

Error Protection for Progressive Image Transmission Over Memoryless and Fading Channels

P. Greg Sherwood and Kenneth Zeger, *Senior Member, IEEE*

Abstract—Product channel codes are proposed to protect progressively compressed and packetized images for noisy channels. Within packets, the product code uses the concatenation of a rate-compatible punctured convolutional code and an error detecting parity check code. Across packets, Reed–Solomon codes are used. Benefits include flexible choice of delay, adaptability of error protection level (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.

Index Terms—Channel coding, fading channels, image coding, Rayleigh channels.

I. INTRODUCTION

NUMEROUS elaborate attempts have been made in the past 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], a system was described for protecting the SPIHT [3] image coding algorithm. It was demonstrated that performance exceeding that of previous coders could be achieved while maintaining the progressive property of the image coder. However, the system in [1] 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 slightly 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 them with the same concatenated channel coder used in [1], and then to add a Reed–Solomon (RS) 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 interleave 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 RS codes. The proposed system is described below, and numerical results are presented afterwards showing its effectiveness. Complete specifications of the codes used in this paper are given in the Appendix.

II. 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 [4]. 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 RS code. The structure of the product code is depicted in Fig. 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, subject to the constraint that it also satisfies the CRC check, from a ranked list of candidates. The correct 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] and [5].

An important property of the row code that is exploited by the column code is that the CRC provides a high probability indication of the decoding success or failure of a packet. 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 lost 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, within the limits of implementation constraints, 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

Paper approved by M. Fossorier, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received April 7, 1998; revised July 18, 1998. This work was supported in part by the National Science Foundation. This paper was presented in part at the International Conference on Image Processing (ICIP), October 1998.

The authors are with Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: {sherwood@code.ucsd.edu; zeger@code.ucsd.edu}).

Publisher Item Identifier S 0090-6778(98)09398-2.

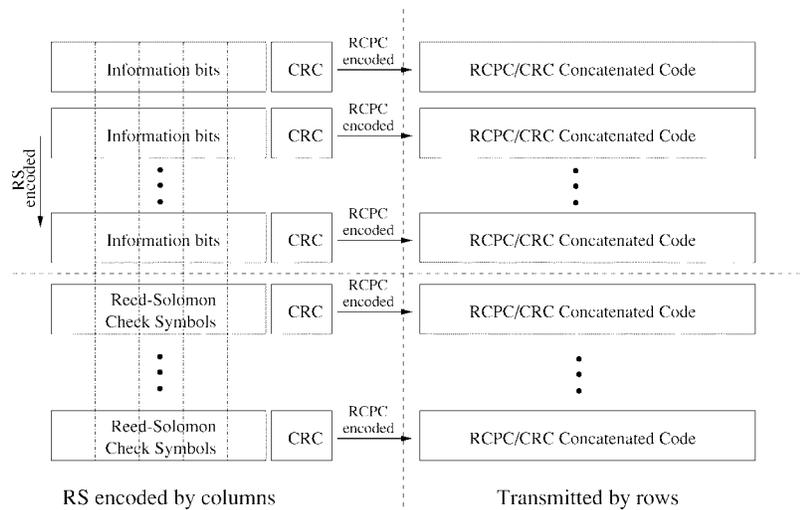


Fig. 1. Schematic diagram of RCPC/CRC and RS product code.

rows of the product code word 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. A row spacing greater than one (i.e., interleaving multiple product codes) is used in some of the examples below to achieve better performance over fading channels at lower decoding complexity than a long code. Another design goal is to minimize delay in order to achieve rapid improvement in progressive image quality, and this constrains the duration of the code. A nice feature of this particular product code is that decoding the 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. When the channel is in a good state, immediate progressive decoding of the image occurs, and when the channel is in a bad state, correct decoding can still occur but perhaps delayed by the number of packets in the product code buffer.

The code presented is well suited for burst errors since entire rows can easily be recovered (since a corrupted row appears merely as a single erasure in each column code). This property is important since even a single bit error in a packet of data from embedded zero tree algorithms usually renders the entire packet (and also the packets to follow) useless. As an additional side benefit, the product channel code also performs well over memoryless channels such as additive white Gaussian noise (AWGN) and binary symmetric (BSC) channels.

We illustrate the effectiveness of this code by way of a simple example on a BSC. The source and channel were selected to allow comparison with the results from [1]; these were the progressive zero tree wavelet coder with arithmetic coding used by Said and Pearlman (SPIHT) [3] and a BSC with bit error rate 0.1. The total block length of the column RS codes with symbols from GF(256) was limited to 20 symbols to reduce the decoding delay and complexity, 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

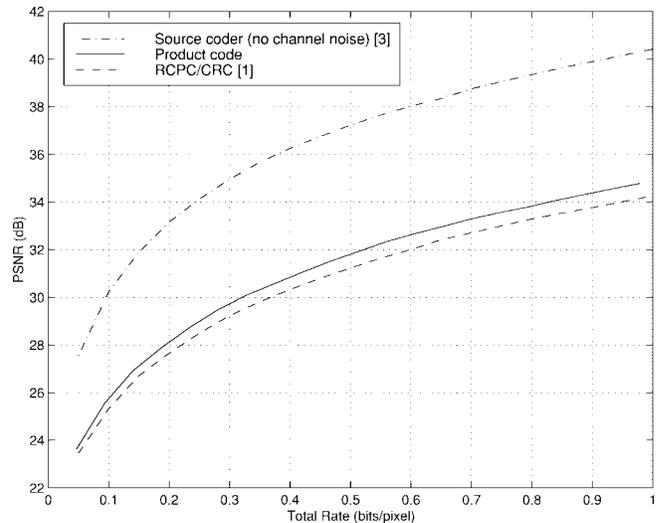


Fig. 2. Comparison of code performance for the 512×512 image "Lena" over a BSC with $\text{BER} = 0.1$.

in [1]. 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% 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 uniformly across all transmission rates. The decoded PSNR values of the Lena image as a function of rate for these codes are shown in Fig. 2 along with the noiseless channel results for the source coder without any channel coding, for comparison.

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 can easily be incorporated into the decoding process giving improved performance.

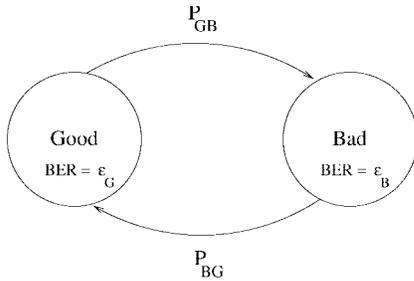


Fig. 3. Gilbert–Elliot two-state channel model.

III. 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 interleave depth before transmission over the channel. The goal is to produce an effective channel, after the de-interleaver, which is nearly 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. Any delay shifts the PSNR versus 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).

A basic channel model incorporating the memory associated with fading channels is the Gilbert–Elliot (GE) two-state model. A diagram of the channel model is shown in Fig. 3. In each state, the channel acts like a BSC with a certain bit error probability (ϵ_B for the bad state and ϵ_G for the good state), and at each bit interval, the channel changes state with probabilities governed by the model transition probabilities $P_{GB} = \Pr(\text{good} \rightarrow \text{bad})$ and $P_{BG} = \Pr(\text{bad} \rightarrow \text{good})$. This same model was used in [6] to model a channel with memory.

The Gilbert–Elliot channel model was used to compare the effectiveness of the product code versus the RCPC/CRC code along with bit interleaving. The selected model parameters were $\epsilon_B = 0.1$, $\epsilon_G = 0.001$, $P_{BG} = 1/400$, and $P_{GB} = \frac{1}{9}P_{BG}$. With these parameters, the steady state probability of being in the bad state is 0.1 and the mean burst duration is 400 bits. The performance requirement we chose for the codes was the same as that used for transmission over a BSC—at least 99% error-free transmission at a total transmission rate of 1.0 bits/pixel. These parameters were chosen to illustrate the performance of this system. In practice, more realistic values would need to be determined of course.

An RCPC/CRC code of rate 0.257 used on a BSC with BER = 0.1 conservatively meets the performance requirement, as it codes for the worst case error rate seen in the bad

state. However, using bit interleaving at the expense of a fixed delay, a higher code rate can achieve the same probability of error requirements. Fully interleaving the channel so that the errors appear memoryless was considered impractical for this channel since the required interleave depth would be excessive when considering the goals of low overall transmission rates and rapid image quality improvement. An interleave depth of 60 was selected as a reasonable value, and although it does not completely remove the memory, an RCPC/CRC code of rate 0.36 is able to meet the probability of error requirements.

The product code was designed by starting with a row RCPC/CRC code with a rate of 0.81 which was selected because it met the performance requirement when the channel was in the good state. The column code was then chosen to handle the row decoding failures that occur when a portion of a packet is transmitted during the bad state. In order to reduce the necessary error correction capability (and thus the redundancy) of the column code, several product codes were interleaved so that a single error burst would not cause multiple row erasures within a single product code. Although interleaving in this manner does increase the delay when row erasures occur versus not interleaving, the delay is not fixed as in the case of bit interleaving and is typically much less than the full extent of the product code. The selected column code consisted of a (20, 11) RS code over GF(256) with a row spacing of 6 (i.e., six interleaved product codes). The rate of the resulting product code is about 0.44 which yields a 23% increase in the available source rate compared with the interleaved RCPC/CRC code. The higher rate of the channel code translates into an increase of about 1.0 dB in PSNR over a range of transmission rates for the Lena image, similar to the gain displayed in Fig. 2.

Further simulations were performed for BPSK transmission over a flat-fading Rayleigh channel using Jakes' method [7] to model the channel. With this model, the channel is characterized by two parameters—the average SNR, which determines the average bit error rate, and the normalized Doppler spread (i.e., the Doppler spread normalized by the data rate), which determines 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 probably near the low end of values of interest. An example leading to a normalized Doppler value of 10^{-5} would include a data rate of 500 kbits/s transmitted at 900 MHz to/from a mobile traveling at about 4 mi/h (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 12 000 bits while the average duration of a fade with a channel bit error rate exceeding 0.01 is on the order of 24 000 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 combated using spatial diversity, frequency diversity, etc.

The results presented in Fig. 2 for the BSC essentially had a single decoded PSNR value at each rate 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 dependent.

Therefore, performance results for the fading channel will be presented using a cumulative distribution of decoded PSNR values for a transmission rate of 0.25 bits/pixel. 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 codes was constructed from the output of the SPIHT [3] image coder with arithmetic coding. The 512×512 Lena image was used in each case and the total rate (including both channel and source coding) considered was 0.25 bits/pixel. 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 eight consecutive information bits made up the symbol values for the RS codes over the finite field GF(256). At least 1000 independent trials are represented in each of the distributions that follow.

The results shown in Fig. 3 compare the performance of: 1) an RCPC/CRC concatenated code with RCPC rate 1/4; 2) the same code with a convolutional interleaver of depth 80; 3) a product code constructed from a rate 1/2 RCPC/CRC row code and a (16, 10) RS column code with row spacing of 4 (i.e., four interleaved product codes); and 4) another product code using the same basic parameters but with additional coding to implement an unequal error protection (UEP) scheme which is described below. The results show that the bit interleaver is not very effective for this channel since the average fade duration is so long (see previous computations). The small reduction of the distribution tail is mitigated by the 0.4-dB reduction in peak PSNR. The product code with uniform protection performs much better than the RCPC/CRC coding with and without bit interleaving 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.

The code structure allows several ways to implement an unequal error protection 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 Fig. 4, 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

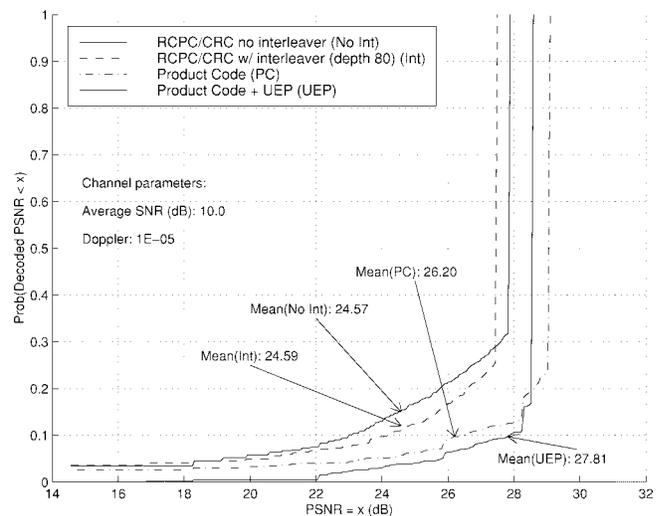


Fig. 4. Cumulative distribution of decoded PSNR for the 512×512 image "Lena" transmitted over a Rayleigh fading channel at rate 0.25 bits/pixel.

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 of the bit stream for the SPIHT coder).

The UEP scheme from Fig. 4 was 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 ten information packets were protected by a shortened (20, 10) RS column code with the parity rows sent as the final ten packets of the image. The extra protection greatly reduced the probability of losing any of the initial ten packets as can be seen by the much lower distribution tail. The result was an increase in mean PSNR of 1.6 dB over the uniform protection product code at the expense of a reduction of 0.5 dB in peak PSNR due to the extra channel coding. This UEP scheme is just one example of how the code structure allows the distribution to be shaped to better match the performance criterion of the application.

IV. CONCLUSION

A product code has been presented for the protection of progressively coded images transmitted over noisy channels with memory. As an added benefit, these codes actually also improve the performance over binary symmetric channels. The code structure is very flexible, allowing properties such as level of protection, decoding delay, and complexity to be tuned according to the performance criterion of the application. Unequal error protection can be implemented in many ways to enhance the performance, especially on fading channels. Finally, the results for fading channels were presented using the cumulative distribution of decoded PSNR values. This method of reporting results is more informative than the mean PSNR, especially for variable channels. We propose that it be adopted by other researchers as a method for reporting the performance of source coders on noisy channels.

TABLE I
RCPC CODES

Rate	Mother Code	Puncturing Matrix	Rate	Mother Code	Puncturing Matrix
8/9	155	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	2/5	155	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
	117			123	
	123			137	
				147	
1/2	155	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	4/11	155	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
	117			123	
	123			137	
				147	

APPENDIX

Polynomials will be expressed in octal notation (e.g., octal 13 is 001011 in binary which translates to the polynomial $X^3 + X + 1$).

All codes used a 16-bit CRC defined by the polynomial 254 465. All product codes in this paper were constructed from RCPC codes with memory 6 mother codes, and each packet was terminated with enough zero bits to flush the state of the convolutional coder (i.e., 6 bits in this case). The search for the correct path in the list-Viterbi algorithm was terminated after 100 candidates as in [1]. However, note that limiting the search depth to ten candidates gives almost the same performance. The puncturing matrices and mother codes are listed in Table I for each rate used in the paper. The product code in the BSC

test used the rate 4/11 code on packets of 200 information bits and a (20, 18) RS column code. In the GE channel tests, the RCPC/CRC code with interleaving used the rate 2/5 code on 200 bit packets, and the product code used the rate 8/9 code on 224 bit packets. In the Rayleigh channel tests, the RCPC/CRC used the mother code of the 4/11 and 2/5 codes listed in Table I and the product code used the rate 1/2 code—all used 200 bit packets.

REFERENCES

- [1] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Processing Lett.*, vol. 4, pp. 198–191, July 1997.
- [2] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
- [3] A. Said and W. A. Pearlman, "A new, fast, and efficient image code based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.
- [4] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [5] N. Seshadri and C.-E. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, pp. 313–323, Feb./Mar./Apr. 1994.
- [6] B. S. Srinivas, E. A. Riskin, R. Ladner, and M. Azizoglu, "Progressive image transmission on a channel with memory," in *Proc. Thirty-Third Annual Allerton Conf.*, 1995, pp. 265–274.
- [7] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley-Interscience, 1974.