



Redmill, D. W., & Bull, D. R. (1996). Error resilient arithmetic coding of still images. In Unknown. (Vol. 2, pp. 109 - 112). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/ICIP.1996.560614

Link to published version (if available):
[10.1109/ICIP.1996.560614](https://doi.org/10.1109/ICIP.1996.560614)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>

Take down policy

Explore Bristol Research is a digital archive and the intention is that deposited content should not be removed. However, if you believe that this version of the work breaches copyright law please contact open-access@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline of the nature of the complaint

On receipt of your message the Open Access Team will immediately investigate your claim, make an initial judgement of the validity of the claim and, where appropriate, withdraw the item in question from public view.

ERROR RESILIENT ARITHMETIC CODING OF STILL IMAGES

David W. Redmill and David R. Bull

Image Communications Group, Centre for Communications Research,
University of Bristol, Bristol. BS8 1UB UK
Tel: [+44] 117 954 5195, Fax: [+44] 117 925 5265
Email: David.Redmill@Bristol.ac.uk Dave.Bull@Bristol.ac.uk

ABSTRACT

This paper examines the use of arithmetic coding in conjunction with the error resilient entropy code (EREC). The constraints on the coding model are discussed and simulation results are presented and compared to those obtained using Huffman coding. These results show that without the EREC, arithmetic coding is less resilient than Huffman coding, while with the EREC both systems yield comparable results.

1. INTRODUCTION

Recently [1, 2, 3] the error resilient entropy code (EREC) has been developed as an efficient and error resilient method for transmitting variable length (entropy coded) blocks of data, found in many image compression systems. Although, some theory [2, 3] exists to describe the error resilience of the EREC, it depends to some degree on the error resilience of the entropy coding method used to create the variable length blocks of data. Previous work has presented results based on schemes employing Huffman coding of blocks of sub-band coefficients. This was motivated by the adoption of this technique in recent standards including baseline JPEG [4] and H.261 [5]. An attractive alternative to Huffman coding is arithmetic coding [6]. Although arithmetic coding can be more complicated, it offers advantages of improved compression efficiency and the potential for adaptively varying the conditioning probabilities. However, arithmetic coding is intrinsically less error resilient, and is much less likely to resynchronize in the same way that many Huffman codes do [7].

2. EREC

Many image and video compression algorithms [4, 5] use a block-based coding structure in which the data is coded as a number of variable length code-blocks. For transmission, these variable length data blocks must

be multiplexed together to form a single compressed signal. This is usually achieved by joining the blocks sequentially.

It should be noted that, although the coding and multiplexing processes can be separated in the encoder, they need to work together in the decoder, since the information needed to separate the blocks is typically contained within the coded data. For example, with the baseline JPEG system, the decoder needs to decode the Huffman coded data to find end-of-block codewords.

The problem with sequential transmission systems, is that channel errors can cause the entropy coder to fail to decode the block boundary information, thus causing errors to propagate to following blocks. Although this propagation can be limited with synchronization codewords [8], they must be used relatively infrequently, to avoid excessive loss of compression.

Error correction coding can also be used at the expense of a reduction of compression performance.

In [1, 2, 3], it was shown that the EREC allows the variable length blocks to be transmitted in a more resilient structure, without any significant loss of compression. The EREC operates by placing the variable length blocks into a fixed length slotted structure. In this way the decoder can independently find the start of each block. The result is that data coded near the beginning of each block is immune to error propagation effects, while data coded near the ends of long blocks suffers from more error propagation. For image and video coding, this implies that the error propagation effects are localized to the higher frequencies around active regions of the image, which are subjectively less important.

Experiments with the EREC in conjunction with Huffman coding [1, 2, 3], showed that the residual channel errors cause the presence of some annoying high frequency artifacts (e.g. figure 6). These are caused by the Huffman decoder falsely producing large high frequency coefficients. Some of these errors can be detected by thresholding (based on statistical limits), and

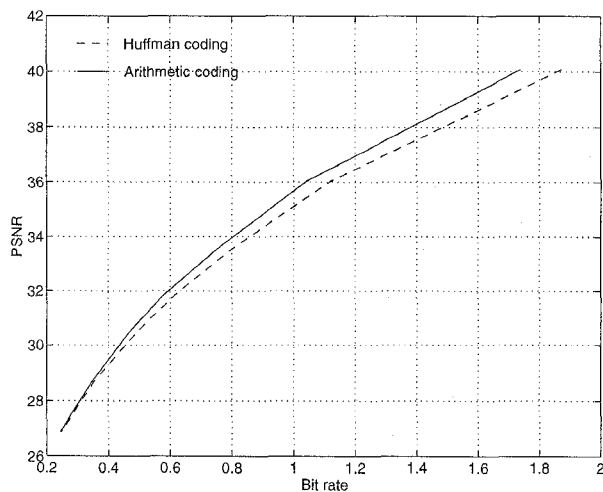


Figure 1: Compression performance (without errors).

concealed by setting them to zero [1, 2].

3. ARITHMETIC CODING

Arithmetic coding provides a form of entropy coding which is both efficient and allows adaptive variation of the symbol conditioning probabilities. These conditioning probabilities represent an estimate of the probability distribution of the value of each symbol. More accurate estimation of the conditioning probabilities, reduces the entropy of the symbol and thereby allows better compression. For correct decoding, the decoder needs have the same conditioning probabilities, which implies that they can be defined as some causal function of the previously coded/decoded data. In this paper, we use a binary arithmetic code [6] and a coding model similar to that of JPEG [4].

4. USE OF ARITHMETIC CODING AND THE EREC FOR IMAGE CODING

In order to take full advantage of the EREC, it is important to code the data as separate code-blocks with the more important features located towards the start of blocks. This can be achieved if image data (represented in sub-band domain) is split into spatial regions (blocks) and scanned from low to high frequencies. An example of this method is the DCT transform with zig-zag scanning as used by JPEG. The coefficients can then be coded sequentially using a separate arithmetic code for each block. It is noted that using a separate arithmetic code for each block introduces a slight redundancy (less than 1 bit per block or $\frac{1}{64}$ bits per pixel), in order to terminate the arithmetic code for that block. The need to use a spatial block-based system precludes the use of a system which codes data in a sub-band by

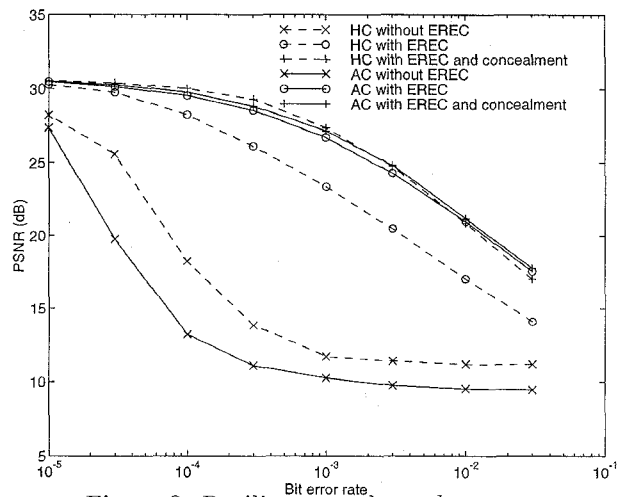


Figure 2: Resilience to channel errors.

sub-band manner.

Due to the parallel nature of the EREC decoding algorithm, it is important that (in the absence of channel errors), the EREC can find the end of each block, independently of data coded in other blocks. This implies that the entropy code used for each block must be independent of other blocks. Inter-block correlations can still be exploited, but only prior to the entropy coding. For an adaptive arithmetic code, this implies that, the conditioning probabilities can only be based on preceding data from the same block. Note that with a sequential system, the codec can also use any data from previous blocks.

5. RESULTS

In order to examine the possible use of arithmetic coding in conjunction with the EREC, we have modified the DCT image codec used in [1, 2] to include arithmetic coding as an alternative to Huffman coding, leaving the DCT and zig-zag scanning processes unchanged. Each coefficient is coded as a series of binary decisions to specify; whether the coefficient is non-zero, its sign, its magnitude and its least significant bits. The conditioning model we have used is a fixed, non-adaptive model, with conditioning probabilities trained by observation of a set of 6 training images (not including the test image 'Lenna').

Figure 1 shows a comparison of the compression performance. For identical (quantized) data, the arithmetic codec was found to offer an improvement in compression of about 5% compared to Huffman coding.

Figure 2 shows a comparison of the error resilience for systems with and without the EREC. Error concealment [1, 2] has also been used to give further refinements.

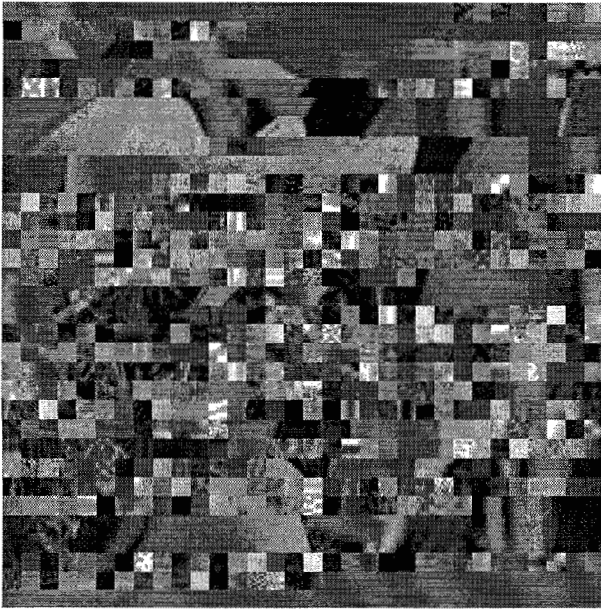


Figure 3: Arithmetic coding without the EREC.

Without the EREC, the arithmetic coding scheme was found to suffer much more from errors than the Huffman coded system. Figures 3 and 4 show typical results at 0.1% random bit error rate (BER). The results demonstrate the inferior resilience to channel errors of arithmetic coding compared to Huffman coding. This difference is mainly due to the resynchronization properties of Huffman coding [7].

With the EREC, but without error concealment, the arithmetic coding scheme performs significantly better than the Huffman coding system. Figures 5 and 6 show typical results at 0.1% BER. Most of the difference in performance is due to the presence of statistically abnormal high frequency artifacts within the Huffman coding system.

The further addition of error concealment, yields little improvement for the arithmetic coded system but improves the Huffman coded system to give comparable results. Figures 8 and 7 show typical results at 0.1% BER.

These results can be explained by assuming that after a channel error, the arithmetic decoder will decode symbols randomly according to their respective conditioning probabilities. This implies that it is improbable that the decoder will accidentally produce improbable coefficients. It was the presence of highly improbable coefficients, within the Huffman coding system, that led to the use of the error concealment.

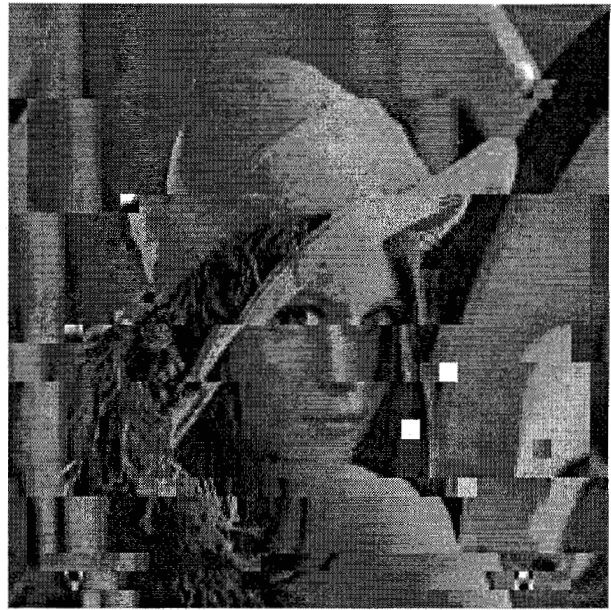


Figure 4: Huffman coding without the EREC.

6. CONCLUSIONS

In this paper we have examined the use of arithmetic coding in conjunction with the EREC, to design an error resilient image codec. In order to take full advantage of the EREC, the arithmetic codec is constrained to be block-based with conditioning probabilities derived from previous coded coefficients in the same block. Experimental results show that without the EREC, Huffman coding is significantly more resilient than arithmetic coding. However, with the EREC and error concealment, both systems give very similar performance in terms of error resilience.

7. REFERENCES

- [1] D. W. Redmill and N. G. Kingsbury. Still image coding for noisy channels. In *IEEE International conference on image processing*, volume 1, pages 1:95–99, 1994 Austin, Texas.
- [2] D. W. Redmill. *Image and video coding for noisy channels*. PhD thesis, Cambridge University Engineering Department., 1994.
- [3] D. W. Redmill and N. G. Kingsbury. The EREC: an error resilient technique for coding variable length blocks of data. *IEEE Transactions on Image Processing*, 5:565–574, 1996.
- [4] *JPEG Digital Compression and Coding of Continuous-Tone Still Images*. International Organization for Standardization ISO 10918, 1991.



Figure 5: Arithmetic coding with the EREC.



Figure 7: Arithmetic coding with error concealment.



Figure 6: Huffman coding with the EREC.



Figure 8: Huffman coding with error concealment.

- [5] *Video Codec for Audiovisual Services at p x 64 kbit/s*. CCITT recommendation H.261, COM XV-R 37-E, 1990.
- [6] G. G. Langdon and J. Rissanen. Compression of black-white images with arithmetic coding. *IEEE Transactions on Communications*, 29:858–867, 1981.
- [7] T. J. Ferguson and J. H. Rabinowitz. Self-synchronizing Huffman codes. *IEEE Transactions on Information Theory*, 30:687–693, 1984.
- [8] W. M. Lam and A. R. Reibman. Self-synchronization variable-length codes for image transmission. In *International Conference on Acoustic Speech and Signal Processing*, pages III-477 – III-480, 1992.