ERROR-RESILIENT CODING AND FORWARD ERROR CORRECTION FOR IMAGE TRANSMISSION OVER UNRELIABLE CHANNELS

Nikolaos V. Boulgouris, Nikolaos Thomos, and Michael G. Strintzis

Information Processing Laboratory
Electrical and Computer Engineering Department
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece

Informatics and Telematics Institute
1st Km Thermi-Panorama Rd,
Thessaloniki 57001, Greece

ABSTRACT

An error-resilient coding scheme is proposed for the transmission of images over unreliable channels. Forward Error Correction is used in conjunction with the error-resilient source coder for the protection of the compressed stream. Unlike almost all other robust coding schemes presented to-date, the proposed scheme is able to decode portions of the bitstream even after the occurrence of uncorrectable errors. The resulting coder will be shown to be very efficient for image transmission over noisy channels.

1 INTRODUCTION

Many new techniques have been recently proposed for the efficient coding of images and video. However, the transmission of the pictorial information over today's heterogeneous, and often unreliable, networks has necessitated the provision of protection methods against possible channel failures. Although, in theory, source and channel coding can be studied independently (Shannon's separation principle, channel coding strategies which take into consideration the structure of the underlying source coder produce significantly better performance.

A variety of coders based on error correcting codes have been proposed in the literature. Sherwood and Zeger [1] divide the bitstream output by the popular SPIHT coder [2] into blocks of constant length. Each packet is protected by a concatenated Rate Compatible Punctured Convolutional code and Cyclic Redundancy Check code (RCPC/CRC).

Man et. al. [3] introduce two methods for coding the location information of significant subband coefficients. The output bitstream is protected by applying RCPC channel codes. Recently, Chande and Farvardin [4] proposed a bit allocation algorithm for application with embedded coders and applied their scheme with the SPIHT source coder.

In all aforementioned algorithms, decoding of the received robust streams stops at the first uncorrectable error. This has the obvious drawback of losing a potentially high portion of the bitstream (i.e. all bits following the first uncorrectable error). This situation deteriorates dramatically with noisier channels, since then the first uncorrectable error may occur very early in the stream.

In this paper, we use an error-resilient source coder which is very suitable for use in joint source/channel coding systems. It is based on the partitioning of information into a number of layers which can be decoded independently provided that some very important and highly protected information is initially errorlessly transmitted to the decoder. The independent bitstreams are subsequently protected using equal or unequal amounts of protection. Forward Error Correction (FEC) based on Rate-Compatible Punctured Convolutional (RCPC) codes is used. This coding approach allows the decoding of the bitstream even after the occurrence of uncorrectable errors, and thus differentiates our scheme from other zerotree-based or block-based robust coders seen so far in the literature.

The organization of the paper is as follows: In Section 2 the proposed error-resilient framework is described. The efficient detection and handling of errors not corrected by the channel code is discussed in Section 3. In section 4 a bit allocation algorithm is presented. Experimental evaluation is presented in Section 5 and finally conclusions are drawn in Section 6.

2 PROTECTION OF COMPRESSED STREAMS

An error-resilient wavelet coder is used for the compression of images. Each image is wavelet transformed, and the wavelet representation is divided into blocks. The bitplanes of all blocks in a subband constitute layers that are individually encoded using the context arithmetic models of [5]. The layers are transmitted from the Most Significant Bitplane to the Less Significant Bitplane in a predefined scan order. Bitplane coding takes place using two processes, namely *significance identification* and *refinement coding*[5]. The resulting coder is termed PSWIC (Predefined scan order Scalable Wavelet Image Coding)

The layers produced as described above are protected using channel coding [6]. Since each bitplane of a block is coded without using information from other blocks, protection can be individualy applied to each such block. A schematic description of the system used for the generation of robust streams is shown in Fig. 1.

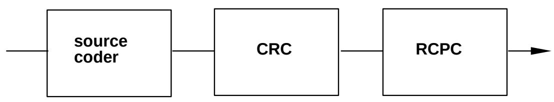

Figure 1: Cascade of operations for the efficient protection of layers.

Specifically, header information is considered very important and is highly protected. Layers $\mathcal{L}^s_{nk}$ and $\mathcal{L}^r_{nk}$ (n,k are the bitplane, block indices) corresponding to *significance* identification and *refinement* coding respectively are also channel coded. The basic structure for adding protection is depicted in Fig. 3. Each layer is independently protected by employing a field in its header which states the size of the source bits used for the coding of that layer. Another field in the header specifies the matrix with which the RCPC codes are punctured [6]. This is very useful in cases where an entire layer has to be discarded (due to uncorrectable errors) since the length of the source+channel rate of the layer can be deduced at the decoder side and, thus, the corrupted layer can be discarded without preventing subsequent layers (that do not depend on the discarded layer) from being decoded correctly (see Fig.2).

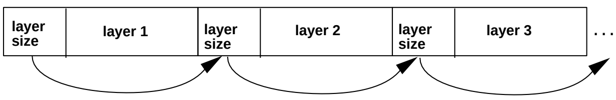

Figure 2: Bitstream structure. Each layer includes a highly protected header indicating the size of the layer. If an uncorrected error occurs in a layer, the corrupted layer can be discarded and the decoding process can proceed with the next uncorrupted layer.

For the efficient protection of layers, each layer $\mathcal{L}$ is partitioned into $N_p(\mathcal{L})$ packets of equal size (apart form the last packet which may be shorter) and protected using the coder shown in Fig 1. The resulting layer structure is shown in Fig. 3. Note that the (non-constant) size of the last packet in a layer can be implicitely calculated from the size of the layer and the puncturing matrix identifier (which are stored in the layer header). Thus, no other side information is needed for its coding and decoding.

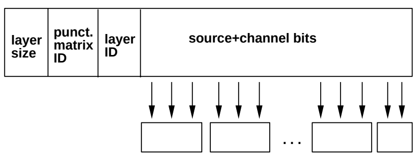

Figure 3: Organization of information in a robust layer.

3 ERROR DETECTION AND HANDLING

A significant feature of a robust coder is its ability to detect and confine errors not corrected by the channel code. Zerotree-based coders are not suitable for error-resilient image transmission since the occurence of a single erroneous bit renders the rest of the bitstream undecodable. In such coders, if an error is not detected, then the quality of the reconstructed image will be totally unacceptable. In our coder, due to the bitstream generation and organization strategy followed, errors not corrected by the channel code, affect usually *only* the packet in which the error occured and occasionally a few subsequent packets.

For the detection of errors, Cyclic Redunduncy Codes (CRC) are employed in conjunction with RCPC codes [6]. For the efficient correction of errors, the serial list Viterbi algorithm [7] was used with a list of 100 paths. When the list-Viterbi algorithm is used, the optimal path in the Viterbi decoding is chosen among those paths that follow the constraints imposed by the CRC.

The detection of an uncorrected error during decoding stimulates the following actions.

- If the error is in layer $\mathcal{L}^s_{nk}$, then this layer is retained up to the first corrupted packet and all subsequent layers $\mathcal{L}^s_{jm}$, $\mathcal{L}^r_{jm}$, $j < n$, $k = m$ for the same block are discarded since the information they contain can not be exploited. This process is illustrated in Fig. 4.
- If the error is in $\mathcal{L}^r_{nk}$, then this layer is retained up to the first corrupted packet. The rest of the packets comprising the layer are discarded, but all subsequent layers $\mathcal{L}^r_{jk}$, $j < n$ are retained (provided that no uncorrectable error occur in those layers) since such errors are localized and do not affect the decoding of subsequent layers.

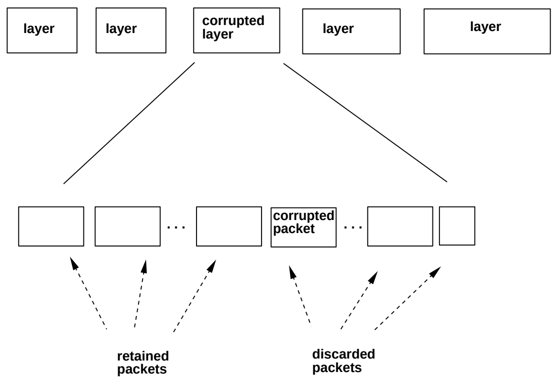

Figure 4: Packet disposal as performed by the proposed coder in case of uncorrectable errors.

The ability of our robust coding methodology to discard corrupted portions of the bitstream in order to confine errors and achieve the best possible reconstruction quality endows the proposed scheme with the capability of achieving superior performance. This will be shown in the experimental results section where a family of robust coders, built using the techniques described so far, are evaluated. The allocation of protection to the source stream is examined in the ensuing section.

4 UNEQUAL ERROR PROTECTION

In order to allocate bits between source and channel we first note that each additional portion of the bitstream that is made available to the decoder reduces the distortion between the original and the reconstructed image. Thus, the problem can be described as that of maximization of the distortion decrease D achieved when bitplanes from $Q(k)$ to $N(k)$ for block k are transmitted

$$D = \sum_{k=1}^{M} D_k = \sum_{k=1}^{M} \sum_{i=Q(k)}^{N(k)} \left(D^s(i,k) + D^r(i,k)\right) \qquad (1)$$

where D_k is the distortion decrease for block k, M is the number of blocks of wavelet coefficients (some blocks may be as large as an entire subband), $N(k)$ is the number of non-zero bitplanes in the kth block and $D^s(n,k)$, $D^r(n,k)$ are the distortion reductions achieved by the transmission of bitplanes $n, \ldots, N(k)$ of the kth block for significance and refinement layers respectively. Finally, $Q(k)$ is the bitplane at which transmission stops for each block k. The average distortion decrease caused by significance layers for the kth block is:

$$D^s(n,k) = (1 - P(n,k)) \sum_{m=n}^{N(k)} D^s_{mk} \quad (2)$$

where D^s_{nk} denotes the individual distortion decrease caused by layers $\mathcal{L}^s_{nk}$ and $1 - P(n,k)$ denotes the probability that *only* layers L^s_{mk}, $m = n, \ldots, N(k)$ are correctly decoded. Since the decoding of a layer is possible only if *all* previous (more significant) layers have been decoded correctly, this probability is equal to

$$1 - P(n,k) = P^s_{n-1}(r^s_{n-1,k}) \prod_{m=n}^{N(k)} (1 - P^s_{mk}(r^s_{mk})) \quad (3)$$

where $P^s_{nk}(r^s_{nk})$ is the *individual* probability that a significant layer is not decoded correctly (i.e. supposing all layers it depends on are correctly decoded) when r^s_{nk} is the channel code rate used for its coding. Similarly, the distortion decrease caused by refinement layers is

$$D^r(n,k) = P^s_{n-1}(r^s_{n-1,k}) \prod_{m=n}^{N(k)} (1 - P^s_{mk}(r^s_{mk})) \cdot$$

$$\cdot \left(\sum_{m=n}^{N(k)} (1 - P^r_{nk}(r^r_{nk})) D^r_{mk} \right) \quad (4)$$

where D^r_{nk} now denotes the individual distortion decrease caused by layer L^r_{nk}.

Each layer $\mathcal{L}_{nk}$ is divided into $N_p(\mathcal{L}_{nk})$ constant-length packets and each packet is individually protected. The probability that a layer is discarded is equal to the probability that at least one packet in this layer is plagued by uncorrectable errors. If p is the probability that a packet is corrupted, then the probability of l corrupted packets among the $N_p(\mathcal{L}_{nk})$ packets that comprise a layer $\mathcal{L}_{nk}$ coded using channel code rate r_{nk} is

$$P(l, N_p(\mathcal{L}_{nk}), r_{nk}) = \binom{N_p(\mathcal{L}_{nk})}{l} p^l (1-p)^{N_p(\mathcal{L}_{nk}) - l} \quad (5)$$

and, therefore, the probability of a layer error (of the existence of at least one packet in the layer in error) is given by the expression

$$P_{nk}(r_{nk}) = \sum_{l=1}^{N_p(\mathcal{L}_{nk})} P(l, N_p(\mathcal{L}_{nk}), r_{nk}) \quad (6)$$

Since the probability of an uncorrectable packet depends on the RCPC code used, this probability is experimentally evaluated for the set of channel codes used.

As seen from (3), refinement layers $\mathcal{L}^r_{nk}$ depend only on previous significant layers $\mathcal{L}^s_{jk}$, $j \geq n$ in the same block. Essentially, (3) expresses the probability that a layer $\mathcal{L}_{nk}$ is not decoded due to errors in previous layers $\mathcal{L}_{jk}$, $n < j \leq N(k)$. Using (2), (3) and (4), (1) becomes

$$D = \sum_{k=1}^{M} \sum_{i=Q(k)}^{N(k)} P^s_{(i-1)k}(r^s_{nk}) \cdot$$

$$\cdot \left(\sum_{n=i}^{N(k)} (D^s_{nk} + (1 - P^r_{nk}(r^r_{nk})) D^r_{nk}) \right) \prod_{n=i}^{N(k)} (1 - P^s_{nk}(r^s_{nk})) \quad (7)$$

where $P^s_{Q(k)-1,k}(r_{nk}) = 1$ and $D^r_{N(k),k} = 0$. The optimization problem then becomes that of maximizing the distortion decrease D given by (7) subject to a channel rate constraint R.

Provided the channel conditions are known, the error probability P_{nk} can be easily calculated for each layer, and optimal selection of the code rates r_{nk} is possible using exhaustive search or dynamic programming techniques. In practice, the allocation of channel bits to source layers is greatly facilitated by the fact that it can be applied on a blockwise basis since the contribution of each block in the quality of the reconstruction image is independent.

After computing r_{nk} for each layer, then the corresponding RCPC code can be applied. The channel bit allocation proceeds for all subsequent blocks and the corresponding allocations can be determined. Since in practice only a limited number of possible code rates is available, the solution is not really optimal. However, in most cases the available code rates are sufficient for achieving high-performance transmission.

5 EXPERIMENTAL EVALUATION

The proposed coders were experimentally evaluated for image transmission over Binary Symmetric Channels (BSCs). The 512×512 "Lenna" image was used in the simulations. Comparison was based on the average PSNR of the reconstructed image for two channel conditions. Specifically, two BSCs were simulated with $BER = 0.01$ and $BER = 0.001$ respectively.

The CRC codes used were taken from [8]. The family of RCPC codes that was used is based on a rate 1/4, memory 6 mother code.

The output of the encoder was punctured (i.e. certain code bits were not transmitted) using the puncturing matrices determined by the allocation process of the previous section. The puncturing matrices change the code rate and hence the correction power of the code according to source and channel needs. Eight puncturing matrices were employed with rates {16/17, 8/9, 16/19, 8/10, 16/21, 8/11, 16/23, 8/12}. In most practical applications, for $BER \leq 10^{-2}$, puncturing with the above matrices is sufficient. Extending the set of available matrices would yield vanishingly negligible gain since the more appropriate protection would be outbalanced by the increase in the cost for the transmission of matrix indices.

The algorithms compared to the present PSWIC coder were those by Sherwood [1], Man [3] and Chande [4]. The results are reported in Tables 1 and 2. Ten thousand PSNR values were averaged for calculating the entries in the tables. As seen, for low BERs (≈ 0.001) the performance of all coders appear to be equivalent. For higher BERs (≈ 0.01) the performance of the coder proposed here is clearly superior to that in [1] and competitive with that in [4]. Reconstructed images for various channel BERs and rates are shown in Fig. 5.

	PSWIC		Sherwood	Chande
	EEP	UEP	EEP	UEP
0.25	32.16	32.28	31.91	**32.30**
0.5	35.25	**35.37**	34.96	35.28
1.00	38.31	**38.36**	38.03	38.28

Table 1: Comparison of the proposed coding scheme for the transmission of images over BSC with BER=0.01. Equal and Unequal Error Protection was used with the proposed schemes.

	PSWIC UEP	Sherwood EEP	Man UEP
0.25	33.10	**33.16**	31.98
0.5	**36.26**	36.25	35.08
1.00	**39.43**	39.34	N/A

Table 2: Comparison of the proposed coding schemes for the transmission of images over BSC with BER=0.001.

Since our source coder perform approximately as well as (and often a little worse than) the SPIHT coder, our superior overall coding results can be primarily attributed to the organization of the bitstream in such a way that enables error localization and decoding beyond the point of an uncorrectable error. This feature alone makes the EEP-based version of our coder perform better than state-of-the-art coders based on unequal error protection. Additionally, the careful allocation of protection among layers makes the UEP variant of the proposed scheme even more efficient.

6 CONCLUSIONS

Novel joint source/channel coding schemes were proposed for the transmission of images over noisy channels. The proposed schemes are based on a source coder which outputs a stream very suitable for robust transmission. Channel coding is applied on the layers of the source bitstream according to their importance. A blockwise optimization algorithm for the efficient unequal error protection of the embedded stream was also proposed.

7 ACKNOWLEDGEMENT

This work was supported by the EU IST projects "OTELO" and "ASPIS".

(a) (b) (c) (d)

Figure 5: Reconstructed "Lenna" when transmitted over noisy channels using the PSWIC algorithm (a) 0.25 bpp, BER=0.001 (33.10 dB). (b) 0.5 bpp, BER=0.001 (36.26 dB). (c) 0.25 bpp, BER=0.01 (32.28 dB). (d) 0.5 bpp, BER=0.01 (35.37 dB).

References

[1] G. Sherwood and K. Zeger, "Progressive Image Coding on noisy channels," *IEEE Signal Processing Letters*, vol. 4, no. 7, pp. 189–191, July 1997.

[2] A. Said and W. A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[3] H. Man, F. Kossentini, and M. J. Smith, "A family of efficient and channel error resilient wavelet/subband image coders," *IEEE Trans. Circuits and Systems for Video Technology*, February 1999.

[4] V. Chande and N. Farvardin, "Progressive Transmission of Images over Memoryless Noisy Channels," *IEEE Journal on Selected Areas in Communications*, June 2000.

[5] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.

[6] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications," *IEEE Trans. Communications*, vol. 36, no. 4, pp. 389–400, Apr. 1989.

[7] N. Seshadri and C-E. Sundberg, "List Viterbi decoding algorithm with applications," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 2/3/4, pp. 313–323, Feb./March./April 1994.

[8] G. Castagnoli, J. Ganz, and P. Graber, "Optimum Cyclic Redundancy-Check Codes with 16-Bit Redundancy," *IEEE Trans. Communications*, vol. 38, no. 1, pp. 111–114, January 1990.