Enabling Error-Resilient Internet Broadcasting using Motion Compensated Spatial Partitioning and Packet FEC for the Dirac Video Codec

M. Tun, K.K. Loo, and J. Cosmas
Electronics and Computer Engineering, School of Engineering and Design, Brunel University
Kingston Lane, Uxbridge, UB8 3PH, UK
Email: myo.tun@brunel.ac.uk, jonathan.loo@brunel.ac.uk, john.cosmas@brunel.ac.uk

Abstract— **Video transmission over the wireless or wired network require protection from channel errors since compressed video bitstreams are very sensitive to transmission errors because of the use of predictive coding and variable length coding. In this paper, a simple, low complexity and patent free error-resilient coding is proposed. It is based upon the idea of using spatial partitioning on the motion compensated residual frame without employing the transform coefficient coding. The proposed scheme is intended for open source Dirac video codec in order to enable the codec to be used for Internet broadcasting. By partitioning the wavelet transform coefficients of the motion compensated residual frame into groups and independently processing each group using arithmetic coding and Forward Error Correction (FEC), robustness to transmission errors over the packet erasure wired network could be achieved. Using the Rate Compatibles Punctured Code (RCPC) and Turbo Code (TC) as the FEC, the proposed technique provides gracefully decreasing perceptual quality over packet loss rates up to 30%. The PSNR performance is much better when compared with the conventional data partitioning only methods. Simulation results show that the use of multiple partitioning of wavelet coefficient in Dirac can achieve up to 8 dB PSNR gain over its existing un-partitioned method.**

***Index Terms*— Error-resilient coding, coefficient partitioning, wavelets, Dirac**

I. INTRODUCTION

The latest video coding standard, H.264 which is being developed by the Joint Video Team (JVT) of the ITU-T Video Coding Expert Group (VCEG) and ISO MPEG-4 groups is aimed to elaborate an open standard that is not application-specific and that perform significantly better than the existing standards in terms of compression, network adaptation and error robustness. In the near future, H.264 will gain wide acceptance on most of the applications especially on broadcasting over wireless, satellite or Internet mediums. However, the improved coding efficiency of H.264 has expensive royalty fees [1] making it too costly for public service broadcasters. Whilst these costs are manageable initially, they could become prohibitive if broadcasters try to scale up to millions of simultaneous users, or if new services are deployed which were not envisaged in the original license agreements.

As an alternative, the British Broadcasting Corporation (BBC) is developing a royalty-free general-purpose video codec called Dirac [2], which is aimed at a wide range of applications from storage of video content to streaming video and supports any frame dimensions from QCIF to HD. Being *Open Source*, Dirac is a very attractive option since the cost of distribution via the Internet increases with the number of users of proprietary systems, and the BBC has ambitions to offer public access to its multimedia archive via internet broadcasting. This wonderful resource remains inaccessible to the public because of the lack of an effective mechanism to suit the public service business model for distribution.

Compressed multimedia data streams transmitted over error prone broadcast channels, such as wireless networks and the Internet is usually corrupted by channel errors. The current alpha release of Dirac codec has been optimized for storage purposes only and still there is no error-resilient encoding mechanism for transmission over the erroneous channels. Our main objective in this paper is to propose a simple, low complexity and patent free error-resilient coding technique in order to fulfil the main non-functional requirements of the Dirac video codec [2].

Several techniques have been developed over the last decades to make video transmission over a wireless or wired network resilient to errors. One approach is to transmit the video sequence into several bit-streams, called descriptions [3][4][5]. In this method, a video sequence will be encoded into two or more bit-streams or descriptions and transmitted over different channels. When all of the descriptions are correctly received, the decoder can reconstruct the video with the best quality. If any of the descriptions are lost during transmission, the decoder can still reconstruct the video with a lower, but acceptable quality. However, transmitting multiple copies of bit-streams would require higher band width and is thus not a suitable method for most applications in which only limited band width is available.

Some consider protecting the transmitted bit-streams against packet losses by applying an unequal amount of Forward Error Correction (FEC) to different data fragments according to the importance of the data [6][7]. However, this technique has the disadvantage of still being vulnerable to packet erasures or channel errors that occur early in the transmission, either of which can cause a total collapse of the decoding process.

Another approach, called coefficient partitioning makes video transmission resilient to channel errors by partitioning the wavelet transform coefficients into groups and independently processing each group. Thus, a bit error in one group does not affect the others, allowing more uncorrupted information to reach the decoder. This method was first reported by Creusere [8] for use with the EZW algorithm in error resilient image transmission. Block based coefficient partitioning method is presented in [9] where each subband is partitioned into an equal number of coefficient blocks. Each coefficient block in a subband carries information about some localized region in the original frames. The components are then formed by grouping from each subband, equal number of coefficient blocks that correspond to different spatial regions of the source. However, none of the above mentioned coefficient partitioning methods survives in the channel having higher packet loss rate more than 5 percent.

To overcome this problem, combined source and channel coding has been considered in most cases where one of the coefficient partitioning methods is used as source coding and combined together with FEC to achieve double level of protection from transmission error [10][11].

In Pearlman's work [12], the wavelet transform coefficients is first broken into a number of spatio-temporal tree blocks according to [8], and the 3-D Set Partitioning in Hierarchical Trees (SPIHT) algorithm is modified to work independently with these blocks. They then apply Kim's method [13][14] of RCPC channel coding as the forward-error correction (FEC) to every packet to protect the data. It is interesting to note that the scheme proposed in [15] could be used inline with any error-resilient coding method to alleviate the effect of error propagation by adding some periodic macroblocks in every fifth inter-frames.

In this paper, the wavelet coefficients partitioning method for error resilient image transmission from [8] is extended in order to work with motion compensated 2D wavelet transformed residual frames and used as the source coding. The idea behind this source coding is that most of the transformed coefficients partitioning techniques in the literature were based upon the intra frames or 3D wavelet transformed frames. By applying the coefficient partitioning upon the residual frames, there would be an extra advantage in reconstructing the corrupted blocks if the reference frame and its motion vector are correctly received. As for the channel coding, RCPC [16] and Turbo Coding (TC) [17] were used in order to investigate the performance for both low and high complexity channel coding mechanisms. Error resilient transmission for the packet erasure wired network can be achieved by using the bitwise interleaver at the output of the encoder. Any type of transform coefficients coding algorithms such as EZW, ZTE, SPIHT, etc., were not used since all of these are heavily patented and Dirac doesn't want to include any patented algorithm in their codec architecture [2]. Moreover, these algorithms do not perform very well in applying to the motion compensated residual frames since most of the coefficients in these frames have already been transformed to zeros.

The organization of this paper is as follows. Section II provides a brief introduction to Dirac video codec and Section III presents the proposed error resilient video encoding technique and section IV explains decoding technique at the receiver. The results and conclusions are presented in section V and VI, respectively.

II. DIRAC VIDEO CODEC

Dirac is an Open Source video codec aimed at resolutions from QCIF (176×144) to HDTV (1920×1080) progressive or interlaced, initially developed by BBC [2]. It aims to be competitive with the other state of the art standard video codecs and performance is very much better than MPEG-2 and slightly less than H.264 even in the Alpha development stage. However, performance was not the only factor driving its design. Dirac is intended to be simple, powerful and modular. It uses hierarchical motion estimation and Overlapped Block-based Motion Compensation (OBMC) to avoid block-edge artefacts. First the motion compensated residual frames are wavelet-transformed using separable wavelet filters and divided into subbands. Then, they are quantized using Rate-Distortion Optimization (RDO) quantizers. Finally, the quantized data is entropy coded using an Arithmetic encoder.

The codec can support any frame dimensions and common chroma formats (luma only, 4:4:4, 4:2:2, 4:2:0) by means of frame padding. The padding ensures that the wavelet transform can be applied properly. Frame padding also allows for any size blocks to be used for motion estimation, even if they do not evenly fit into the picture dimensions.

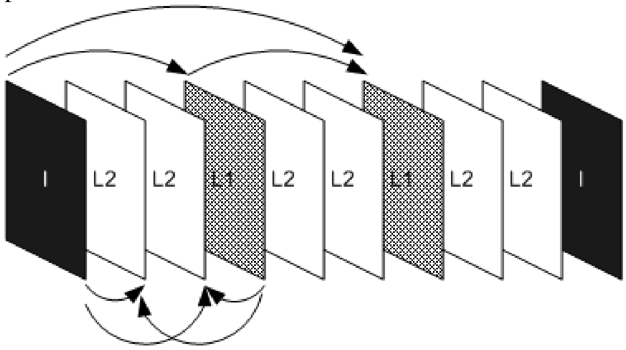


Figure 1. Prediction of L1 and L2 frame

Dirac defines three frame types. Intra frames (I frames) are coded independently without reference to other frames in the sequence. Level 1 frames (L_1 frames) and Level 2 frames (L_2 frames) are both inter frames, which are coded with reference to other previously (and/or future) coded frames. The definition of the L_1 and L_2 frames are the same with P and B frames in H.264. The encoder operates with standard Group of Picture, (GOP) modes whereby the number of L_1 frames between I frames, and the separation between L_1 frames, can be specified depending on the application.

A prediction method for frame coding using a standard GOP structure is shown in Figure 1. In this figure, the number of L_1 frames between I frames is 2 and the L_1 frame separation is 3.

Current version of Dirac can be used only for the storage purpose. The encoder is still lacking the other facilities e.g. error-resilient transmission and rate control which are the essential features for real time video broadcasting. Scalability is also another important feature that Dirac still requires. The main objective of this research is to enable the error-resilient transmission of the encoder in order to be able to use in real time broadcasting.

III. Proposed Error-Resilient Video Coding Technique

A. Coefficient Partitioning

The basic idea of the coefficient partitioning for error resilient coding is to divide the wavelet coefficients at the output of the DWT process of the Dirac codec into S groups and then quantize and code each of them independently so that S different bitstreams are created [8]. By coding the wavelet coefficients with multiple, independent bitstreams, any single bit error truncates only one of the S bitstreams while the others are still correctly received. Therefore, the wavelet coefficients represented by the corrupted bitstreams are reconstructed at reduced accuracy, while those represented by the error-free streams are reconstructed at the full encoder accuracy. The partitioning method used here is the extension of [8]. In which, the wavelet coefficients partitioning method is applied to the motion compensated residual frames instead of applying to the intra coded frames in [8] for the image transmission and 3D wavelet transformed frames in [12]. By doing so, the quality of the reconstructed frames particularly at the higher packet loss rate becomes much better than the original scheme in [8] and [12] especially when the motion vector data and reference frames are correctly received. It is because the corrupted data can simply be replaced with the shifted version of the data from the reference frame.

The Figure 2 graphically illustrates this wavelet coefficient partitioning for $S = 4$ bitstreams for four levels wavelet decompositions. In this figure, each coefficient with the same shade of grey maps the same group. If the image is of size $X \times Y$ and L levels of wavelet decomposition are used, then the maximum number of independent bitstreams allowed is

$$S = \left(X \times Y\right) / 4^{L} \tag{1}$$

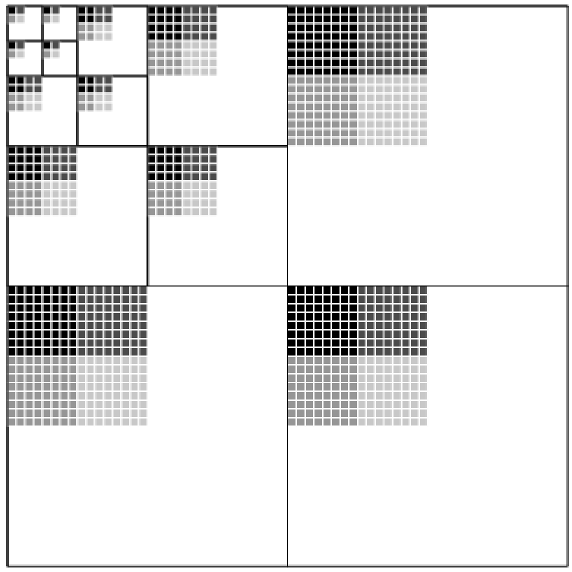


Figure 2. Wavelet Coefficient partitioning for S = 4, with four levels wavelet transform.

B. Error Resilient Video Coding

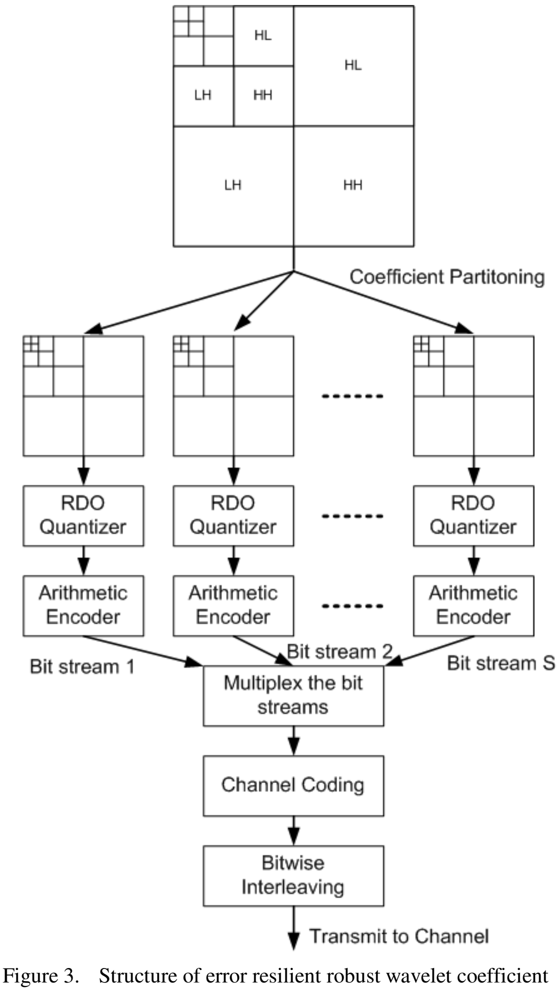


Figure 3. Structure of error resilient robust wavelet coefficient partitioning and encoding procedure

Figure 3 shows the block diagram of error resilient encoding procedure of the Dirac video codec. The output of the DWT process of Dirac encoder is divided into S sub-frames according to the wavelet coefficient partitioning method shown in the section III.A. Then, these sub-frames are processed independently, i.e. by employing the RDO quantization and arithmetic encoding before entering the multiplexer. In the multiplexer, all the independent parallel bitstreams are combined to obtain a serial stream starting from bitstream 1, followed by bitstream 2 and so on until bitstream S is reached.

The bitstream syntax of the original Dirac codec and the proposed method with S number of partitions are shown in Figure 4 and 5 respectively. The resulting bitstream syntax after multiplexing no longer follows the original syntax because of the introduction of multiple partitioning. Then, FEC is applied to the output serial bitstream by using rates 2/3, 1/2, 1/3 and 1/4 RCPC encoder or rate 1/2 of Turbo Encoder.

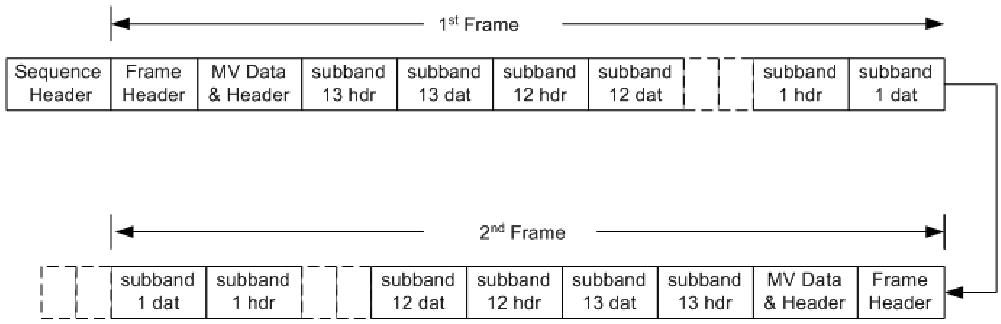


Figure 4. Bit-stream Syntax of original Dirac codec

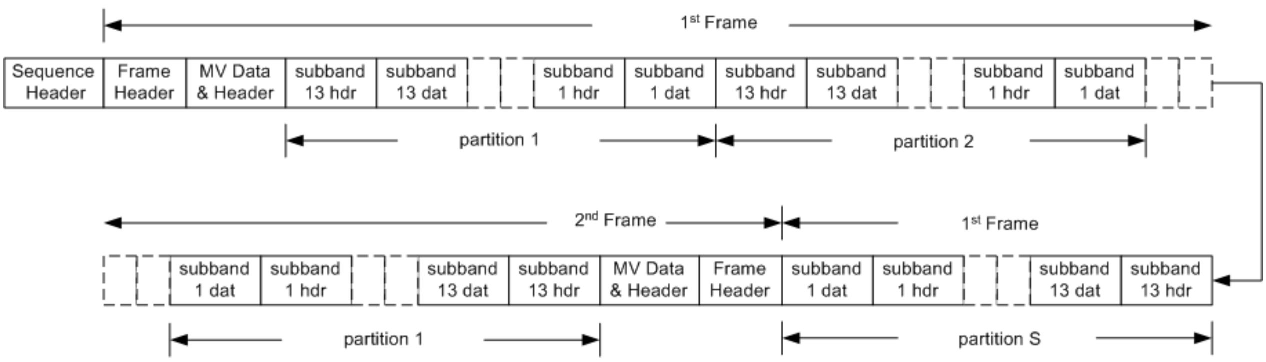


Figure 5. Bit-stream Syntax of Dirac codec with proposed method, for S number of partitions

In order to incorporate channel coding to the output bitstream, first, the bitstreams are breaked into equal length segments of L bits. The checksum bits, $c = 16$ of the Cyclic Redundancy Code (CRC) is generated for each segment of L bits and then appended to each segment. Next m bits where $m = 8$ are padded at the end of each $L + c$ bit segment to flush the memory of the encoder (i.e. to terminate the trellis at original stage). Finally, each segment of the $L+c+m$ bits is passed through the rate R of the channel encoder. The generator polynomial of the CRC used here is from [12] where $g(x) = X^{16} + X^{14} + X^{12} + X^{11} + X^{8} + X^{5} + X^{4} + X^{2} + 1$.

The generator matrix and puncturing tables of the RCPC encoder are given in the appendix. The memory of the associated convolutional encoder, M is 4 and punctured periodically with period, $P = 8$. At the receiver, the Viterbi Algorithm is used to decode the received data.

Turbo encoder is the parallel concatenation of two recursive systematic convolutional encoders having generator polynomials $g_1 = 31$, $g_2 = 27$ with memory $M = 4$. The puncturing is performed at the output of the encoder by taking only odd parity bit and even parity bit from the upper and lower convolutional encoder output correspondingly. The encoder interleaver is a pseudorandom interleaver having a length of $L+c+m$ bits. In the decoder, the symbol by symbol MAP algorithm is used with the number of iterations set to 6.

A bitwise interleaver is placed at the output of the RCPC encoder before the packet is constructed. The role of the bitwise interleaver is to distribute the series of information bits into several different locations so that a packet lost in the packet erasure network does not affect the error correcting capability of the RCPC decoder (i.e. to avoid the formation of error burst). The bitwise interleaver length is set to 100 times the length of the packet, where packet length is equal to $R^{-1} \times (L + c + m)$ bits. So that a packet loss in the packet erasure channel does not mean losing the whole packet instead the loss is only 1/100 of a packet. In the receiver, the RCPC or Turbo decoder can effectively correct those errors since the possibility of error burst formation have been eliminated by using bitwise interleaving.

VI. DECODING TECHNIQUE

It is assumed here that the channel is a packet erasure channel and generates no bit errors inside each packet except the loss of the whole packet because of network congestion. If a packet loss has occurred, an all zero data packet is created at the decoder to replace the lost one and undergo a bitwise deinterleaving process. The pseudo-code of the decoding algorithm is as follow.

```
//S is the number of partitions in a frame
//Function call for Bitwise de-interleaving
BitwiseDeinterleaver();

//Function call for Channel Decoder
//(RCPC or Turbo)
ChannelDecoder();

   //Function call for De-Multiplexing Operation
//After that S number of bitstreams are
//generated
DeMultiplexer();

for j = 1:S
int i = 1;
do
{
    //Checking CRC for erroneous packet
    CheckCRC(packet(i));

    if (CRCfail)
        ERROR_CODE = 1;
    else
        ERROR_CODE = 0;
    end

    if (!ERROR_CODE)
           //Function call to Arithmetic Decoder
        ArithmeticDecoder();
    else
        //Error in Received Packet
        //Jump out of do-while loop
        break;
    end
    i++;
}while (num_packet_left != 0)

if (ERROR_CODE)
      //Fill the rest of the subband coefficients
       //corresponds to erroneous bitstream with
    //zeros
    ZeroPadding();
end

end//end of S loop

   //Reverse Process of Coefficient Partitioning
//at the Encoder
Multiplexer();

//Inverse Discrete Wavelet Transform
  //after that frame reconstruction is performed
IDWT();
```

As shown in pseudo-code, the channel decoding and demultiplexing process follows after bitwise deinterleaving so that S numbers of sub bitstreams are generated. The channel decoder (i.e. RCPC or Turbo decoder) normally tries to correct the errors. CRC error checking mechanism is used to check whether the packet is correctly received or not. If it is erroneous packet, the corresponding ERROR_CODE for this packet is set so that arithmetic decoder can stop decoding for this packet and the rest, and jump to the next bitstream (just out of the do-while loop in pseudo-code). Zero Padding stage is required to fill the rest of the subband coefficients data corresponds to erroneous bitstream with zeros since the arithmetic decoder stop decoding for this bitstream once the ERROR_CODE is set. It then continues to decode the packets of the other bitstreams so that the decoder still have clean packets already decoded up to that point and lose only the remaining packets of the corrupted bitstream.

On the other hand, if only the single bitstream is transmitted without partitioning, the whole remaining bitstream becomes useless if there is any single bit error in the middle of the bitstream. Therefore, by coding the wavelet coefficients with multiple and independent bitstreams, any single bit error affects only one of the S bitstreams, while the others are received unaffected.

In Figure 6, if the error is found in the packet number 10, this packet and rest of the packets in this bitstream are simply discarded. After decoding, the normal un-partitioned case has only 9 clean packets while in the proposed method, it still have 14 clean packets. Obviously, the proposed method could deliver more clean packets than in the normal un-partitioned case since it just stops decoding at the step of first error occurrence. A better error resilient performance can be achieved if the maximum possible number of bitstreams are transmitted, which should be the power of 4 and can be calculated by using Equation 1 according to Creusere's work in [8].

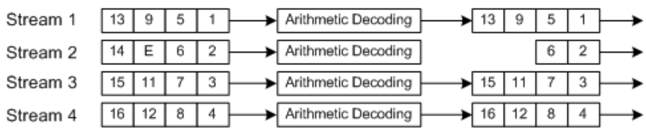


Figure 6. Example of decoding failure at packet number 10

V. RESULTS

A. Simulation Setup

The performance of the proposed model for error resilient video transmission is tested with CIF format canal vertical pan street sequence which can be downloaded from [2] with chroma format 4:2:0 and GOP length 36, i.e. the number of L_1 frames between I frames is 11 and L_1 frame separation is 3. The number of L_2 frames can be calculated by using the equation 2 as follow.

$$Num\_L_2 = (Num\_L_1 + 1) \times L_1\_Sep \qquad (2)$$

TABLE I. GENERAL PARAMETERS

General Parameters	
Block Length ($B = L + c + m$)	200 bits
Number of CRC bits (c)	16 bits
Number of Encoder Tail bits (m)	8
Number of Information bits/Packet (L)	B - c - m
Packet Length	$1/R(L+c+m) = B/R$
Bitwise Interleaver Length	100×Packet Length

TABLE II. CHANNEL CODING PARAMETERS

Channel Coding Parameters	
Rate Compatible Punctured Code (RCPC)	
Number of Memory (M)	4
Number of Encoder Output (N)	4
Encoder Rates (R)	2/3, 1/2, 1/3, 1/4
Puncturing Interval (P)	8
Decision Depth ($> P \times N$)	100 bits
Turbo Code (TC)	
Number of Memory (M)	4
Number of Encoder Output (N)	2
Interleaver Length	Block Length (B)

The total number of the wavelet coefficient partitions, *S,* used in the proposed error resilient coding format is 33 where the partitioning is done to the DWT output of the original frame to get 33 sub-frames.

The distortion is measured by the peak signal-to-noise ratio (PSNR). The goal of the test is to find the performance of proposed error resilient model and so there has been no attempt to conceal the error (error concealment) at the decoder. In order to introduce the unequal error protection, bitstream of the Dirac encoder output is divided into two layers namely: layer 1 and layer 2. Layer 1 includes header information which is most sensitive part of the compressed bitstream and layer 2 corresponds to data layer. Figure 7 illustrates the idea of the separation of two layers. Furthermore, it is assumed that the Dirac header is not corrupted from the packet loss of the packet erasure channel, i.e. any packet error was not introduced to the header since it will be protected by using stronger channel code in actual delivery. All the PSNR curves are averaged over 10 independent runs.

The general and channel coding parameters that are used in the experiments are summarized in Table 1 and 2.

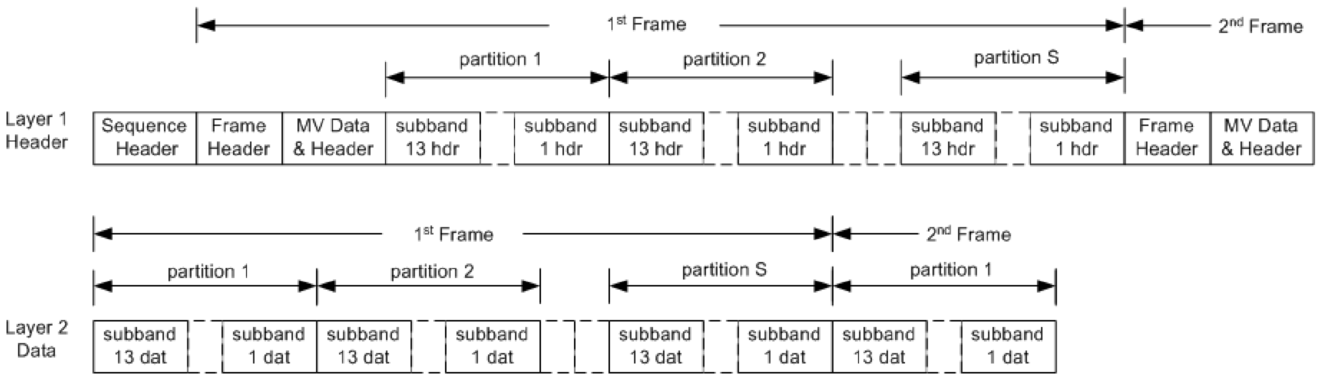


Figure 7. Separation of Layer 1, Header and Layer 2, Data for un-equal error protection

B. Numerical Results and Discussions

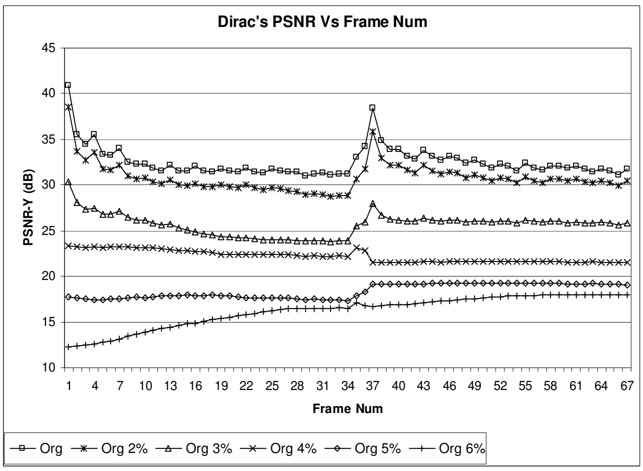


Figure 8. PSNR Performance comparisons between different percentages of packet error for un-partitioned format with rate 1/2 RCPC

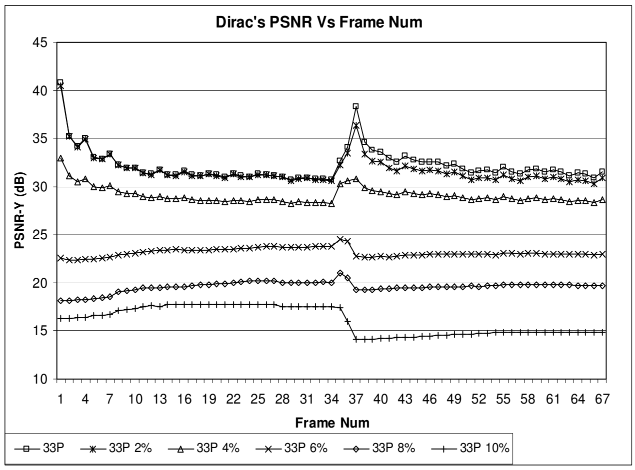


Figure 9. PSNR Performance comparisons between different percentages of packet error for 33-partitioned format with rate 1/2 RCPC

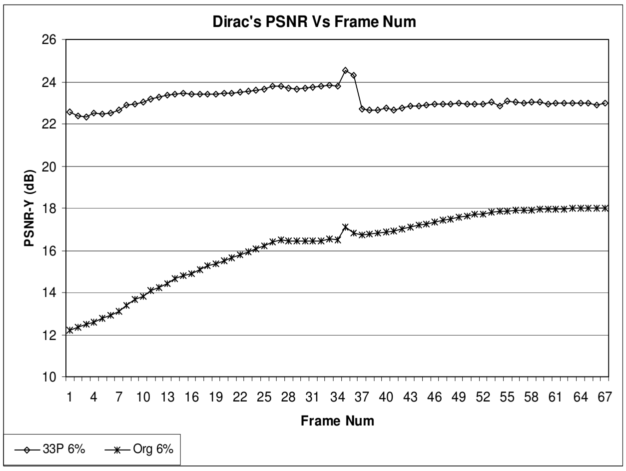


Figure 10. PSNR Performance comparisons between 33-partitioned and un-partitioned (Original) formats for 6% packet loss with rate 1/2 RCPC

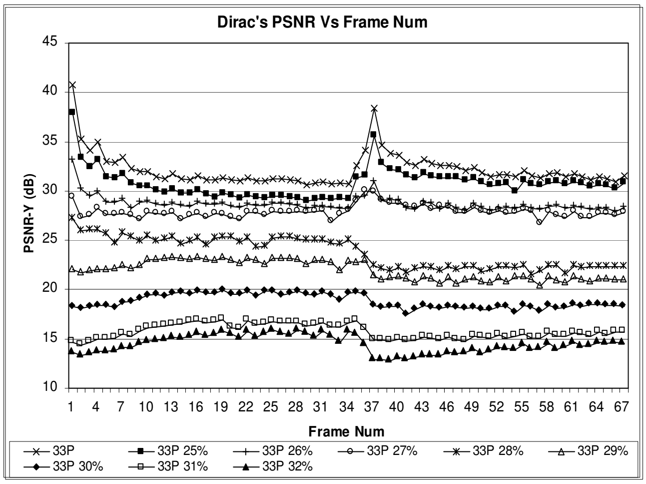


Figure 13. PSNR Performance comparisons between different percentages of packet error for 33-partitioned format with rate 1/2 Turbo coding

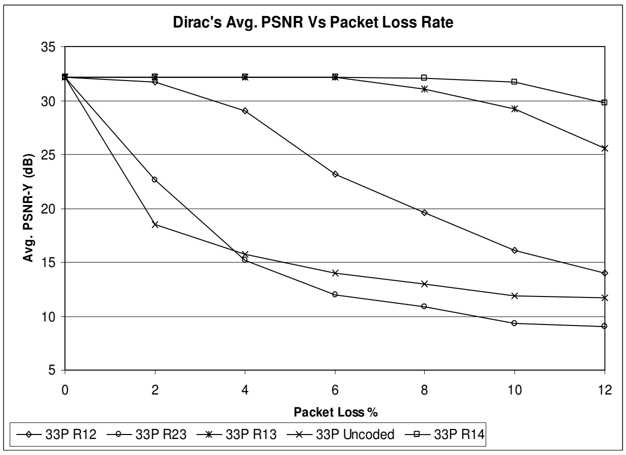


Figure 11. Average PSNR Performance comparisons between Un-coded, Rate 2/3, Rate 1/2, Rate 1/3 and Rate 1/4 of 33-partitioned formats for the packet loss rates from 0 to 12% with RCPC

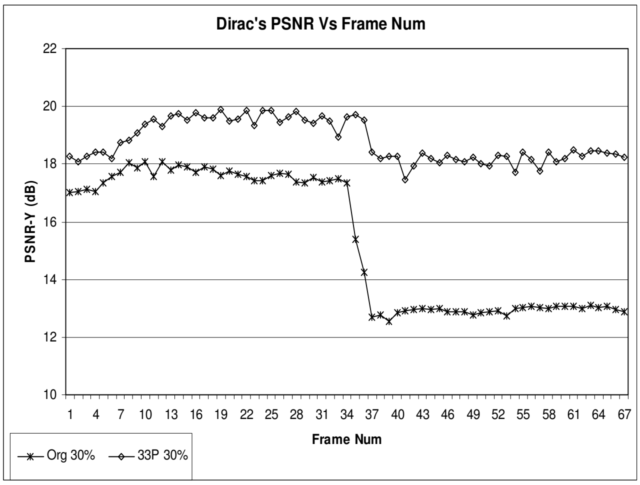


Figure 14. PSNR Performance comparisons between 33-partitioned and un-partitioned (Original) formats for 30% packet loss with rate 1/2 Turbo coding

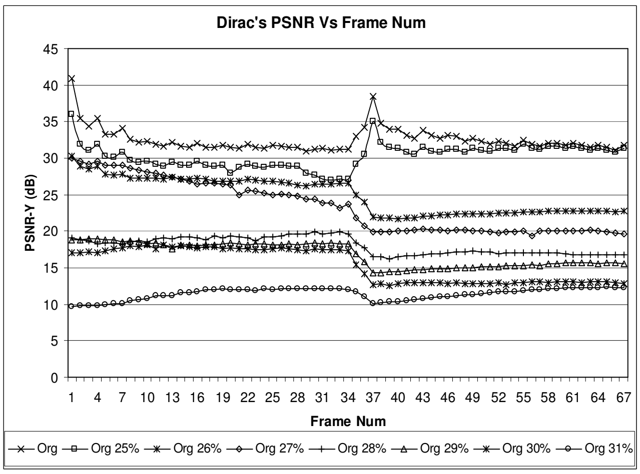


Figure 12. PSNR Performance comparisons between different percentages of packet error for un-partitioned format with rate 1/2 Turbo coding

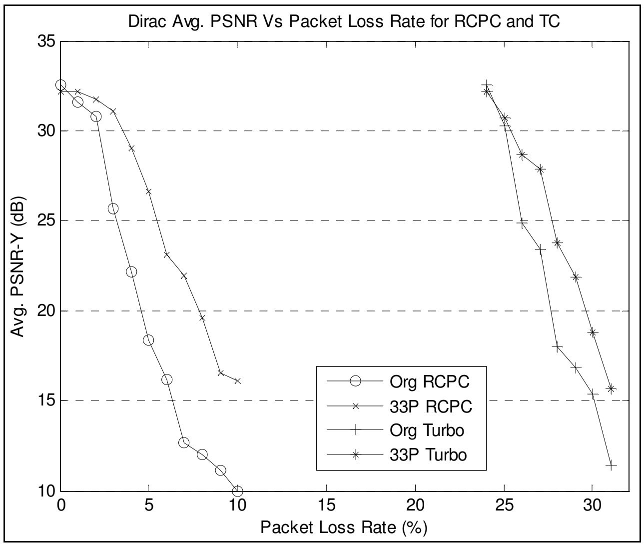


Figure 15. Average PSNR Performance comparisons between 33-partitioned and un-partitioned (Original) formats with Rate 1/2 RCPC coding and Turbo coding

Figure 8 and 9 show the PSNR performance comparison between different percentage of packet errors for both un-partitioned and 33 partitioned formats with rate 1/2 RCPC coding. There is no bit error at the RCPC decoder output for 1% packet loss rate in both cases, i.e. with the use of bitwise interleaver at the encoder; the RCPC decoder can effectively correct the resulting error pattern at the output of the bitwise de-interleaver. From the figures, it can be seen that the PSNR curve for 2% packet loss case is just below the error free curve with a slight performance degrading which is the result of having a few bit errors in the received sequence after RCPC decoding.

Figure 10 shows the PSNR performance comparison between two partitioning formats with the same percentage of packet error. It is clear that the 33 partitioned format achieves at least 5 dB gains over the un-partitioned one when the percentage of packet loss is 6%.

Figure 11 shows the average PSNR performance comparison between un-coded, rate 2/3, rate 1/2, rate 1/3 and rate 1/4 of 33 partitioned format using RCPC coding. It is interesting to note that the performance of the rate 2/3 encoding case achieves a few dB gains over the un-coded one for the packet error loss rates of less than 4% and then cross over occurred after that. This is because the error correcting capability of the rate 2/3 RCPC decoder is relatively low and cannot correct the errors effectively when the packet error loss rate increases. At this point, because of the usage of bitwise interleaving in rate 2/3 case, packet errors are spread over the interleaving length making the PSNR performance even lower than the un-coded one. On the other hand, rate 1/2, 1/3 and 1/4 offer better error correcting capabilities and achieve much higher PSNR performance gain than un-coded one. The coding gain over un-coded case is around 4 dB, 17 dB and 20 dB for the rate 1/2, 1/3 and 1/4 respectively at the 10% packet error loss rate. From the same figure, it is clear that there are no losses in terms of PSNR performance in the rate 1/3 and 1/4 encoding and according to the simulation results, there are no bit errors at the output of the RCPC decoder for the packet error percentages from 1% to 6% at these encoding rates. So, it is safe to conclude that encoder rates 1/3 and 1/4 are able to be used to protect the header layer, layer 1 of the Dirac's compressed bitstream for the lower packet error loss rates less than 6%.

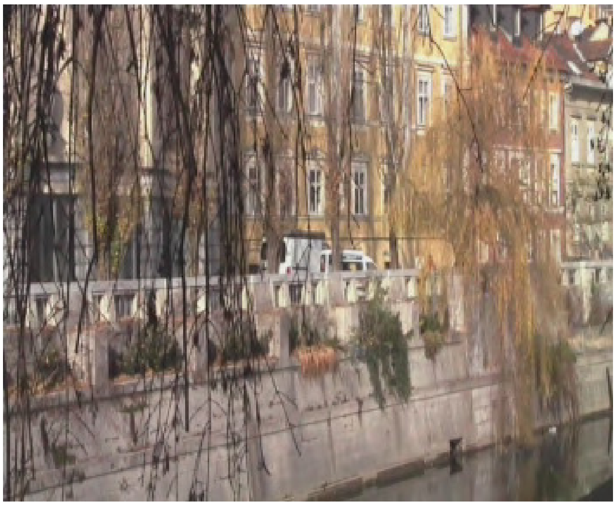
Figure 12 and 13 show the PSNR performance comparison between different percentage of packet errors for both un-partitioned and 33 partitioned format using rate 1/2 Turbo coding. Clearly, with the use of Turbo coding, the packet error percentage of more than 30% can be introduced in both cases. There is no bit error at the decoder output for the packet error less than 25%. This shows that the combined effect of the bitwise interleaver and channel encoder is much more efficient with the use of powerful channel coding mechanism.

Figure 14 shows the PSNR performance comparison between two partitioning formats with the same percentage of packet error. It is clear that the proposed 33 partitioned format also achieve at least 5 dB gains over the un-partitioned format in Turbo coding case as well.

Figure 15 shows the average PSNR performance comparison between the un-partitioned and 33 partitioned formats for the packet loss rate from 1% to 31% with rate 1/2 RCPC and the rate 1/2 Turbo coding. From this figure, it can be seen that the performance gain of RCPC tends to increase gradually starting from the 2% packet loss rate and the maximum performance gain achieved for 33 partitioned format is 8 dB at the 8% packet loss rate. On the other hand, it can be seen that the maximum performance gain of Turbo coding is approximately 6 dB over the un-partitioned format at the percentage of packet loss around 28%. A better error resilient performance can be achieved if the higher number of partitions is used at the expense of lower compression efficiency. But in both channel coding types, the average partition gain is approximately 5dB over un-partitioned format, which can be seen clearly in Figure 10 and 14 for RCPC and Turbo coding respectively. As expected, the Turbo coding can protect the transmitted packet sequence much more then RCPC with the expense of relatively higher decoding complexity and iteration delay at the receiver.

Figure 16 (a) - (d) shows the frame number 37 (I frame) of un-partitioned and the 33 partitioned format for 2% and 6% packet loss with rate 1/2 RCPC. The corresponding PSNR values for Figure 16, (a) to (d) are 38.45, 38.34, 19.45 and 22.22 dB respectively. A vertical black patch at the lower left corner of the Figure 16 (d) is the result of the loss of the whole partition. This happened when the bit error occurred in the lowest subband (DC suband) of a particular sub-frame or partition since it is required to discard the whole remaining bitstream of this sub-frame starting from the error location. It is occurred only in the *I* frame coding case since there is no reference frame in order to compensate this error.

Figure 17 (a) and (b) show the frame number 20 (inter frame) of un-partitioned and the 33 partitioned format with 30% packet loss using rate 1/2 Turbo coding. The corresponding PSNR values for the Figure 17 (a) and (b) are 18.92 and 24.15 dB, respectively. Being inter frame, Figure 17 (b) does not show the black patch like in Figure 16 (d) instead showing degraded subjective quality in the area where the corresponding partition is suffered from serious channel error. It is because, for the inter frame coding in the proposed method, the corrupted data can still be replaced with the exact replica pointed by the motion vector in the reference frame. In this case, the quality of reconstructed frame at the corrupted area mainly depends upon the accuracy of the motion estimation at this particular location and the quality of the reference frame. Since the motion compensated residual data is completely lost, decoder has to rely only on the data from the reference frame and motion vector data in order to reconstruct the corrupted area.



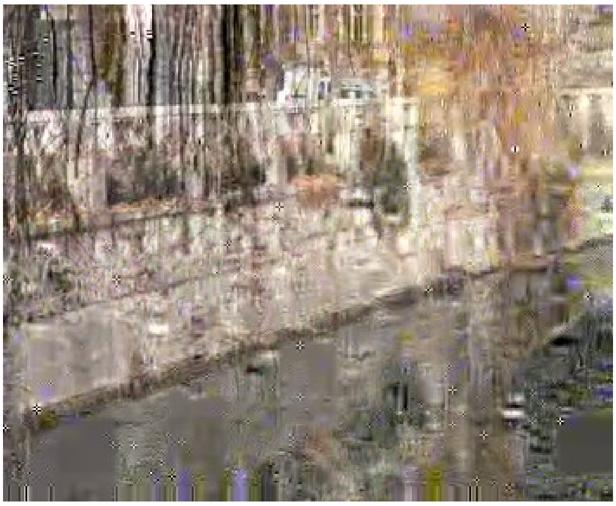
(a) Un-partitioned format, 2% packet loss rate

(b) 33-partitioned format, 2% packet loss rate

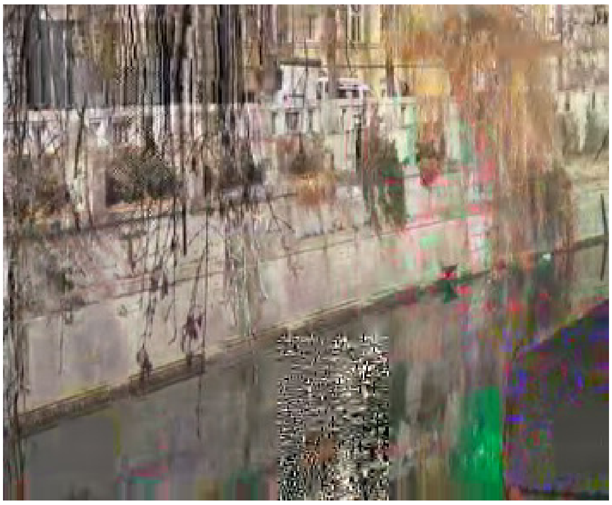
(c) Un-partitioning format, 6% packet loss rate

(d) 33-partitioned format, 6% packet loss rate

Figure 16. 4:2:0, CIF format canal vertical pan street sequence (frame 37, I frame) with rate 1/2 RCPC



(a) Un-partitioned format, 30% packet loss rate



(b) 33-partitioned format, 30% packet loss rate

Figure 17. 4:2:0, CIF format canal vertical pan street sequence (frame 20) with rate 1/2 Turbo Coding

From the figures, it can be seen that the proposed method offers very good reconstructions for all packet loss rates compared to the un-partitioned format especially for the 6% packet loss in the Figure 16 (c) and (d) where the proposed technique provides an excellent protection while the reconstructed quality of the original format is completely corrupted even though the PSNR difference is only 2.77 dB. The reconstruction quality of the figure 17 (b) is also very much better than 17 (a) in Turbo coding.

VI. CONCLUSION

In this paper, a technique to achieve the error resilience transmission of the compressed bitstream of Dirac video encoder over the packet erasure wired network is proposed. With the proper extension of source coding method for image transmission from [8] with appropriate channel coding, the proposed technique could achieve a performance gain of at least 5 dB over the un-partitioned one for both RCPC and Turbo coding. A better error resilient performance can be achieved if the maximum possible number of partitions (which is 99 according to equation 1) is used, at the expense of lower compression efficiency. The chosen number of partitions, 33 is optimum in terms of compression efficiency and performance gain since it offers relatively better compression efficiency compared with maximum number of partitions and acceptable performance gain. Apart from that, the received video sequence also has a good reconstruction without having any serious degrading in subjective quality as shown in Figure 16 and 17. In terms of complexity, the additional processing load is only at the coefficient partitioning or de-multiplexing at the encoder and multiplexing at the decoder in source coding and the channel coding complexity comes from either RCPC or Turbo. As an overall, the process of the coefficient partitioning does not introduce much complexity to the encoder and the usage of relatively less complex forward error control channel coding mechanism, RCPC offers the whole error resilient coding process to be simple and effective way of combating the channel errors for a network which has lower packet loss rate less than 10%. On the other hand, the proposed method is also suitable for the congested network which has high level of packet loss rate which is up to 30% by employing more powerful channel coding method i.e. Turbo Code with the expenses of relatively higher level of complexity and decoding delay time at the decoder. However the Turbo decoding complexity and iteration delay can be greatly reduced by employing the state of the art hardware technology which is available now a day so that it would be possible for the real time decoding. Moreover, the performance of the proposed method can be effectively increased by applying the various types of error concealment techniques at the decoder. Therefore, the proposed method of error resilient coding could be a suitable solution for transmission of wavelet transform based video codec's compressed bitstream over the packet erasure wired network. Moreover, the proposed method is essential tool for Dirac video codec in transporting the larger volume of compressed video files to the end users. Broadcasters can also offer highly competitive and attractive package to their customers because of the open source nature of Dirac.

APPENDIX

Generator Matrix for Convolutional Encoder

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Puncturing Matrices used for Puncturing the output of the Convolutional Encoder

Rate 2/3

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Rate 1/2

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Rate 1/3

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Rate 1/4

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

ACKNOWLEDGMENT

The authors gratefully acknowledge the joint financial support from BBC R&D and Brunel University, UK. We would also like to thank Dr Tim Borer, Dr Thomas Davies and Dr Peter Bleackley from BBC R&D for their insightful and constructive comments on this work.

REFERENCES

[1] H.264 Video Coding License Fees. [Online]. Available: http://www.vialicensing.com/products/AVCH264VC/license.terms.html
[2] BBC Research & Development, "Dirac Video Codec". [Online]. Available: http://dirac.sourceforge.net/
[3] J. Zheng, and L. P. Chau, "Multiple Description Coding using Multiple Reference frame for robust Video Transmission," *IEEE Int. Symp. Circuits and Systems*, *ISCAS* 2005, vol. 4, pp. 4006 – 4009, May 2005.

[4] C. M. Chen, Y.C. Chen, and C. M. Chen, “Multiple Description Motion Compensation Video Coding for MPEG-4 FGS over Lossy Packet Networks,” *IEEE Int. Conf. Multimedia and Expo, ICME*2004, vol. 2, pp. 1143 – 1146, June 2004.

[5] M. Schaar, and D. S. Turaga, “Multiple Description Scalable Coding using Wavelet-Based Motion Compensated Temporal Filtering,” *IEEE Int. Conf. Image Processing*, *ICIP*2003, vol. 2, pp. 489-92, Sept. 2003.

[6] P. G. Sherwood and K. Zeger, “Progressive image coding for noisy channels,” *IEEE Signal Processing Lett*., vol 4, pp. 189-191, July 1997.

[7] W. R. Heinzelman, and M. B. R. Talluri, “Unequal Error Protection of MPEG-4 Compressed Video,” *Int. Conf. Image Processing*, *ICIP*99, vol. 2, pp. 530-534, Oct. 1999.

[8] C. D. Creusere, “A new method of robust image compression based on the embedded zerotree wavelet algorithm,” *IEEE Trans. Image Processing*, vol. 6, pp. 1436-1442, Oct. 1997.

[9] W. Tan, and A. Zakhor, “Resilient Compression of Video for Transmission over the Internet,” 32nd *Asilomar Conf. Signals, Systems & Computers*, vol. 1, pp. 243 - 247, Nov. 1998

[10] N. V. Boulgouris, N. Thomos, and M. G. Strintzis, “Transmission of Images Over Noisy Channels Using Error-Resilient Wavelet Coding and Forward Error Correction,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 12, pp.1170-1181, Dec. 2003.

[11] J. Kim, R. M. Mersereau, and Y. Altunbasak, “Error-Resilient Image and Video Transmission Over the Internet Using Unequal Error Protection,” *IEEE Trans. Image Processing*, vol. 12, no. 2, pp.121-131, Feb. 2003.

[12] S. Cho and W. A. Pearlman, “A Full-Featured, Error-Resilient, Scalable Wavelet Video Codec Based on the Set Partitioning in Hierarchical Trees (SPIHT) Algorithm,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 157-171, March 2002.

[13] B.-J. Kim, Z. Xiong, W. A. Pearlman, and Y. S. Kim, “Progressive video coding for noisy channels,” *J. Vis. Commun. Image Repres*., vol. 10, pp. 173-185, 1999.

[14] Z. Xiong, B.-J. Kim, and W. A. Pearlman, “Progressive video coding for noisy channels,” in Proc. *IEEE Int. Conf. Image Processing* (*ICIP* ‘98), vol. 1, Oct. 1998, pp. 334-337.

[15] J. Zheng and L. P. Chau, “Error-Resilient Coding of H.264 Based on Periodic Macroblock,” *IEEE Transactions on Broadcasting* : Accepted for future publication, vol. , pp. , January 2006.

[16] J. Hagenauer, “Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications,” *IEEE Trans. Commun*., vol. 36, pp. 389-400, April 1988.

[17] C. Berrou, A. Glavieux, and P. Thitimasjshima, “Near Shannon limit error correcting coding and decoding: Turbo codes” *in proc IEEE Int. Conf. on Commun.* Geneva, Switzerland, May 1993, pp 1064-1070.

Myo Tun received his B.Eng. in Electronics from Yangon Institute of Technology, Yangon, Myanmar and M.Eng. in Telecommunication from Asian Institute of Technology, Bangkok, Thailand in 1996 and 2000 respectively. He worked as a laboratory supervisor and research associate at Telecommunication Program, Asian Institute of Technology from 2001 to 2005 in the field of wireless and multimedia communication. Currently, he is a PhD candidate in the School of Engineering and Design, Brunel University, UK. He is involved in a joint project between Brunel University and British Broadcasting Corp (BBC) on the Dirac research and development. His research interests include video compression and transmission, error control coding, multi-carrier CDMA and wireless communication.

Kok-Keong Loo (M’01) a.k.a. Jonathan Loo received his MSc degree (Distinction) in Electronics at University of Hertfordshire, UK in 1998 and PhD degree in Electronics and Communication at the same university in 2003. After completing his PhD, he works as a lecturer in multimedia communications at Brunel University, UK. He is also a course director for MSc Digital Signal Processing. Besides that, he currently serves as principle investigator for a joint project between Brunel University and British Broadcasting Corp (BBC) on the Dirac research and development. He also serves as research engineer for the IST-FP6 PLUTO project. His current research interests include visual media processing and transmission, digital/wireless signal processing, software defined radio, and digital video broadcasting and networks..

John Cosmas (M’86) received his BEng in Electronic Engineering at Liverpool University, UK in 1978 and a PhD in Image Processing at Imperial College, University of London, UK in 1986. He worked for five years in industry first with Tube Investments and then with Fairchild Camera and Instruments. After completing his PhD, he worked for 13 years as a lecturer in digital systems and telecommunications at Queen Mary College, University of London. Since 1999, he has worked for Brunel University, first as a reader and then in 2002 as a Professor of multimedia systems. His current research interests are concerned with multimedia broadcast communications systems, which evolved from his longstanding interests in video codecs (MPEG-1/2/4) and mobile communication systems.

Prof. Cosmas currently serves as Associate Editor for IEEE Trans. Broadcasting. He also contributes to the Digital TV Group’s “Mobile Applications” sub-group and to the Digital Video Broadcast’s Technical Module: Converged Broadcast and Mobile Services (CBMS).