured convolutional codes linearly increases with the length of the codeword [24]. In addition, the maximum trellis complexity of a random block interleaver linearly increases with the length of the interleaver [25]. On the other hand, since we use the LDPC encoding structure given in [27] and the Max–product algorithm at the receiver, the computation complexity of both LDPC encoding and decoding is linear with increasing block length. If we assume that the complexity of deinterleaving is minor compared with the decoding operation, the maximum trellis complexity of the decoding procedure of the proposed scheme is found to be on average at most M times more complex than the decoding operations of the systems in [2], [3], [9], and [16].

IV. Numerical Results and Discussions

We use the following channel code sets: 1) a convolutional code set that consists of the 13 RCPC codes with memory 6 found in [23] and used in [2] and 2) an LDPC code set that consists of the nine RC-LDPC codes used in [9]. The convolutional code is treated as a block code by using zero-tail padding with $z = 6$ zeros at the end of each information block.

A. Simulations With RCPC Codes

Throughout this section, the RCPC channel code set is used. First, we show some simulation results that illustrate the role of the interleavers in our system design. We set $M = 2$ and choose two pairs of codes (r_1, r_2), i.e., (2/3, 1/2) and (4/9, 1/2), and information block size (b_1, b_2), i.e., $(10^4, 10^4)$. In Fig. 7, the decoded BER is plotted for the bits in $\mathcal{I}_1$ as a function of crossover probability ϵ_0 with two different permutation table sizes, i.e., $|\mathcal{P}_1| = 1500$ and $|\mathcal{P}_1| \approx 10^4$. As expected, if we increase the size of the random permutation table, we get better BER performance. Also shown in the same figure is the decoded BER for the bits in $\mathcal{I}_2$ (single code rate 1/2). The curve named *non-interleaved* shows that, when there is no interleaver, we obtain very minor performance improvement over the single code rate scheme. This is because correlated error patterns at the output of the Viterbi decoder significantly affect the performance of the following decoding stage. At higher crossover error probabilities, the decoded BER for $\mathcal{I}_1$ is worse than that for the single code rate scheme. To explain this, first observe in Fig. 7(a) that toward the right-hand edge of the plot (roughly $\epsilon_0 > 0.1$), the outer decoder is overwhelmed by the high raw channel error rate. As we move to the left, the outer decoder is not overwhelmed. However, the decoded BER of the outer decoder has to be low enough so that the inner encoder is not overwhelmed. Finally, toward the left-hand edge of the plot (roughly $\epsilon_0 < 0.09$), neither decoder is overwhelmed and give lower decoded BERs than ϵ_0.

Finally, we obtained the curve denoted *i.i.d. Assump.* in Fig. 7 in the following way: For any given raw channel error rate ϵ_0, we first decoded outer code r_2 and obtained decoded BER ϵ_2

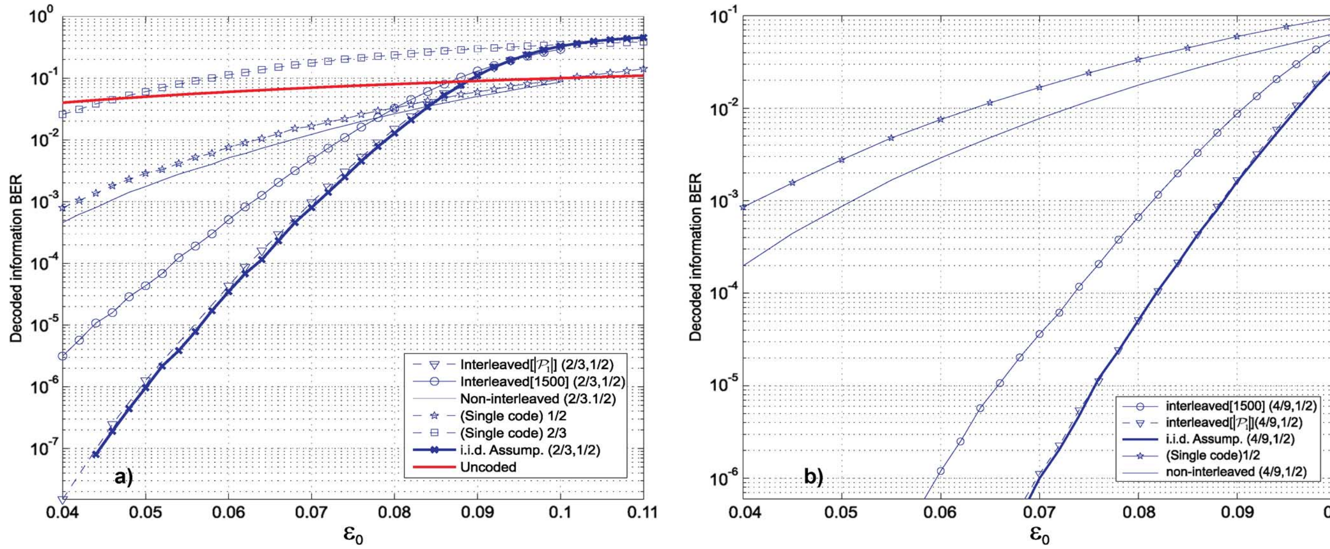


Fig. 7. Effect of interleaving with different sizes of the random permutation tables.

by simulation. After finding ϵ_2, we sent the inner codeword (information block encoded with r_1) through a BSC with crossover probability ϵ_2 and obtained decoded BER ϵ_1 by simulation again (argument given in Section III). The curve *i.i.d Assump.* given in Fig. 7 is the set of ϵ_1s for different values of ϵ_0 shown in the abscissa. It is verified that an interleaver depth of $|\mathcal{P}_1|$ using a block length of 10^4 gives a very close approximation to the i.i.d. assumption.

In the rest of the simulations, we use an embedded bit stream, and we use our protection scheme to send it over a BSC. We compare four transmission schemes based on embedded bit streams.

1) *ShZeg:* This is based on [2]. It uses a single optimal code with *FixedInfo* formatting. The information block size is 200 bits, and a single optimal code rate is chosen from the code set that minimizes the average distortion for all the transmission rates in consideration. In addition, the list Viterbi decoder (LVA) of [2] is replaced with the ordinary Viterbi algorithm.
2) *NosLu:* This is the approach presented in [3]. It uses an optimal code rate for each information block with *FixedInfo* formatting [21]. The information blocks have size of 202 bits.
3) *Product Code:* This system was originally proposed in [12] for a correlated fading channel, but the effectiveness of the scheme was shown for a BSC as well. It uses RS codes in concatenation with convolutional codes as a 2-D code structure.
4) *Concatenated:* This is the proposed concatenated coded system with related optimization as defined.

We produced the embedded bit stream by encoding the grayscale 512 $\times$ 512 *Lena*, *Barbara*, and *GoldHill* images using SPIHT with arithmetic coding and JPEG2000 *Part 1* (with no error-correction capability) progressive image coders. Due to space limitations, we only present the results of *Lena* encoded using SPIHT with arithmetic coding and *Goldhill* encoded with JPEG2000. Other combinations exhibited similar results. We tested two crossover error probabilities, namely, $\epsilon_0 = 0.01$ and $\epsilon_0 = 0.05$. We define the total transmission rate $r_{\rm tr}$ in bits per pixel (bpp) for a given $L_x \times L_y$ image as $r_{\rm tr} = B/(L_x \times L_y)$. Quality assessment of the decoded images is given in terms of the average peak signal-to-noise ratio (PSNR). PSNR results for all the systems are shown as a function of $r_{\rm tr}$ in bpp, where

$$\text{PSNR} = 10 \log_{10}\left(\frac{MAX_I^2}{\mathbb{E}[\mathcal{D}]}\right) \tag{12}$$

and $\text{MAX}_I = 255$ is the maximum possible intensity value of the image.

For a given BSC with crossover probability ϵ_0 and sets $\mathcal{B}$ and $\mathcal{R}$, we produce the total bit stream c_M, as explained in Section II, and send it through the BSC. We obtain $\{\epsilon_i\}_{i=1}^M$ by simulating M sequential decoding stages. Note that, given sets $\mathcal{B}$ and $\mathcal{R}$, the sizes of the random permutation tables are automatically determined. We should also notice that $\{\epsilon_i\}_{i=1}^M$ are assumed to be independent of $\{b_i\}_{i=1}^M$, as the decoded BER of RCPC codes only slightly changes with varying block size [30]. Finally, we calculate $\mathbb{E}[\mathcal{D}]$ using the approximations given in (2) and (5). After optimization, we find the optimal parameters to be used in the proposed scheme. Finally, using the optimal set of parameters $(M^*, \mathcal{B}^*, \mathcal{R}^*)$, we simulate the system to find the expected distortion and convert it to PSNR using (12). As shown in Fig. 8, the proposed system is more effective at higher channel error rates and higher transmission rates.

Another observation is that, although *NosLu* optimizes channel code rates for each information block, because of the flat region of the R–D curve of the source, *NosLu* does not improve much over *ShZeg* at higher transmission rates. However, at smaller transmission rates, *NosLu* gives around 0.35-dB gain over the *ShZeg* scheme as reported in [3]. In addition, note that [16] does not use the approximation in [3] and finds the optimal rate schedule using dynamic programming instead of Lagrangian multipliers. However, we experimentally observed that the performance loss due to the approximation in [3] is minor. *Product Code* for the *Lena* image gives around 0.47-dB gain over *ShZeg* for all transmission rates. The advantage of

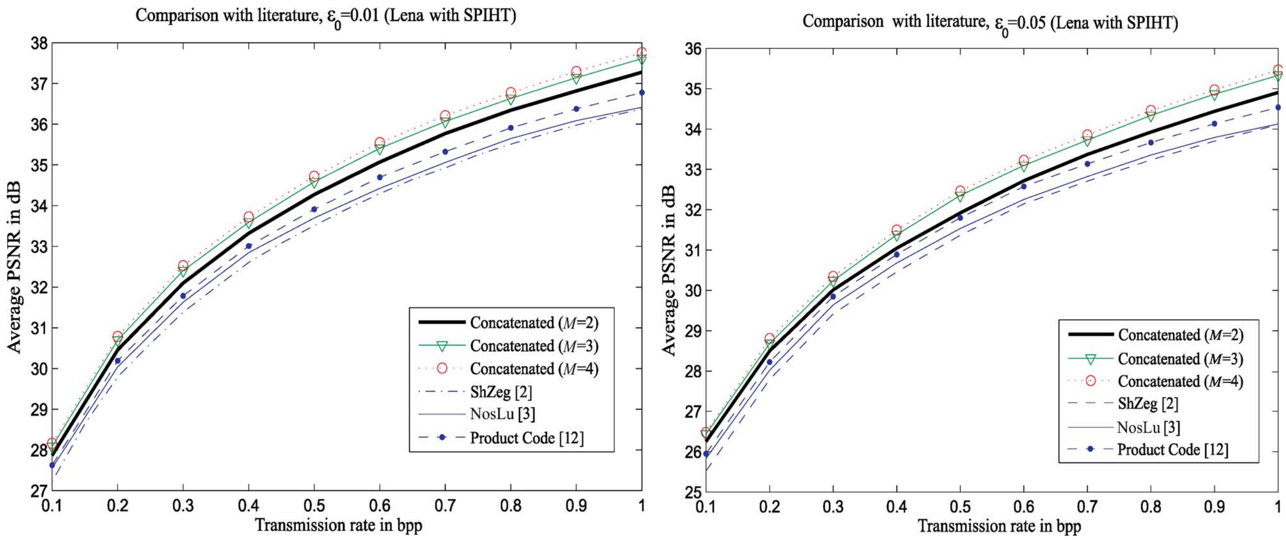


Fig. 8. Different systems are compared over a BSC with BER $= 0.01$ and BER $= 0.05$ using a 512 × 512 *Lena* image.

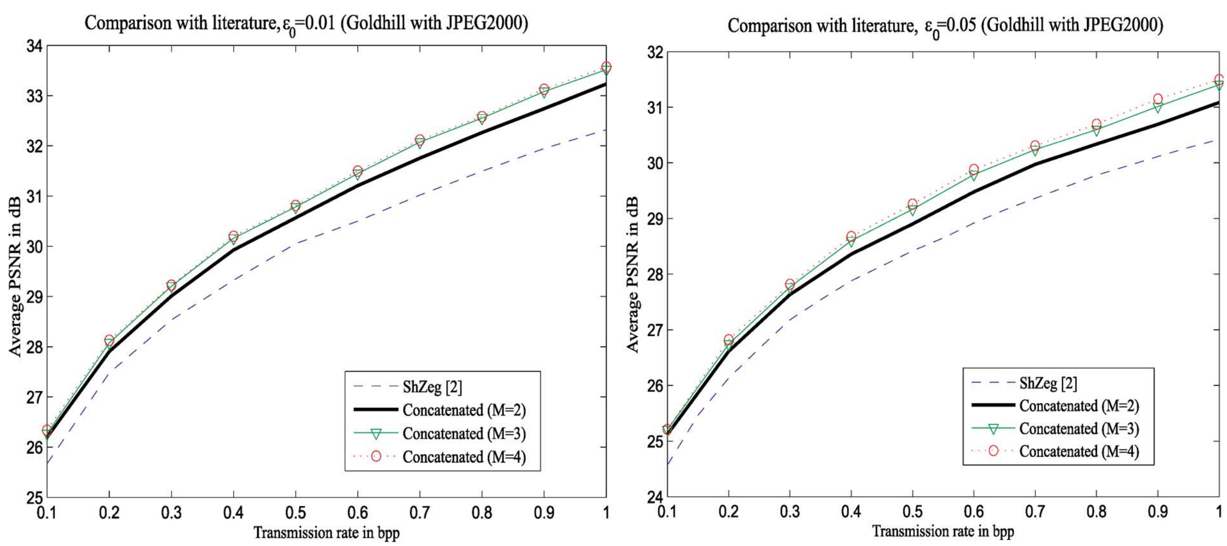


Fig. 9. Different systems are compared over a BSC with BER $= 0.01$ and BER $= 0.05$ using a 512 × 512 *Goldhill* image.

Product Code is its 2-D coding structure that provides the same level of protection of *ShZeg* using an overall higher code rate (RCPC + RS). This allows more information content in the bit budget and the potential for performance improvement. *Concatenated* provides more than 1-dB gain over *ShZeg* and 0.5-dB gain over *Product Code* at all transmission rates. The advantage of *Concatenated* can be attributed in part to packet size adjustment (and, thus, PER adjustment) and its flexible protection by multiple channel codes. Using multiple code rates enables protection levels that are not possible using the discrete code set. We tried information block sizes of 100, 200, 400, 600, 800, and 1000 and verified that choosing other than 200 bits as the information block size of conventional methods does not help their performance much. In Fig. 9, we show average PSNR performances of some of those systems using *Goldhill* encoded with JPEG2000. The results of *Concatenated* exhibit similar gains compared with *ShZeg*, i.e., they are more significant at lower ϵ_0 and higher transmission rates. The jaggedness of the curves is due to the discrete code set and the source encoder.

Next, we provide and compare results for *Concatenated* using *Lena* at various values of M, as shown in Table I. There are two observations with regard to the proposed formatting and channel code rate allocation of *Concatenated*. First, when $M = m \in \{1, 2, 3, 4, 5\}$, we have $r_1^{*(m)} \geq r_2^{*(m)} \geq \cdots \geq r_m^{*\,(m)}$, where the superscript denotes the current value of M (we refer to this as the First Condition). This is because baseline protection is achieved by the outermost code, and then, incremental protection is added by each encoding stage. Our results show that incremental protection is enough to obtain the necessary UEP with a less powerful outermost code. Note that this result can be used to restrict the search space of our exhaustive search in finding optimal code rate set $\mathcal{R}^*$. It leads to complexity reductions, as shown in Table II. Second, if we consider two different values for m, namely, m_1 and m_2, where $m_2 = m_1 + 1$, we observe that $r_{m_2}^{*\;(m_2)} \geq r_{m_1}^{*\;(m_1)}$ (we refer to this as the Second Condition). This is because, as we increase M, we observe that the outermost code rate did not need to be more powerful than the one used in the system with one less encoding stage, since

TABLE I
OPTIMAL ALLOCATION: $\mathcal{R}^*$ FOR M-PACKET TRANSMISSION. OPTIMAL PARAMETER M^* IS SHOWN IN BOLD

$\epsilon_0 = 0.15$, $r_{tr} = 0.3$ bpp						
M	r_1^*	r_2^*	r_3^*	r_4^*	r_5^*	PSNR (dB)
1	1/4	N/A	N/A	N/A	N/A	14.48
2	4/7	1/4	N/A	N/A	N/A	26.96
3	8/9	2/3	1/4	N/A	N/A	27.32
4	**1**	**8/9**	**2/3**	**4/15**	**N/A**	**27.38**
5	1	1	8/9	2/3	4/15	27.37

$\epsilon_0 = 0.1$, $r_{tr} = 0.3$ bpp						
M	r_1^*	r_2^*	r_3^*	r_4^*	r_5^*	PSNR (dB)
1	1/4	N/A	N/A	N/A	N/A	20.44
2	2/3	1/3	N/A	N/A	N/A	28.45
3	8/9	4/5	4/13	N/A	N/A	28.71
4	**8/9**	**8/9**	**4/5**	**1/3**	**N/A**	**28.79**
5	1	8/9	8/9	4/5	1/3	28.75

$\epsilon_0 = 0.05$, $r_{tr} = 0.6$ bpp						
M	r_1^*	r_2^*	r_3^*	r_4^*	r_5^*	PSNR (dB)
1	1/4	N/A	N/A	N/A	N/A	31.65
2	8/9	1/3	N/A	N/A	N/A	32.72
3	8/9	4/5	4/9	N/A	N/A	33.10
4	**8/9**	**8/9**	**8/9**	**4/9**	**N/A**	**33.22**
5	1	8/9	8/9	8/9	4/9	33.18

$\epsilon_0 = 0.01$, $r_{tr} = 0.7$ bpp						
M	r_1^*	r_2^*	r_3^*	r_4^*	r_5^*	PSNR (dB)
1	2/5	N/A	N/A	N/A	N/A	34.60
2	8/9	4/7	N/A	N/A	N/A	35.77
3	8/9	8/9	2/3	N/A	N/A	36.07
4	**1**	**8/9**	**8/9**	**2/3**	**N/A**	**36.08**
5	1	1	8/9	8/9	2/3	36.06

$\epsilon_0 = 0.005$, $r_{tr} = 0.5$ bpp						
M	r_1^*	r_2^*	r_3^*	r_4^*	r_5^*	PSNR (dB)
1	1/2	N/A	N/A	N/A	N/A	33.80
2	8/9	2/3	N/A	N/A	N/A	34.90
3	**1**	**8/9**	**2/3**	**N/A**	**N/A**	**34.97**
4	1	1	8/9	2/3	N/A	34.97
5	1	1	1	8/9	2/3	34.96

$\epsilon_0 = 0.001$, $r_{tr} = 0.3$ bpp						
M	r_1^*	r_2^*	r_3^*	r_4^*	r_5^*	PSNR (dB)
1	2/3	N/A	N/A	N/A	N/A	33.02
2	8/9	4/5	N/A	N/A	N/A	33.43
3	**1**	**8/9**	**4/5**	**N/A**	**N/A**	**33.49**
4	1	1	8/9	4/5	N/A	33.48
5	1	1	1	8/9	4/5	33.45

TABLE II
NUMBER OF POSSIBILITIES WITH AND WITHOUT THE FIRST CONDITION. WE OBTAIN COMPLEXITY REDUCTIONS WITH INCREASING M. FOR EXAMPLE, WHEN $M = 5$, IT IS ENOUGH TO SEARCH ONLY 1/62 OF ALL POSSIBILITIES

$\forall\epsilon_0, \forall r_{tr}$	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$
No condition	14	196	2744	38416	537824	7529536
${r_1^*}^{(m)} \geq \cdots \geq {r_m^*}^{(m)}$	14	105	560	2380	8568	27132

TABLE III
NUMBER OF POSSIBILITIES WITH TWO CONDITIONS WHEN $\epsilon_0 = 0.01$ AND $r_{\text{tr}} = 0.7$ bpp. WE OBTAIN MORE COMPLEXITY REDUCTIONS WITH INCREASING M. FOR EXAMPLE, WHEN $M = 5$, IT IS ENOUGH TO SEARCH ONLY 1/3055 OF ALL POSSIBILITIES

$\epsilon_0 = 0.01$, $r_{tr} = 0.7$bpp	$m_1 = 1$	$m_1 = 2$	$m_1 = 3$	$m_1 = 4$	$m_1 = 5$	$m_1 = 6$
No condition	14	196	2744	38416	537824	7529536
${r_1^*}^{(m_1)} \geq \cdots \geq {r_{m_1}^*}^{(m_1)}$	14	105	560	2380	8568	27132
$\{{r_{m_2}^*}^{(m_2)} \geq {r_{m_1}^*}^{(m_1)}\} \bigwedge \{{r_1^*}^{(m_1)} \geq \cdots \geq {r_{m_1}^*}^{(m_1)}\}$	14	50	85	120	176	260

going from m_1 to m_2 encoding stages gives a certain level of additional protection for more important bits. Therefore, it is enough to choose the outermost code ${r_{m_2}^*}^{(m_2)}$ to be equal to or greater than ${r_{m_1}^*}^{(m_1)}$ to provide baseline protection for the total bit stream. Having a larger outermost code rate also allows more source bits into the bit stream and increases the potential for performance improvement. However, as we increase M, the CRC and tailing bits will start to overwhelm the allowed bit budget, giving rise to performance degradation due to lack of source bits. Using the condition ${r_{m_2}^*}^{(m_2)} \geq {r_{m_1}^*}^{(m_1)}$, we can obtain additional reductions in complexity, particularly at low ϵ_0. One particular example is shown in Table III.

Table I also shows the average PSNR performance of *Concatenated* as a function of M. For all the cases shown, we pick up more than 0.5-dB gain using the optimal value M^*. For $1 \leq M \leq M^*$, we get diminishing returns as we approach M^*. When $M > M^*$, there can be slight degradation in the performance, because with larger M, the optimization results in a code rate of unity for some of the initial information blocks. This leads to an increase in the number of CRC bits, which, in turn, decreases the total number of source and channel coding bits in the allowed bit budget. In addition, since the sizes of the permutation tables are chosen to be equal to the payload size in this paper, smaller information blocks mean shorter permutation table size, which significantly impact the overall performance of *Concatenated*.

Finally, we compare the proposed scheme with some of the reported results found in [2], [3], and [12]. As shown, those studies use the LVA for increased performance. Therefore, we use our proposed scheme with the LVA for comparison. In other words, the ordinary Viterbi algorithm is replaced with the LVA in the proposed formatting and system model. The LVA finds the "best path" that has the smallest accumulated metric in the trellis and satisfies the CRC at the same time. We constrained the path search depth to 70 candidate paths. If none of the 70 candidate paths satisfies the CRC, decoding failure is declared. Table IV shows some results for *Lena* and *Goldhill* images for $\epsilon_0 = 0.1$ and $\epsilon_0 = 0.01$.

As a perspective on PSNR, if one sets the information rate equal to that which corresponds to the capacity of a BSC with crossover probability ϵ_0 and assumes that the data are sufficiently coded such as to produce no errors, then the results corresponding to this idealized scenario are listed in the rows entitled "error-free case." Note that the capacity of a BSC is given

TABLE IV
PSNR (IN DECIBELS) FOR 512 × 512 IMAGES TRANSMITTED OVER A BSC AT VARIOUS TRANSMISSION RATES WITH LVA DECODING. $\{0.252, 0.505, 0.994\} \times C(\epsilon_0)|_{\epsilon_0=0.1} \approx \{0.1338, 0.2682, 0.5278\}$ bpp AND $\{0.252, 0.505, 0.994\} \times C(\epsilon_0)|_{\epsilon_0=0.01} \approx \{0.2316, 0.4642, 0.9137\}$ bpp

r_{tr}(bpp)	Image	System	Channel raw BER (ϵ_0)		Image	System	Channel raw BER (ϵ_0)	
			0.1	0.01			0.1	0.01
0.252	Lena	Proposed	29.46	32.74	Goldhill	Proposed	27.43	29.54
		[2]([3], [12])	28.4 (28.61, 28.88)	32.0		[2]	26.7	29.0
		"error-free case"	31.29	33.67		"error-free case"	28.63	30.31
0.505	Lena	Proposed	32.33	35.83	Goldhill	Proposed	29.24	31.79
		[2]([3], [12])	31.1 (31.27, 31.6)	35.2		[2]	28.6	31.2
		"error-free case"	34.39	36.91		"error-free case"	30.78	32.8
0.994	Lena	Proposed	35.27	38.84	Goldhill	Proposed	31.41	34.72
		[2]([3],[12])	34.2 (34.25, 34.66)	38.0		[2]	30.7	34.0
		"error-free case"	37.39	39.9		"error-free case"	33.4	35.98

TABLE V
PSNR (IN DECIBELS) FOR 512 × 512 *LENA* AND *GOLDHILL* IMAGES ENCODED WITH JPEG2000 AND TRANSMITTED OVER A BSC AT VARIOUS TRANSMISSION RATES. "ERROR-FREE CASE" CORRESPONDS TO $\{0.252, 0.505, 0.994\} \times C(\epsilon_0)|_{\epsilon_0=0.1} \approx \{0.1338, 0.2682, 0.5278\}$ bpp AND $\{0.252, 0.505, 0.994\} \times C(\epsilon_0)|_{\epsilon_0=0.01} \approx \{0.2316, 0.4642, 0.9137\}$ bpp. BOLD DENOTES THE WINNING SCHEME FOR A GIVEN TRANSMISSION RATE AND RAW CHANNEL ERROR RATE. *ConRCPC* USES THE PROPOSED CODING PARADIGM AND RCPC CODE FAMILY WITH ORDINARY VITERBI DECODING, AND *ConLDPC* USES THE PROPOSED CODING PARADIGM AND AN RC-LDPC CODE FAMILY WITH A BELIEF PROPAGATION ALGORITHM

Image	r_{tr}(bpp)	System	Channel raw BER (ϵ_0)				Image	Channel raw BER (ϵ_0)			
			0.01	0.03	0.08	0.1		0.01	0.03	0.08	0.1
Lena	0.252	*ConRCPC*	31.39	30.12	28.31	27.78	Goldhill	28.77	27.96	26.86	26.44
		ConLDPC	**32.83**	**32.59**	**31.29**	**30.86**		29.74	**29.41**	**28.63**	**28.39**
		RC-LDPC [9]	32.77	32.42	31.03	30.48		**29.79**	29.35	28.49	28.09
		IRA [7]	32.75	32.09	30.24	29.92		29.74	29.23	28.01	27.90
		RCTC [5]	32.56	31.90	29.94	29.40		29.64	29.16	27.88	27.69
		"error-free case"	33.59	32.88	31.67	31.17		30.30	29.88	28.89	28.57
	0.505	*ConRCPC*	34.47	33.1	31.14	30.37		30.88	30.08	28.69	28.07
		ConLDPC	36.12	**35.67**	**34.31**	**34.08**		32.21	**31.79**	**30.69**	**30.41**
		RC-LDPC [9]	36.08	35.41	34.00	33.32		**32.28**	31.65	30.52	30.13
		IRA [7]	**36.18**	35.48	33.37	33.13		32.10	31.65	30.12	29.99
		RCTC [5]	35.67	35.15	33.20	32.76		31.79	31.38	30.04	29.89
		"error-free case"	36.82	36.16	34.86	34.34		32.84	32.31	31.14	30.72
	0.994	*ConRCPC*	37.53	36.17	34.19	33.42		33.63	32.3	30.63	30.21
		ConLDPC	39.01	**38.77**	**37.38**	**37.13**		**35.30**	**34.65**	**33.41**	**33.11**
		RC-LDPC [9]	**39.03**	38.34	36.91	36.19		35.23	34.33	32.99	32.42
		IRA [7]	38.93	38.26	36.26	36.03		35.00	34.18	32.30	32.20
		RCTC [5]	38.78	37.74	36.15	35.85		34.81	34.05	32.24	32.10
		"error-free case"	39.84	39.24	37.90	37.36		36.06	35.40	33.92	33.39

by $C(\epsilon_0) = 1 - h(\epsilon_0)$, where $h(\epsilon_0) = -\epsilon_0 \log_2(\epsilon_0) - (1 - \epsilon_0)\log_2(1-\epsilon_0)$ is the binary entropy function, and ϵ_0 is the raw error rate of the channel. As shown, the proposed scheme not only improves the performance of [2] but also gives results that are close to the "error-free case."

B. Simulations With RC-LDPC Codes

Throughout this section, the RC-LDPC channel code set is used. An extra byte (instead of the CRC) is added in each encoding stage to inform the RC-LDPC decoder about the channel coding rate used, as in [9]. Our mother code rate is 8/13, and through puncturing and extending the mother code [31], we obtain nine code rates in total, namely, 8/22, 8/20, 8/18, 8,16, 8/15, 8/13, 8/12, 8/11, and 8/10. This yields similar simulated PER performance to that of the irregular RC-LDPC construction in [9] for a BSC. In what follows, we present the performance of the proposed coding scheme using the *Lena* and *Goldhill* images (encoded with JPEG2000) with the class of RCPC (*ConRCPC*) and RC-LDPC (*ConLDPC*) codes. The results are given in Table V using a maximum of 250[3] iterations of the Max–product algorithm at the receiver. As shown, similar gains are obtained, except that the results are now closer to the "error-free case." In addition, note that the proposed scheme is more powerful at higher crossover error probabilities, as it virtually increases the size of the available code set and allows better protection than would be possible with the strongest code rate in the code set. Finally, we note that, at $\epsilon_0 = 0.1$, the proposed scheme is at most only ≈0.3 dB away from the maximum achievable PSNR for all the simulations presented.

[3]Reducing the maximum number of iterations to 100 as in [9] will only reduce the reported PSNRs in Table V by about 0.15 dB.

C. Discussion on the Advantages of the Proposed Transmission Scheme

Progressive image reconstruction has several advantages. One of these advantages is not preserved by our coding structure, but the others are. Progressive coding allows rapid display of a low-quality version of the image at early stages of the reception of the bit stream. The receiver progressively obtains better image quality as more bits reliably arrive and can redisplay the image multiple times at increasingly higher quality, allowing one to abort the image early in the reception if one determines, based on the low quality version, that the image is not the one desired. This might allow for faster browsing of a remote image database. This feature of allowing redisplaying multiple times is not preserved by our coding structure, as decoding of the most important information chunk cannot proceed immediately when it is received, as it is interleaved with other information chunks and their parity bits. This advantage of progressive image coding does not in any case extend to scalable video coding, because a single frame of video, even if it were scalably encoded, is not redisplayed multiple times at increasing quality levels.

The other advantages of progressive image transmission (which do extend to scalable video) are supported by our encoding structure. Progressive coding allows for simple downsizing of the total bit stream in case of congestion at an intermediate router. This feature is retained. The router would need to deinterleave but would not need to do either source decoding or channel decoding in order to strip off sections of information bits and parity bits and transmit onward the beginning portions of the progressive stream. Finally, progressive encoding has the advantage of being very suitable for UEP; thus, the initial information bits are more heavily protected for graceful degradation of the transmitted source quality at the receiver. We note that, if the proposed system were used for scalable video coding, the overall delay incurred on the encoder side would be dominated by the delay of the scalable video source encoder and not by the delay of the proposed channel encoder and interleaver. Although some scalable video coders incur significant delay (such as 3-D SPIHT, which might need to buffer a group of 16 frames), for a lower delay encoder such as H.264 fine-grain scalability, if the M codewords are all part of one frame, then interleaving is all done within one frame. In this case, the proposed system will incur negligible processing delay.

V. Conclusion

We have considered an embedded bit stream using concatenated block encoding in conjunction with random block interleavers. Novel formatting is introduced for embedded bit streams with better reconstruction properties at the receiver. Both the information block size and the channel code rates are subject to optimization, which makes the proposed system more flexible. A low-complexity descent search for block size adjustment and a constrained exhaustive search in finding the optimal code rates are presented. In addition, using the proposed image coding system, we use multiple code rates for the protection of information blocks, which makes additional BERs achievable beyond those possible using conventional methods with a discrete channel code rate set. The proposed scheme achieves this performance gain at the expense of loss of progressive display capability and increased complexity.

References

[1] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[2] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Process. Lett.*, vol. 4, no. 7, pp. 189–191, Jul. 1997.

[3] A. Nosratinia, J. Lu, and B. Aazhang, "Source–channel rate allocation for progressive transmission of image," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 186–196, Feb. 2003.

[4] L. Li and M. Salehi, "Hierarchical image coding matched to unequal error protection rate compatible punctured convolutional codes," in *Proc. IEEE Int. Conf. Robot., Intell. Syst. Signal Process.*, Oct. 2003, vol. 1, pp. 238–243.

[5] B. A. Banister, B. Belzer, and T. R. Fisher, "Robust image transmission using JPEG2000 and turbo codes," *IEEE Signal Process. Lett.*, vol. 9, no. 4, pp. 117–119, Apr. 2002.

[6] T. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Wireless image transmission using turbo codes and optimal unequal error protection," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1890–1901, Nov. 2005.

[7] C. Lan, T. Chu, K. R. Narayanan, and Z. Xiong, "Scalable image and video transmission using irregular repeat–accumulate codes with fast algorithm for optimal unequal error protection," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1092–1101, Jul. 2004.

[8] X. Pan, A. H. Banihashemi, and A. Cuhadar, "Combined source and channel coding with JPEG2000 and rate-compatible low-density parity-check codes," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1160–1164, Mar. 2006.

[9] X. Pan, A. H. Banihashemi, and A. Cuhadar, "Progressive transmission of images over fading channels using rate-compatible LDPC codes," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3627–3635, Dec. 2006.

[10] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 819–860, Jun. 2000.

[11] G. D. Forney, *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[12] P. G. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," in *Proc. ICIP*, Oct. 1998, vol. 1, pp. 324–328.

[13] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 9, pp. 36–58, Sep. 2001.

[14] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Product code error protection of packetized multimedia bitstreams," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sep. 2003, pp. I-77–I-80.

[15] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Wireless transmission of images using JPEG2000," in *Proc. ICIP*, 2004, pp. 2523–2526.

[16] V. Chande and N. Farvadin, "Progressive transmission of image over memoryless noisy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 850–860, Jun. 2000.

[17] V. Stankovi, R. Hamzaoui, and D. Saupe, "Fast algorithm for rate-based optimal error protection of embedded codes," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1788–1795, Nov. 2003.

[18] G. D. Forney, "Burst-correcting codes for the classic bursty channel," *IEEE Trans. Commun.*, vol. COM-19, no. 5, pp. 772–781, Oct. 1971.

[19] G. Castagnoli, J. Ganz, and P. Graber, "Optimum cyclic redundancy check codes with 16-bit redundancy," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 111–114, Jan. 1990.

[20] D. J. Costello and O. Y. Takeshita, "On the packet error rate of convolutional codes," in *Proc. Inf. Theory Netw. Workshop*, Metsovo, Greece, Jun. 1999, p. 29.

[21] L. Cao, "On the unequal error protection for progressive image transmission," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2384–2388, Sep. 2007.

[22] J. A. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. New York: Springer-Verlag, 2005.

[23] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1997.

[24] R. J. McEliece and W. Lin, "The trellis complexity of convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1855–1864, Nov. 1996.

[25] R. Garello, G. Montorsi, S. Benedetto, and G. Cancellieri, "Interleaver properties and their applications to the trellis complexity analysis of turbo codes," *IEEE Trans. Commun.*, vol. 49, no. 5, pp. 793–807, May 2001.

[26] R. G. Gallager, *Low Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[27] L. Ping, W. K. Leung, and N. Phamdo, "Low density parity check codes with semi-random parity check matrix," *Electron. Lett.*, vol. 35, no. 1, pp. 38–39, Jan. 1999.

[28] R. Yazdani and M. Ardakani, "An efficient analysis of finite-length LDPC codes," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 677–682.

[29] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[30] S. Dolinar, D. Divsalar, and F. Pollara, Code performance as a function of block size JPL, Pasadena, CA, JPL TDA Progress Report 42–133, May 1998.

[31] S. F. Zaheer, S. A. Zummo, M. A. Landolsi, and M. A. Kousa, "Improved regular and semi-random rate-compatible low-density parity-check with short block lengths," *IET Commun.*, vol. 2, no. 7, pp. 960–971, Aug. 2008.

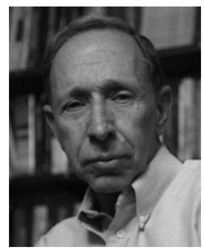
Suayb S. Arslan (S'06) received the B.S. degree in electrical and electronics engineering from Bogazici University, Istanbul, Turkey, in 2006, and the M.S. degree in electrical and computer engineering from the University of California at San Diego (UCSD), La Jolla, in 2009. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, UCSD.

During the summer of 2009, he was with Mitsubishi Electric Research Laboratory, Boston, MA, where he was involved in research and development of image/video processing algorithms for biomedical applications. In the summer of 2011, he joined Quantum Corporation, Irvine, CA, where he conducted research on advanced detection algorithms and postprocessing for increased capacity tape drives. His research interests include wireless/wireline digital multimedia communications, joint source–channel coding, information theory, image/video processing, and cross-layer design optimization.

Pamela C. Cosman (S'88–M'93–SM'00–F'08) received the B.S. (Hons.) degree in electrical engineering from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1989 and 1993, respectively.

She was a National Science Foundation (NSF) Postdoctoral Fellow with Stanford University and a Visiting Professor with the University of Minnesota, Minneapolis, during 1993–1995. In 1995, she joined the faculty of the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, where she is currently a Professor. She was the Director of the Center for Wireless Communications from 2006 to 2008. Her research interests are in the areas of image and video compression and processing, and wireless communications.

Prof. Cosman is a member of Tau Beta Pi and Sigma Xi. She was a Guest Editor of the June 2000 Special Issue on Error-Resilient Image and Video Coding of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego. She was an Associate Editor of the IEEE COMMUNICATIONS LETTERS (1998–2001) and an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS (2001–2005). She was the Editor-in-Chief (2006–2009) and a Senior Editor (2003–2005, 2010–present) of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She was the recipient of the ECE Departmental Graduate Teaching Award, a Career Award from the NSF, a Powell Faculty Fellowship, and a Globecom 2008 Best Paper Award.



Laurence B. Milstein (S'66–M'68–SM'77–F'85) received the B.E.E. degree from the City College of New York, New York, in 1964, and the M.S. and Ph.D. degrees in electrical engineering from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1966 and 1968, respectively.

From 1968 to 1974, he was with the Space and Communications Group, Hughes Aircraft Company, and from 1974 to 1976, he was a member of the Department of Electrical and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. Since 1976, he has been with the Department of Electrical and Computer Engineering, University of California, San Diego (UCSD), La Jolla, where he is the Ericsson Professor of Wireless Communications Access Techniques and former Department Chairman, working in the area of digital communication theory with special emphasis on spread-spectrum communication systems. He has also been a Consultant to both government and industry in the areas of radar and communications.

Dr. Milstein was the Vice President for Technical Affairs in 1990 and 1991 of the IEEE Communications Society and is a former Chair of the IEEE Fellows Selection Committee. He was an Associate Editor for Communication Theory of the IEEE TRANSACTIONS ON COMMUNICATIONS, an Associate Editor for Book Reviews of the IEEE TRANSACTIONS ON INFORMATION THEORY, an Associate Technical Editor of the IEEE COMMUNICATIONS MAGAZINE, and the Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He was a recipient of the 1998 Military Communications Conference Long-Term Technical Achievement Award, an Academic Senate 1999 UCSD Distinguished Teaching Award, an IEEE Third Millennium Medal in 2000, the 2000 IEEE Communication Society Armstrong Technical Achievement Award, and various prize paper awards, including the 2002 MILCOM Fred Ellersick Award.