ERROR RESILIENCE METHODS FOR FGS VIDEO ENHANCEMENT BITSTREAM

Rong Yan[1], Feng Wu[2], Shipeng Li[2], Ran Tao[1]

[1] Beijing Institute of Technology, #7 Baiyi RD, Haidian, Beijing, 100080, China
[2] Microsoft Research China, #49 ZhiChun RD, Haidian, Beijing, 100080, China

ABSTRACT

Fine Granularity Scalable (FGS) video coding scheme that has been recently adopted in the MPEG-4 streaming video profile provides a layered and fine granularity scalable bitstream with different importance at different layers. This property makes FGS bitstream suitable for transmitting over the error-prone channels with unequal error protection. However, there are no tools currently available for error resilience in the FGS enhancement bitstream. This causes two problems when delivering the enhancement bitstream through an error-prone channel. First, there is only a weak error detection capability in the enhancement layer. The decoder would put a lot of wrong data into the decoded video before it knows an error has occured in the bitstream. Secondly, in the current FGS scheme, once the decoder detects an error, it would discard the rest of the bitstream in that frame. The effective transmitted video data will be determined by the channel error rate, but not by the channel bandwidth. In this paper, we propose to include some error-resilience tools into the FGS enhancement layer bitstream. We design a hierarchical enhancement layer bitstream structure with resynchronization markers and Header Extension Code (HEC). The experimental results show that the proposed enhancement layer bitstream has stronger error detection and resynchronization capabilities when transmitted over error-prone channels. Thus it provides much better decoded video.

1. INTRODUCTION

With the rapid development of mobile communications, wireless channel becomes an increasingly popular and convenient means to access Internet. Specially, the new 3G mobile communications standard can provide broadband wireless connections for video transmissions (384kbits/s — 2Mbits/s). As we know, wireless channels have different characteristics than the wired Internet channels. They are typically noisy channels, and suffer from a number of channel degradations such as random bit errors and burst bit errors due to fading and multiple path reflection. When compressed video data is sent over these channels, the effect of channel errors on compressed video bitstream can be very severe. As a result, the video decoder that is decoding the corrupted video bitstream often loses synchronization. Moreover, predictive coding techniques such as motion compensation used in various video compression standards make the situation even worse. The decoders based on these techniques would quickly propagate the effects of channel errors across the video sequence and rapidly degrade the video quality.

For applications transmitting video over error-prone channels, such as wireless channels, the FGS provides a more robust solution [1][2]. Firstly, the FGS provides an inherent error recovery feature that can immediately recover from any enhancement layer errors. Secondly, the FGS provides a layered bitstream structure with different importance at different layers. In this layered bitstream structure, the most important information can be sent separately and with increased error protection compared to the less important enhancement information. There are basically two bitstreams in the FGS video: the base layer bitstream and the enhancement layer bitstream. The base layer bitstream is very sensitive to channel error. Any random errors or burst errors may cause the decoder to lose synchronization and the decoded errors will be propagated to the start of next GOP. However, the enhancement layers can tolerate the channel errors. When there are errors in enhancement layer bitstream, a decoder can simply drop the rest of the enhancement bitstream of this frame and search for the next synchronization marker. There should be neither obvious visual artifacts nor error propagation due to the error recovery feature of the FGS.

MPEG-4 included some error resilience techniques from the source coding viewpoint to enable robust transmission of compressed video data over noisy communication channels, such as Data Partitioning, Resynchronization, Reversible variable-length codes, Head Extension Code (HEC), Adaptive intra refresh and NEWPRED [3][4]. These techniques can greatly improve the decoder error detection and recovery capabilities. Since the encoding of the FGS base layer is the same as that of the visual object in MPEG-4, the FGS base layer can use these techniques mentioned above in the same way. However, there are no syntax and semantics currently available for error resilience in the FGS enhancement bitstream. This will cause two very serious problems when delivering the enhancement bitstream through an error-prone channel. First, there is only a weak error detection capability in the enhancement layer. The decoder would put a lot of wrong data into the decoded video before it knows an error has occured in the bitstream. Secondly, in the current FGS scheme, once the decoder detects an error, it would discard the rest of the bitstream in that frame. The effective transmitted video data will be determined by the channel error rate, but not by channel bandwidth. Although the channel bandwidth is very broad, the actual decoded data by the video decoder may be a very small portion of the total received data. The reason is that the decoder has to discard a great deal of enhancement bitstream due to transmission errors.

This paper is organized as follows. In section 2, we use the wireless channel model to analyze the average length of actual decoded bits in the original FGS enhancement bitstream when transmitting them through a simulated wireless channel. Section 3 describes the hierarchical enhancement bitstream structure with HEC and resynchronization markers. Finally, some experimental results verifying the performance the proposed error-resilience tools in the FGS enhancement layer bitstream are presented in Section 4. Section 5 concludes this paper.

2. ANALYSIS OF TRANSMITTING FGS VIDEO OVER A WIRELESS CHANNEL

A wireless channel model shown in Figure 1 is used to analyze statistically how many bits can be correctly decoded for each frame when transmitting the enhancement bitstream through an error-prone noisy channel. It is essentially a two-state Markov model proposed by Gilbert [5], which can be used to simulate both packet losses in Internet channels and symbol errors in wireless channels. This model can be used to characterize the loss or error sequences generated by the data transmission channels. In the good state (G) losses or errors occur with low probability while in bad state (B) they occur with high probability. The losses or errors occur in cluster or bursts with relatively long error free intervals (gaps) between them. The state transition is shown in Figure 1 and summarized by its transition probability matrix:

$$P = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{bmatrix}$$

This model can be used to generate the cluster and burst sequences of packet losses or symbol errors. In this case, it is common to set $\alpha \approx 1$ and $\beta = 0.5$. Generating random packet losses and symbol errors is a special case for the model in Figure 1 where the model parameters can be set as $\alpha \approx 1$ and $\beta = 0$. The error rate of this channel model is $1-\alpha$ in this case.

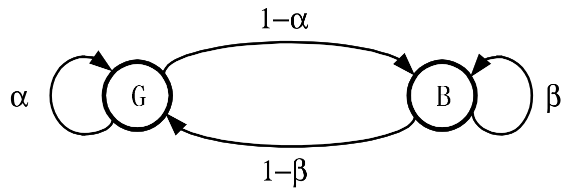

Figure 1: Two-state Markov model for simulating packet losses and channel errors.

This channel model is used to simulate a wireless channel with burst errors in this paper. The occupancy time in good state G is very important for delivering the enhancement layer bitstream through the noisy channel. It means how many continuous symbols can be correctly transmitted. So we define the Good Run Length (GRL) as the length of good symbols between adjacent error points. It is obvious that the distributions of the GRL are subject to a geometrical distribution given by Yajnik [6],

$$p(k) = (1-\alpha)\alpha^{k-1} \qquad k = 1, \quad 2, \quad \ldots, \quad \infty .$$

Thus the mean of GRL should be

$$m = \lim_{N\to\infty} \sum_{k=1}^{N} k \times p(k)$$

$$= \lim_{N\to\infty} \frac{1-\alpha^N}{1-\alpha} .$$

Since α is always less than 1, the above mean of GRL is close to $(1-\alpha)^{-1}$ when N goes to infinite. In other words, the average length of continuous good symbols in the received bitstream is $(1-\alpha)^{-1}$ when transmitting an enhancement bitstream over the simulated channel. Similarly, the mean of Bad Run Length (BRL) is close to $(1-\beta)^{-1}$ when N goes to infinite, which denotes the average length of continuous bad symbols. Obviously, the occupancy times for good state and bad state are both geometrically distributed with respective means of $(1-\alpha)^{-1}$ and $(1-\beta)^{-1}$. Thus the average symbol error rate produced by the two-state Markov model is

$$er = \frac{1-\alpha}{1-\alpha+1-\beta} . \tag{1}$$

As we do know, there are no additional error detection and protection capabilities in the FGS enhancement bitstream. Once there is an error in the enhancement bitstream, the decoder simply drops the rest of the enhancement layer bitstream of this frame and search for the start of the next frame. Therefore, the correctly decoded enhancement bitstream in each frame should be between the enhancement bitstream header and the location where the first error occurred. According to the simulated channel above, although the channel bandwidth may be very broad, the average decoded length of enhancement bitstream is only $(1-\alpha)^{-1}$ symbols.

Now we'll use an example to show how much $(1-\alpha)^{-1}$ normally could be. For a typical wireless channel, the average symbol error rate is 0.01 and its fading degree is 0.6. The corresponding parameter β of the two-state Markov model is 0.6 (equal to the fading degree) and the parameter α is about 0.996 if calculated using the formula (1). In such a wireless channel, the average continuously correctly transmitted data (equal to the mean of GRL) is always about 250 symbols for the enhancement bitstream in each frame. Generally, each transmitted symbol consists of 8 bits in practical channel coding and transmission. Thus the correctly transmitted enhancement data for each frame is around $250 \times 8 = 2000$ bits in such a wireless channel. If the encoded frame rate is 10 Hz, the actual decoded enhancement data rate is only 20 kbits/s. However, the channel bandwidth may be far larger than this rate.

3. ERROR RESILIENCE TOOLS IN THE FGS ENHANCEMENT LAYERS

Clearly from the analysis of the previous section, there needs to add some error detection/protection capabilities into the FGS enhancement layers to improve the robustness and efficiency when transmitted over a wireless channel. In order to get a good trade off between the overhead information and the error robustness of the FGS enhancement bitstream, only some simple error detection and resynchronization tools should be added. The resynchronization marker is the first technique added to the enhancement bitstream. In the present enhancement layer bitstream, once an error is detected, the rest of the bitstream in the current frame is simply dropped. If there are some resynchronization markers in the enhancement layer bitstream, the decoder could just discard the part of the bitstream between two resynchronization markers and continue decoding the rest of the bitstream to minimize the error effects. On the other hand, the original enhancement bitstream provides very weak capability on the error detection just based on the VLC tables and the DCT coefficient number in a block. The resynchronization markers can improve the error detection of FGS enhancement bitstream. The decoder can determine whether a video packet is correctly decoded or not by checking the next resynchronization marker after decoding this video packet. This will prevent the client from decoding some error bits. HEC (Header Extension Code) is the second technique added to the enhancement bitstream. For each frame, the important information that the decoder needs to know to be able to decode the enhancement bitstream is the header data. The header data contains information about the time-stamps, the maximum number of enhancement layers, and the VOP type. If some of this information is corrupted due to channel errors, the decoder has no other choice but to discard all the enhancement bitstream belonging to this frame. HEC technique can recover this header data from other video packets.

We propose a hierarchical structure of the enhancement bitstream with resynchronization marker and HEC as shown in Figure 2. There are three levels from top to bottom in the hierarchical enhancement bitstream: VOP level, BP (Bit Plane) level and VP (Video Packet) level. Every level starts with the header information. The first field in these headers is a unique symbol, such as VOP start code, BP start code and the inserted resynchronization marker. That is, no valid combination of the VLC codes can produce these unique symbols. All these unique symbols can be used for resynchronization purpose.

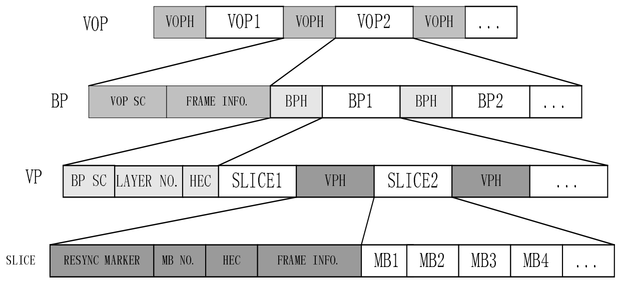

Figure 2 The hierarchical structure of the proposed enhancement layer bitstream

In the new enhancement bitstream, firstly, resynchronization markers are added to the enhancement bitstream at various locations. When the decoder detects an error it will search for this resynchronization marker and regain synchronization. Similar to the GOB in H.261 and H.263 standard, the images may be divided into many video packets that consist of one or more rows of macroblocks. This structure is very suitable to enhancement layer bitstream. Any erroneous marcoblocks in the lower enhancement bit plane layers will cause the marcoblocks at the same locations in higher enhancement layers undecodable due to the dependency among bit planes. Therefore, if one error is detected in one video packet in a lower enhancement layer, the corresponding video packets in higher enhancement layers have to be dropped. On the other hand, the bits used in lower bit plane layers are fewer and the bits used in higher bit plane layers are more. The same number of video packets in each bit plane can provide stronger detection and protection capabilities to the lower enhancement layers because the lower enhancement layers are more important. Moreover, the proposed enhancement layer syntax also allows packing any number of macroblocks into a video packet for flexibility, since the video packet header contains the index of the first macroblock in this video packet. Meanwhile, the start codes of bit planes are used as special resynchronization markers. The header of the first video packet in each bit plane is the bit-plane header instead of resynchronization header.

The HEC field can appear in both the BP header and VP header, which is a one-bit field used as an indicator. As we know, some of the most important information the decoder needs to know to be able to decode the enhancement bitstream is in the VOP header. The VOP header contains information about the time-stamps associated with the decoding and presentation of the video data, and the mode in which the current video object is encoded (whether Inter or Intra VOP). If some of this information is corrupted due to channel errors, the decoder has to discard all the data belonging to the current video frame. The HEC technique is to duplicate the important data in VOP header to some bit-plane headers or video packet headers. Once the VOP header is corrupted by channel errors, the decoder can still recover these data from the bit-plane header or the video packet header. On the other hand, by checking the data in BP header and VP header, the decoder can ascertain if the VOP header is received correctly. If HEC field in a BP header or a VP header is set to 1, this means the important data in VOP header is duplicated in the BP header or the VP header. It is clear that only a few HEC fields can be set to 1 in order to avoid too much overhead.

4. EXPERIMENT RESULTS

Extensive simulations have been performed to test the performance of the proposed error resilience syntax in the enhancement bitstream. The sequences Akiyo, Coastguard and Foreman (CIF format) are used in these experiments. Every I frame followed by 59 predicted frames including P frames and B frames. There are two B frames between two P frames. The encoded frame rate is 30Hz. The limitation on the length of the motion vectors is set to ±32 pixels. The bit rate of base layer is 256 kbits/s with TM5 rate control. The enhancement bitstream is truncated at 128kbits/s, 256kbits/s, …, until 896kbits/s. The channel model generates a simulated wireless channel with burst errors. The fading degree of the wireless channel is 0.6 and its error rate is 0.01. The goal of the simulations is to verify the robustness of the new enhancement bitstream syntax. Here we assume that there are no errors in the transmission of the base layer bitstream. In addition we assume that no channel coding is applied to the enhancement bitstream. In our experiments, three sequences have different slice structure in the first enhancement layer. Each video packet of Akiyo, Foreman and Coastguard sequences consists of six rows, three rows and two rows of macroblocks, respectively. For other enhancement layers, three sequences have the same slice structure. Each video packet consists of one row of macroblocks.

Since the results are very similar among these three sequences. Only the results for Foreman and Coastguard are given in Figure 3 and Figure 4. These results are the actual decoded bits for two types of enhancement bitstreams with different bit rates. Firstly we define the actual decoded bits per frame in the original enhancement bitstream are the bits from the frame start code of the enhancement layer to the location where an error is detected by the video decoder (not by any channel decoder). Secondly, the actual decoded bits per frame in the proposed enhancement bitstream are defined as all the bits in all correctly decoded video packets. Note that the number of overhead bits introduced by the proposed syntax has already been subtracted from the total number of bits correctly decoded in this definition. The dot-dash line curves in Figure 3 and Figure 4 are the average length of continuous good bits of the simulated channels, which are very consistent with the theoretic conclusions.

The experimental results are the average values of 10 experiments to smooth the randomness of the simulated channel. Initially, the actual decoded bits in the original enhancement bitstreams are about 3500 bits. As the channel bandwidth increases, the actual decoded bits close to a constant number. The constant number is larger than that of the analytical results in Section 2. The reason is that there are no additional error detection/protection tools in the original enhancement bitstream. Only VLC tables and the number of DCT coefficients in a block provide a very weak capability to detect errors. Therefore, the location in the bitstream where an error is detected is not the same location where the error has actually occurred. Generally, the location where an error is detected is far away from the location where the error actually occurred. From the experimental results, the average length from the frame start code to the location where one error occurs is about 2000 bits. However, the average length from the frame start code to the location where the decoder detects the error is about 5500 bits. In other words, the decoded bits shown in these figures (Figure 3 and Figure 4) for the original enhancement layer bitstream may already contains errors and the actual correctly received bits may be much fewer than the number shown. For the same reason, the actual decoded bits of the original enhancement bitstream may be a little more than that of the proposed enhancement bitstream in the low bit rate case. But it cannot help to improve the decoded video quality of the original enhancement bitstream since there are wrongly decoded video data between the location where the error occurs and the location where the error is detected. On the contrary, it may cause more quality loss.

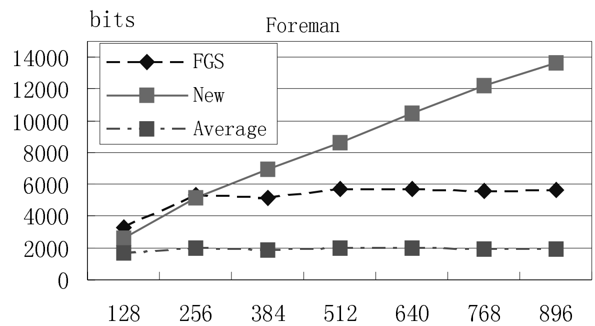

Figure 3 The actual decoded bits vs enhancement layer bit rate for Foreman sequence.

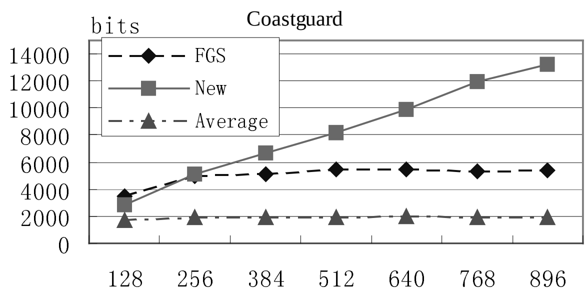

Figure 4 The actual decoded bits vs enhancement layer bit rate for Coastguard sequence

We can clearly see that the number of actual decoded bits of the proposed enhancement bitstream linearly increases as the channel bandwidth or bit rate increases from 128 kbits/s to 896 kbits/s. The decoded bits of the new enhancement bitstream are significant more than that of the original enhancement bitstream from 384 kbits/s to 896kbits/s. The actual decoded bits for the new enhancement bitstream are nearly 3 times more than that of the original enhancement bitstream at 896 kbits/s.

The overhead bits used in the enhancement bitstream are shown in Table 1. These overhead bits are caused by inserting resynchronization header and HECs to the enhancement bitstreams. We can see that the overhead bits at low bit rate are about 350 bits. However at high bit rates the overhead bits are about 1000 bits. If we defined the overhead percent as the number of overhead bits versus the total number of bits, including the base layer bits and the enhancement bits, the overhead percent is about 3%.

5. CONCLUSIONS

We proposed to put some simple but very flexible error resilience tools into the FGS enhancement bitstream. Inserting the resynchronization markers and HEC fields into the FGS enhancement bitstream only cause a very small overhead. But the efficiency and error robustness of the enhancement bitstream is significantly improved when transmitting them over error-prone channels. Due to the flexibility and effectiveness of the proposed scheme, a revised version of this proposal has been accepted in MPEG-4 standard for the streaming video profile.

Table 1 The overhead bits and percent in the proposed enhancement bitstream

Bit rate	Akiyo		Coastguard		Foreman	
	Bits	Percent	Bits	Percent	Bits	Percent
384	373	2.80	359	2.73	349	2.91
512	572	2.91	496	3.45	589	3.35
640	755	2.75	587	3.24	692	3.54
768	858	2.63	674	3.09	790	3.35
896	926	2.59	775	3.00	895	3.10
1024	984	2.63	897	2.93	999	2.88
1152	1035	2.48	952	2.83	1085	2.70

6. REFERENCES

[1] F.Ling, W.P.Li and H.Q.Sun, "Bitplane coding of DCT coefficients for image and video compression", Proceedings of SPIE VCIP'99, San Jose, Jan.25-27, 1999.

[2] W.P.Li, "Fine granularity scalability in MPEG-4 for streaming video", ISCAS 2000, Geneva, Switzerland, May 28-31, 2000.

[3] Raj Ralluri, "Error-Resilient Video Coding in the ISO MPEG-4 Standard", *IEEE Communications Magazine*, pages 112-119, Jun. 1998.

[4] Yao Wang, Qin-Fan Zhu, "Error control and concealment for video communications: A reviews," *Proc. IEEE.vol.86, no. 5*, pages 974-997, May 1998.

[5] J.R.Yee and E.J.Weldon, "Evaluation of the performance of error correcting codes on a Gilbert channel", IEEE trans. Comm. Vol 43, no 8, 2316-2323, 1995.

[6] M.Yajnik, "Measurement and Modeling of the Temporal Dependence in Packet Loss", *UMASS CMPSCI technical report #98-78*.