Intra Frame Relay in ECC Video

Bing Du, Anthony Maeder and Miles Moody
School of Electrical and Electronic Systems Engineering
Queensland University of Technology, Brisbane, Australia

Abstract

A new error resilient video coding tool has been proposed [4], which uses ECC (Error Correction Code) to encode further a bitstream generated by a video coding standard such as MPEG series or H.26x series. Simulation results show a significant improvement over the current error resilient video coding tools in MPEG-4. However, ECC video still has some disadvantages: if a single error bit within a video frame escapes protection with ECC, the quality of subsequent frames will suffer greatly. IFR (intra frame relay) is proposed here to overcome this problem by using upstream messaging from the decoder. In an upstream message the first corrupted macroblock number of an I (Intra) frame is transmitted to the encoder, so that in the next frame all the macroblocks associated with the corrupted macroblocks are encoded in Intra mode to provide the decoder with sound reference frames for reconstructing subsequent frames. Simulation results show a significant improvement over the basic ECC scheme.

Keywords: video coding, error correction codes, mobile communications

1 Introduction

Diverse error resilience and error concealment techniques have been developed to cope with residue errors [5]. Residue errors are the errors delivered to the application layer by the transmission system of a network after first error control (conventional error control) takes place in the data link layer, while error resilience refers to operations to remove residue errors at the application layer. To maintain interoperability, error resilience tools compatible with current video coding standards are desirable. In the MPEG-4 [1] standard, five error resilience coding tools have been incorporated. The key technique among these error resilience tools is resynchronization, in which a compressed video bitstream is packetized by inserting resynchronisation markers in the bitstream. This lets the decoder regain synchronization after an error occurs in the bitstream by looking for another resynchronisation point, thereby limiting the error effect to one packet. With packetization combined with RVLC (reversible variable length coding) and DP (data partitioning), the decoder can recover some data within the packet which would otherwise be discarded.

Packetization for error resilience at the application layer simply means that resynchronization markers are inserted periodically in a video bitstream. Packet size (for error resilience) usually refers to the number of bits between two resynchronization markers. Packet size in a bitstream may vary slightly from packet to packet, as the start and end of a packet needs to be aligned with the start and end of a macroblock. NEWPRED is another error resilience tool in the MPEG-4 standard, in which a reference picture for inter-frame coding is replaced adaptively according to upstream messages from the decoder.

While resynchronisation, DP and RVLC do give protection to an encoded bitstream, they have their limitations. First, they are passive in the sense that the error bits in a video bitstream cannot be corrected before final video decoding. Instead the packets containing error are discarded, resulting in loss of information. The lost information is unrecoverable, so with the inter-frame error propagation effect, the reconstructed video quality rapidly declines to unrecognizable if no other measure is taken. Second, while bringing error resilience, they also introduce vulnerability. If an error happens to be within a marker (including resynchronisation marker, DC marker or motion marker), the decoder will lose synchronization, meaning one or more packets must be discarded. Finally, there is an associated loss of coding efficiency with this scheme. In our simulation with video sequences Salesman and Akiyo, with a packet size of 600 bits, resynchronization combining DP and RVLC results in a bit rate increase of more than 9.9% in the final bitstream. It may be argued that employing only resynchronization without DP and RVLC can reduce the coding rate. However, this combination is really necessary to exploit the potential of resynchronization.

To address the passiveness and disadvantages of current error resilience tools, ECC video [4] has been proposed. Basically in ECC video, a video bitstream encoded using a current video coding standard (including all MPEG series and H.26x series) is not protected by resynchronization: instead it is further encoded using ECC. This is an active error protection approach in the sense that it can recover a corrupted bitstream by correcting the errors in the bitstream. The ECC in our proposal is achieved with a punctured convolutional code [2,3]. Due to the very effective error correction capability a punctured convolutional code can achieve, the proposed scheme shows significant improvement over the packetization approach in the current MPEG-4 standard, in terms of reconstructed video quality and coding efficiency.

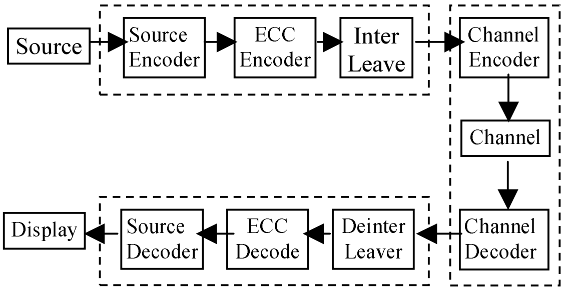

Figure 1: Video Transmission System employing ECC

A video transmission system employing ECC is shown in figure 1. The interleaving operation in the system is to cope with bursty errors and packet loss. Figure 1 shows that ECC with interleaving used as an error resilience tool is implemented at the application layer as part of source coding; therefore it is not a channel coding scheme, even though a punctured convolutional code is commonly used at the data link layer as a forward error control scheme in practice. Both DP and RVLC can still be used with ECC video. If the case when a whole frame contains only one packet, the motion vectors and other important information will be in the first part of the packet, while the texture coefficients will be in the second part of the packet.

However ECC video has its disadvantages. The only synchronization point in an ECC video bitstream is at the Picture Start Code. If a single error bit within a frame escapes protection with ECC (though this is very rare if the error correction code is properly designed according to residue error conditions), the decoder will lose synchronization with the encoder. Consequently the macroblock within which the error occurs and all the following macroblocks within the frame, will be undecodable because the decoder is not able to locate the next codeword due to the use of VLC. The quality of the reconstructed frame and all subsequent P frames will suffer greatly until the next I frame, due to inter-frame error propagation effects. Note that the NEWPRED will not work when the base I frames collapse. A new error resilience tool, IFR, is proposed in this work to solve this problem. Simulation results show a significant improvement for this approach over the basic ECC scheme.

2 ECC with IFR

In the IFR scheme, when transmitting an I frame, the starting number of the corrupted macroblocks due to a residue error is transmitted to the encoder by the decoder through a back channel message. The encoder then knows that the picture area in the I frame from the starting number to the end of the frame has not been successfully decoded. All the macroblocks associated with the corrupted macroblocks in the next frame can then be encoded in Intra mode, so as to ensure that all the subsequent frames will have decent reference frames. There are two reasons to employ the proposed scheme only for I frames. First, an I frame is the most important frame for decoding. If an error happens within an I frame, all the subsequent P (predicted) frames will be affected due to inter-frame error propagation. Second, to encode macroblocks in a P frame in Intra mode can reduce the coding efficiency significantly. If an error happens within a P frame, using NEWPRED will be more effective.

It should be noted that the encoder not only needs to encode the macroblocks starting from the starting macroblock to the end of the frame in the first P frame following the I frame in Intra mode, it also needs to encode those macroblocks in Intra mode which may use part or all of the corrupted macroblocks as references for motion estimation. For instance, if the maximum search range for motion estimation is 16 pixels, the Intra mode should start from the macroblock immediately above and left of the starting macroblock in the P frame following the I frame. Transmission delay effects may need to be considered. For instance, when the back channel message arrives at the encoder, the encoder may already be encoding the second P frame following the I frame. In this case the encoder should start to encode in Intra mode from the macroblock two rows above and two columns left of the starting macroblock transmitted to the encoder by the decoder in the second P frame, if the maximum search range for motion estimation is 16 pixels.

To make the proposal realistic, a video decoder must have some capability to detect errors in a bitstream after ECC operations. The error detection process used in this work is based on the following mechanisms. During the decoding process, if one of the events below occurs, the bitstream will become undecodable and the decoder will know that errors are present.

- Invalid VLC (MCBPC, CBPY, MVD, and TCOEF) code is detected.
- Quantizing information goes out of range.
- Invalid INTRA DC code is detected.
- Escaped TCOEF with level 0 is detected.
- Coefficient overrun occurred.

- Motion vector refers out of picture or maximum search range (for P frame error detection).

The decoder then starts some error concealment for the rest of the macroblocks within the frame and sends back the starting number of the broken macroblocks to the encoder. However, errors can occur in a way that the bitstream is still decodable even though the bitstream contains errors. In this case, error detection is conducted after the decoding process using redundancy information inherent in the neighboring macroblocks. More detailed discussion on error detection after decoding can be found in [7].

If DP and RVLC are employed, both the first and last corrupted macroblock numbers can be transmitted to the encoder. There will then be fewer corrupted macroblocks compared with not employing RVLC and DP, so the coding efficiency will be improved while coding the next frame and subsequent frames, as there will be less macroblocks associated with corrupted macroblocks needing to be encoded in Intra mode. Another reason for using DP and RVLC in ECC video is that when motion information is available, it is easier to conduct error concealment. However, to employ RVLC in ECC video reduces the coding efficiency. With an increase in bits equivalent to the increase for using RVLC, the power of ECC can be further increased. The choice of technique needs to be based on several factors including residue error conditions, video content, networking protocols, etc.

3 Simulation Results

To evaluate the effectiveness of the proposed IFR approach, two video sequences, Akiyo with relatively slow motion and Salesman with fast movement, are chosen as the test sequences. The goal is to compare the PSNR of ECC video with and without IFR. The tests are conducted based on the following conditions.

i) 50 frames of each video sequences are encoded with the first frame coded as I frame followed by P frames without rate control.

ii) When ECC is employed, the ½ rate base convolutional code (561, 752) is chosen which has a constraint length of $K = 9$. This base code is punctured to rate 13/14, which means that for every 13 bits in the compressed video bitstream, another bit is added after convolutional encoding. After transmission through a noisy channel, the convolution encoded bitstream is decoded using the hard decision Viterbi decoding algorithm with trellis depth of $17\text{x}K$.

iii) Packet size of both video sequences is set to 600 bits when resynchronization is used. DP and RVLC are employed with resynchronization, while are not employed in ECC scheme.

iv) Same quantization parameters are used in all experiments, which means that correctly decoded bitstreams protected using ECC and ECC plus IFR should have same visual quality on same video sequence in error free environments.

v) In each test, the residue errors are simulated by random errors with Gaussian distribution with Bit Error Rate (BER) set to $1\text{x}10^{-4}$. Back channel messages are transmitted error free.

vi) After the corrupted bitstreams are decoded, the erroneous motion vectors and texture information are replaced by 0. This means that when the motion vectors are not available, motion compensation is implemented by using the motion vectors in the same position in the previous frame, and when the texture information is not available, the blocks in question are reconstructed using the texture information in the blocks pointed at by the motion vectors.

The parameters used to generate the simulation results are designated as follows. By ECC(13/14) we mean the ECC scheme is used at the application layer with ECC rate set to 13/14, and by Pack(600) we mean the resynchronization scheme is used with packet size set to 600 bits. The final results, obtained by averaging results from 100 individual tests, are shown in figures 2 and 3. The number of bits used for coding each frame of the video sequences are shown in tables 1 and 2. The advantage of using ECC instead of resynchronization is clearly seen. Encoding each frame of both Akiyo and Salesman with ECC uses fewer bits on average than with packetization

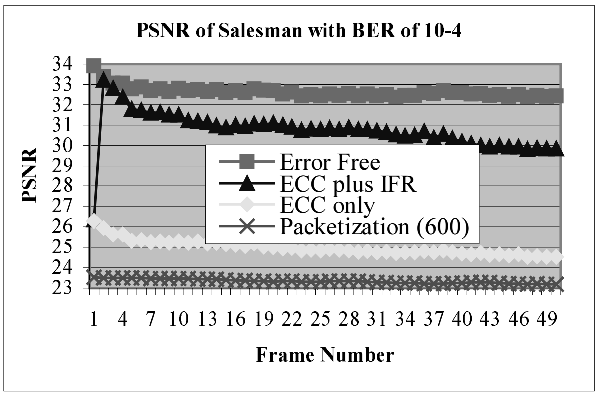

Figure 2: PSNR for Salesman at BER of $1\text{x}10^{-4}$

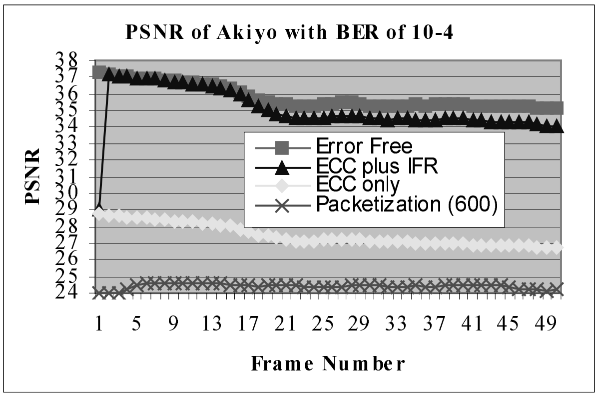

Figure 3: PSNR for Akiyo at BER of $1\text{x}10^{-4}$

(4806.72 and 4959.52 respectively for Akiyo; 11543.84 and 11768.48 respectively for Salesman), while the ECC method has a PSNR gain around 3 dB for Salesman and 4 dB for Akiyo over the resynchronization scheme.

The improvement using IFR over ECC alone is also clearly shown. The quality of the first frame (I frame) for both sequences is not good, due to the existence of residue errors, but the qualities of all the subsequent P frames are lifted when IFR is employed with PSNR gain of about 7 dB for Salesman and 9 dB for Akiyo, while the ECC video output quality without employing IFR remains relatively low. It should be noted that IFR can only be effective when employed together with ECC. The reason is that the packetization approach and associated RVLC and DP in the current video coding standards are passive and they do not have the capability to correct error bits in a video bitstream.

4 IFR Delay Analysis

One obvious problem with IFR is that the data rate for the first P frame following an Intra frame will be increased due to the employment of IFR, compared with not using IFR. From table 1, it can be seen quantitatively that the employment of IFR results in the number of bits for the P frame following the first Intra frame of the video sequence Akiyo increasing from 520 to 7315, when ECC(13/14) is used in the residue error condition, where the BER of the bitstream is 10^{-4}. However, the IFR data rate of the first P frame is still less than the peak date rate of all P frames among the 50 frames, which occurs in frames 18 and 19. So IFR does not pose any special difficulty for transmitting this sequence, if the transmission channel for P frames is allocated according to the peak data rate of the P frames for the particular sequence.

For the video sequence Salesman, the employment of IFR results in the number of bits for the first P frame following the I frame being increased from 8488 to 27227 as shown in table 2, which is less than twice the peak rate of all P frames and introduces a transmission delay of about one frame. One solution is to drop the following P frame, i.e. only one P frame (the first one) is transmitted instead of two if the channel allocation is fixed. Another solution is to modify the transmission protocol to allow more channel capacity for the first P frame following an I frame. This can be implemented easily because of the periodic structure of the encoded video bitstream. We can treat the first P frame as a “half I frame” and if we can periodically update the channel allocation for I frames, it is easy to cater for a periodic “half I frame” following an I frame.

The bit rate increase of the first P frame mentioned above is the average of the results of the 100 tests. More generally, the transmission delay to the P frame following an Intra frame caused by the employment of IFR depends on the following conditions. First, the residue error condition has a significant influence on the data rate increase on the first P frame following an I frame, if the ECC rate does not match the residue error conditions. When the residue error condition is good, the delay is small. Otherwise if the residue error condition is poor, the delay will be lengthy. Second, the content of the video sequence is also an important factor if the ECC rate does not match the residue error conditions completely. The more complex the content is, the more bits it will produce after compression; consequently, there are more chances that the bitstream get corrupted by residue errors and more likely the data rate of the P frame following an I frame is increased. However, if the ECC rate matches the residue error conditions perfectly, the ECC operations can correct all the transmission errors, so the content of video will not have any influence on the data rate of the first P frame following an I frame. In such cases the employment of IFR does not result in any data rate increase because it has no chance to function, as all the residue errors are corrected by ECC.

Obviously increasing ECC capability will reduce the data rate of the first P frame following an I frame. Either increasing the ECC rate or using the soft-decision Viterbi decoding algorithm instead of the hard-decision Viterbi decoding algorithm to decode the punctured convolutional code can increase the ECC capability. If soft-decision Viterbi decoding is used, the network needs to deliver the soft-decision output to the application layer [6]. Using this approach to decode we have established that when ECC(9/10) is applied rather than ECC(13/14), the ECC operation corrects all the residue errors in the bitstreams of both sequences Akiyo and Salesman for all the 100 tests when BER of the residue errors is set to 10^{-4}, delivering transmission-error free reconstructed video output. Based on 100 tests, other experiments have revealed that ECC(7/8) corrects all the residue errors in the bitstreams of these two video sequences when the BER of the residue errors reaches 10^{-3}. Increasing the ECC rate will reduce coding efficiency. However, if DP and RVLC are not used with ECC, applying ECC(9/10) results in a final bit rate increase of 11.21%. Compared with a 9.9% overhead increase for packetization with DP and RVLC (with packet size set to 600 bits) in the final bitstream, the 1.31% overhead increase introduced by using ECC(9/10) instead of using Pack(600) is negligible taking into account what ECC(9/10) can achieve and Pack(600) cannot.

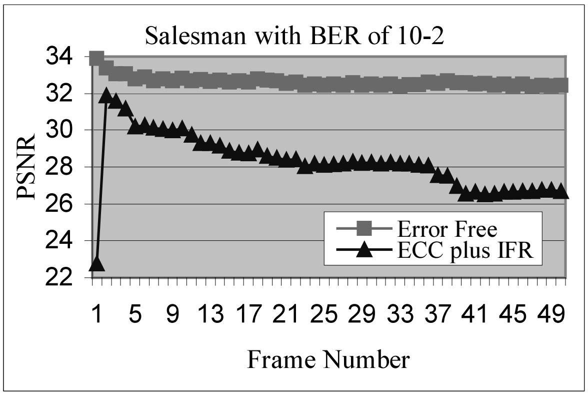

Figure 4: PSNR for Salesman at BER of 1x10^{-2}

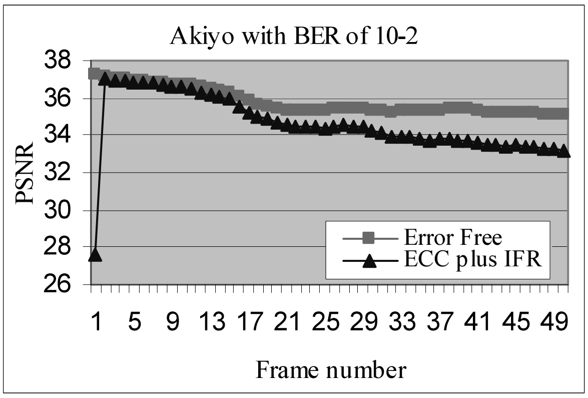

Figure 5: PSNR for Akiyo at BER of 1x10^{-2}

Another set of simulation results obtained by averaging results from 100 individual tests conducted in the residue errors with BER of the residue error set to 10^{-2} is shown in figures 4 and 5. This further demonstrates how effective ECC enhanced with IFR can be. Soft decision Viterbi decoding is used here and all the other experiment conditions are the same as listed in section 3, except that the BER of the residue errors is changed from 10^{-4} to 10^{-2}. In this simulation, ECC(7/8) is applied where the BER of the residue errors stays at 10^{-3} most of the time and occasionally drops to 10^{-2}. In these conditions, ECC(7/8) enhanced with IFR still delivers decent video for both sequences Salesman and Akiyo, while the bitstreams protected with resynchronization are undecodable no matter how long the packet size. ECC(7/8) increases the bit rate of the final bitstream by 14.29% for all frames except the first P frames after the I frames in ECC without DP and RVLC. However, it allows video transmission when BER of the residue errors drops to 10^{-2}, which would be impossible if resynchronization is used instead of the ECC scheme.

5 Conclusions and Future Work

Building on the ECC scheme [4] for residue errors, this work introduces a new scheme, IFR, using back channel messages to enhance the performance of the basic ECC scheme. Simulation results show that the IFR technique improves over the original ECC scheme significantly. To stop inter-frame error propagation caused by I frame errors, IFR is very effective for ECC video, and to stop it when caused by P frame errors, NEWPRED is an effective alternative. In a wired network, where the residue error conditions are stable, it is easy to design an ECC rate matching the residue error conditions; therefore the employment of IFR is not so important. In a wireless situation, where the residue error conditions vary, it happens frequently that some errors escape protection by ECC; therefore it is preferable to employ the IFR technique with ECC. Future work will consider design and implementation of adaptive ECC for video communication in mobile environments, where the ECC coding rate can follow the change of residue error conditions dynamically. More effective error detection techniques after ECC will improve performance of the ECC video with IFR. This work was supported by the Commonwealth of Australia via the Cooperative Research Centre scheme.

REFERENCES

[1] ISO/IEC 14496-2, “Information Technology – Coding of Audio-Visual Objects: Visual” (2001).

[2] Viterbi, A.J., “Convolutional Codes and Their Performance in Communication Systems”, *IEEE Trans. on Comm. Tech.*, COM-19(5), pp 751-772 (Oct 1971).

[3] Yasuda, Y., Kashiki, K., Hirata, Y., “High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding”, *IEEE Trans. on Comm.*, COM-32(3), pp 315-319 (Mar 1984).

[4] Du, B., Ghanbari, M., “ECC video in bursty channel errors and packet loss”, in *Proc. 12th Picture Coding Symp.*, Saint-Malo, pp 99-103 (Apr 2003).

[5] Wang, Y., Zhu, Q., “Error Control and Concealment for Video Communication: A Review”, *Proc. of the IEEE*, 86(5), pp 974-997 (May 1998).

[6] Hagenauer, J., Hoher, P., “A Viterbi algorithm with soft-decision output and its applications”, in *Proc. IEEE Global Telecommunications Conf.*, Dallas, pp 47.1.1-7 (Nov 1989).

[7] Khan E., Gunji, H., Lehmann, S., Ghanbari, M., “Error Detection and Correction in H.263 coded video over wireless network”, in *Proc. Packet Video Workshop 2002*, Pittsburgh (Apr 2002).

Table 1: Number of bits per frame for Akiyo

Frame No	ECC(13/14)	ECC&IFR(13/14)	Pack(600)
0	46192	46192	50352
1	520	7315	488
2	656	651	616
3	656	653	616
4	1104	1094	1112
5	1040	1043	1040
6	1096	1185	1088
7	1280	1268	1272
8	1520	1518	1552
9	1712	1715	1736
10	1608	1601	1632
11	1432	1431	1472
12	2040	1987	2024
13	1992	1961	1976
14	2848	2796	2912
15	4112	4084	4208
16	5688	5658	5840
17	6896	6876	7000
18	7656	7685	7896
19	7512	7493	7704
20	6744	6719	6880
21	6216	6179	6352
22	5152	5110	5232
23	4288	4277	4368
24	4240	4188	4312
25	3136	3151	3184
26	3056	3059	3104
27	3760	3755	3848
28	4752	4731	4816
29	5424	5429	5512
30	5928	5913	6048
31	5984	6007	6112
32	5096	5083	5176
33	4264	4295	4344
34	4600	4591	4688
35	4912	4887	5008
36	5144	5158	5256
37	5016	4976	5128
38	5216	5187	5272
39	5288	5243	5408
40	4768	4771	4840
41	4752	4716	4872
42	4240	4214	4328
43	3688	3688	3752
44	3736	3721	3784
45	4328	4311	4416
46	4696	4684	4760
47	4920	5935	5040
48	5008	5978	5096
49	4424	4402	4504

Table 2: Number of bits per frame for Salesman

Frame No	ECC(13/14)	ECC&IFR(13/14)	Pack(600)
0	76896	76896	81168
1	8488	27227	8744
2	13992	13579	14120
3	16112	16006	16256
4	15392	15392	15464
5	12376	12404	12568
6	8776	8720	8872
7	8376	8351	8520
8	9160	9071	9312
9	9152	9169	9328
10	10144	10105	10344
11	9360	9274	9480
12	8360	8418	8456
13	9304	9257	9496
14	9992	10023	10088
15	10064	10066	10080
16	8576	8507	8640
17	9752	9818	9920
18	14288	14077	14320
19	16528	16514	16624
20	14048	13918	14224
21	10576	10470	10904
22	10400	10342	10696
23	8928	8876	9032
24	6360	6296	6440
25	5648	5570	5712
26	6936	6868	7032
27	7880	7933	8024
28	6760	6770	6840
29	5032	4948	5080
30	5856	5822	5912
31	6744	6701	6848
32	7400	7484	7624
33	8168	8067	8416
34	10904	10683	11120
35	13696	13664	13848
36	15792	15714	15920
37	14040	14001	14088
38	14704	14756	14880
39	13208	13167	13416
40	10952	10927	11120
41	11200	11189	11328
42	10592	10594	10648
43	10080	10096	10248
44	10720	10695	10840
45	11344	11120	11600
46	9344	9345	9520
47	8848	8851	8976
48	8256	8172	8432
49	7688	7649	7856