Submission to Image Communication Special Issue on Mobile Image / Video Transmission

Bandwidth Efficient Transmission of MPEG-II Video over Noisy Mobile Links

Robert Swann

Cambridge Consultants Ltd,
Science Park, Milton Road Cambridge CB4 4DW, U.K.

Email: `rswann@iee.org`

Nick Kingsbury

Signal Processing Group, Cambridge University Engineering Department,
Trumpington Street, Cambridge CB2 1PZ, U.K.

Email: `ngk@eng.cam.ac.uk`

Abstract

We consider the performance of MPEG-II compressed video when transmitted over noisy channels. We present the results of bit sensitivity and resynchronisation sensitivity measurements, and propose techniques for substantially improving the resilience of MPEG-II to transmission errors without the addition of any extra redundancy into the bitstream. We find that it is errors in variable length encoded data which cause the greatest artifacts as errors in these data can cause loss of bitstream synchronisation.

We develop the concept of a 'black box transcoder' where we losslessly transcode MPEG-II into a different structure for transmission. We achieve bitstream-resynchronisation using a technique known as error-resilient entropy coding (EREC). Finally we improve the error-resilience of differentially coded information by replacing the standard 1D-DPCM with a more resilient hierarchical pyramid predictor.

We consider the transmission of MPEG-II over three separate channels: a channel subject to random bit errors, a channel subject to burst errors, and a channel subject to ATM cell losses.

1 INTRODUCTION

This paper develops concepts presented in [1, 2, 3], which consider channels subject to random bit errors. Here we extend this work, and consider the resilient transmission of MPEG-II video over channels subject to burst errors and packet errors.

The massive growth in telecommunications during the last few years has created a demand for many video services including mobile multimedia communications. These services are only viable with substantial video compression. Whereas existing compressed video standards such as MPEG-II[4] facilitate the transmission of high quality video over error-free channels, they tend to suffer greatly from error propagation effects when transmitted over noisy channels. Only one or two bit errors can cause a large area of the picture to be corrupted. It is a property of most compressed video systems that they fail abruptly without warning. The aim of this research is to modify a digital system (MPEG-II) so it behaves more like an analogue system, where performance degrades gracefully with increasing channel noise.

Standard techniques for improving the MPEG-II noise performance [5, 6, 7] concentrate on the addition of forward error correction (FEC), or the introduction of frequent synchronising codewords. Although these methods require low complexity hardware, they add redundancy thereby lowering the coding efficiency, thus the error-free picture quality is lower for a fixed bitrate channel. FEC also causes abrupt failure as the error-rate increases. This is because FEC can add more errors to a bitstream once the correcting capability of the code has been exceeded. Instead we consider the lossless 'black box' approach of figure 1 where MPEG-II data is transcoded [1, 2] into a more resilient structure without increasing the bit rate, transmitted over a noisy channel, and finally recoded back into a compliant MPEG-II format, suitable for input to an MPEG-II decoder. The lossless transcoders are designed to provide substantial resilience to errors.

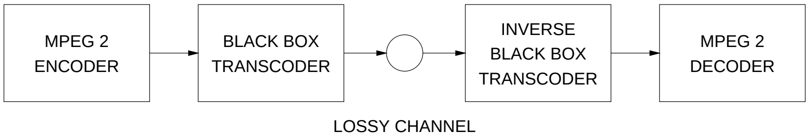


Figure 1: The MPEG-II Transcoder

Figure 2 is a stylized graph to illustrate the difference between channel coding and resilient coding schemes. The channel coded scheme (- -) exhibits a substantially constant quality until the channel noise has reached the error-correcting capability of the code, at which point, the error correction adds even more errors, and the quality degrades sharply. The resilient coding scheme (–) shows a higher quality for a noiseless channel as bits are spent on finer quantisation instead of unused error correction. The resilient scheme exhibits a video quality which degrades gracefully as the channel quality degrades. We believe that the resilient coding approach can be of benefit in many wireless applications. The resilient coding scheme works particularly well where the channel characteristics are complicated, unpredictable or non-stationary, as FEC can only be used effectively where the channel characteristics are predictable. It can also provide the user with warning that the channel is deteriorating before the quality loss becomes unacceptable.

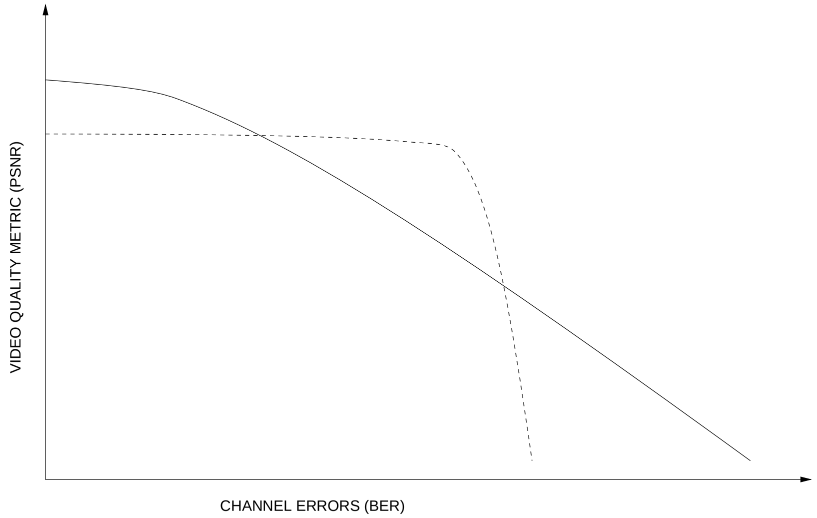
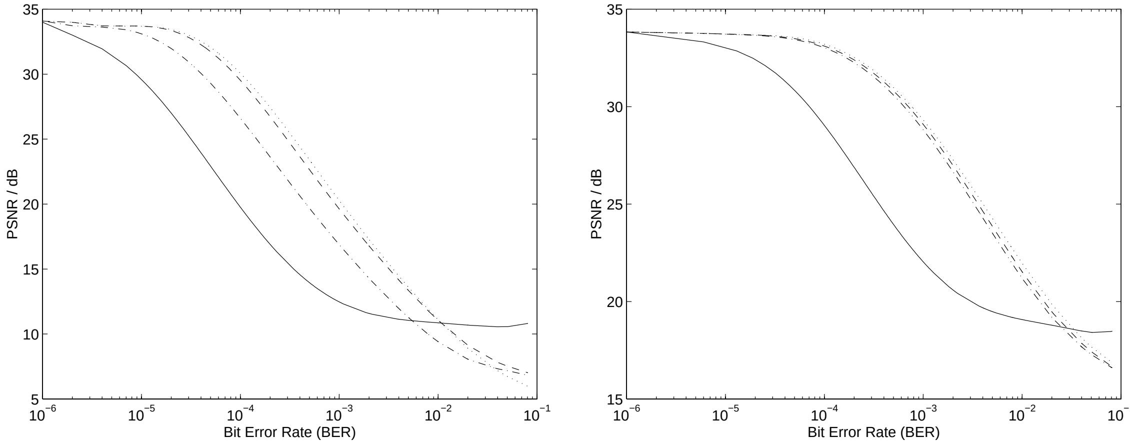


Figure 2: The relative performances of channel coding and resilient coding schemes

2 ERROR PROPAGATION

From error sensitivity measurements on MPEG-II bitstreams, we conclude that it is not the loss of any particular type of bit, but the loss of bitstream synchronisation which is the primary cause of corrupted pictures. Figure 3 shows the performance of resynchronisation at the slice layer as in normal MPEG-II(–) , the macroblock layer (·—), the block layer (- -) and after each independent coefficient (··). We find the most useful resynchronisation points to be at the beginning of each block in intra pictures and at the beginning of each macroblock in predicted pictures.



(a) Intra (I) Pictures (b) Predicted (P) Pictures

Figure 3: The performance of different resynchronisation schemes

3 ERROR RESILIENT SYNCHRONISATION

To achieve block synchronisation, we ensure that each variable length block starts at a known position in the transmitted bitstream. We achieve this using a technique known as error-resilient entropy coding (EREC). This is described in detail in [3], so we only give a brief summary here: The bitstream is re-ordered into fixed length slots without adding redundancy such that longer blocks fill up the spaces left by shorter blocks. Figure 4 shows N variable length blocks of data (for $N = 10$). Each block can be considered to be the data required to code an image block, and its height corresponds to the number of bits in that block. The idea behind the EREC is to fit these variable length blocks into the fixed slot structure of figure 5.

The EREC slot height corresponds to the average block length. Therefore the number of slots is N, and the total number of spaces in the slot structure is equal to the total number of bits in all of the variable length blocks. Since the average block length is often non-integer, we transmit the number of bits per EREC frame, making the first few slots one bit longer than the rest of the slots. It is essential that the decoder receives this information without error, and so

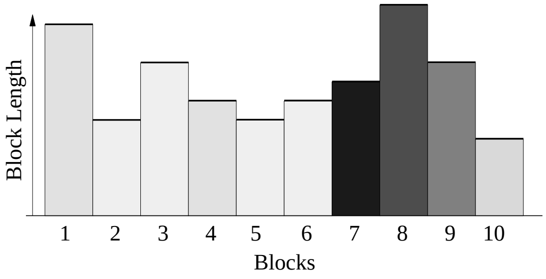


Figure 4: Variable Length Data Blocks

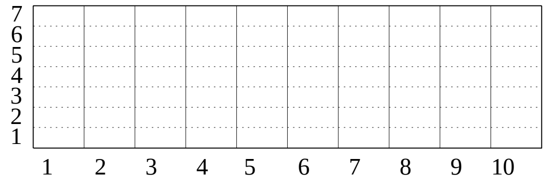


Figure 5: Fixed Length EREC Slot Structure

the number of bits per frame is heavily protected.

Initially as much as possible of each block of data is placed in the corresponding slot. Any data which overfills its slot is retained. In the subsequent stages of the EREC encoding process, the remaining data is shifted to the right. Here it attempts to fill any spaces left by smaller than average blocks. Figure 6 shows this after the first shift. This process continues, until, after $N-1$ shifts, all the bits of data will have been placed in the slot structure, as in figure 7.

The decoding process is similar to the encoding process except it operates in reverse: the decoder starts at the bottom of each slot and decodes the data going up the block until either the top of the slot is reached, or the end of block codeword is received. If the end of block

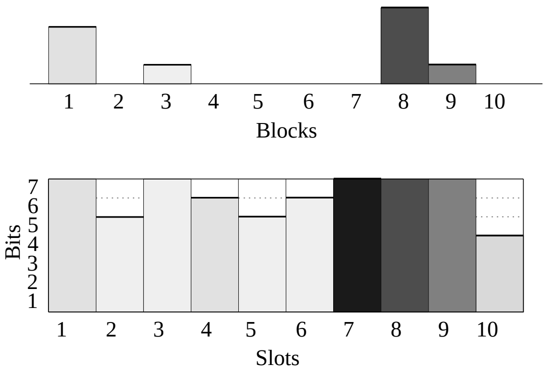


Figure 6: EREC Stage 1

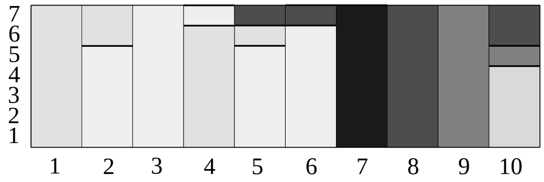


Figure 7: Final EREC State

is detected, then that block has been successfully decoded. If, however, the top of the slot is reached, then the decoder knows that the rest of the data exists on top of another slot. The decoder shifts all the excess bits back to the left, and continues the process until all the data has been successfully decoded. No additional information is required to correctly decode the EREC data.

4 ENHANCEMENTS TO THE EREC

The further a codeword is from the beginning of a variable length block of data, the greater the probability of that codeword being corrupted. This is because data which is placed in the slot structure later is more susceptible to error propagation than data placed early on. It therefore makes sense to place important data at the start of blocks, or indeed to create separate slots for different blocks of data. We therefore create separate slots for the DC, and macroblock type data in intra pictures.

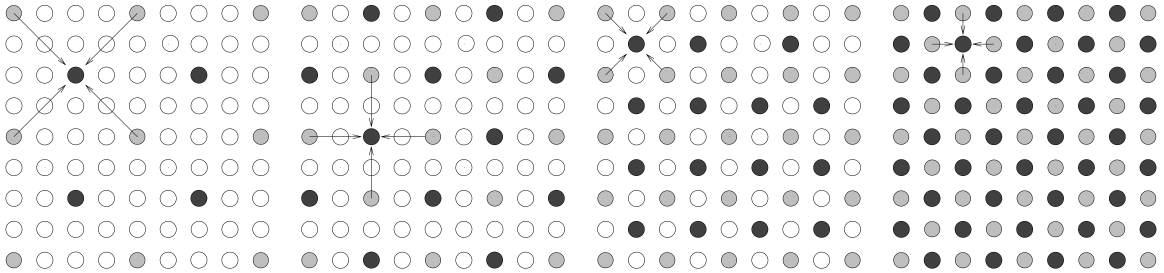
There is no reason why each slot should be the same length. Hence we specify smaller slots for the macroblock type data and the DC coefficients. Indeed it is possible to give more protection to more important data such as the DC coefficients by making the DC coefficient slots equal to the maximum length of a DC codeword. This adds more resilience to the DC data at the expense of the less important high frequency AC coefficients.

Because of the spatial correlation in images, high activity blocks tend to be clustered. This can mean that some long blocks get fitted into many EREC slots. We therefore use a pseudo-random shift sequence rather than the simple regular one illustrated in figures 4 to 7, so that neighbouring slots are filled by blocks from all over the EREC frame. This causes blocks to be fitted more rapidly into the slot structure and hence decreases the error extension in high frequency AC DCT coefficients, as well as decreasing the computational effort in the EREC decode process.

In channels subject to packet loss, we find that gains can be made by prohibiting any EREC shifts to or from slots which have suffered erasures from the lost packets. This has the effect of lowering error extension in high frequency DCT coefficients.

5 RESILIENT DIFFERENTIAL CODING

Given block resynchronisation, we find that it is the MPEG-II differential coding of the DC coefficients and motion vectors which causes the most visible artifacts. The corrupted DC coefficients appear as corrupted horizontal stripes (slices). Instead we employ a hierarchical pyramid based coding scheme using a four point median predictor to code these parameters. This greatly increases the error resilience, for no extra coding cost. Indeed the coding efficiency is usually improved. The pyramid coder [8] is a quincunx coder which forms alternate square and diamond shaped predictions (see figure 8).



(a) Stage 1 (b) Stage 2 (c) Stage 3 (d) Stage 4

Figure 8: Hierarchical Quincunx Coding. At each stage the black pels are predicted from the four nearest grey pels, and the prediction errors are coded

Initially, we code one point in 64 without reference to any other points. Although this is error-resilient, it does not yield high compression. Therefore we only send a small fraction of bits in this manner.

For the differential coding, we use a four point median predictor except at the edges where fewer points are used. The median filter has the inherent property of rejecting outliers, so errors will not usually propagate far and will only propagate within the squares or diamonds in which they occur. With proper attention to rounding, this predictor can be designed to operate losslessly on the original 1D-DPCM coefficients.

6 BITRATE, BUFFERING AND COMPLEXITY

MPEG-II is naturally very sensitive to errors in the sequence header. One error in this header can cause the whole sequence to be lost. The sequence header contains information such as the picture display size, and chroma format. The sequence layer also carries information such as user-defined quantisation scales; timing and bitrate data; the profile and level; and the video format. Although it is common practise to assume the sequence header information to be *a priori* knowledge, the picture size and chroma format information amount to only 30 bits, which may easily be protected using FEC without a incurring a large overhead.

MPEG-II is also very sensitive to corrupted picture headers. The picture header contains, among other information, the intra DC precision and motion vector ranges; the DCT scan pattern; the picture structure; and picture coding type. The essential information in this layer ammounts to fewer than 40 bits, and should also be heavily protected using FEC.

We propose a scheme whereby each picture is encoded using EREC frames of four MPEG-II slices. A high number of slices per EREC frame results in fewer bits required to specify the frame sizes, and a high coding efficiency. However, large EREC frames increase complexity, memory requirement, and delay. At compressed bitrates of the order of 5Mb/s, the choice of four slices per EREC frame represents a reasonable tradeoff between these two factors.

EREC frames are byte aligned, and so the EREC frame lengths may be specified using an 18-bit code. This is protected up to BER=10% using the (32,6) augmented Reed-Muller (Distance=16) code. This results in a 96-bit EREC frame specifier code.

As four slices are coded per EREC frame, four MPEG-II 32-bit slice start codes totalling 128 bits are eliminated, to be replaced by the 96-bit EREC frame spicifier and leave 32 bits spare. These spare bits amount to 288 bits per picture, assuming 9 EREC frames in a picture.

In addition, we save 68 bits from the picture start codes, and so we have 356 bits per picture in which to gain picture synchronisation, and to convey the picture layer data. We propose a 128-bit pseudo-random synchronisation sequence at the beginning of each picture. This sequence is used to achieve picture synchronisation, and since it replaces the need for MPEG-II start codes, the MPEG-II requirement for no start code emulation is relaxed. The remaining 228 bits are used to protect picture layer information, and so of the order of 50 bits of picture layer data may be protected up to BER=10% using the (32,6) Reed-Muller code. This protects the 40 bits mentioned above, leaving 10 spare bits which may be protected.

The 128-bit picture synchronising code is followed immediately by the heavily protected picture data and EREC frame specifier codes. This data ammounts to around 1.5kbits, and must be received error-free. We therefore interleave this data over the first 12kbits of the transcoded data.

In this scheme, we introduce a delay of one picture, to the forward transcoding process. Memory is required in the forward transcoder, as EREC frame information for a whole picture is transmitted at the beginning of a picture. However, this buffer is not required in the inverse transcoder, since the frame sizes will have been recieved at the beginning of the picture.

In delay-critical applications, the delay may be reduced to the EREC frame size of four slices (64 lines), which typically corresponds to little over 10% of the picture. This would require a different scheme for the protection of the critical data.

The transcoder adds little complexity to the encoder, but it does add some complexity to the decoder. However, the complexity of the transcoder is still small compared to the complexity of a compliant MPEG-II decoder, because the black box does not require any transform or motion compensation operations, and memory requirements are limited to relatively small EREC frames.

7 CONCEALMENT

Due to the inherent error extension towards the end of EREC blocks, many errors will appear as incorrect high valued AC coefficients. We use a simple error concealment strategy whereby large DCT coefficients which are statistically unlikely (based on a training set of various images) are suppressed at times of high bit error-rate. Following each suppressed coefficient, all remaining coefficients in that DCT block are also suppressed since these would also be expected to be corrupted. This scheme is still lossless as the concealment is only switched on in the presence of errors. Figure 9 shows the maximum AC DCT coefficients averaged over many sequences. The dotted line shows the threshold used to switch on concealment. Enhanced performance could be achieved by raising and lowering this threshold depending on the Bit Error Rate.

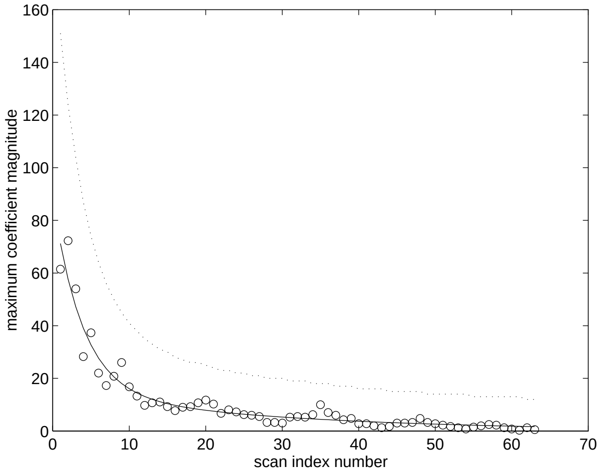


Figure 9: The maximum values of AC coefficients depending on the scan index number

There are various ways in which the bit error rate can be estimated for concealment purposes. A BER estimate can be formed by looking at the number of errors corrected in the protected header. Estimates can also be achieved from the number of failures to find the end of a block in the EREC decoding process, or the number of illegal variable length codes received. An estimate of the bit error rate may also be available from the MPEG-II systems layer.

8 RANDOM BIT ERROR MODEL

In the next three sections, we consider three separate error models. First, we consider a channel subject to random bit errors. This represents the worst case situation for error resilient systems, since these methods used are found to perform even better if the errors occur in bursts (for the same mean bit error rate) because after the first error in a block, subsequent errors in that block have little effect. Figure 10 shows the relative performances of standard MPEG-II and the transcoded scheme for a group of 12 pictures in the sequence ‘calendar and mobile’ coded

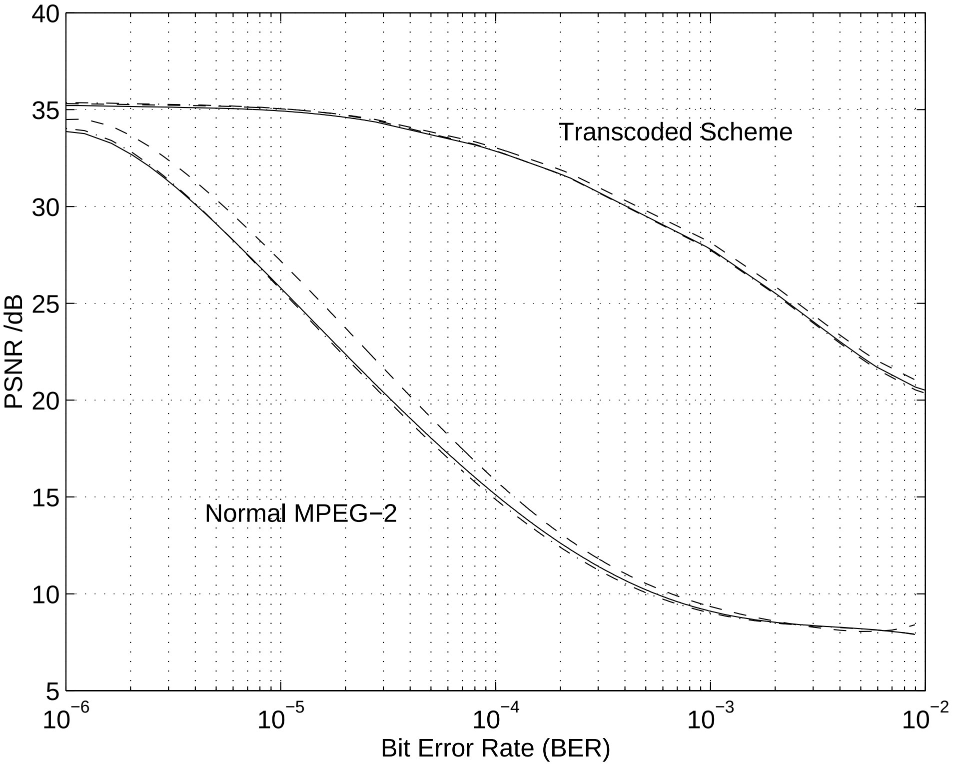


Figure 10: Degradation of signal to noise ratio vs. channel bit error rate for a group of pictures of 'mobile and calendar'. (- -) denotes intra pictures, (·—) denotes I and P pictures, and (–) denotes an average over all pictures (I+P+B).

at 5Mb/s. We show results for three cases: I pictures only, I and P pictures, and all pictures (I, P and B). This graph shows that for a constant peak signal to noise ratio, the transcoded scheme can withstand up to a hundred times as many errors as can the standard MPEG-II stream (e.g. for 25dB PSNR, standard MPEG-II requires BER$< 10^{-5}$ but the transcoded scheme can withstand errors up to 2.10^{-3}). It also shows that the main degradations occur in I pictures and that little change in distortion occurs as these degradations propagate to P and B pictures. Figure 13 shows the same information for an intra picture from the 'calendar and mobile' sequence. Figure 13(a) shows the original error-free decoded picture, and 13(b) MPEG-II at BER$= 10^{-3}$. Figure 13(c) shows the transcoded scheme at BER$= 10^{-3}$, and 13(d) the transcoded scheme at BER$= 10^{-2}$.

9 BURST ERROR MODEL

Now we consider a channel subject to burst errors. We consider geometric distributions for burst lengths and error-free gaps, thus the channel may be characterised by two parameters: the average bit error rate (BER) and the average burst length (ABL).

Figure 11 shows the effect of varying the burst length at a constant BER. It can be seen that the greater the burst length, the greater the performance (for a constant average BER). This can be explained by considering an error in an EREC slot. If we assume that an error causes loss of synchronisation until the next EREC slot, then subsequent errors in that slot will have no further effect.

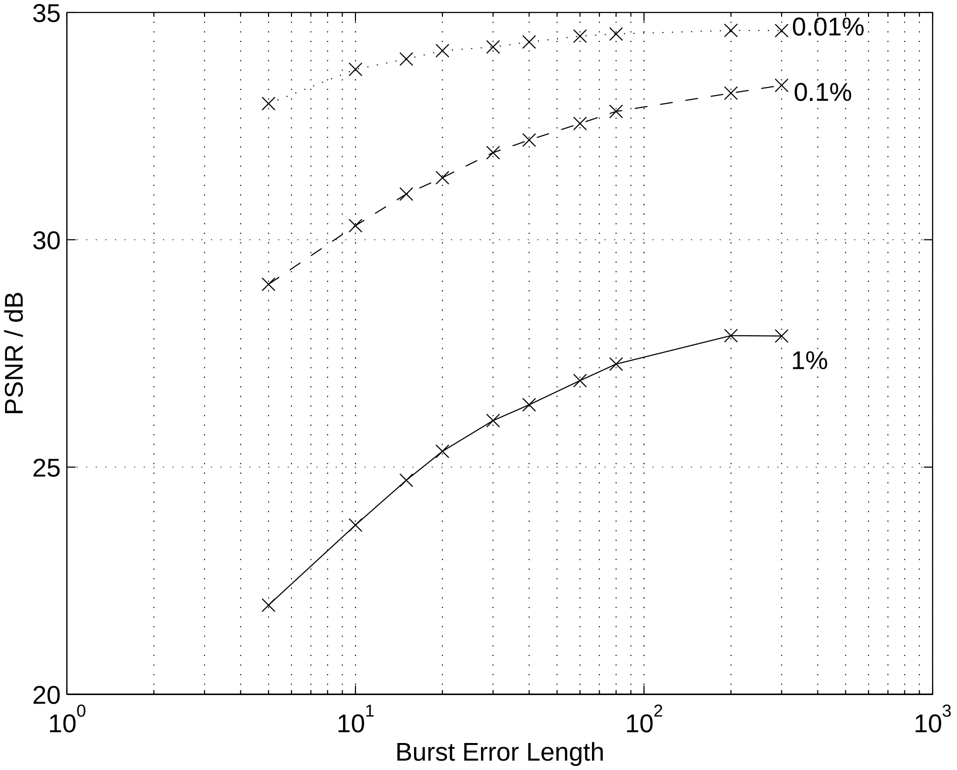


Figure 11: The effect of varying the average burst error length (ABL), while the BER is constant. (··) corresponds to BER= 10^{-4}, (- -) BER= 10^{-3}, and (–) corresponds to BER= 10^{-2}. The BER is assumed to be $\frac{1}{2}$ inside the bursts, and 0 outside the bursts.

10 ATM CELL LOSS MODEL

Now we consider a more realistic model for packet errors. We assume the transcoded data is packaged into 53 byte ATM cells of which 47 bytes are available for video data. We also assume that this layer provides sequencing, and an indication of which cells have safely arrived. Cells are lost according to a cell loss ratio (CLR). Lost cells are considered to be erasures and cells which are not lost are considered to have arrived with no bit errors. Lost cells are flagged to the decoder for concealment purposes. In intra pictures, the AC coefficients of lost blocks are removed, and only DC coefficients are used. The differential DC coefficients are set to zero, and so the DC coefficients are predicted from neighbouring blocks according to the quincunx predictor in section 5. This has the effect of spatially interpolating the DC coefficients. A similar prediction effect occurs automatically when predicted motion vectors are lost. In inter pictures, lost macroblocks are considered to be skipped (uncoded). This then corresponds to motion compensated temporal replacement.

Figure 12 shows the performance of our scheme compared to standard MPEG-II. It can be seen that for a constant 30dB output PSNR, the transcoded scheme can withstand fifty times as many lost cells as standard MPEG-II. Figure 13 shows the same information for an intra picture from the 'calendar and mobile' sequence. Figure 13(e) shows MPEG-II at CLR= 1%, 13(f) shows the transcoded scheme also at CLR= 1%, and figure 13(g) shows the transcoded scheme at CLR= 10%.

Although we have presented results for sequences coded at 5Mb/s. We find very similar results to be obtained using sequences coded in the range 1.5Mb/s to 8Mb/s. We have also achieved similar performance gains using our techniques in an h.261 scheme at 64kb/s.

11 CONCLUSION

We have introduced the concepts of a lossless black box transcoder for video elementary stream, to provide error resilience *without* changing the bit rate. It employs a resilient technique for gaining block-synchronisation, a more resilient technique for coding differential data, and error

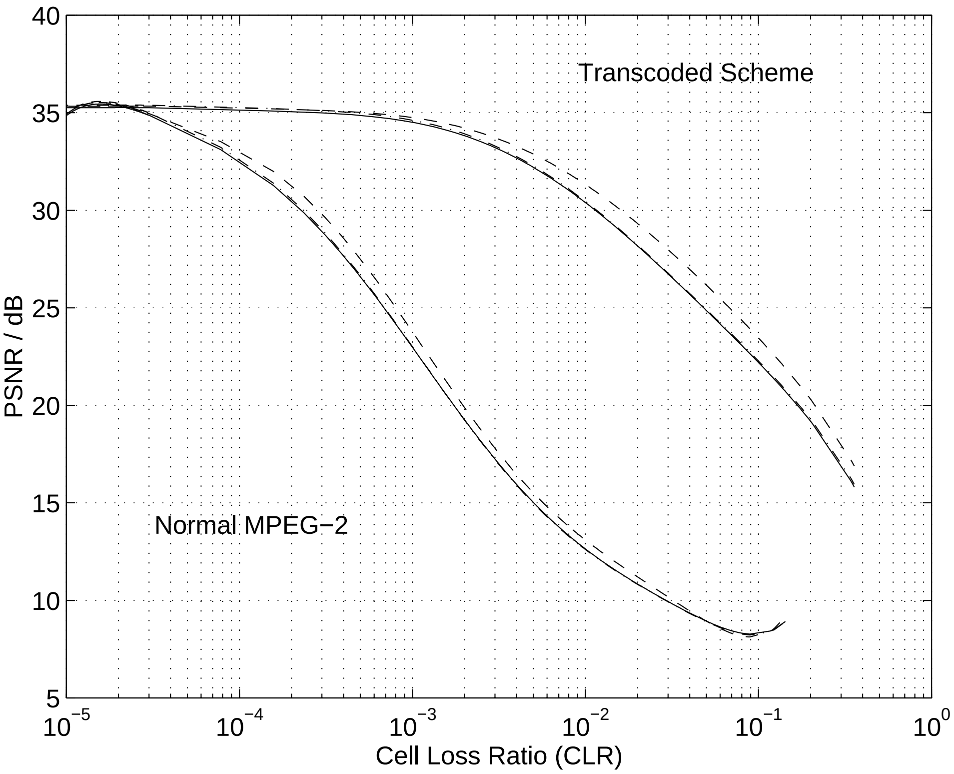


Figure 12: Degradation of signal to noise ratio vs. channel cell loss rate (CLR) for a group of pictures of 'mobile and calendar' coded at 5Mb/s. (- -) denotes intra pictures, (·—) denotes I and P pictures, and (–) denotes an average over all pictures (I+P+B).

concealment. Using these techniques, graceful performance degradation appears possible to random bit error rates exceeding 1 in 100, or cell loss ratios exceeding 10% in the MPEG-II video elementary stream.

References

[1] R.E.M. Swann and N.G. Kingsbury. Techniques for improving the error resilience of MPEG2 codecs. In *Proceedings of the International Broadcasting Convention*, pages 450–455, September 1996.

[2] R.E.M Swann and N.G. Kingsbury. Transcoding of MPEG2 for enhanced resilience to transmission errors. In *Proceedings of the International Conference on Image Processing*, pages II.813–816, September 1996.

[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*, 4(5):565–574, April 1996.

[4] ISO/IEC JTC1/SC29/WG11 N0702 Rev. Generic coding of moving pictures and associated audio.

[5] O. A. Aho and J. Juhola. Error resilience techniques for MPEG-2 compressed video signals. In *Proceedings of the International Broadcasting Convention*. IEE, September 1994.

[6] Y. Ninomiya. Error management in digital terrestrial television broadcasting. *SMPTE Journal*, pages 595–607, September 1994.

[7] E. Ayanoglu, P. Pancha, A. Reibman, and S. Talwar. Forward error control for MPEG-2 video transport in a wireless ATM lan. In *International Conference on Image Processing 1996*, pages II.833–836, September 1996.

[8] B. Zeng, Y. Neuvo, and A. Venetsanopoulos. Interpolative BTC image coding. In *ICASSP 92*, pages III–95–99, San Francisco, March 1992.

12 Acknowledgments

This work was supported by an the EPSRC CASE award in conjunction with the British Broadcasting Corporation. The authors would like to thank the BBC Research Department for their support.

Figure 13: (a) Original Error-Free Decoded Intra Picture

Figure 13(b): MPEG-II with protected header, BER=10^{-3}

Figure 13(c): Transcoded Scheme, random BER= 10^{-3}

Figure 13(d): Transcoded Scheme, random BER= 10^{-2}

Figure 13(e): MPEG-IIwith protected header, Cell Loss Ratio =1%

Figure 13(f): Transcoded scheme, Cell Loss Ratio=1%

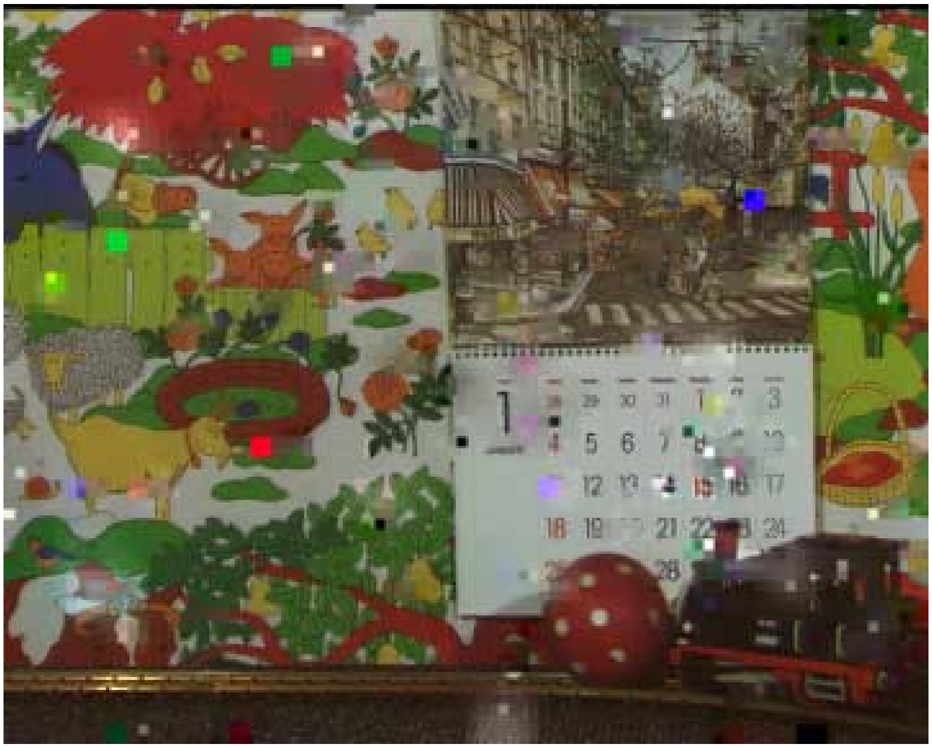


Figure 13(g): Transcoded scheme, Cell Loss Ratio=10%