Progressive Image Coding on Noisy Channels *

P. Greg Sherwood Kenneth Zeger

Dept. of ECE, University of California at San Diego, La Jolla, CA 92093-0407
{sherwood, zeger}@code.ucsd.edu

Abstract

Numerous sophisticated techniques have been developed over the last several decades to efficiently transmit images across noisy channels. Here, we cascade an existing image coder with carefully chosen error control coding, and thus produce a progressive image compression scheme whose performance on a noisy channel is significantly better than that of previously known image compression techniques. The main idea is to trade off the available transmission rate between source coding and channel coding in an efficient manner. This coding system is easy to implement and has acceptably low complexity. Furthermore, effectively no degradation due to channel noise can be detected; instead, the penalty paid due to channel noise is a reduction in source coding resolution. As an example, for the 512x512 Lena image, at an overall transmission rate of 1 bit per pixel, and for binary symmetric channels with bit error probabilities 10^{-3}, 10^{-2}, and 10^{-1}, the proposed system typically outperforms other existing systems by at least 2.6 dB, 2.8 dB, and 8.9 dB, respectively.

1 Introduction

Image compression has been extensively studied for the past several decades for both noiseless and noisy channels. However, considerably more progress has been achieved for the noiseless channel case.

One of the most successful and practical image coders today for the noiseless channel was originally developed by Shapiro [1] and later refined by Said and Pearlman [2]. Their image coding technique first takes a wavelet transform of the input image

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

and then codes the pixels in the wavelet domain in a clever manner using the tree-structured dependencies that result between the pixels in different subbands. In this manner a "progressive" mode of transmission is achieved, namely that as more bits are transmitted, better quality reconstructed images can be produced at the receiver. The receiver need not wait for all of the bits to arrive before decoding the image; in fact, the decoder can use each additional received bit to improve somewhat upon the previously reconstructed image.

These wavelet-based encoders have been shown to perform better than almost any other existing compression scheme. In addition, they have the nice features of being progressive and computationally simple. However, to obtain the high quality compression that they achieve, variable-length coding is used with significant amounts of "state" built into the coder. The result is that channel errors can cause a non-recoverable loss of synchronization between the encoder and decoder. Total collapse of the reconstructed image often results from loss of synchronization. In fact, a vast majority of images transmitted using this progressive wavelet algorithm will frequently collapse if even a single transmitted information bit is incorrectly decoded at the receiver.

As an example, consider transmitting a 512x512 image encoded at 1 bit per pixel across a binary symmetric channel with a bit error probability of 10^{-4}. On average any such image would have about 25 bit errors, nearly guaranteeing that every such transmitted image cannot be reconstructed due to synchronization loss. Even at a bit error probability of 10^{-5}, an average of 2.5 bit errors per image would occur, again assuring that a very large majority of the transmitted images could not be received reliably. In fact, many useful digital channels have bit error probabilities in the range 10^{-3} to 10^{-1}.

One approach to circumventing loss of synchronization on noisy channels is to use fixed rate image compression techniques, and those not based upon finite state algorithms (see for example [3, 4, 5, 6, 7, 8]). However, some of these techniques have the disadvantages of not being progressive, not performing as well for good quality channels, or having extremely high computational complexity. Other techniques designed for protecting transmission of medical images across noisy channels include [9]. In [8], some analytic justification is given for a joint source-channel coding design method. Two of the most competitive techniques for protecting images from channel noise are found in [10] and [11, 12].

Another approach to protecting image coders from channel noise is to divide the transmitted bit stream into two classes, the "important" bits and the "unimportant" bits, based upon the effects of channel errors on these bits. The important bits can then be sent as header information using good error control codes and the remaining bits can be sent with weaker channel codes. This type of technique was used in [13, 14].

A more traditional approach to protecting source coder information from the effects of a noisy channel is to cascade the source coder with a channel coder. This method can improve the overall quality of the system if the channel code rate is wisely chosen. Otherwise, some portion of the transmission rate is either under-utilized, in which case the source coding quality diminishes, or else too much of the transmission rate is devoted to source coding, in which case too many undetected channel errors are decoded. In each of these cases the resulting image quality suffers. Analytical results have recently been obtained in [15] as guidance in choosing the optimal tradeoff between source coding and channel coding. In fact, these results roughly follow those that we use in the present system.

In this paper we present a low-complexity technique that preserves the encoding power of the progressive wavelet schemes of Shapiro-Said-Pearlman, preserves the progressive transmission property, and is simple to implement in practice. We focus on very noisy channels, such as binary symmetric channels with bit error probabilities between 10^{-1} and 10^{-3}, typical of many wireless communication applications.

One nice feature of the proposed coding system is that its performance for a given image remains constant with probability near one over all possible received channel error patterns. Effectively no degradation due to channel noise can be detected. In fact, the effect of channel noise is to force the transmitter to encode the image at a lower source coding resolution and devote more bits to channel coding. Thus, on very noisy channels, the reconstructed image quality will be that of the noiseless channel encoder, but at a lower source coding rate.

The system does not have to be designed for any particular transmission rate, and in fact works quite well over a broad range of transmission rates. At very low transmission rates, such as around 0.2 bits/pixel, and on very poor channels such as those with bit error rates of 10^{-1}, transmitted images with this system are "just recognizable," whereas other coding systems would generally produce an unrecognizable decoded image. At these lower transmission rates, one use of the proposed system is to allow the user to obtain early recognition of an image under severe channel conditions, so that he/she can make many fast decisions to stop viewing certain images. This feature is useful in communication systems with multiple sensors and slow transmission rates.

2 System Description

One powerful method of error control coding is to use a concatenated code consisting of a Reed-Solomon outer code followed by a convolutional inner code (e.g. [16]). This approach, for example, was used in [13] for transmitting JPEG images across noisy channels. It was also used in [5, 6] for transmission across a Gaussian channel, although the implementation complexity appears extremely large in this case. The results in [5, 6] are also difficult to compare since it is not clear whether soft-decision decoding was used and some overhead bits seem to have been neglected in the overall bit rate computation. We adopt a related concatenated coding scheme with somewhat more flexibility and lower complexity to protect the output of the Said-Pearlman coder.

The main idea is to partition the output bit stream from the Said-Pearlman image coder into consecutive blocks of length N (typically we use $N = 200$ although its value is completely flexible). Then a collection of c checksum bits are derived based only on these N bits (we use $c = 16$). Finally m zero bits, where m is the memory size of the convolutional coder, are added to the end to flush the memory and terminate the decoding trellis at the zero state. The resulting block of $N + c + m$ bits is then passed through a rate r rate-compatible punctured convolutional (RCPC) coder [17].

The fact that the outer code is a cyclic redundancy code (CRC) used for error detection has the advantages of extremely low computational complexity and great flexibility in selecting block lengths (block lengths are unconstrained, in contrast to Reed-Solomon block lengths). The resulting bit stream is transmitted across a binary symmetric channel with bit error probability ϵ and then is decoded.

The decoder consists of a Viterbi decoder with the added feature that the "best path" chosen is the path with the lowest path metric that also satisfies the checksum equations. This additional constraint eliminates certain paths from consideration. In fact, whenever an undetected error would occur in the ordinary Viterbi decoder without the checksum bits, the correct path through the trellis is usually the one with the second lowest path metric. When the check bits indicate an error in the block, the decoder usually fixes it by finding the path with the next lowest metric. Systems of this type were analyzed in [18].

3 Algorithm Complexity

The complexity of the proposed channel decoder is quite reasonable, usually requiring little computation beyond that of the usual Viterbi decoding algorithm for the convolutional code. In addition, since punctured convolutional codes are used in this system, the trellis decoding is simplified because the trellis is that of a $1/N$ rate code for all the punctured rates. For a code of rate K/N this translates into a computation savings by a factor of $2^{K-1}/K$ per decoded bit.

In order to easily search for other likely trellis paths if necessary, some additional storage is required over the normal Viterbi algorithm. Usually only the survivor path and its metric must be stored for each state at the current trellis stage, as the decoding algorithm progresses. In this case, we store all paths and their metrics for each state and each trellis stage. The storage requirement amounts to 2 paths and metrics at each state and each stage of the trellis because the codes are punctured rate $1/N$ codes. For the packets of length 200 bits and a convolutional code with memory 6, the resulting memory requirement is about 40 Kbytes which is reasonable for an image application.

Each candidate trellis path is checked by computing a 16 bit CRC. The CRC polynomial was selected from those listed in [19] and [20] based on the number of information bits in a packet. For example, with 200 information bits in a packet the selected CRC polynomial was $X^{16} + X^{14} + X^{12} + X^{11} + X^{8} + X^{5} + X^{4} + X^{2} + 1$. Computing the check bits is very simple, only requiring bit shift and xor operations, and table lookups can be used to trade off some of the computation with space requirements. For a 200 bit path, the byte based algorithm requires about 50 xors, 25 byte shifts, and 25 lookup operations. Typically the CRC only needs to be computed for the first candidate, but in the $0.3 - 3.0\%$ (which means at most about 3-4 packets of a rate 1.0 bpp image) of cases that require further paths to be checked, the correct path is among the top 10 candidates $98+\%$ of the time. Figure 1 shows a typical distribution of path search depths given that the output of the Viterbi algorithm did not satisfy the CRC check.

The search for alternate candidate paths is accomplished using a tree-trellis search algorithm. The same algorithm is used in speech recognition with hidden Markov models [21]. First, the maximum number of candidates, M, that will be decoded is selected since this determines storage requirements of a stack which will contain the M most likely paths as the search runs. The depth of the stack is selected to provide a good compromise among decoding time, probability of locating the correct path, and the probability of undetected errors due to an incorrect candidate path satisfying the CRC checks. In general, after determining the n^{th} best path, the next best path is determined by searching backwards in the trellis along the n^{th} best path starting from the trellis stage where it first branches from a higher ranking path. The search amounts to comparing the metric of a new path, whose initial segment consists of the path eliminated at that stage in the Viterbi algorithm and merges with the path in question thereafter, to the metrics stored on the stack. Since the metric and path values are stored as the Viterbi algorithm runs, the amount of computation is small. If the metric is low enough, the path is placed on the stack at the appropriate location according to the metric value. After searching to the beginning of the n^{th} path, the $(n+1)^{st}$ will be stored at position $n+1$ on the stack. Also, we note that the search is terminated as soon as a path satisfies the CRC checks, so usually very few candidate paths need to be determined as indicated by the distribution in Figure 1.

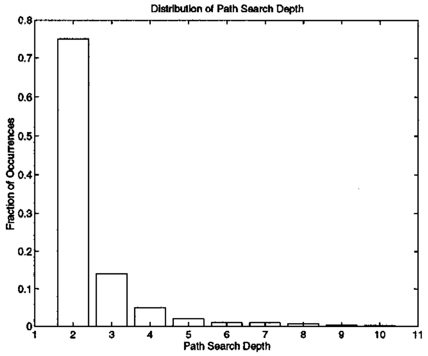

Figure 1: Distribution of path search depths given that the Viterbi decoder output fails the CRC check.

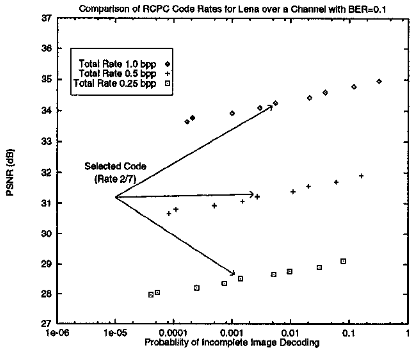

Figure 2: Tradeoff between the likelihood of incomplete decoding and image quality given complete decoding, for convolutional code rates from 1/4 to 1/3.

4 Experimental Results

The system was tested on two 512x512 images, the standard "Lena" and "Goldhill" images from the USC database. These particular test images were chosen to allow for some comparison with existing techniques. Each image was coded at bit error probabilities of $\epsilon = 10^{-1}$, $\epsilon = 10^{-2}$, and $\epsilon = 10^{-3}$, and at transmission rates ranging from 0 bpp up to 1 bpp, in increments of 0.05 bpp. All the RCPC codes used in testing were selected from tables in [17] or based on convolutional codes listed in [16]. In particular, a rate 2/7 memory 6 (punctured rate 1/4) code was used on the $\epsilon = 10^{-1}$ channel, a rate 2/3 memory 6 (punctured rate 1/3) code was used on the $\epsilon = 10^{-2}$ channel, and a rate 8/9 memory 6 (punctured rate 1/3) code was used on the $\epsilon = 10^{-3}$ channel. This scheme retains the progressive nature of the underlying source coder allowing almost a continuum of transmission rates. Each increment in the transmission rate can be as small as about 0.001 bpp which corresponds to the packet size we selected containing 200 source bits. Many of the previously reported image coding systems for noisy channels are not progressive.

For each image and each bit error probability, many thousands of independent trials were simulated on a computer for the various transmission rates which translates into millions of packets. In these tests, the path search depth was limited to 100 candidate paths, and if none of these 100 paths satisfied the CRC check, then decoding for that image was stopped at that packet giving incomplete decoding. The effect of incomplete decoding is often acceptable since the output is simply the image decoded at a lower source rate. Figure 2 shows the tradeoff between typical image PSNR and probability of incomplete image decoding for inner RCPC code rates of $\{1/4, 8/31, 4/15, 8/29, 2/7, 8/27, 4/13, 8/25, 1/3\}$. These results are for the Lena image coded at total (i.e. source plus channel coding) rates 1.0, 0.5, and 0.25 bpp using a channel with BER $= 10^{-1}$. For our purposes, the inner RCPC codes were selected so that the probability of incomplete image decoding was below 0.01 for the highest transmission rate of interest (1.0 bpp) for each channel BER. The highest incidence rate of incomplete image decoding in our tests was actually around 0.005. Higher acceptable incidence rates for incomplete decoding would allow higher typical PSNR values while more stringent incidence rates would lower typical PSNR values.

The curves in Figures 3-8 show the resulting PSNR of the cascaded source coding and channel coding system as a function of the overall transmission rate across the channel. Figures 3, 5, and 7 are for Lena, and Figures 4, 6, and 8 are for Goldhill. Each figure is for a fixed bit error probability ϵ on a binary symmetric channel. The other points in the plots represent the performances of the best known image codes for noisy channels, prior to the proposed system. These include those of Tanabe and Farvardin [10], Chen and Fischer [11], as well as those in [8, 22, 23]. Numerous other results exist in the literature, but all of them appear to be inferior to the results reported in [10, 11], or else do not provide results for the test images we considered.

It can be seen from the Lena graphs at a transmission rate of 1 bpp, that our proposed system improves the performance over the other reported SNRs by at least 2.6 dB, 2.8 dB, and 8.9 dB, for bit error probabilities of 10^{-3}, 10^{-2}, and 10^{-1}, respectively. At a transmission rate of 0.5 bpp, the gains for these three channels are 2.3 dB, 2.6 dB, and 6.4 dB, respectively. At a transmission rate of 0.25 bpp, the gains for these three channels are 2.2 dB, 1.9 dB, and 4.4 dB, respectively. For the Goldhill image at 1 bpp, the gains for these three channels are 2.2 dB, 2.3 dB, and 5.7 dB, respectively. At a transmission rate of 0.5 bpp, the gains for these three channels are

1.4 dB, 1.4 dB, and 4.1 dB, respectively. At a transmission rate of 0.25 bpp, the gains for these three channels are 1.3 dB, 1.1 dB, and 3.0 dB, respectively.

5 Acknowledgment

The authors wish to thank Javier Garcia-Frias and John Villasenor for providing a preprint of their DCC97 paper.

References

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[2] 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, pp. 243–250, June 1996.

[3] T. S. Huang *et al.*, "Design considerations in pcm transmission of low-resolution monochrome still pictures," *Proceedings of the IEEE*, vol. 55, pp. 331–335, Mar. 1967.

[4] J. W. Modestino and D. G. Daut, "Combined source-channel coding of images," *IEEE Transactions on Communications*, vol. 27, pp. 1644–1659, Nov. 1979.

[5] M. J. Ruf, "A high performance fixed rate compression scheme for still image transmission," in *Proceedings DCC '94. Data Compression Conference*, pp. 294–303, 1994.

[6] M. J. Ruf, "Source and channel coding of images using the operational rate-distortion function," in *International Symposium on Information Theory and its Applications (ISITA)*, (Victoria, Canada), pp. 751–754, Oct. 1996.

[7] M. Ruf and J. Modestino, "Rate-distortion performance for joint source and channel coding of images," in *Proc. IEEE Int. Conference on Image Processing*, vol. II, pp. 77–80, 1995.

[8] J. Garcia-Frias and J. D. Villasenor, "An analytical treatment of channel-induced distortion in entropy coded image subbands," in *1997 Data Compression Conference*, 1997.

[9] R.-Y. Wang, E. A. Riskin, and R. Ladner, "Codebook organization to enhance maximum a posteriori detection of progressive transmission of vector quantized images over noisy channels," *IEEE Transactions on Image Processing*, vol. 5, pp. 37–48, Jan. 1996.

[10] N. Tanabe and N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," *IEEE Journal on Selected Areas in Communications*, vol. 10, pp. 926–943, June 1992.

[11] Q. Chen and T. R. Fischer, "Robust quantization for image coding and noisy digital transmission," in *Proceedings DCC '96. Data Compression Conference*, pp. 3–12, 1996.

[12] Q. Chen and T. R. Fischer, "Image coding using robust quantization for noisy digital transmission," *Submitted to IEEE Transactions on Image Processing*, 1996.

[13] T. P. O'Rourke, R. L. Stevenson, Y.-F. Huang, and D. J. Costello, Jr., "Improved decoding of compressed images received over noisy channels," in *Proceedings ICIP-95*, pp. 65–68, 1995.

[14] D. W. Redmill and N. G. Kingsbury, "Still image coding for noisy channels," in *Proceedings ICIP-94*, pp. 95–99, 1994.

[15] B. Hochwald and K. Zeger, "Tradeoff between source and channel coding," *IEEE Trans. on Information Theory*, 1997 (to appear).

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

[17] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, pp. 389–400, Apr. 1988.

[18] N. Seshadri and C.-E. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Transactions on Communications*, vol. 42, pp. 313–323, February/March/April 1994.

[19] T. V. Ramabadran and S. S. Gaitonde, "A tutorial on CRC computations," *IEEE Micro*, vol. 8, pp. 62–75, Aug. 1988.

[20] G. Castagnoli, J. Ganz, and P. Graber, "Optimum cyclic redundancy-check codes with 16-bit redundancy," *IEEE Transactions on Communications*, vol. 38, pp. 111–114, Jan. 1990.

[21] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, New Jersey: Prentice-Hall, 1993.

[22] A. C. Hung and T. H.-Y. Meng, "Error resilient pyramid vector quantization for image compression," in *Proceedings ICIP-94*, pp. 583–586, 1994.

[23] Z. Mohdyusof and T. R. Fischer, "Subband image coding using a fixed-rate lattice vector quantizer," in *Proceedings ICIP-95*, pp. 101–104, 1995.

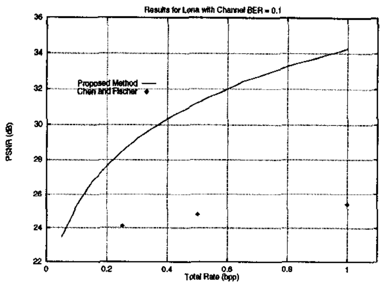

Figure 3: Results for 512x512 Lena over a BSC with BER = 0.1.

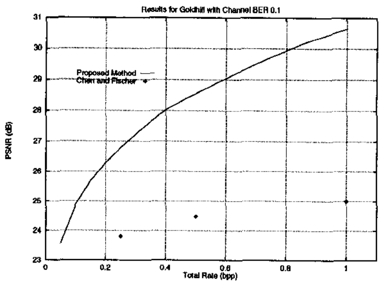

Figure 4: Results for 512x512 Goldhill over a BSC with BER = 0.1.

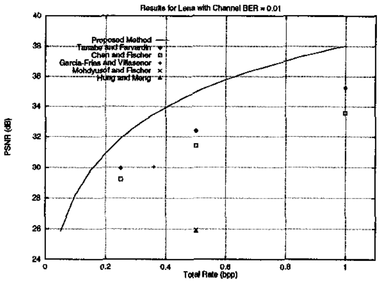

Figure 5: Results for 512x512 Lena over a BSC with BER = 0.01.

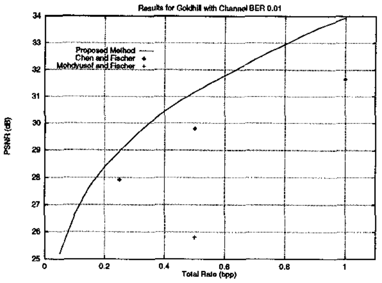

Figure 6: Results for 512x512 Goldhill over a BSC with BER = 0.01.

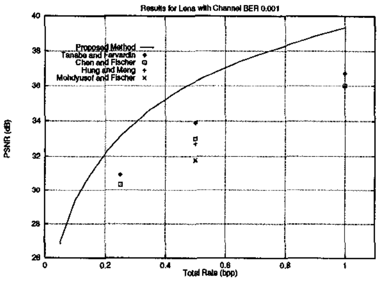

Figure 7: Results for 512x512 Lena over a BSC with BER = 0.001.

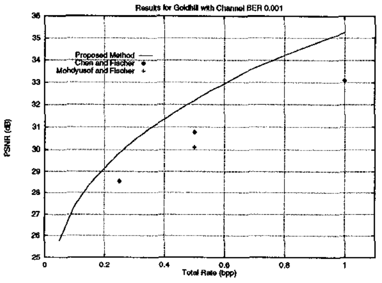

Figure 8: Results for 512x512 Goldhill over a BSC with BER = 0.001.