Application Layer Error Correction Scheme for Video Header Protection on Wireless Network

Chia-Ho Pan, I-Hsien Lee[1], Sheng-Chieh Huang[2], Chih-Chi Cheng, Chung-Jr Lian, Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

[1]Computer & Communications Research Laboratories, Industrial Technology Research Institute, HsinChu, Taiwan

[2]Electrical and Control Engineering National Chiao-Tung University, HsinChu, Taiwan

Abstract

In wireless video streaming application, video information may be corrupted by a noisy channel. By introducing error resilience and error concealment techniques, many researchers have tried to eliminate quality degradation of reconstructed picture in decoding a corrupted data. On the contrary, there are relatively fewer works discussing the ways to diminish the corruption. Hence, system designers need to use different methods to restrict the error cause by channel within a tolerable extent. In other words, the system will be difficult to be implemented in practical design. In this paper, we propose a way to protect the video header information in application layer without modifying standardized syntax. Beside, we also consider channel condition of wireless transmission and propose a way to reduce redundant bits used in channel coding. By doing this, the bitstream can be simply transmitted over practical wireless network and the reconstructed picture quality outperforms the original one.

1. Introduction

Multimedia communication has already played an important role in our daily life. Amount them, providing a real-time video service over wireless network is one of the most demanding applications. It's well known that fading channel and multi-path effect may introduce different kinds of interference in wireless transmission and these noisy channels usually corrupt transmitted data and cause an unpredictable error. To ease this problem, Forward Error Correction (FEC) has been proposed. This scheme will embed some redundant bits in the transmission data, and decoder will use these information to correct the error caused by the channel interference. However, the correcting ability of FEC depends on the amount of redundancy embedded in the system. In other words, there will be a limitation of the correcting ability under the given redundant bitrate. Hence, transmitting information with FEC still cannot ensure the original information will not be corrupted by the noisy channel.

Unfortunately, the compressed video bitstream is very sensitive to errors. This is because the property of extracting video information into a small amount of bits. Every single bit in the video stream takes important information and losing it can make serious problem in reconstructing video. For example, video coding usually uses Variable Length Code (VLC) scheme to compress video information. If an error occurs at a VLC symbol, decoders will not correctly identify this symbol. This error may also make the following data unusable because of the dependency of these VLC symbols. It will be more serious if this single bit error occurs at the header of a bitstream. As we know, the header information takes the information that will be used in video decoding. Losing this information can terminate a regular decoding procedure. In order to solve these problems, many error-resilience and error-concealment methods have been proposed [1][2][3][4]. Nevertheless, these works focus on using spatial or temporal redundancy to conceal the error that corrupt macroblocks (MBs) instead of discussing the error occurred in header information.

Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM'05)
0-7695-2489-3/05 $20.00 © 2005 IEEE

It's clear that all headers and extensions of the higher syntactic layers, i.e. the picture layer, group of pictures and sequence layer, contain critical information for decoding. In the audio services of Digital Audio Broadcasting (DAB) [5], transmitter will use Cyclic Redundancy Checks (CRCs) to detect critical audio frame header information because rendering an audio frame with errors in the header is likely to result in audibly unpleasant effects. Hence, DAB decoder will ignore the frame if CRCs reports a potential corruption of its content. This method reveals the fact that protecting the header information of a multimedia bitstream is an essential need of applicable multimedia service. However, the characteristic of temporal prediction in video coding renders the scheme of CRCs not suitable to be used. Simply detecting a header error and discard the corrupted frame in video decoding can still make serious performance degradation. As a consequence, the ability of correcting header error is an important requirement and critical issue in practical video transmission service.

In this paper, we propose a method to protect the header information of video bitstream in application layer. To satisfy the urgent of providing real-time video service over wireless network, this method is dedicatedly designed for the environment of wireless system that we practically used today. Although many researchers have provided ways to solve this problem in different layers of Open Systems Interconnections (OSI), these works either require a special designed network procedure to support its idea or do not take wireless channel effect into account. On the contrary, our method can be applied in any networks or transmission environments since the effort that we made is confined in the application layer of OSI. Besides, the proposed method can reduce redundancy used in channel coding since a corrupted bitstream can still be corrected by our method. In our simulations, we will demonstrate the proposed method outperforms other methods under a limited bit-rate. Before describing the detail of our method, we will briefly introduce the ways that have been used in protecting video information.

2. Review of Header Protection Schemes

As we have mentioned above, every bit of a multimedia bitstream can represent different importance of a reconstructed video sequence. Depending on the importance of various parts of a multimedia bitstream, Unequal Error Protection (UEP) [6] techniques can adapt different amount of effort devoted to protect the sub-bitstreams to errors. This technique can be used in several different applications. For example, scalable video coding contains a base layer which provides a low but acceptable level of quality, and several additional enhancement layers to incrementally improve the video quality. UEP will assign a more reliable subchannel, a stronger FEC code, or allowing more retransmissions [7][8] to ensure the correction of base layer information. By doing this, we can expect the important part of the bitstream will be well protected. However, the existed network and protocol environments may not support UEP at the network level, and this situation is not likely to change in the near future, for both technical and economical reasons.

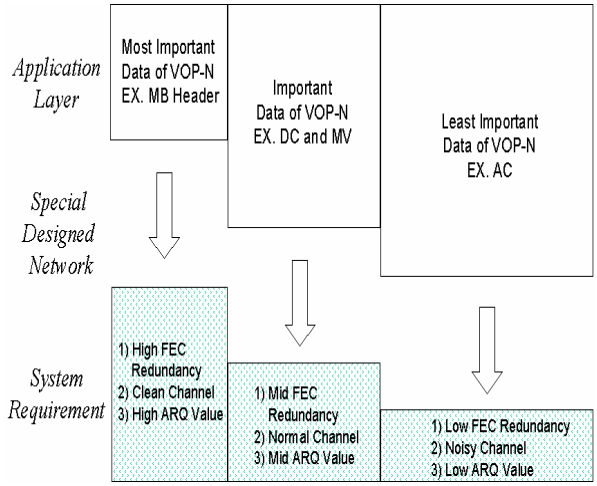

Figure 1. Unequal Channel Protection in Combination with Data Partition.

As we have mentioned in section 1, losing the header information of a bitstream can cause serious performance degradation in video transmission service. To solve this problem, Data partition technique [9] has been introduce in MPEG-4 and many following up video coding standards. In this technique, video bitstream is split into two partitions, the first containing coding mode information for each macroblock together with DC coefficients of INTRA block or motion vectors for INTER blocks. The remaining data are placed in the second partition. In combining with the technique of UEP, we can pay more effort on protecting the information of the first partition that is adequate to decode an acceptable video quality. As we can see in Figure 1, the combination of data partition and unequal error protection requires a more complex communication network to achieve its goal. This phenomenon raises many doubts over how effective data partitioning is in real streaming systems. It is argued that unequal error protection is either not easy to implement, or incur a large overhead in bit rate, or may not be effective at all because the required up-to-date channel information is not available.

In addition to the difficulty of implementing, data partition only pay attention to the important information of MB layer; the headers of higher syntactic layers are not transmitted in a highly protective way. In the existed standard, there is a way dedicated to protect the headers of higher syntactic layers. Header Extension Code (HEC) [9] allows important header information to be duplicated at slice level. Since a VOP consists of one or more video packets, the system attempts to start a packet with a slice synchronization point of the video bitstream by following up with a HEC. Hence, every coded packets carry, in HEC, information relevant to all slices of a frame. If the first packet is lost, it will still be possible to decode any of the slices of the frames by using the HEC information in other packets. However, this scheme is only applicable in some specific networks. Furthermore, it cannot be always applied in different video codecs. For example, H.263 contains no syntax element that allows including redundant picture header information at the slice level. This serious drawback requires to be compensated by accompanying protocols such as RFC2429 [10].

3. Header Protection Scheme in Application Layer

All methods that we mentioned in the section 2 try to protect the header information of a video bitstream. However these works are confined to use any other layers of OSI but application layer. Because applying a dedicated network procedure to achieve video service can be an all-consuming work, these solutions may not be effective ways in implementing a real-time video service. In this section, we will introduce a way to protect video header in application layer. Because our method will only embed slight redundant information in original information, system provider can process this protected data as a part of normal video bitstream. Nevertheless, video decoder can use the redundant information embedded in the bitstream to correct header information. As a consequence, the quality of reconstructed pictures will outperform that of the original one.

3.1. Error Correction in Application Layer

In the proposed method, we also combine the similar concepts mentioned in data partition and unequal error protection. Specifically, the most critical information of the video bitstream, header information, will be placed in a separated partition and error correction code will be used to protect it. However, the proposed process will be accomplished in the application layer instead of using a complex system. After the error correction coding, the redundancy of the error correction coding will be transmitted in combination with video information as shown in Figure 2. As we can see, this method only uses a single transport and physical channel that we practically used. In other words, it is no need to adjust the FEC coding mode, transmission power, or number of ARQ to increase correction probability of important information.

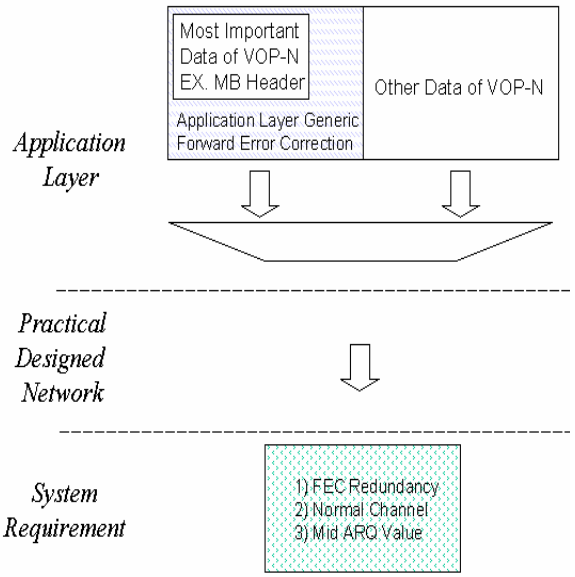

Figure 2. Proposed Application Layer Error Protection Method.

Since we only take the header information of a video bitstream into the first partition, the amount of redundancy will be limited into a small value. In our design, only 1 kbps additional bit-rate is required while we use Hamming Code as the error correction method in application layer. In addition, original header information will not be affected after error correction coding because of the characteristic of Hamming Code. These phenomena raises the possibility to embed this redundancy into the original bitstream. As we know, there already have many popular video codecs in the market. It is not reasonable to modify the syntax of the existed standards. Hence, embedding the redundancy into "user_data" if the video codec, such as MPEG-4, will support the proposed idea. Because header information will not be affected by the Hamming Code coding procedure, this information will be still put into the bitstream by following original syntax. That is to say, a decoder that does not support application layer error correction can still decode a bitstream, which is encoded by the proposed method.

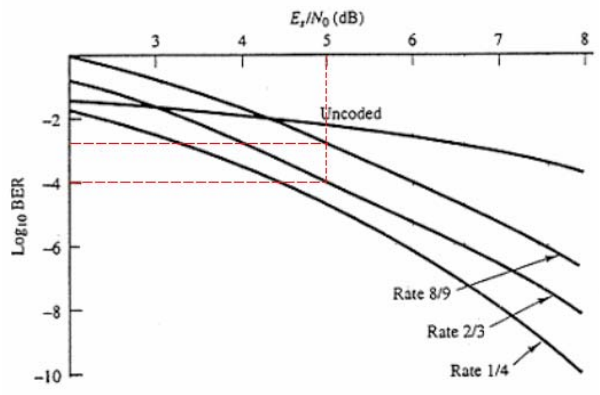

Figure 3. Redundant Code Rate Requirement of Channel Coding by using Convolution Code [11]

3.2. Transmission Rate Reduction

Although applying the proposed method will slightly increase the amount of data by 1%, the transmission rate will be reduced in wireless application. On system point of view, the information of application layer need to be further protected by FEC code. The amount of FEC redundancy, code rate of channel coding, depends on the Bit Error Rate (BER) that a video decoder can tolerate. Specifically, a looser BER requirement from application layer can result in less redundant bit used in channel coding. Since different information represents different importance in a video bitstream, the requirement to have a correct header information can determine the bound of BER value.

To demonstrate the advantage of the proposed idea, we operate the video decoder at the requirement of BER=10e-3. It's a relatively loose BER constrain in providing real-time video service. Under such a bad channel condition, header information will be frequently corrupted, hence, serious quality degradation will occur. Nevertheless, the proposed method can correct most of the corrupted header information while video information is transmitted in this condition. To facilitate the problem of corruption in header information if header information is not well protected, people generally operate the system at BER=10e-4. Figure 3 represents the code rate of channel coding that should be selected to satisfy a fixed BER. As we can see, while the signal to noised ratio is fixed at 5 dB, only 12.5% redundant information, rate 8/9, is required to meet BER=10e-3 requirement. However, code rate 2/3 will be select to ensure application layer will have a BER value of 10e-4. That means 50% redundant bits will be used in channel coding. Hence, the proposed method can also save total transmission bit rate in practical use.

Table 1. Average PSNR comparison of the reconstructed vide sequences with the header protection scheme has been applied and has not been applied.

dB / Sequence	PSNR(Y)	PSNR(U)	PSNR(V)
Foreman	36.11/27.72	40.18/38.44	40.98/38.88
Mobile	33.51/28.83	34.68/33.70	34.39/33.43
Mother and Daughter	37.41/35.55	41.38/41.27	42.16/41.99

4. Experimental Results

In this section, we will demonstrate the proposed methods by using two scenarios, which are practically used in real-time video service. The first scenario is often used in broadcasting service, that is to say, encoder do not have feedback information to adapt its encoding process. Hence, errors will propagate all the way until video decoder has a new INTRA block. Besides, we also provide the second scenario that a system has a feedback channel in video service to compare with our design. In this case, video encoder will be notified that a picture has been lost at decoding side due to a corruption of picture header. By doing this, video encoder will use proper information as reference data and the error propagation will be confined in limited frames.

In the following simulations, we use Hamming Code as the error correction code to protect header information of a H.264 video bitstream [12]. These header information include sequence layer header, picture layer header and slice layer header. The amount of redundant bit to protect these headers is about 1kbps. Errors are generated by using random noise at the BER of 10e-3 in the following experiments. Then, this error will be used to corrupt a header protected and not protected bitstream. Since both bitstreams have the same error probability, the error occurred in the bitstream except header information will be handled in the same way. On the other hand, when the error occurred in the header is detected, the following information belongs to this header will be dropped. The corrupted frame will be replaced by copying the information from the nearest frame. Also, the header information in first frame is excluded from errors. In order to evaluate the performance, we have considered 3 video sequences namely: Mother & Daughter(slow motion), Mobile(complex motion) and Foreman(fast motion). All of these sequences contain 300 frames and are coded as IPPPPP..., that is the first frame is intra-coded and all other frames are intercoded.

Figure 4. Illustrations of header protection effect at average PSNR when QP = 28. (a) Frame 91 of protected Foreman sequence. (b) Frame 91 of unprotected Foreman sequence. (c) Frame 190 of protected Mobile sequence. (d) Frame 190 of unprotected Mobile sequence. (e) Frame 110 of protected Mother and Daughter sequence. (f) Frame 110 of unprotected Mother and Daughter sequence.

Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM'05)
0-7695-2489-3/05 $20.00 © 2005 IEEE

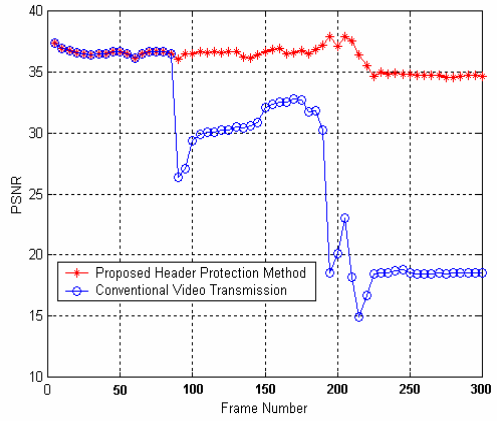

Figure 5. Frame by frame luminance PSNR performance comparison with header protected and header unprotected Foreman sequence at QP = 28.

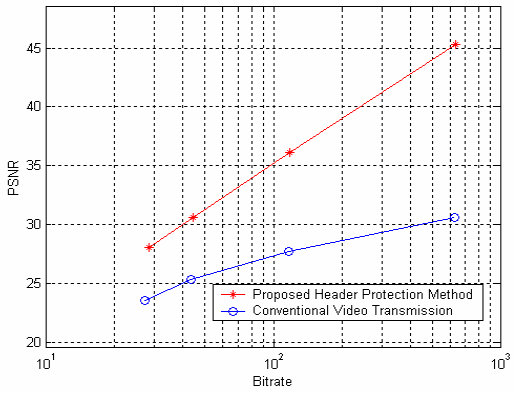

Figure 6. PSNR degradation caused by the corrupted higher layer header information for Foreman sequence at QP=16, 28, 36, and 40.

Table 1 shows the PSNR value of decoding both bitstream by using H.264 reference software JM9.6[13]. Using neighboring frame information due to a corrupted header will seriously corrupt reconstructed picture in fast motion sequence because there have quite a difference between two neighboring frames. As we can see in Figure 4, the motion feature of a video sequence also affects the amount of degradation in reconstructed frames. Fast motion video sequence, Foreman, obviously more sensitive to the lost of picture header and the reconstructed frames will be serious degraded in this case. Figure 5 shows frame by frame luminance PSNR value of decoding Foreman sequence. Although some INTRA blocks in the following frames will recover the corrupted picture to slightly improve PSNR performance, these INTRA blocks still cannot compensate the error caused by the corrupted header. This error will propagate to all the other following frames and further degrades PSNR performance. In order to evaluate the performance of the proposed method in different bitrate, we encode the Foreman sequence by using different Qp. Figure 6 is the average PSNR value by decoding these bitstreams under different bit-rates. As we can see in this figure, the proposed method can at least outperform the original one by 5dB. Furthermore, we can also observe that video quality improvement will not be proportional to the log scale of the encoding bit-rate if the higher header information of bitstream has been corrupted.

Table 2. Average PSNR of the reconstructed vide sequences by decoding a bitstream with header protection and a bit stream without header protection except an additional feedback channel.

dB / Sequence	PSNR(Y)	PSNR(U)	PSNR(V)
Foreman	36.11/35.77	40.18/40.03	40.98/40.85
Mobile	33.51/33.32	34.68/34.64	34.39/34.34
Mother and Daughter	37.41/37.35	41.38/41.40	42.16/42.16

For the system that has a feedback mechanism, the feedback channel can inform video encoder that particular information has been corrupted and should not be used as reference information. Here, we assume that this round-trip time of the system only takes a time interval of two frames. The following frames will choose a proper reference information to refer and the error propagation will be halted immediately. Table 2 shows the results of this scenario and our scheme. Although this scenario provides a more complex procedure in avoiding error propagation, the proposed method still has a better PSNR performance.

5. Conclusion

In this paper, we propose a way to protect the header information of video bitstream in application layer. In an error-prone channel, this scheme will prevent error occurred in header information so as to use its parameters in video decoding. Because we will not modify the syntax of the existed standard and the redundant bits can be embedded in the bitstream, this scheme can be used in combination with any video codecs. Besides, our method can also be applied in any networks or transmission environments since the effort that we made is confined in the application layer. From experimental results, the scheme proposed in this paper provides a better performance in comparing with the method without header protection and the scheme with a feedback channel. This makes the proposed method become one of the better solutions in transmitting video sequence in practical use.

References

[1] G.-S. Yu, M. M.-K. Liu, and M. W. Marcellin, “POCS-based error concealment for packet video using multi-frame overlap information,” IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 4, pp. 422–434, Aug. 1998.

[2] Y. J. Chung, J. W. Kim, and C.-C. J. Kuo, “Real-time streaming video with adaptive bandwidth control and DCT-based error concealment,” IEEE Trans. Circuits Syst. II, vol. 46, no. 7, pp. 951–956, Jul. 1999.

[3] H. Sun andW. Kwok, “Concealment of damaged block transform coded images using projections onto convex sets,” IEEE Trans. Image Process., vol. 4, no. 4, pp. 470–477, Apr. 1995.

[4] V. Parthasarathy, J.W. Modestino, and K. S. Vastola, “Design of a transport coding scheme for high-quality video over ATM networks,” IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 2, pp. 358–376, Apr. 1997.

[5] European Telecommunication Standard Institute, ETSI: Radio broadcast systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers, ETS 300 401, May 1997.

[6] M.G. Martini and M.Chiani, ” Proportional unequal error protection for MPEG-4 video transmission”, IEEE International Conference on Communications, 2001.

[7] K. N. Ngan and C. W. Yap, “Combined source-channel video coding,” in Signal Recovery Techniques for Image and Video Compression and Transmission, A. K. Katsaggelos and N. P. Galatsanos, editors, ch. 9, pp. 269-297, Kluwer Academic Publishers, 1998

[8] L. Kondi, F. Ishtiaq, and A. K. Katsaggelos, “Joint source-channel coding for scalable video,” Proc.2000 SPIE Conf. On Visual Communications and Image Processing, San Jose, CA, Jan. 2000.

[9] MPEG-4 Video Group, "MPEG-4 Video Verification Model Version 18.0", ISO/IEC JTC1/SC29/WG11 N3908, January 2001, Pisa.

[10] RFC2429: C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullivan, S. Wenger, and C. Zhu: "RTP payload format for the 1998 version of ITU-T Rec. H.263 video (H.263+)", RFC2429, May 1999.

[11] Stephen B. Wicker, “Error Control Systems for Digital Communication and Storage” pp. 330, 1995, Prentice Hall International, Inc.

[12] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), 2003.

[13] http://iphome.hhi.de/suehring/tml/download/, H.264/AVC Reference Software version 9.6.