

Error Protection of Wavelet Coded Images Using Residual Source Redundancy *

P. Greg Sherwood and Kenneth Zeger
University of California San Diego
9500 Gilman Dr MC 0407
La Jolla, CA 92093
{sherwood,zeger}@code.ucsd.edu

Abstract

We investigate the residual redundancy in the Shapiro and Said-Pearlman wavelet-based image compression algorithms. The goal is to use redundancy inherent in the output of these source coders to provide joint source-channel decoding in order to improve error correction/detection performance for transmission across a noisy channel.

1. Introduction

Shannon's separation principle states that source and channel coding can be treated separately with no performance loss - at least with unlimited complexity and delay. However in practical systems where some redundancy remains at the output of the source coder, the receiver can exploit the redundancy to improve channel decoding performance. Practical source coding algorithms for voice/audio, images, and video do not operate perfectly, and the remaining redundancy at the output of the source coder, referred to as *residual redundancy*, can be exploited by joint source-channel decoding. This paper investigates residual redundancy for certain progressive image compression algorithms. Specifically, we report here on experimental observations and progress in our of on-going research to ultimately improve channel decoding using source information.

Shapiro's [6] embedded zerotree wavelet (EZW) image compression algorithm exploits the dependency that exists in typical images among spatially-related wavelet coefficients in different subbands. As bit planes of the image are encoded with the EZW algorithm, a "zerotree" symbol allows efficient coding of large groups of low-magnitude coefficients. While the EZW algorithm and the refined algorithm by Said and Pearlman [4] (SPIHT) achieve excellent compression results, they are highly susceptible to transmission errors due to the use of variable length coding and

a large amount of state information.

Even single-bit transmission errors often lead to the loss of synchronization and a total collapse of decoded image quality. Two examples demonstrating the effects of errors are shown in Figures 1 and 2. Figure 1 shows the effect of an error in a single "sign bit" of a wavelet coefficient with large magnitude. The bit error, in this example, only affects one wavelet coefficient. A second example, in Figure 2, demonstrates a case where the bit error results in loss of synchronization in the decoder. In both cases, the transmission error occurred in the first 100 transmitted bits, and the reconstructed image is shown at rate 0.25 bpp. In a typical image, those bits where errors usually lead to loss of synchronization make up a significant fraction of the total bits - often 70 to 80 percent of the bits for images compressed to rates of 0.5 bits per pixel and below. This observation is demonstrated in the stacked bar graph in Figure 3 for several bit planes of the "Lena" image. The visual effect of a bit error generally decreases as a function of how far into the bit stream the error occurs, due to the progressive nature of the coder.

The fact that bit errors so often result in extremely unnatural looking images suggests that decoding a corrupted bit stream (i.e. into EZW or SPIHT symbols) should result in parsed symbols that are very unlikely for natural images. *It is precisely the frequent occurrence of improbable symbol sequences resulting from bit-error corrupted image transmissions that we study, in an effort to use knowledge of source information to combat transmission errors.* In what follows, we investigate the residual redundancy in the output of a zerotree wavelet coder based on the SPIHT algorithm, together with the effects of uncorrected channel errors.

2. Joint Source-Channel Decoding

The idea behind joint source-channel decoding is to use the residual redundancy in a compressed source to assist

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



Figure 1. 512x512 Lena (at 0.25 bpp) with a single sign bit error in a large magnitude coefficient.

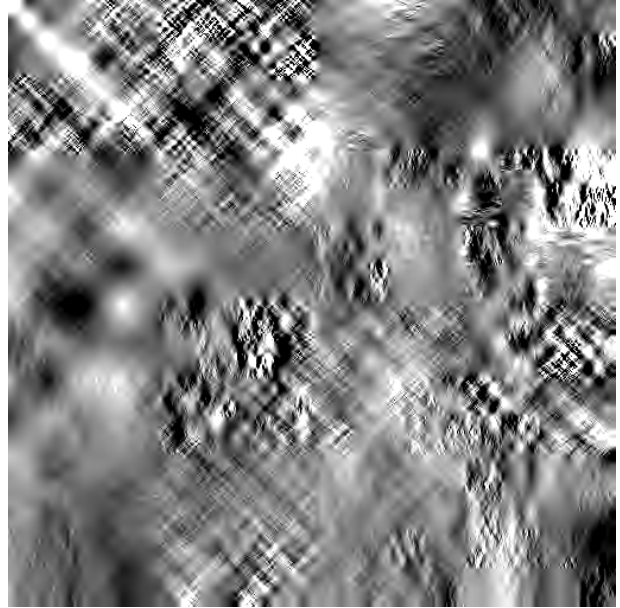


Figure 2. 512x512 Lena (at 0.25 bpp) with a single sorting bit error leading to error propagation.

in the channel decoding operation. If the source coder were ideal for a noiseless channel, then the compressed bit stream would consist of independent equally-likely bit values. In practice, the source coder is not able to perfectly compress the data, and often it is infeasible to add additional lossless compression at the encoder, such as arithmetic coding, due to complexity constraints or compliance with standards. Also, the results in [7] indicate that it may be better to use the residual redundancy at the receiver than remove it in a noisy environment. The residual redundancy in a source can be utilized by a channel decoder to correct bit errors that might otherwise go unnoticed. This concept was exploited, for example, in speech and image coding applications in [1, 3, 5, 7]. As noted in [3], when smoothing of the decoded output can be used to mitigate the effects of channel errors, there is residual redundancy which can be exploited by the channel decoder to directly improve channel decoding. An important problem for a given source and a given source coding algorithm is to characterize the nature of the residual redundancy.

3. Illegal Sequences

If a source coder has the property that not every possible bit stream is a potential output of the coder, then there is some inherent redundancy in the coder (e.g. Huffman cod-

ing does *not* have this property). In some cases it can prove more convenient to leave this redundancy in the coder because the extra transmission rate is very small and removing it can be difficult. Also the decoder can use knowledge about such illegal sequences to detect errors - that is, if the decoder parses an illegal sequence (one the encoding algorithm would never choose), then the decoder can be certain an error occurred.

The Said and Pearlman algorithm has some illegal sequences which can all be used for bit error detection. The illegal sequences are all related to the extension of the set of significant coefficients (i.e. those coefficients with magnitude at least as large as the current bit plane) through the extension of the tree of coefficients under consideration. The code syntax allows for the possibility of specifying that a significant coefficient lies in a given branch of the tree while declaring all such coefficients insignificant - clearly the encoder would never use such a sequence.

Specifically, the algorithm processes 2x2 groups of sibling wavelet coefficients (i.e. those coefficients with the same parent in the tree) as a unit when checking for new significant descendants. If *none* of these four coefficients had a significant descendant in an earlier bit plane then a single bit is used to indicate whether *any* of the four has a significant descendant in the current bit plane. In cases where a significant descendant is indicated, four bits specify which members of the 2x2 group have significant descendants. Only 15 of the possible 16 sequences of 4 bits

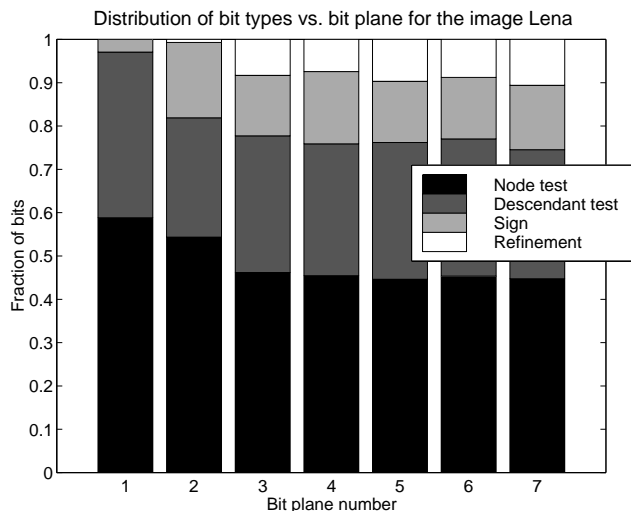


Figure 3. Bit type distribution vs. bit plane number for the 512x512 Lena image.

are allowed since at least one of the coefficients has a significant descendant.

The other illegal sequence is more dispersed in the bit stream. Immediately after declaring that a coefficient has significant descendants, the significance of the 2x2 group of its children is reported. Then the group of children is placed on the end of the current list of sibling groups so that it can be checked for significant descendants at a later time. An illegal condition occurs if a sibling group is added but none of its coefficients or descendants are reported to be significant. The bits leading to this condition tend to be dispersed in the bit stream with a spacing that is dependent on the number of sibling groups in the list.

These illegal conditions all occur during the descendant test phase of the SPIHT algorithm. Processing in the SPIHT algorithm is accomplished with a few sub-passes through the data so that all current insignificant nodes in the list are tested for significance, then all elements of the list are checked for significant descendants, and finally all significant coefficients are refined. This grouping of tests can lead to a large number of bits between an error and an error detection by an illegal condition, especially as the coding rate increases since the sub-passes become longer. Interleaving the tests reduces this problem since the descendant tests will be dispersed throughout the bit stream instead of in blocks. The impact of this re-ordering on the source coding performance is very small and may even improve it for some images at certain rates. The performance of the two will of course be equal at the point where each bit plane is complete.

In order to evaluate the error detection performance of illegal sequences, both the probability of detection and the

number of bits from the error to the point of detection were examined. The conditional cumulative distributions of the number of bits before detection (conditioned on the event that the error was detected) for several images is shown in Figure 4. These distributions show the fraction of errors detected within a certain number of bits for single bit errors. The performance is similar over a range of images with median detection delays on the order of 100-200 bits and mean detection delays on the order of 200-300 bits. The conditional distribution was considered because errors in sign and refinement bits do not lead to error propagation so cannot cause illegal sequences.

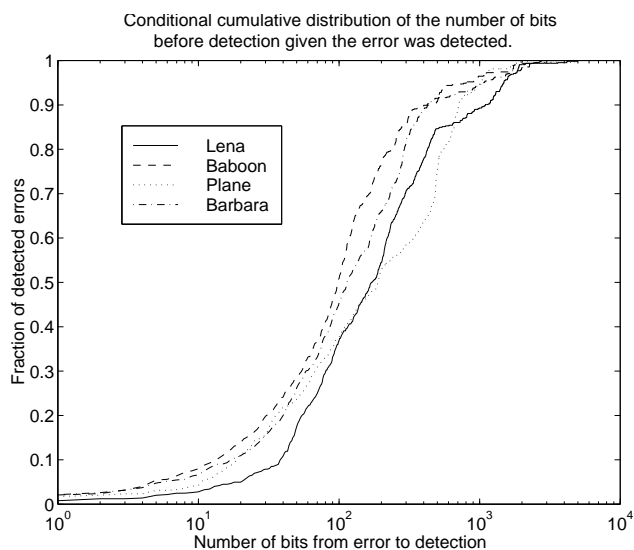


Figure 4. Detection performance for single bit errors.

An interesting question to consider is how the importance of an error (in terms of its effect on PSNR) is related to the detection delay. Those errors which go undetected or lead to long delays will hopefully also have less of an impact on decoded image quality. The scatter plot in Figure 5 shows the detection delay and decoded PSNR for many single-bit error trials. The points in the plot with detection delay equal to 66000 (i.e. those in the upper right corner of the plot) correspond to undetected errors since the entire compressed image contained fewer bits. While some of these undetected errors occurred in descendant or node test bits, it can be noted that all had a relatively small impact on image quality. The plot also shows that in general the errors with longer delays are the ones which result in less degradation of PSNR. This relationship is very different from conventional error detection using parity checks on the bit values where performance would be independent of decoded PSNR.

The performance was also tested with burst errors, which

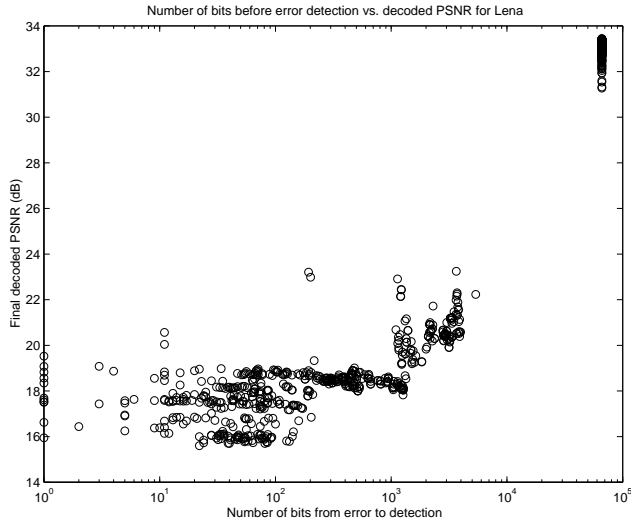


Figure 5. Detection performance for single bit errors.

are typical of the errors at the output of a decoder for convolutional codes (i.e. a Viterbi decoder). Bursts of duration 10 bits, with probability 0.5 that an individual bit is in error, were inserted in independent trials at many different locations in the bit stream. The error detection performance was very similar to the single bit error case.

4. Wavelet Coefficient Dependencies

In order to provide soft likelihood information to the channel decoder, the probability of any particular input sequence must be evaluated. Since the encoding operates on wavelet coefficients, it is natural for the decoder to utilize dependencies in the wavelet domain to evaluate the likelihood of a sequence. An important question is to determine what dependencies exist and how they affect the distribution of code sequences.

The zerotree method already takes advantage of some dependency among wavelet values by using the fact that if a coefficient is found to have magnitude below a given threshold, then all its descendants in the wavelet tree also tend to have magnitudes below this threshold. For a bit plane encoder like the EZW algorithm, this implies that parent nodes tend to become significant before descendants. In order to estimate likelihood information, it is important to estimate the distribution of the bit plane where a child becomes significant, given the parent value (or its quantized representation available at the decoder). Examining the joint distribution of parent and child log-magnitudes for several natural images reveals that there is a dependency for large values of the parent which is the typical region for low rate image

compression. Figure 6 shows a contour plot of the joint distribution of parent and child log-magnitudes for the Lena image. The same parent-child dependency was exploited in [2] for image compression.

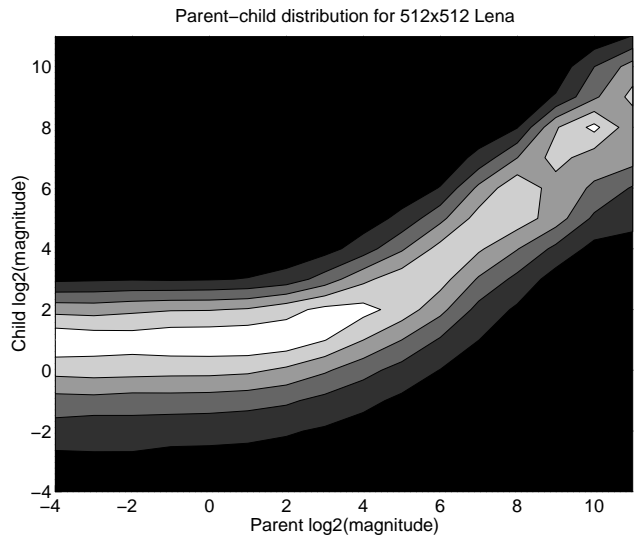


Figure 6. Joint distribution of parent-child log magnitudes.

The joint distribution is typically very similar over a large range of natural images. In the progressive decoder, the distribution will provide some information on when a child should become significant given the parent's decoded value.

Transmission errors tend to cause more coefficients to be declared significant in a single bit plane than is normal for a natural image. Also, the tree of wavelet coefficients under consideration tends to extend deeper in a single bit plane. In a natural image, new significant coefficients are usually children of current leaf nodes of the tree or perhaps grandchildren, but rarely any deeper descendants. After a loss of synchronization due to bit errors, the decoded symbols lead to unusual parent-child magnitude relationships. Therefore, a metric based on probability estimates from the parent-child joint distribution can allow errors to be recognized. Initial results are encouraging and more extensive testing is underway in order to draw more conclusions about the effectiveness of the metric.

5. Conclusion

We have identified illegal symbol sequences in the SPIHT zerotree coder which can be used for error detection. Detection performance was evaluated in terms of probability of detection and detection delay, since quick detection

can help reduce the number of candidate decoded sequences considered in the channel decoder. The results reveal detection delays on the order of 100-200 bits. Smaller delays occur for errors that lead to more severe image quality degradation. We have also proposed the use of the parent-child magnitude joint distribution to construct a soft likelihood metric.

On-going and future work includes evaluation of the performance of the parent-child distribution metric in distinguishing between noisy and natural decoded images, inclusion of other dependencies in the metric computation, integration of the metric with channel decoding algorithms, and evaluation of the performance of the joint source-channel decoder.

References

- [1] F. I. Alajaji, N. C. Phamdo, and T. E. Fuja. Channel codes that exploit the residual redundancy in celp-encoded speech. *IEEE Transactions on Speech and Audio Processing*, 4(5):325–336, Sept. 1996.
- [2] R. W. Buccigrossi and E. P. Simoncelli. Progressive wavelet image coding based on a conditional probability model. In *Proc. ICASSP 97*, pages 2957–2960, 1997.
- [3] J. Hagenauer. Source-controlled channel decoding. *IEEE Transactions on Communications*, 43(9):2449–2457, Sept. 1995.
- [4] 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*, 6(3):243–250, June 1996.
- [5] K. Sayood and J. C. Borkenhagen. Use of residual redundancy in the design of joint source/channel coders. *IEEE Transactions on Communications*, 39(6):838–846, June 1991.
- [6] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, Dec. 1993.
- [7] W. Xu, J. Hagenauer, and J. Hollman. Joint source-channel decoding using the residual redundancy in compressed images. In *Proceedings of ICC/SUPERCOMM '96. International Conference on Communications*, pages 142–148, 1996.