

Channel Code Blocklength and Rate Optimization for Progressive Image Transmission

P. Greg Sherwood, Xiaodong Tian, and Kenneth Zeger
 UC San Diego, Department of ECE
 La Jolla, CA 92093-0407

Abstract—We investigate the problem of selecting blocklength and code rate for progressive image transmission, motivated by turbo coding methods where performance improves with blocklength. The problem is to balance the tradeoff among error protection, source coding rate, and delay. We propose a general performance measure for evaluating progressive transmission and use dynamic programming to determine the channel code parameters based on the progressive performance. Performance results are provided for the evaluation of the gains over less complex methods as a function of channel error rate.

I. INTRODUCTION

Systems composed of embedded wavelet-based image coders followed by channel coding result in some of the best systems currently known for transmitting images over certain noisy channels [1], [2], [3], [4]. Selecting the proper channel code rate is important so that bits are not wasted on unnecessary redundancy, while keeping the error probability sufficiently low. Adjusting the blocklength can further improve the performance of certain channel codes.

It is known from information theory that long blocklengths can achieve good performance at rates close to capacity, and many known codes exhibit improved performance as the blocklength is increased. Examples include turbo codes, low-density parity check codes, and Reed-Solomon codes on erasure channels. The results presented in [4] for image transmission on a packet erasure channel demonstrate that excellent performance at a target transmission rate can be achieved by using the maximum possible channel codeword blocklength. For an erasure rate ϵ and channel code rate $r \leq 1 - \epsilon$, a useful upper bound on the block error probability is [5, p.531]

$$P_E(r, \epsilon, N) \leq 2^{-ND_2(1-r|\epsilon)}$$

where N is the blocklength and $D_2(a||b) = a \log_2(a/b) + (1-a) \log_2((1-a)/(1-b))$ is the information divergence. This upper bound shows the advantage of long blocklengths (i.e., large N) for rates below capacity.

If decoding at multiple rates is not important, then using long blocklength channel codes provides the best performance in many cases. One penalty associated with long blocklength codes is increased decoding complexity. For progressive coding, there is the additional penalty of decoding delay. Since the decoder typically must wait until the entire block has been

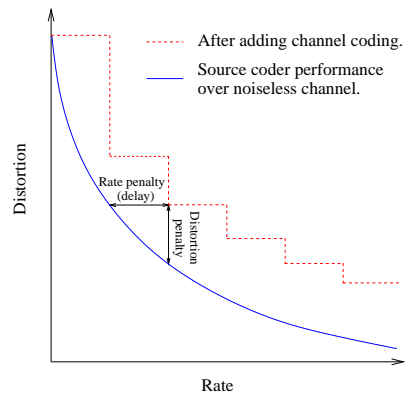


Fig. 1. Distortion vs. rate with and without channel coding.

received and decoded before reliable source bits are available, the distortion vs. rate curve flattens out over the duration of a blocklength and decreases only at the end of the block. The result is that the curve is staircase shaped, where the size of the steps depends on the channel code blocklength as illustrated in Figure 1. The tradeoff is that long blocklengths improve error protection but they also decrease progressivity.

Previous work [3], [6], [7] has considered methods of selecting the best channel code rate for image transmission on noisy channels. A gradient-based technique was used in [7] to determine the best rate allocation between source and channel codes for a given transmission rate. In [6], Lagrangian methods were used to select the channel code rate schedule for fixed length information blocks to minimize distortion at a final target rate. Although progressive source coders were used, progressive performance was not considered in [6], [7].

In [3], optimization was also performed for a specific transmission rate, but the optimization criterion and rate-compatible properties of the channel codes allowed optimal transmission at many lower rates. A dynamic programming approach was presented for determining channel code rate schedules for fixed-length information blocks using optimization criteria based on MSE, PSNR, and number of available source bits. The optimization based on number of available source bits was suggested as the best approach since it reduces complexity, eliminates the need to transmit the rate schedule, and allows a single rate schedule to incorporate the optimized rate schedules for

†This work was supported in part by the National Science Foundation

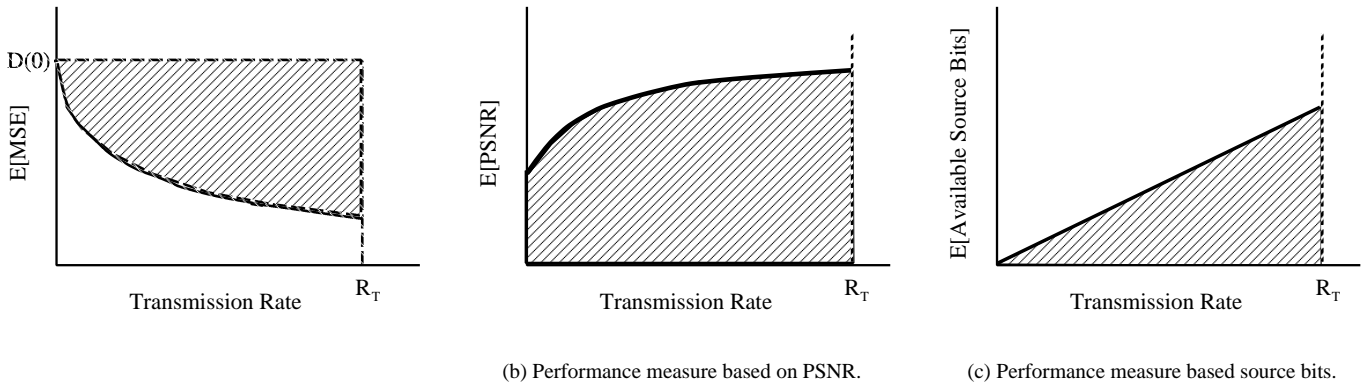


Fig. 2. Three possible progressive performance measures each equal to the area of the shaded region.

many lower transmission rates as prefixes by using the rate-compatible properties of the codes considered.

We extend the work in [3] to consider the optimization of blocklength and channel code rate, and we use a more general performance function to characterize progressive performance. Specifically, this paper includes the following contributions:

- A general performance measure is presented for evaluation of progressive performance;
- A dynamic programming solution is presented for optimizing blocklength and rate;
- Results for a number of error rates compare optimizations based on PSNR and optimizations based on available source bits with and without fixed information block sizes;
- Results for Reed-Solomon codes on erasure channels and turbo codes on bit error channels, demonstrate the importance of optimizing blocklengths.

II. A PERFORMANCE MEASURE FOR PROGRESSIVE TRANSMISSION

In order to optimize the channel code parameters, it is necessary to define a performance measure for progressive transmission. In image coding, typically the expected MSE or PSNR at a target transmission rate is used. For progressive coding, the desire is generally to maximize the expected performance at all rates up to the target transmission rate. We propose a family of performance measures given by

$$\mu = \int_0^{R_T} W(x)F(x)dx \quad (1)$$

where R_T is the terminal transmission rate, $F(x)$ is the expected performance for a given channel at transmission rate x , and $W(x)$ is a non-negative, real-valued “weighting function” which allows unequal emphasis for different rates. The performance function $F(x)$ is large for good performances and small for poor performances, in contrast to the more typical use of a “distortion” function. The performance measure μ includes,

as a special case, the conventional approach of measuring performance at a single rate (or possibly a small set of rates), by using impulse functions as part of the weighting function (e.g., $W(x) = \delta(x - r_1) + \delta(x - r_2) + \dots$ where $\delta(x)$ is the Dirac delta function).

Possible performance measures include the expected MSE, expected PSNR, or number of available source bits as noted in [3]. The PSNR-based performance is closely tied to image quality but does not emphasize the large distortions at low rates as much as the MSE-based performance. Performance based on available source bits assumes all bits as equally important but it allows simplifications in the optimization and eliminates overhead information. In both of these cases, the goal (assuming $W(x) = 1$) is to maximize the area under $F(x)$. If we define $F(x) = D(0) - D(x)$, where $D(x)$ is the expected MSE at rate x , then the goal is also to maximize μ . In the context of dynamic programming algorithms discussed in the next section, μ is the reward. Figure 2 shows examples of these performance measures where the shaded region represents the area computed by the integral with $W(x) = 1$.

III. OPTIMIZATION USING DYNAMIC PROGRAMMING

The optimal schedule of channel code parameters can be determined using dynamic programming where the goal is to maximize the reward based on the progressive performance measure (i.e., μ in (1)). The notation is based on that in [8].

Given a transmission rate constraint R_T , the goal of the optimization problem is to determine the *code schedule*, $\pi = \{(N_1, K_1), (N_2, K_2), \dots, (N_{M(\pi)}, K_{M(\pi)})\}$, subject to $\sum_{i=1}^{M(\pi)} N_i \leq R_T$ that maximizes μ . Each component of the code schedule contains a blocklength parameter, N_i , and an information size parameter, K_i . For a given system consisting of source coder, channel coder, and channel, each code schedule, π , determines an expected performance-rate function as in (1), denoted $F_\pi(x)$. Since the transmitted data consists of a sequence of codewords, the overall reward will be based on the incremental reward associated with each codeword. The incremental reward of a codeword with parameters (N_i, K_i) under

code schedule π is given by

$$r_\pi(N_i, K_i) = \int_{\tilde{R}_{i-1}}^{\tilde{R}_i} W(x)(F_\pi(x) - F_\pi(\tilde{R}_{i-1}))dx + (F_\pi(\tilde{R}_i) - F_\pi(\tilde{R}_{i-1})) \int_{\tilde{R}_i}^{R_T} W(x)dx \quad (2)$$

where $\tilde{R}_i = \sum_{j=1}^i N_j$ is the number of transmitted bits up to and including codeword i , and $\tilde{R}_0 \equiv 0$. The first term of (2) is the incremental reward during transmission of the codeword while the second term is the accumulated incremental reward from the end of the codeword to the target rate. The integrals typically simplify to sums due to the discrete nature of the problem, and for certain performance measures such as the one based on available source bits, the integrand simplifies due to the constant reward for each bit.

The underlying assumption used in combining the incremental rewards from multiple codewords into the overall reward is the serial decoding requirement of the source decoder. A source decoder is said to have a *serial decoding requirement* if a substantial portion of the bit stream must be decoded in a sequential and uninterrupted fashion for correct interpretation. Many embedded coders, especially those employing adaptive entropy coding, have this property. The effect is that decoding terminates at the first uncorrectable (and detected) error so no reward is accumulated from subsequent correctly received codewords. A “reward-to-go” function associated with codeword i is defined as

$$J_i(R_T, \pi) = r_\pi(N_i, K_i) + (1 - P_E(N_i, K_i))J_{i+1}(R_T, \pi) \quad (3)$$

where $P_E(N_i, K_i)$ is the probability of block decoding error for codeword i . The $(1 - P_E(N_i, K_i))$ term multiplying J_{i+1} is the result of the serial decoding requirement. The goal of the optimization problem is to find the optimal code schedule π^* that satisfies

$$\pi^* = \arg \max_{\pi} J_1(R_T, \pi) \quad (4)$$

with the condition that $J_i(R_T, \pi) = 0$ for some sufficiently large i under all admissible policies due to the transmission rate constraint R_T .

In the following sections, this general framework is applied to specific channel conditions and performance results are provided for the resulting rate schedules.

IV. PROGRESSIVE TRANSMISSION OVER ERASURE CHANNELS

This section considers the problem of transmitting an image over a packet erasure channel with erasure rate ϵ . The packets are assumed to be fixed length, and this length is a parameter of the optimization. The information is protected using erasure

correcting codes across the packets (similar to that found in [2], [4]), such as Reed-Solomon codes, which allow for error-free transmission of K information packets out of a block of length N as long as any K packets are received.

The codes are systematic with the information packets transmitted first so these packets are available immediately to the source decoder until the first erasure. After the first erasure, further source decoding is delayed until K packets of the current codeword have been received at which point erasure correction is possible. Assuming the weighting function from (2) is unity (i.e., $W(x) = 1$) for simplicity, the incremental reward is derived below for this case.

For the first K_i packets of the codeword, the incremental reward is based on the expected number of packets before the first erasure. Let $f(x)$ be the performance of the source coder at transmission rate x on a noiseless channel with no channel coding (e.g., number of available source packets or bits (where $f(x) = x$), PSNR, etc). Note that the rate parameter is in units of packets rather than bits, for simplicity, and the conversion to bits involves a constant multiplicative factor due to the constant packet size. The expected performance over this range of transmitted packets under code schedule π is given by

$$F_\pi(M + \tilde{R}_{i-1}) = \sum_{j=0}^{M-1} f(\tilde{K}_i + j)(1 - \epsilon)^j \epsilon + f(\tilde{K}_i + M)(1 - \epsilon)^M \quad 1 \leq M \leq K_i \quad (5)$$

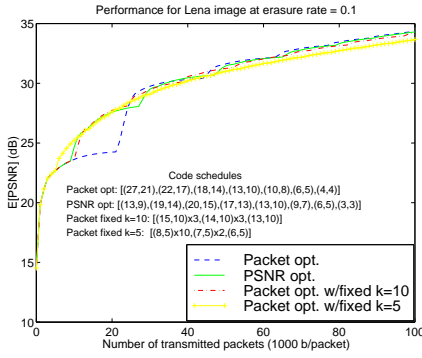
where $\tilde{K}_i = \sum_{j=1}^i K_j$ is the number of cumulative information packets and ϵ is the packet erasure rate.

During the transmission of the $N_i - K_i$ parity packets, the performance only changes at the point where a total of K_i packets have been received for that codeword (at which point erasure correction is possible). The expected performance over this range is given by

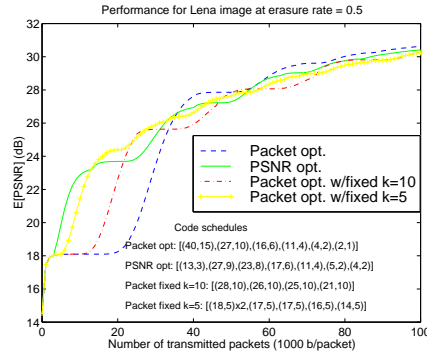
$$F_\pi(M + \tilde{R}_{i-1}) = \sum_{j=0}^{K_i-1} P_{me}(M - j - 1, M - K_i) f(\tilde{K}_i + j)(1 - \epsilon)^j \epsilon + (1 - P_E(M, K_i)) f(\tilde{K}_i + K_i) \quad K_i + 1 \leq M \leq N_i \quad (6)$$

where $P_{me}(x, y) = \sum_{j=y}^x \binom{x}{j} (1 - \epsilon)^{x-j} \epsilon^j$, is the multiple erasure probability (i.e., the probability of at least y erasures in x packets) and $P_E(M, K_i) = P_{me}(M, M - K_i + 1)$ is the block decoding error probability of a code with parameters (M, K_i) .

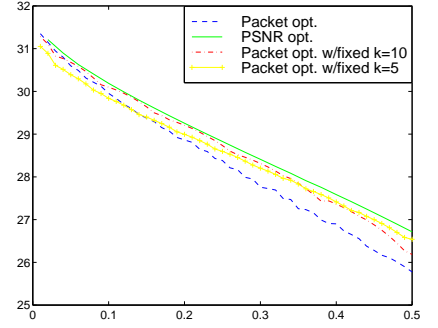
Optimization based on number of available source bits is attractive due to the reduced complexity and the elimination of the need to transmit the code schedule (i.e., the code schedule is image independent and can be computed by the receiver). However, performance measures based on MSE and PSNR are better indicators of image quality, so experiments were performed to compare the relative performance of the optimization methods. The overhead required for transmission of the code



$\epsilon = 0.1$.



$\epsilon = 0.5$.



$D(R) = \sum_{i=1}^n a_i e^{-\lambda_i R}$ with as few as two or three terms, as noted in [7]. This functional approximation could be used in determining the code schedule, and the function parameters could be transmitted instead of the actual schedule which would allow the receiver to compute the schedule.

Figure 3 shows the results of experiments using the SPIHT [9] image coder with arithmetic coding along with code schedules computed from performance measures based on PSNR and available source bits or equivalently packets. The resulting code schedules are evaluated in terms of expected PSNR for two different erasure rates in Figures 3(a) and 3(b). In addition, a method similar to that in [3], referred to as packet optimization with “fixed k ,” is applied to Reed-Solomon codes. Results are provided for code schedules with $k = 5$ and $k = 10$ information packets per codeword.

In the “fixed k ” method, the information size of each codeword is fixed and the blocklength (equivalently rate) is selected, the performance measure is available source bits (packets), and the optimization is for the final transmission rate so $W(x) = \delta(x - R_T)$ in (2). The resulting code schedules from the optimization are monotonic non-increasing in blocklength (or non-decreasing in channel code rate) due to the serial decoding requirement. Initial information packets are protected more heavily than later packets, even though every packet contains one unit of reward, since a packet can only be used if all earlier information packets are correctly received. The order of transmission of the codeword packets can be altered so that optimal (in terms of available source bits) code schedules are achieved at certain lower transmission rates as suggested in [3]. This progressive transmission is achieved by transmitting punctured versions of the codewords starting with the last

codeword parameters in the schedule. Additional parity packets are sent as needed during transmission to achieve all the intermediate code rates specified in the schedule between the last codeword and the codeword in question. Optimal schedules in terms of available source packets are achieved for transmission rates which equal the cumulative sum of the blocklengths starting from the last codeword and proceeding to the first codeword in the schedule. There are a few points to mention regarding the reordering for progressive transmission in [3]:

- The optimality is for the particular value of k and it corresponds to available source packets, not expected PSNR;
- The optimality is only guaranteed at certain transmission rates (equal to the cumulative sum of blocklengths starting from the last codeword), so other transmission orders may be better at other intermediate transmission rates;
- The best order for performance at all intermediate transmission rates is dependent on the erasure rate, and it is generally better to transmit according to the natural schedule order (i.e., all packets of each codeword in sequence) under high erasure conditions;
- The transmission rates of optimality are not selectable, but are instead determined by the code schedule computed for the final transmission rate.

The results in Figures 3(a) and 3(b) show that the optimization based on available source packets and that based on expected PSNR achieve nearly the same expected PSNR at the final transmission rate. The code schedules are also listed in the graphs. Generally, the code schedules for the packet-based optimization consist of codewords with non-increasing blocklengths and nearly identical rates with the possible exception of the final codewords due to the transmission rate constraint. The PSNR-based optimization selects a shorter but lower-rate initial codeword to reduce the delay while still keeping the error probability low, and then the remainder of the schedule is similar to the packet-based schedule. As the first two graphs in Figure 3 demonstrate, the relative performance of the different methods depends on the erasure rate. Figure 3(c) shows the per-

formance in terms of the expected PSNR averaged over all the intermediate transmission rates (essentially a normalized version of the area under the curve) over a range of erasure rates. The PSNR-based optimization has the best performance at all rates as expected, while the other curves cross at different erasure rates. Also the spread in relative performance increases with erasure rate.

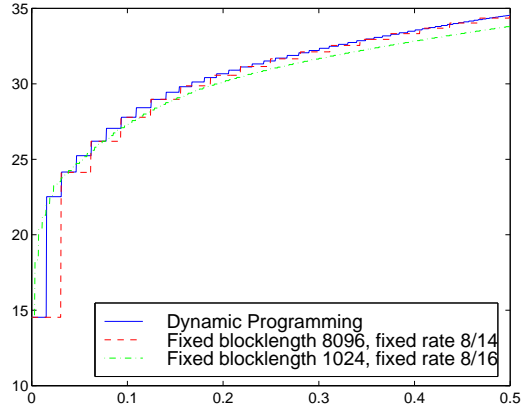
V. OPTIMIZATIONS FOR TURBO CODES

The same optimization techniques can be applied for turbo coding over bit error channels by simply changing the incremental reward function. For the erasure channel, information bits can be immediately used by the source decoder up to the first erasure since any data that arrives on this channel is known to be good. However, in this case, the information bits cannot be used before the entire codeword has been received since they are likely to contain errors which cannot be corrected until the remainder of the codeword arrives. The result is that the incremental reward of a codeword only has the second term from (2), and the performance curves have more of a staircase shape. The incremental reward for a codeword with parameters (N_i, K_i) under code schedule π is given by

$$r_\pi(N_i, K_i) = (1 - P_E(N_i, K_i))(f(\tilde{K}_i + K_i) - f(\tilde{K}_i))(R_T - \tilde{R}_{i+1}). \quad (7)$$

It is important that decoding errors are detected, because the source decoder will lose synchronization and corrupt the image with high probability if its input contains errors. Therefore an outer cyclic redundancy code (CRC) is used for error detection as well as providing a stopping criterion for the iterative turbo decoding. Also it is the block error probability that is important here and not the bit error probability so choice of interleavers becomes important for large blocklengths. For this reason, s-random interleavers [10], which enforce a minimum spread between adjacent information bits after interleaving, were used because they result in better block error probabilities than random interleavers.

Figure 4 shows the performance-rate curves comparing a schedule determined by a PSNR-based optimization, along with two fixed rate and blocklength turbo coding schemes. The turbo codes consisted of a family of punctured codes produced from a rate 1/3 mother code. Note that using punctured codes allows the possibility of optimizing only the rate (i.e., the “fixed-k” method mentioned in the previous section) and using the progressive transmission order, but recent results suggest that non-punctured codes perform significantly better for the same code parameters [11]. The optimized code schedule has a similar structure to those computed for the erasure channel, where the code rate is essentially constant and the blocklength decreases monotonically. As can be seen in the graph, the optimized schedule strikes a balance between performance at low and high transmission rates relative to the fixed coding methods. There are also decoding complexity benefits to the shorter blocklengths at the end of the optimized schedule.



0.05.

VI. CONCLUSION

We have proposed a general performance measure for evaluating progressive image transmission systems operating over noisy channels. Using this performance measure, a dynamic programming optimization algorithm was presented for determining the best code schedule where both block length and rate are optimized. While the performance gains of the image dependent PSNR-based optimization may not warrant the additional complexity in every application, the method is useful for evaluating the relative performance of simpler coding schemes. Future work will extend the investigation with turbo codes and include comparisons with low density parity check codes.

REFERENCES

- [1] P. G. Sherwood and K. Zeger, “Progressive image coding for noisy channels,” *IEEE Sig. Proc. Letters*, vol. 4, no. 7, pp. 189–191, July 1997.
- [2] P. G. Sherwood and K. Zeger, “Error protection for progressive image transmission over memoryless and fading channels,” *IEEE Trans. Comm.*, vol. 46, no. 12, pp. 1555–1559, Dec. 1998.
- [3] V. Chande and N. Farvardin, “Joint source-channel coding for progressive transmission of embedded source coders,” in *Proc. DCC ’99*, 1999.
- [4] A. E. Mohr, E. A. Riskin, and R. E. Ladner, “Graceful degradation over packet erasure channels through forward error correction,” in *Proc. DCC ’99*, 1999.
- [5] R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, Inc., 1968.
- [6] J. Lu, A. Nosratinia, and B. Aazhang, “Progressive source-channel coding of images over bursty error channels,” in *Proc. ICIP 98*, 1998, pp. 127–131.
- [7] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Konzentsev, “Joint source channel matching for a wireless communication link,” in *Proc. ICC-98*, 1998, pp. 482–486.
- [8] D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*, Prentice Hall, 1987.
- [9] 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, no. 3, pp. 243–250, June 1996.
- [10] D. Divsalar and F. Pollara, “Turbo codes for PCS applications,” in *Proc. ICC-95*, 1995, pp. 54–59.
- [11] S. Benedetto, R. Garelo, and G. Montorsi, “A search for good convolutional codes to be used in the construction of turbo codes,” *IEEE Trans. Comm.*, vol. 46, no. 9, pp. 1101–1105, Sept. 1998.