Error Protection for Progressive Image Transmission over Memoryless and Fading Channels *

P. Greg Sherwood
Dept. of ECE, UCSD
La Jolla, CA 92093-0407
sherwood@code.ucsd.edu

Kenneth Zeger
Dept. of ECE, UCSD
La Jolla, CA 92093-0407
zeger@ucsd.edu

Abstract

A product channel code is proposed to protect progressively compressed and packetized image data that is transmitted across noisy channels. Across packets, the product code is composed of Reed-Solomon codes. Within packets, the product code uses the concatenation of a rate compatible punctured convolutional code and an error detecting parity check code. The benefits include flexibility in terms of delay, the ability to easily adapt the level of protection based on importance (i.e., unequal error protection), and scalable decoding complexity. The system outperforms the best known image coders for memoryless channels and performs well on fading channels.

1. Introduction

Numerous elaborate attempts have been made in the past decade to protect transmitted images from the effects of channel noise. The best known image coders tend to behave very poorly in the presence of channel noise, often because of the finite-state nature of the compression algorithms. In contrast, suboptimal image coders are often very robust to channel noise. In [1], an implementable system was described for protecting the successful Embedded Zerotree Wavelet coding algorithm from channel noise. It was demonstrated that performance exceeding that of previous coders could be achieved while maintaining the progressive nature of the image coder. However, the system was designed exclusively for use on a memoryless channel.

In the present paper, we extend the work in [1] to the case of a fading channel. We do so by using a channel coding system which is specifically designed for use with packetized output data from one of the best wavelet based algorithms known. In addition to working well on a fading channel, the system turns out to actually improve upon the performance in [1] for the memoryless case as well.

The main idea is to break the image coder bit stream into packets, encode it with the same concatenated channel coder used in [1], and then to add a Reed-Solomon code across the packets. Thus it provides a second layer of protection and is specifically suited to the progressive wavelet based algorithms, since no fixed interleaver delay is needed. If, for example, every bit in a packet (i.e., a row in the product code) is corrupted after row decoding because of a short term error burst, this shows up as a single symbol erasure in each column and can easily be corrected by the column Reed-Solomon codes. The proposed system is described below, and numerical results are presented afterwards showing its effectiveness.

2. A Product Channel Code

A product code is often described as a two-dimensional code constructed by encoding a rectangular array of information digits with one code along rows and with another code along columns [2]. In the product code used here, the row code is a concatenated code consisting of an outer cyclic redundancy code (CRC) and an inner rate-compatible punctured convolutional (RCPC) code while the column code is a systematic shortened and/or punctured Reed-Solomon (RS) code. The structure of the product code is depicted in Figure 1. Note that the RCPC codes used for the rows are not systematic, and the symbols for the RS codes are constructed from consecutive information bits of a row prior to encoding with the RCPC/CRC code.

The RCPC/CRC concatenated code allows substantial flexibility in choosing the code rate and block length of the rows. The row codes are decoded using the list-Viterbi algorithm which selects the trellis path with the best metric, that also satisfies the CRC check, from a ranked list of candidates. The correct

---

*This work was supported in part by the National Science Foundation.

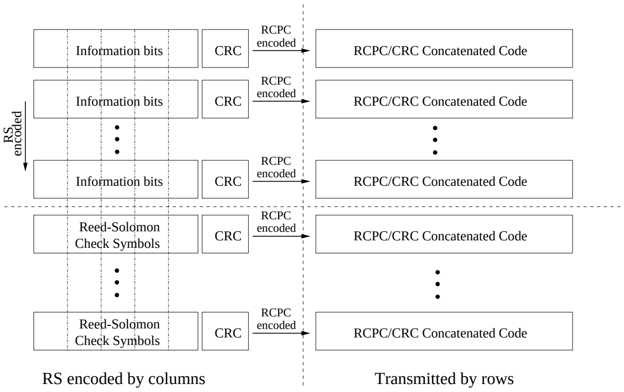

Figure 1: Schematic diagram of RCPC/CRC and RS product code.

path is typically among the first few top candidates so a long search is rarely necessary. Also a sequential version of the algorithm can reduce computational complexity by only searching for the next best path after higher ranking candidates have failed. More thorough discussions of the list-Viterbi decoding algorithm and applications can be found in [1, 3], and recent applications of list-based decoding to turbo codes can be found in [4].

An important property utilized by the product code is that the CRC provides a high probability indication of the decoding success or failure. The decoding failures in the row codes are treated as erasures when decoding the column RS codes. Since the column codes typically only need to correct erasures, the computational complexity is reduced and twice as many incorrect rows can be recovered compared to a decoder without error detection on the rows. The Galois field of the RS code symbols should be chosen as large as possible to reduce the number of columns in a block and to give the most flexibility in choosing the block lengths via shortening and/or puncturing. The RS codes are in systematic format with information symbols transmitted first, so the final rows of the product codeword will be the RCPC/CRC encoded parity symbols of the RS column codes.

There is no requirement that the rows be consecutive in the bit stream, and, in fact, spreading out the rows is important for good performance over fading channels. Of course, another design goal is to minimize delay in order to achieve rapid improvement in image quality, and this will provide some constraints to limit the duration of the code. A useful feature of this particular product code is that decoding columns is unnecessary unless a decoding failure is detected in a row code. Therefore, the decoded bits from a row can be used immediately if no decoding failure is detected in the row, eliminating the delay cost when the channel is clear.

The product code is well suited for burst errors since entire rows can easily be recovered. This property is important since even a single bit error in a packet of data from the embedded zerotree algorithms usually renders the entire packet (and also the packets to follow) useless. The product channel code also performs excellently over memoryless channels like the additive white Gaussian noise (AWGN) or binary symmetric (BSC) channels. This observation is an additional side benefit.

We illustrate the effectiveness of this code by way of an example on a BSC. The source and channel were selected to allow comparison with the results from [1] and consist of the progressive zerotree wavelet coder with arithmetic coding by Said and Pearlman (SPIHT) [5] transmitted over a BSC with error rate 0.1. The total block length of the column RS codes with symbols from GF(256) was limited to 20 symbols to limit the decoding delay, and the source bit stream was protected uniformly by selecting the combination of RCPC/CRC and RS code rates which gave equivalent probability of error results as the RCPC/ CRC code in [1]. Also the path search depth of the list-Viterbi algorithm was limited to 100 as in [1], but we note that a search depth of 10 will provide almost all the performance gain of list decoding. Under these conditions, the same probability of decoding error can be achieved with an overall channel coding rate of 0.295 for the product code versus a rate of 0.257 for the RCPC/CRC code alone. Therefore, nearly 15 percent more rate is available for source coding for a given overall transmission rate. This results in a gain in decoded image quality of about 0.5 dB in PSNR. The decoded PSNR values for the Lena image as a function of rate for these codes are shown in Figure 2 along with the noiseless channel results for the source coder without any channel coding for comparison.

Several modifications are possible to tune the properties of the code. In general, more efficient codes can be created using longer RS codes (i.e., more rows) at the expense of more delay required to correct row decoding failures. Also, some improvement in error performance can be obtained at the expense of complexity by using an iterative decoding algorithm (e.g., turbo decoding). Finally, while the above results for the BSC use hard decision decoding, soft decisions could easily be incorporated into the decoding process giving improved performance.

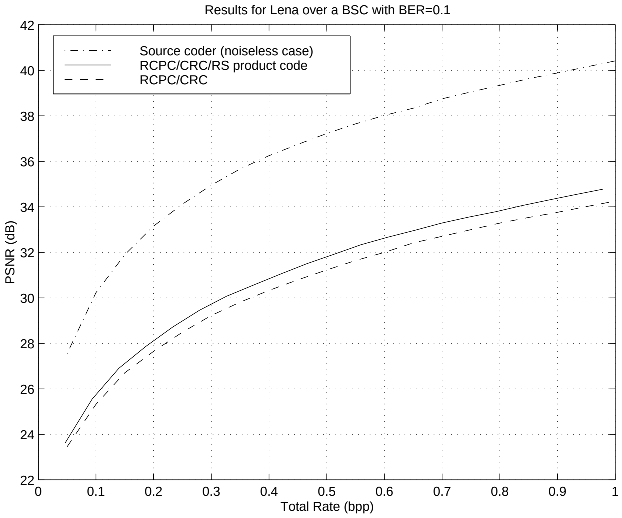

Figure 2: Comparison of code performance for the 512x512 image 'Lena' over a BSC with BER=0.1.

3. Fading Channels

While the performance of the product code is good over memoryless channels, one of its most important features is its suitability for fading channels which arise in wireless applications. A typical approach to error control for fading channels is to introduce a bit interleaver which spreads out adjacent bits by the interleaver depth before transmission over the channel. The goal is to produce an effective channel, after the de-interleaver, which is almost memoryless and then to use conventional error control coding (e.g., convolutional codes) to deal with the errors. One problem with this approach is that an interleaver of depth n introduces a delay on the order of n^2, and this delay is constant regardless of the channel conditions. Any delay impacts the performance of the progressive coder because the goal is to improve quality (PSNR) as rapidly as possible and delay shifts the PSNR vs. rate curve to the right, lowering the PSNR for a given rate. As mentioned earlier there is no significant delay with the product code unless there is a row decoding failure. In that case, the delay depends on the number of bits that must be received before the necessary number of RS check symbols are available (i.e., equal to the number of erased rows).

Simulations of BPSK transmission over a flat-fading Rayleigh channel were performed using Jakes' method [6] to model the channel. With this model, the channel is characterized by two parameters - the average SNR, which characterizes the average bit error rate, and the normalized Doppler spread (i.e., the Doppler spread normalized by the data rate), which characterizes how quickly the channel changes over time. The results presented are for a channel with average SNR of 10 dB and a normalized Doppler spread of 10^{-5} which is near the low end of many typical applications. An example leading to a normalized Doppler value of 10^{-5} would include a data rate of 500 Kbits/sec transmitted at 900 MHz to/from a mobile traveling at about 4 miles/hour (e.g., a person walking). The average fade duration is dependent on the fade margin which characterizes the amount that the SNR can be reduced before communication becomes unreliable. For example, with the parameters mentioned above, the average duration of a fade with a channel bit error rate exceeding 0.1 is on the order of 12000 bits while the average duration of a fade with a channel bit error rate exceeding 0.01 is on the order of 24000 bits. The strength of the RCPC/CRC channel codes in this case will determine the channel error rate which can be handled and thus the fade margin. Normalized Doppler values lower than 10^{-5} result in fade durations that include such a large portion of the bits that the channel is either good or bad for almost the entire image transmission - a situation better combatted using spatial diversity, frequency diversity, etc.

The results presented in Figure 2 for the BSC essentially had a single decoded PSNR value at each rate (despite being an "average" value) for each of the two codes since the probability of decoding failure was so low. However, when examining the performance of a code after transmission over a noisy channel, it is often insufficient to consider only the mean decoded PSNR, especially for time-varying channels. Instead, a distribution of decoded PSNR values for each rate of interest is more appropriate since it shows the variability of the decoded image quality. A realistic performance measure should include some combination of the expected PSNR and a measure of the variability, although the best relative weighting is probably viewer and application dependent.

Therefore performance results for the fading channel will be presented using a cumulative distribution of decoded PSNR values at a particular total transmission rate. In this type of plot, curves with better performance will generally lie closer to the bottom and right edges of the plot indicating a higher frequency of large PSNR values. Note that reported mean PSNR values are computed by averaging decoded MSE values and then converting the mean MSE to the corresponding PSNR value rather than averaging the PSNR values directly.

For the results presented in this section, each of the

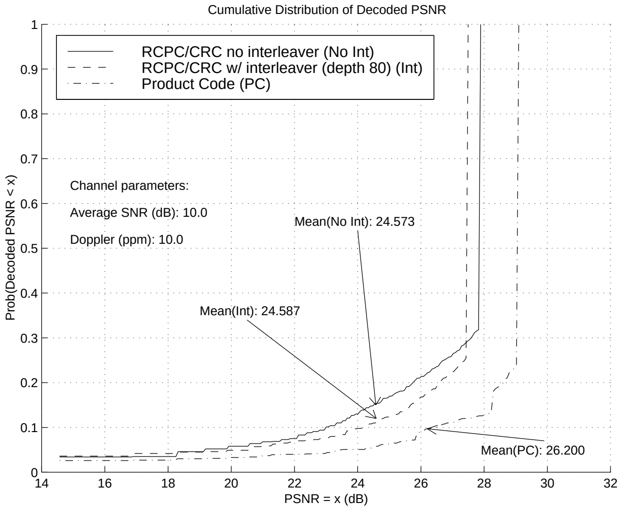

Figure 3: Comparison of code performance for the 512x512 image 'Lena' transmitted over a Rayleigh fading channel with total rate 0.25 bpp.

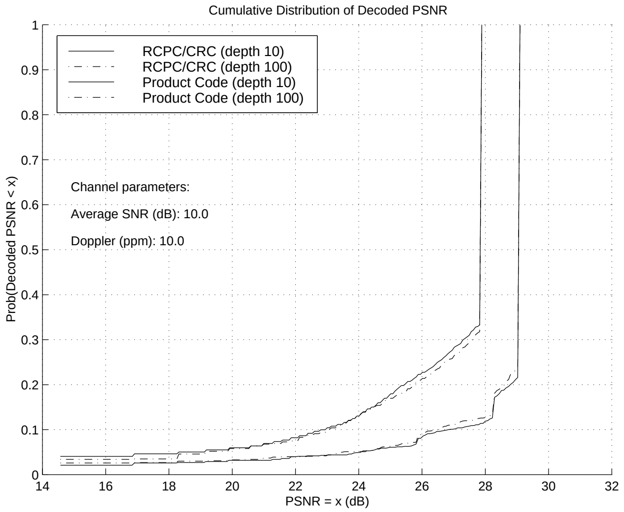

Figure 4: Comparison of code performance with different trellis search depths for the 512x512 image 'Lena' transmitted over a Rayleigh fading channel with total rate 0.25 bpp.

codes was constructed from the output of the SPIHT[5] image coder with arithmetic coding. Each cumulative distribution curve represents at least 1000 independent trials. The 512x512 Lena image was used in each case and the total rate (including both channel and source coding) considered was 0.25 bpp. Blocks of 200 information bits were protected by a 16 bit CRC and encoded by RCPC codes of various rates. For the product codes, groups of 8 consecutive information bits made up the symbol values for the RS codes over the finite field GF(256). The path search depth of the list-Viterbi algorithm was limited to 100 paths.

The first set of results shown in Figure 3 compares the performance of a RCPC/CRC concatenated code with RCPC rate 1/4, the same code with a convolutional interleaver of depth 80, and a product code using RCPC rate 1/2, a shortened (16, 10) RS code, and a row spacing of 4 (i.e., 4 interleaved product codes). The results show that the bit interleaver does not really improve the performance since the interleaver depth is small relative to the average fade duration given above. The use of the interleaver does reduce the tail of the distribution slightly, but the peak PSNR is also reduced due to the interleaver delay so the mean is only improved by 0.014 dB. The product code performs much better than either of the other two codes giving both a higher peak PSNR and a lower tail which results in an improvement in peak PSNR of 1.2 dB and in mean PSNR of 1.6 dB.

As with the BSC, limiting the path search depth of the list-Viterbi algorithm to 10 paths makes very little difference in the distribution of decoded PSNR values for either the RCPC/CRC-only code or the product code on the fading channel. For the channel conditions described above, changing the path search depth from 100 to 10 results in almost no difference in the PSNR distributions. Figure 4 shows cumulative distributions for the RCPC/CRC code without interleaving and the product code shown in Figure 3 with trellis search depths 10 and 100 paths. Clearly, the distributions for each code are nearly identical using either 10 or 100 paths. In fact, limiting the search depth to 10 paths may slightly benefit the product code performance because searching deep in the list of candidate trellis paths increases the likelihood of mistakenly selecting an incorrect path which is worse for column decoding than simply declaring an erasure.

4. Unequal Error Protection

The results presented in the previous sections were for channel protection applied uniformly to the source coder output. However, the bits do not have equal importance in determining the final decoded PSNR. For progressive image coders like Shapiro's EZW [7] and Said and Pearlman's SPIHT [5] algorithms, the importance of the bits roughly decreases monotonically throughout the bit stream. A better rate-distortion tradeoff can be achieved by tailoring the channel protection based on the importance (i.e., unequal error protection).

Due to the use of variable length and adaptive arithmetic coding in the source coder, uncorrected channel errors lead to loss of synchronization at the decoder. Therefore, the decoder must stop decoding at the first detected but uncorrected error. This situation leads to the long flat tail in the distributions shown in Figure 3 where the minimum decoded PSNR value of 14.53 dB represents the case where the first packet was lost and only the image mean was used for reconstruction.

The code structure allows several ways to implement an unequal error protection (UEP) scheme including: varying the rate of the row RCPC/CRC code, varying the rate of the column RS code, or including important information rows in multiple product codes. For the channel conditions and codes used in Figure 3, the best approach for reducing the long tail of the distribution is probably to include the initial information packets in additional product codes. The reason is that additional RCPC coding may still have problems correcting the errors associated with a deep fade, and simply increasing the redundancy of the column codes may not allow tailoring the additional protection to the importance as accurately (i.e., the rows in a given product code can have a large variation in importance, especially at the beginning).

Results for two UEP schemes as well as the product code from Figure 3 are shown in Figure 5. Both UEP schemes were constructed by protecting the first two information packets with an additional shortened (4, 2) RS column code and transmitting the parity rows after half of all packets had been sent. In addition, the first 10 information packets were protected by a shortened (20, 10) RS column code with the parity rows sent as the final 10 packets of the image. The difference between the two was in the other coding parameters where UEP scheme 1 used RCPC rate 2/3, a shortened (15, 8) RS code, and row spacing of 4; and where UEP scheme 2 used RCPC rate 1/2, a shortened (16, 10) RS code, and row spacing of 4.

As can be seen in Figure 5, both UEP schemes greatly reduce the tail of the distribution which is reflected in the increased mean values with gains of 0.9 dB and 1.6 dB compared with the uniform product code. The fact that the mean for UEP 2 is greater than for UEP 1 comes at the expense of decreased peak (and typical) PSNR, which makes it difficult to judge which of the two is preferable and illustrates the point that mean PSNR may not always be the best metric.

5. Conclusion

A product channel code has been proposed for use in progressive image transmission over memoryless and fading channels. Simulation results have been provided to demonstrate that it performs well for these channels. The code structure offers flexibility in choosing parameters which control decoding delay, unequal error protection, and decoding complexity. Also, on good channels there is no added decoding delay, in contrast to using an interleaver.

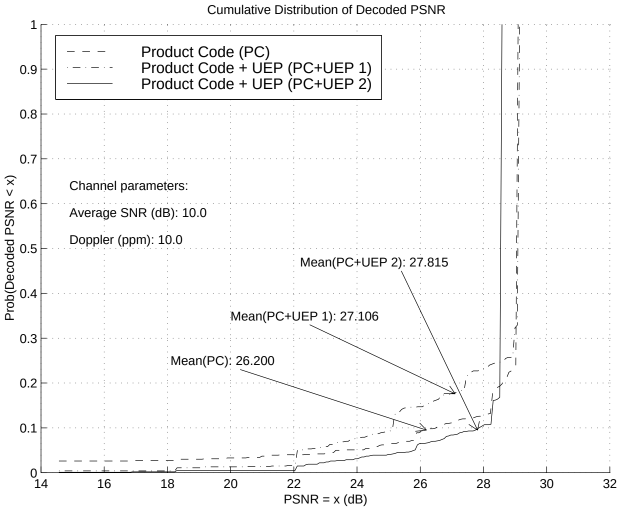

Figure 5: Comparison of code performance for the 512x512 image 'Lena' transmitted over a Rayleigh fading channel with total rate 0.25 bpp.

References

[1] P. G. Sherwood and K. Zeger. Progressive image coding for noisy channels. *IEEE Sig. Proc. Letters*, 4(7):198–191, July 1997.

[2] S. Lin and D. J. Costello, Jr. *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[3] N. Seshadri and C.-E. W. Sundberg. List Viterbi decoding algorithms with applications. *IEEE Trans. on Comm.*, 42(2/3/4):313–323, Feb/Mar/Apr 1994.

[4] K. R. Narayanan and G. L. Stüber. List decoding of turbo codes. *IEEE Trans. on Comm.*, 46(6):754–762, June 1998.

[5] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on CSVT*, 6(3):243–250, June 1996.

[6] W. C. Jakes. *Microwave Mobile Communications.* Wiley-Interscience, 1974.

[7] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Sig. Proc.*, 41(12):3445–3462, December 1993.