

AN EFFICIENT JOINT SOURCE-CHANNEL RATE ALLOCATION SCHEME FOR JPEG2000 CODESTREAMS

Zhenyu Wu, Ali Bilgin and Michael W. Marcellin

Dept. of Electrical and Computer Engineering,
The University of Arizona, Tucson, AZ, 85721

ABSTRACT

A two-level, hybrid-optimization scheme is proposed for rate allocation of JPEG2000(J2K) transmission over noisy channels. It combines forward error correction (FEC) and J2K error resilience modes to minimize the expected end-to-end image distortion. Subject to a total target bit rate, it forms fixed-length source packets, and can be adapted to fixed-length channel packets in a straightforward manner. The proposed scheme utilizes the source distortion-rate characteristics and the BER statistics of the available channel codes. It has complexity less than $O(R^2)$ for a specific J2K codestream, where R is the number of available channel code rates. Performance results with both RCPC and turbo codes for fixed-length source packets are provided to show the effectiveness of the scheme.

1. INTRODUCTION

Highly efficient and progressive wavelet-based image codecs, such as SPIHT [1] and J2K [2], are very attractive for image transmission over various communication channels. However, their bitstreams suffer high sensitivity to channel errors due to error propagation at their decoders. Joint source and channel coding (JSCC) has emerged as an effective way to address this problem.

In JSCC, rate allocation is used to distribute available total rate between the source and the channel coders, in order to optimize some desired measure with a specific channel condition. Finding an optimal rate allocation is important but challenging. In [3], SPIHT codestreams were protected equally with a carefully chosen channel code at each channel error rate. In [4], a dynamic programming method was proposed to determine the rate allocation with fixed-length information packets using optimization criteria based on MSE, PSNR and available source rate. The last criterion was suggested as the best approach, since it reduces complexity, eliminates the need to transmit the rate schedule, and allows optimal transmission at many rates lower than the optimized target rate. In [5], a Viterbi-based method was presented for J2K codestreams. This method minimizes the expected distortion subject to a rate constraint with fixed-length channel packets. It has complexity growing as $O(N^2)$ with N the number of transmitted packets, and its result is a good approximation to that from optimal brute force search. In [6], the authors used a local search algorithm that starts from a rate-optimal solution and converges to a locally distortion-optimal solution for fixed-length channel packets. It yields comparable performance to that obtained by [5] with reduced complexity. In [7], a new performance measure for progressive transmission was proposed

for SPIHT codestreams. By dynamic programming, the optimal blocklengths and the associated channel rates were obtained for Reed-Solomon and turbo codes.

In this paper, a two-level, hybrid-optimization rate allocation scheme is proposed for J2K transmission based on [8]. The proposed scheme combines the error correction capability of FEC and the error-resilient source decoding capability provided by J2K error resilience modes to minimize the expected end-to-end image distortion. It forms fixed-length source packets for transmission, and can be adapted to fixed-length channel packets in a straightforward manner. Two different optimization methods are applied to two different J2K geometric levels, with complexity less than $O(R^2)$ for a specific codestream, where R is the number of available channel code rates. By utilizing the features of J2K codestreams, a multi-layer source/channel encoded codestream is formed at the desired total rate. Rate compatible punctured convolutional codes (RCPC) and turbo codes (RCPT) are chosen and applied to a BSC with $\epsilon = 10^{-2}$ for the fixed-length source packets case. Experimental results indicate that the proposed scheme compares favorably with other schemes.

The paper is organized as follows: in Section 2, the general formalism is defined for the J2K codestream transmission problem. In Section 3, a dynamic programming based approach is developed. In Section 4 experimental results are provided and Section 5 gives the conclusions.

2. GENERAL J2K RATE ALLOCATION PROBLEM

The goal of joint source-channel optimization is to find an optimal rate allocation scheme V , in order to minimize the expected distortion subject to a designated total rate. For J2K codestreams without channel noise, this is equivalent to minimizing the sum of distortions contributed from the codeblocks, while keeping the sum of their bitstream lengths within the designated total length:

$$\min \sum_b D_b \quad \text{s.t.} \quad \sum_b L_b \leq L_T \quad (1)$$

where D_b and L_b are the distortion and bitstream length of codeblock b , respectively, and L_T is the designated file size [9, 10, 2].

With channel noise, for each codeblock b , its expected distortion with some rate allocation scheme \vec{V}_b can be written as:

$$E[D_b(\vec{V}_b)] = D_{b,0} - E[\Delta D_b(\vec{V}_b)] \quad (2)$$

where $D_{b,0}$ is the initial (zero rate) distortion, $E[\Delta D_b(\vec{V}_b)]$ is the expected distortion reduction when \vec{V}_b is employed, resulting in a length $L_b(\vec{V}_b)$ bitstream for codeblock b .

With Eq. (2), in the noisy channel case, Eq. (1) can be rewritten as:

$$\min \sum_b (D_{b,0} - E[\Delta D_b(\vec{V}_b)]) \quad \text{s.t.} \quad \sum_b L_b(\vec{V}_b) \leq L_T$$

or equivalently,

$$\min \sum_b -E[\Delta D_b(\vec{V}_b)] \quad \text{s.t.} \quad \sum_b L_b(\vec{V}_b) \leq L_T \quad (3)$$

Adapting the results from [11], we can solve Eq. (3) using Lagrange multiplier method:

$$\min\left\{\sum_b -E[\Delta D_b(\vec{V}_b)] + \lambda \sum_b L_b(\vec{V}_b)\right\} \quad (4)$$

by minimizing each term (corresponding to each b) independently. By sweeping λ over the range of zero to infinity, sets of $\{V\}$ and $\{\sum_b L_b\}$ can be created. If a $(\sum_b L_b)$ happens to equal L_T , then a desired rate allocation scheme V has been found.

Minimizing each term in Eq. (4) corresponds to an optimization task at the coding pass level within a single codeblock. For codeblock b , denote N_c as the maximum number of coding passes that may be included. For each coding pass i , there is an associated distortion reduction d_i with length l_i bytes, where $i \in [1, N_c]$. From Eq. (4), for a given λ , it is desired to find a rate allocation scheme \vec{V}_b which minimizes:

$$-E[\Delta D_b(\vec{V}_b)] + \lambda L_b(\vec{V}_b) \quad (5)$$

Denote r_i as the channel code rate assigned for coding pass i ($0 \leq r_i \leq 1$) and $P(r_i, \frac{l_i}{r_i})$ as the probability that there are one or more uncorrected errors in coding pass i if channel coding rate r_i is employed, and let N'_c ($N'_c \leq N_c$) be the number of coding passes included by \vec{V}_b . When decoding a J2K codestream with error resilience mode switches “RESTART” and “ERTERM” [2], a J2K decoder can decode all correct coding passes within a given codeblock prior to the coding pass containing the first bit error. Decoding of other codeblocks is unaffected by such an error. With this condition, the expected distortion reduction for codeblock b including N'_c coding passes using channel code rates $r_1, r_2, \dots, r_{N'_c}$ is

$$\begin{aligned} E[\Delta D_b(\vec{V}_b)] &= \sum_{j=1}^{N'_c-1} \left(\sum_{k=1}^j d_k \right) \left(\prod_{k=1}^j \left[1 - P\left(r_k, \frac{l_k}{r_k}\right) \right] \right) P\left(r_{j+1}, \frac{l_{j+1}}{r_{j+1}}\right) \\ &\quad + \left(\sum_{k=1}^{N'_c} d_k \right) \left(\prod_{k=1}^{N'_c} \left[1 - P\left(r_k, \frac{l_k}{r_k}\right) \right] \right). \end{aligned} \quad (6)$$

With Eq. (5) and (6), given any λ , the target function to be minimized for each codeblock b becomes:

$$\begin{aligned} f(\vec{V}_b) &= - \sum_{j=1}^{N'_c-1} \left(\sum_{k=1}^j d_k \right) \left(\prod_{k=1}^j \left[1 - P\left(r_k, \frac{l_k}{r_k}\right) \right] \right) P\left(r_{j+1}, \frac{l_{j+1}}{r_{j+1}}\right) \\ &\quad - \left(\sum_{j=1}^{N'_c} d_j \right) \left(\prod_{j=1}^{N'_c} \left[1 - P\left(r_j, \frac{l_j}{r_j}\right) \right] \right) + \lambda \sum_{j=1}^{N'_c} \frac{l_j}{r_j}. \end{aligned} \quad (7)$$

Eq. (7) can be evaluated for any rate allocation scheme $\vec{V}_b = (r_1, r_2, \dots, r_{N'_c})$. Notice that \vec{V}_b must jointly optimize N'_c and the channel code rates $r_1, r_2, \dots, r_{N'_c}$.

In [8], an exhaustive search based scheme was proposed to find the optimal \vec{V}_b for Eq. (7) over a reduced search space with $\binom{N_c+R-1}{R}$ candidates, where R is the number of available channel codes. But it becomes impractical to search when R is large. Thus a dynamic-programming based optimization is proposed, which aims to reduce the overall computational complexity. The notation is based on that in [12].

3. A DYNAMIC PROGRAMMING BASED APPROACH

In [5], a Viterbi-based scheme was used to optimize J2K codestreams. Its optimization was done at the level of an existing J2K codestream. Our proposed scheme optimizes at the coding pass and codeblock levels, and then uses the rate allocation results to drive a J2K encoder to generate a codestream with proper quality layers corresponding to protection levels.

By proper rearrangement, Eq. (7) can be written as:

$$\begin{aligned} f(\vec{V}_b) &= \sum_{i=1}^{N'_c} \left\{ \left(\prod_{j=1}^i [1 - P(r_j, \frac{l_j}{r_j})] \right) (-d_i) \right\} + \lambda \sum_{i=1}^{N'_c} \frac{l_i}{r_i} \\ &= \sum_{i=1}^{N'_c} \left\{ \left(\prod_{j=1}^i [1 - P(r_j, \frac{l_j}{r_j})] \right) (-d_i) + \lambda \frac{l_i}{r_i} \right\} \end{aligned} \quad (8)$$

In light of Eq. (8), for the proposed backward dynamic programming algorithm, the cost function at stage k , state r_k is defined as:

$$g_k(r_k) = [1 - P(r_k, \frac{l_k}{r_k})](-d_k) + \lambda \frac{l_k}{r_k} \quad (9)$$

where stage k corresponds to the coding pass k ($1 \leq k \leq N_c$) and state r_k belongs to the set of available channel code rates at stage k denoted as $R(k)$.

The cost-to-go function at stage k , state r_k is defined as:

$$J_{N_c}(r_{N_c}) = [1 - P(r_{N_c}, \frac{l_{N_c}}{r_{N_c}})](-d_{N_c}) + \lambda \frac{l_{N_c}}{r_{N_c}} \quad (\text{when } k = N_c)$$

and

$$\begin{aligned} J_k(r_k) &= g_k(r_k) + \\ &\min_{r_{k+1} \in R(k+1)} \left\{ [1 - P(r_k, \frac{l_k}{r_k})] [J_{k+1}(r_{k+1}) - \lambda (\frac{l_{k+1}}{r_{k+1}} + \sum_{p=k+2}^{N_c} \frac{l_p}{r_p^*})] \right. \\ &\quad \left. + \lambda (\frac{l_{k+1}}{r_{k+1}} + \sum_{p=k+2}^{N_c} \frac{l_p}{r_p^*}) \right\} \\ &\quad (\text{when } 1 \leq k \leq N_c - 1) \end{aligned} \quad (10)$$

where

$$r_p^* = \begin{cases} \operatorname{argmin}_{r_{k+2} \in R(k+2)} J_{k+1}(r_{k+1}) & \text{when } p = k + 2 \\ \operatorname{argmin}_{r_p \in R(p)} J_{p-1}(r_{p-1}^*) & \text{when } k + 3 \leq p \leq N_c \end{cases} \quad (11)$$

The optimized rate allocation scheme \vec{V}_b for codeblock b is then determined by:

$$V_b(k) = \begin{cases} \operatorname{argmin}_{r_1 \in R(1)} J_1(r_1) & \text{when } k = 1 \\ \operatorname{argmin}_{r_k \in R(k)} J_{k-1}(V_b(k-1)) & \text{when } 2 \leq k \leq N_c \end{cases} \quad (12)$$

Note that for the cost-to-go function J_k at each stage k , in addition to the states corresponding to the available channel codes, there is a value-0 entry indicating the possible choice that only the first $k - 1$ coding passes are included by \vec{V}_b . So $\vec{V}_b = (r_1, r_2, \dots, r_{N'_c})$ where N'_c is the last coding pass index with non-zero channel code rate.

Without any code rate constraint, each state can choose any state from the previous stage, with complexity of $O(R^2)$. If the available channel codes are from a punctured code family, which are different in their code rates, then the non-decreasing rate constraint can be imposed as $r_i \leq r_{i+1}$ ($1 \leq i \leq N_c$). That is, each state can only choose its optimal predecessor in its reduced code rate space, yielding even much lower complexity.

Based upon the above discussion, the proposed rate allocation scheme can be divided into two levels. At the lower level, it deals with coding passes in each codeblock. Given a $\lambda \geq 0$, it uses dynamic-programming method to find an optimal code-rate vector which minimizes Eq. (7), resulting in a bitstream length L_b for each codeblock b . At the higher level, it deals with codeblocks. Using the Lagrange multiplier method, it performs a search over λ to achieve $\sum L_b(\lambda)$ equal to the desired total length. Because of the monotonic relationship between λ and the resulting bitstream length, a bisection method can be utilized to find a particular designated length.

In Eq. (10), a term $[1 - P(r_k, \frac{l_k}{r_k})]$ of stage k is multiplied with the adjusted cost-to-go function J_{k+1} of stage $k + 1$. This is because generally the J2K decoder must decode the bitstream in a sequential and uninterrupted fashion inside a codeblock, which is referred to as a "sequential decoding requirement" in [7]. With the presence of this multiplicative term, the dynamic programming scheme may not be able to generate the optimal solution in some cases. But it can be alleviated by carefully choosing the channel codes, as mentioned in [5]. In our case, the possible suboptimality is localized within each codeblock's own codestream.

4. EXPERIMENTAL RESULTS

In the following experiments, J2K codestreams targeted at different total rates are transmitted through a binary symmetric channel (BSC) with $\epsilon = 10^{-2}$ for the fixed-length source packets case. Rate compatible punctured convolutional (RCPC) codes and turbo (RCPT) codes are chosen as the channel codes.

In the RCPC case, the same channel encoder/decoder structure is used as the one in [3]. From simulations, we found that the rate $\frac{2}{3}$ RCPC code punctured from a memory 6, rate $\frac{1}{3}$ code is strong enough to yield an extremely small probability of uncorrected error for the channel. So from [13], code rates $\frac{8}{9}, \frac{4}{5}, \frac{2}{3}$ are selected, together with the no channel coding (rate 1), forming 4 available code rates. Their error rate statistics based on fixed-length source packets are simulated and tabulated for further computation.

In the RCPT case, a rate $\frac{1}{3}$ parallel concatenated convolutional code is used with $(33/31)_{oct}$ as its encoder generator polynomials. Each channel block size is chosen to be 500 bytes, as a tradeoff between the channel code performance and the source code's progressiveness. Each channel block is decoded with 15 iterations. Using the puncturing patterns from [14], rate $\frac{2}{3}$ has close to error free performance with the above settings. So code rates $\frac{2}{3}, \frac{8}{11}, \frac{4}{5}$ and $\frac{8}{9}$ are selected, together with no channel coding (rate 1), forming 5 available code rates. Similarly, their error rate statistics based on fixed-length source packets are simulated and

tabulated.

Goldhill (512×512) and Lenna (512×512) are used as our test images. Simulations are done at the total rate 0.10, 0.25, 0.50, 0.75 and 1.00 bpp, and 1000 trials for each case, respectively. During the rate optimization, the non-decreasing code rate constraint ($r_i \leq r_{i+1}$) is imposed to simplify the final codestream structure with negligible loss. With the same conditions, the proposed dynamic programming (DP) optimization results are compared with those of the exhaustive search (ES) results in [8], as shown in Tables 1 and 3. In the RCPC case, DP yields identical results as ES in rate and PSNR, with exceptions at 1.00 bpp for both images, incurring negligible differences. At the same time, the computation time for the 4 available RCPC rates is reduced by a factor of a hundred from ES to DP. With 5 available channel rates for the turbo codes, ES is unmanagable. For these codes, we show results for only DP, and demonstrate a gain of roughly 0.5 dB at 1.00 bpp over RCPC.

In Fig. (1), the scaled minimal values of Eq. (7) are shown for both of RCPC and turbo codes. Turbo codes yield significant improvement over the entire λ range as a result of their stronger protection. Thus it can be expected that with the proposed rate allocation scheme, better performance can be achieved with even stronger channel codes.

In the RCPC optimization, rate $\frac{2}{3}$ and $\frac{4}{5}$ codes are almost always preferred over the other two weaker codes. J2K layer functionality is utilized to collect the separate coding passes with the same protection level from all the codeblocks and put them together continuously, forming a 2-layer codestream. Thus, the rate $\frac{2}{3}$ code is applied to the first layer and rate $\frac{4}{5}$ to the second. The main header (containing packed packet headers) is protected with rate $\frac{4}{3}$ together with the first layer. Note that to increase the codestream's quality progressiveness, more layers can be formed from each of the two equally protected portions of the bitstream.

In the RCPT optimization, rates $\frac{2}{3}$, $\frac{8}{11}$ and $\frac{4}{5}$ are preferred over the other two weaker codes, so a 3-layer codestream is similarly formed. As in RCPC case, the main header is protected with the strongest code together with the first layer.

In Tables 2 and 4, our unequal error protection (UEP) results based on exhaustive search and dynamic programming are compared with equal error protection (EEP) SPIHT and J2K using the same scheme as in [3]. At each rate, our UEP scheme outperforms the corresponding EEP cases. With 5 testing rates for both images, only rate 1.00 bpp cases have different rate allocations between ES and DP based schemes. Even then, the resulting PSNR differences are negligible, which indicates that the suboptimality from DP is subtle.

The above experiments are based on the fixed-length source packets case. By varying the source length inside each packet according to different channel code rates, fixed-length channel packets can be formed. With the same scheme proposed in Section 2 and 3, jointly source/ channel coded codestreams based on fixed-length channel packets can be obtained in a straightforward manner.

In [5], optimization is based on fixed-length channel packets as opposed to fixed-length information packets as employed here. During the decoding in [5], when the channel decoder detects any uncorrected erroneous packets, the source decoder only decodes the correct bits up to that packet, with the remainder of the codestream being discarded. In our case, the source decoder only stops the decoding within a single codeblock. Decoding of other codeblocks can still continue to the end of the codestream by discarding bits selectively to prevent error propagation. This comes with a small cost in error free PSNR due to the use of error resilience mode switches. Apparently, our gain due to error localization is

Bit Rate (bpp)	0.1	0.25	0.50	0.75	1.00
ES RCPC: $\frac{2}{3}, \frac{4}{5}$ rates bitstream length ratio	1:5.27	1:5.39	1:5.46	1:4.05	1:3.93
ES RCPC: Source Rate(bpp) noise-free PSNR(dB)	0.08 26.78	0.19 29.30	0.36 31.63	0.53 33.26	0.69 34.42
DP RCPC: $\frac{2}{3}, \frac{4}{5}$ rates bitstream length ratio	1:5.27	1:5.39	1:5.46	1:4.05	1:3.93
DP RCPC: Source Rate(bpp) noise-free PSNR(dB)	0.08 26.78	0.19 29.30	0.36 31.63	0.53 33.26	0.70 34.45
DP Turbo: $\frac{2}{3}, \frac{8}{11}, \frac{4}{5}$ rates bitstream length ratio	1:0:4.05	1:2.00:4.85	1:2.50:9.00	1:3.00:8.53	1:3.67:12.20
DP Turbo: Source Rate(bpp) noise-free PSNR(dB)	0.09 27.00	0.21 29.62	0.40 31.94	0.59 33.66	0.76 35.02

Table 1: Rate Optimization Results Comparison for Goldhill (512 x 512).

mostly offset by this loss, since our results are very nearly identical to those of [5].

On the other hand, with more source decoding robustness introduced in the codestream by the error resilience mode switches, the proposed scheme offers more graceful degradation in the case of mismatched channel BER, as compared to [5]. This comparison is shown in Table 5. In Fig.(2), the PSNR distributions with mismatched BER for Lenna and Goldhill at 1.00 bpp total bit rate are presented. The designed channel BER is 0.01 and the actual BER is 0.03. In both figures, the x-axis denotes the PSNR, and the y-axis denotes the cumulative percentage of simulations that result in PSNR values lower than the corresponding x value. This figure demonstrates that relatively few simulations result in PSNRs significantly below the average values (32.72, 28.54 dB) reported in Table 5.

5. CONCLUSION

In this paper, a two-level, hybrid-optimization rate allocation scheme is proposed for J2K codestream transmission over noisy channels. It minimizes the expected end-to-end distortion with fixed-length source packets, and can be adapted to fixed-length channel packets case straightforwardly. It can provide close to optimal solutions with low computational complexity. Experimental results with two different strength channel codes are provided. This scheme can be applied to other types of channel codes as well.

6. REFERENCES

- [1] A. Said and W. A. Pealman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits Syst. Video Tech*, pp. 243–250, June 1996.
- [2] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Practice and Standards*, Kluwer Academic Publishers, Massachusetts, 2002.

Bit Rate (bpp)	0.1	0.25	0.50	0.75	1.00
EEP SPIHT [3]	26.63	28.98	31.16	32.69	33.94
EEP J2K [3]	26.59	29.02	31.17	32.81	33.93
UEP ES-RCPC J2K [8]	26.72	29.19	31.48	33.08	34.22
UEP DP-RCPC J2K	26.72	29.19	31.48	33.08	34.21
UEP DP-Turbo J2K	26.95	29.56	31.84	33.54	34.85

Table 2: PSNR Comparison with Different Schemes for Goldhill (512 x 512).

Bit Rate (bpp)	0.1	0.25	0.50	0.75	1.00
ES RCPC: $\frac{2}{3}, \frac{4}{5}$ rates bitstream length ratio	1:4.06	1:4.31	1:2.95	1:3.31	1:2.82
ES RCPC: Source Rate(bpp) noise-free PSNR(dB)	0.08 28.16	0.18 32.06	0.35 35.30	0.52 37.21	0.70 38.46
DP RCPC: $\frac{2}{3}, \frac{4}{5}$ rates bitstream length ratio	1:4.06	1:4.31	1:2.95	1:3.31	1:2.81
DP RCPC: Source Rate(bpp) noise-free PSNR(dB)	0.08 28.16	0.18 32.06	0.35 35.30	0.52 37.21	0.70 38.45
DP Turbo: $\frac{2}{3}, \frac{8}{11}, \frac{4}{5}$ rates bitstream length ratio	1:0:1.49	1:1:4.28	1:2:5.30	1:2.75:5.56	1:2.17:5.17
DP Turbo: Source Rate(bpp) noise-free PSNR(dB)	0.08 28.50	0.20 32.59	0.40 35.76	0.59 37.60	0.78 38.92

Table 3: Rate Optimization Results Comparison for Lenna (512 x 512).

Bit Rate (bpp)	0.1	0.25	0.50	0.75	1.00
EEP SPIHT [3]	28.21	31.92	35.00	36.79	38.07
EEP J2K [3]	27.93	31.75	34.92	36.78	38.06
UEP ES-RCPC J2K [8]	28.08	31.90	35.14	37.02	38.28
UEP DP-RCPC J2K	28.08	31.90	35.14	37.02	38.26
UEP DP-Turbo J2K	28.48	32.51	35.66	37.49	38.79

Table 4: PSNR Comparison with Different Schemes for Lenna (512 x 512).

Images	Designed/Actual BER	
	0.01/0.01	0.01/0.03
Lenna [5]	38.78	30.28
Lenna (Proposed)	38.79	32.72
Goldhill [5]	34.81	24.26
Goldhill (Proposed)	34.85	28.54

Table 5: PSNR Comparison for Mismatched BER.

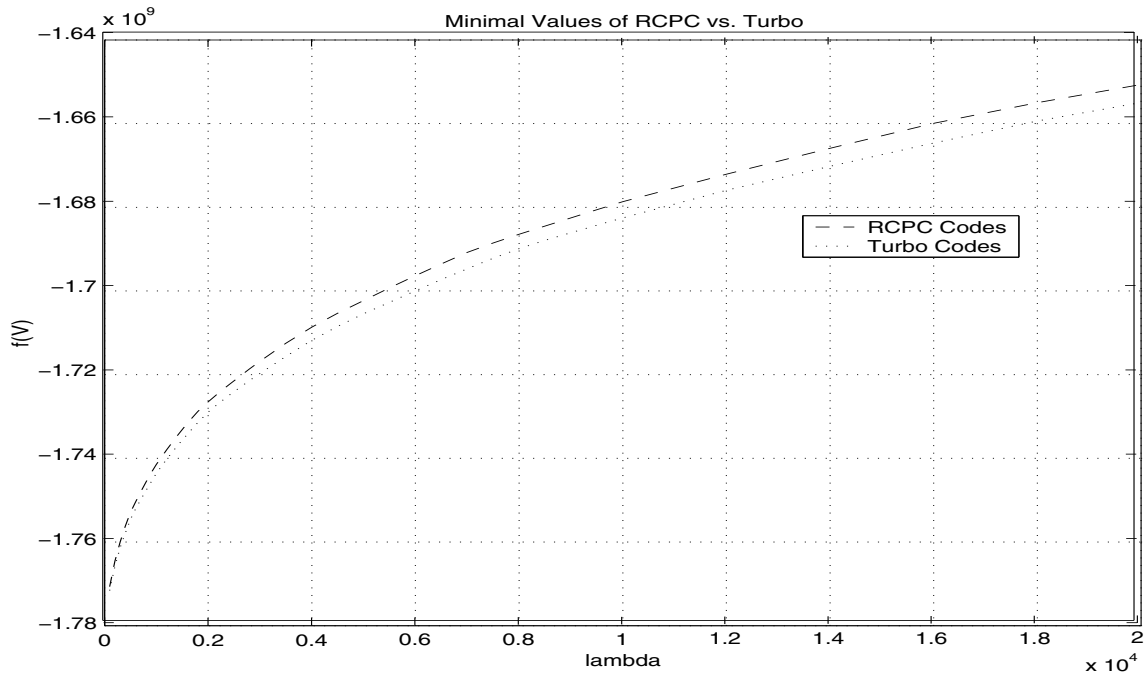


Figure 1: Optimized Minimal Values of RCPC and Turbo Codes for Goldhill (512 x 512).

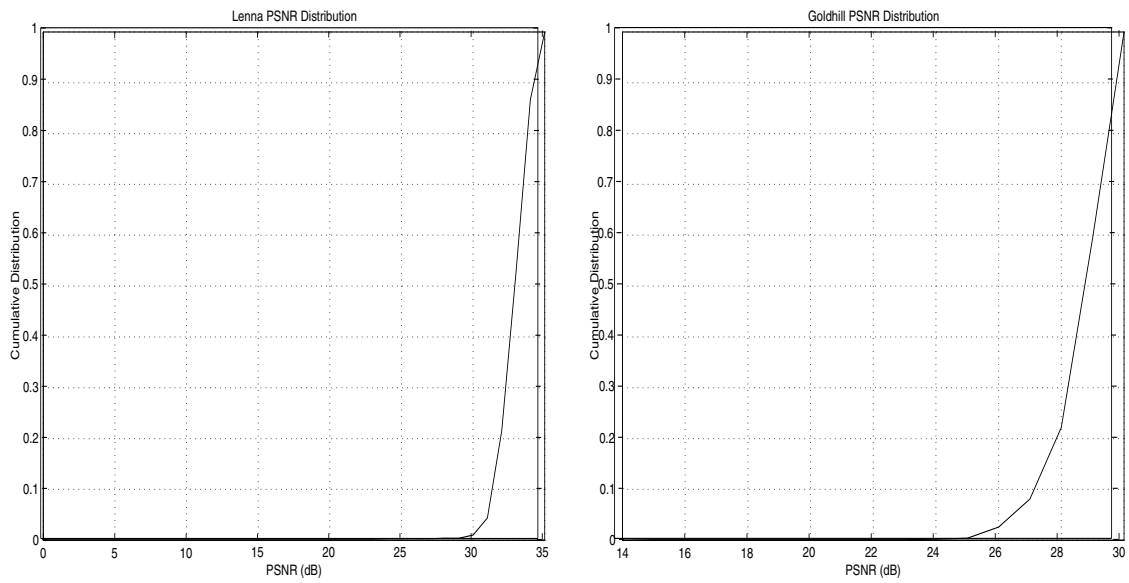


Figure 2: Lenna and Goldhill PSNR Distribution with Mismatched Channel BER.

- [3] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Processing Letters*, vol. 4, pp. 189–191, July 1997.
- [4] V. Chande and N. Farvardin, "Progressive transmission of images over memoryless channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 127–131, June 2000.
- [5] B. A. Banister, B. Belzer, and T. R. Fischer, "Robust image transmission using JPEG2000 and turbo-codes," in *Proc. IEEE ICIP-2000*, Oct 2000, vol. 1, pp. 375–378.
- [6] R. Hamzaoui, V. Stankovic, and Z. Xiong, "Rate-based versus distortion-based optimal joint source-channel coding," in *Proc. DCC-2002*, Mar 2002.
- [7] P. G. Sherwood, X. Tian, and K. Zeger, "Channel code blocklength and rate optimization for progressive image transmission," in *Wireless Communications and Networking Conference*, 1999, vol. 2, pp. 978–982.
- [8] Z. Wu, A. Bilgin, and M. W. Marcellin, "Unequal error protection for transmission of JPEG2000 codestreams over noisy channels," in *Proc. IEEE ICIP-2002*, Sept. 2002, vol. 1, pp. 213–216.
- [9] *JPEG 2000 Part I Final Draft International Standard*, ISO/IEC JTC 1/SC 29/WG1, Doc. No. N1855, 2000.
- [10] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Proc.*, vol. 9, pp. 1158–1170, July 2000.
- [11] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acous. and Sig. Proc.*, vol. 36, pp. 1445–1453, Sept. 1988.
- [12] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, Mass, 1995.
- [13] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. on Communications*, vol. 36, pp. 389–400, Apr. 1988.
- [14] D. N. Rowitch and L. B. Bilstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," *IEEE Trans. on Communications*, vol. 48, pp. 948–959, June 2000.