MPEG-4 FGS CODING PERFORMANCE IMPROVEMENT USING ADAPTIVE INTER-LAYER PREDICTION

Su-Ren Chen, *Chen-Po Chang*, *and Chia-Wen Lin*

Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi 621, Taiwan, R.O.C.
E-mail: cwlin@cs.ccu.edu.tw

ABSTRACT

MPEG-4 Fine Granularity Coding (FGS) has been introduced as a standard video coding tool for video streaming over heterogeneous networks (e.g., the Internet and wireless networks) for its flexibility of supporting a large-range of bit-rates and good error robustness. However, the flexibility and error robustness of MPEG-4 FGS come with the expense of significantly lower coding efficiency than single-layer coding. To improve the coding efficiency, we propose a three-mode codec architecture which introduces part of the enhancement-layer (EL) bitplanes into the motion-compensated prediction (MCP) loops of base-layer (BL) and EL. We propose a novel two-pass adaptive inter-layer prediction scheme which collects coding statistics of macroblocks (MBs) in the first-pass encoding, and then, in the second pass, dynamically chooses the prediction mode by properly reaching a good tradeoff between the estimated coding gain and drifting error with the fine prediction. We also propose an adaptive bit-allocation method which truncates the FGS EL bitstream at the streaming server according to the channel bandwidth to further improve the coding efficiency. Experimental results show that the proposed method can achieve average PSNR improvement by 1.2~1.4 dB over the baseline FGS, while keeping low drifting error at low channel rates.

1. INTRODUCTION

With the proliferation of online multimedia content, the popularity of multimedia streaming technology, and the establishment of video coding standards, people are able to ubiquitously access and retrieve various multimedia contents via the Internet, promoting networked multimedia services at a fast speed. However, there could be a large degree of heterogeneity on network bandwidth, transmission quality, type of codecs, computing resources and display capability, etc. among client terminals. The heterogeneity of present communication networks and user devices poses difficulties in delivering theses bitstreams to the receivers. Scalable coding tools in current coding standards can achieve dynamic bitrate or resolution conversions to support heterogeneous video communications. Most of existing scalable coding tools, however, usually only provide a very limited support of heterogeneity of bit-rates and resolutions (e.g., MPEG-2 and H.263+). To serve video streaming over heterogeneous networks, the MPEG-4 committee has developed the FGS coding [1] which can support a wide range of bit-rates by truncating the EL at an arbitrary location, and provide good error robustness since no temporal dependency is exploited in coding the EL.

Although FGS can support a wide range of bit-rates and provide good error resilience to ease the adaptation of channel variations, this comes with the expense of significant decreased coding efficiency because the lack of temporal redundancy at the EL. In such a way, only low-quality reconstructed video is used in the prediction of BL and EL, leading to relatively higher prediction errors, thereby achieving significantly lower coding efficiency than single layer coding at the same transmission bit-rate, since the single layer coding uses the full-quality video for prediction. The performance degradation can be up to 1.5 to 2.5 dB as reported [2,3]. To address this problem, there have been several relevant works proposed for enhancing the coding efficiency of FGS coding.

For example, Wu et al. proposed a progressive FGS (PFGS) coding scheme [3], where the EL can take reference from the combination of BL and part of previous EL data. An additional MCP loop is employed for the EL coding to improvement coding efficiency. Schaar and Radha [4] proposed a new method, namely , Adaptive Motion Compensated FGS (AMC-FGS), which is featured with two simplified scalability structures: one-loop and two-loop MC-FGS with different degrees of coding efficiency and error resilience. The two-loop MC-FGS employs an additional MCP loop at the EL coder for only B-frames, while the one-loop structure introduces fine predictions for P- and B-frames, leading to relatively higher coding efficiency and lower error robustness. An adaptive decision algorithm is used in AMC-FGS to dynamically switch over the two prediction schemes to achieve better tradeoff in terms of coding efficiency and error robustness. Huang et al. [7] proposed a Robust FGS (RFGS) which adopts an additional MC loop in the EL coding with leaky prediction. The extra MC loop can improve the temporal coding efficiency by referencing to high quality frame memory, and the accompanied drifting error are handled by multiplying the temporal prediction information with a leaky factor α ($0 \leqq \alpha \leqq 1$). In RFGS, the number of referenced bit-planes is also adjustable. By adjusting α and the number of EL bit-planes used for prediction, the RFGS can provide flexible encoding schemes between the baseline FGS [1], MC-FGS [4], and PFGS [3]. Reibman at al. [6] proposed a scheme to control the drift when introducing a weighted average of the fine and coarse predictions into the MCP loops of BL and EL. A rate-distortion optimization framework was developed to determine the weighting factors of fine and coarse predictions which can efficiently improve the coding efficiency. However, the drift at low bit-rates is still quite significant.

The rest of this paper is organized as follows. Sec. 2 introduces the proposed three-mode FGS codec. An adaptive inter-layer prediction scheme and a bit-allocation method for EL bit-

truncation are also described. Sec. 3 shows the experimental results and performance comparisons. Finally, the conclusion is drawn in Sec. 4.

2. PROPOSED FGS CODING SCHEME

2.1. Proposed FGS codec architecture

As shown in Fig. 1, the proposed FGS encoder contains two frame-memories: the coarse frame-memory which stores the BL reconstructed image, and the fine frame-memory which combines the reference MBs (from either of the two frame-memories) with several bit-planes of FGS EL coding residues. The fine prediction is usually more precise than the coarse one, since several bit-planes of EL coding residues are included in the contents of fine frame memory, leading to significantly higher coding efficiency. In the FGS coding, the sending order of EL bit-planes is arranged from MSB to LSB. The first few MSB bit-planes (usually 2~3 MSB planes [2,3]) are very useful for enhancing the image quality, while the effect of the LSB bit-planes is of relatively much less importance. Therefore the proposed coder uses a bit-plane divider to divide the EL bit-planes into two parts: one with the first *m* MSB bit-planes, and the other with the remaining LSB bit-planes. The first part of EL together with the predicted MBs is fed into the fine frame memory for prediction as illustrated in Fig. 1. In the MCP loops of the BL and EL coders, the two prediction blocks for the two coders are obtained from either of the fine and coarse frame-memories. Using the fine prediction to encode the video can achieve significant coding gain than using the coarse prediction; however, the video quality may be injured seriously by the drifting error when the bit-planes used for the fine prediction are not received completely by the decoder due to insufficient bandwidth or data loss in transmission. Such drifting error will result in error propagation until reaching an intra refresh point.

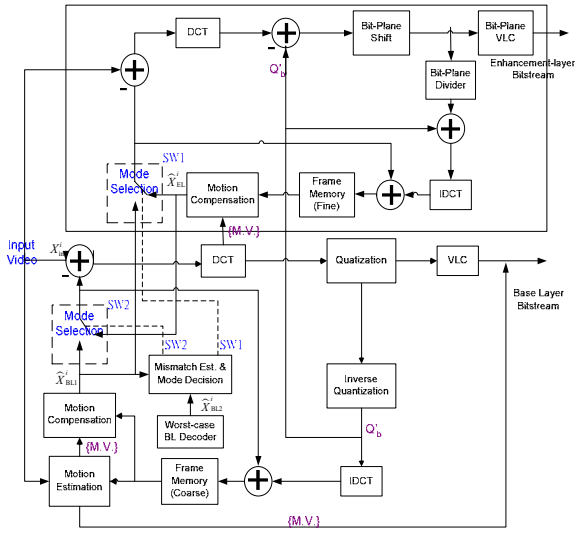


Fig. 1. Proposed FGS encoder architecture.

The proposed encoder contains two switches, SW1 and SW2, for configuring the prediction modes of the two motion-compensated prediction loops in the EL and BL coders, respectively. The upper switch SW1 is used to choose the prediction from either of fine and coarse memories for the MCP loop at the EL coder; while SW2 is for choosing the prediction for the BL (SW = 1: fine prediction; SW = 0: coarse prediction). Three coding modes are provided in the proposed encoder at the MB-level: All-Fine Prediction (AFP: SW1 = 1 and SW2 = 1), All-Coarse Prediction (ACP: SW1 = 0 and SW2 = 0), and Mixed Prediction (MP: SW1 = 1 and SW2 = 0). These coding modes have different characteristics in terms of coding efficiency and error robustness. If the AFP mode is selected, both the BL and EL exploit predictions from the fine frame memory, leading to the highest coding efficiency among the three modes. This, however, runs a high risk of introducing drifting error, since the receiver may not be able to completely receive the EL bit-planes used in the fine predictions due to insufficient channel bandwidth or packet loss. On the contrary, the ACP mode uses coarse predictions for both the BL and EL. This mode guarantees no drifting error should the base-layer bitstream be received completely, but its coding efficiency is the lowest among the three modes. The MP mode compromises on the coding efficiency and error robustness by adopting the fine prediction for the EL and the coarse prediction for the BL, respectively. With this mode, drifting error may occur at the EL when part of EL bit-planes used for fine predictions is lost; while the BL can be drift-free under the assumption that the decoder receives the whole BL data. In the proposed method, to avoid performing motion re-estimation and sending one extra motion vector for each MB, we reuse the motion vectors obtained from the BL encoder for the MCP operation at the EL coder.

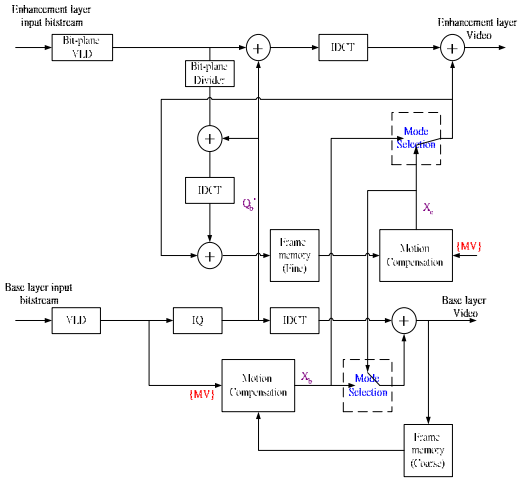


Fig. 2. Proposed FGS decoder architecture.

2.2. FGS coding with adaptive inter-layer prediction

As discussed above, encoding with the coarse prediction (i.e., the ACP mode) is less efficient than that with the fine prediction (i.e., the AFP and MP modes), while drifting error may occur if the fine prediction is utilized but some of EL bit-planes used for prediction are not received by the decoder. This leads to a tradeoff between coding efficiency and error robustness. Our approach is to statistically estimate the best choice of prediction modes without any a priori knowledge of the user channel condition by using a two-pass encoding procedure. In the first-pass encoding of one video frame, we collect the encoding parameters of all MBs in the frame, including prediction error values with the fine and coarse predictions, respectively, and the estimated mismatch error introduced with the fine prediction in the case that the EL data used for fine-prediction cannot be completely received at the decoder. Among these parameters,

the difference between the prediction error values of the two predictions reflects their coding gain difference, while the mismatch error will result in error propagation to the subsequent frames. For example, the coding gain with the fine prediction can be significantly higher than that with the coarse one, which can be estimated as the difference between the fine and coarse prediction errors of the as follows:

$$G_i = \sum_{m=0}^{15}\sum_{n=0}^{15}\left(\left\|X_{\text{in}}^i(m,n)-\widehat{X}_{\text{BL1}}^i(m,n)\right\|-\left\|X_{\text{in}}^i(m,n)-\widehat{X}_{\text{EL}}^i(m,n)\right\|\right) \quad (1)$$

where X_{in}^i stands for the ith incoming MB; $\widehat{X}_{\text{BL1}}^i$ and $\widehat{X}_{\text{EL}}^i$ represent the associated coarse and fine predictions of X_{in}^i, respectively. Note, the two norms in (1) represent the energy values (e.g., the magnitudes) of the fine and coarse prediction errors, respectively. A large G_i value for one MB implies that the fine prediction is much more accurate than the coarse one.

However, the coding gain comes with the risk of introducing drifting error as discussed above. In order to capture such drifting effect, we propose to evaluate the following two estimates of mismatch error of a lost MB:

$$D_i^{\text{B}} = \sum_{m=0}^{15}\sum_{n=0}^{15}\left\|\widehat{X}_{\text{BL1}}^i(m,n)-\widehat{X}_{\text{EL}}^i(m,n)\right\| \quad (2)$$

$$D_i^{\text{W}} = \sum_{m=0}^{15}\sum_{n=0}^{15}\left\|\widehat{X}_{\text{BL2}}^i(m,n)-\widehat{X}_{\text{EL}}^i(m,n)\right\| \quad (3)$$

where D_i^{B} and D_i^{W} stand for the best-case and worst-case estimates of mismatch errors, respectively, under the assumption of zero motion-vector error concealment being used. $\widehat{X}_{\text{BL2}}^i$ is the coarse prediction from another BL coder (not shown in Fig. 1) which encoded at the base-layer bit-rate (i.e., without receiving any EL bits). The mismatch estimates indicate the bounds of concealment error. The best-case estimate D_i^{B} evaluates the lower bound of mismatch error since it assumes all the BL data in previous frames are received correctly. In contrast, the worst-case estimate D_i^{W} is to calculate the accumulated drift should the decoder have only the base-layer (lowest) bandwidth. These two measures can be used to characterize the effect of drifting error, since they reflect the difference between the two frame memories of encoder and decoder. A MB with a large mismatch value implies that it is likely to result in more drifting error if lost. Note, it is difficult to accurately estimate the actual mismatch while encoding without the knowledge about the decoder channel condition. However, we know that the actual mismatch error is bounded by the best and worst estimates, that is, $D_i^{\text{B}} \le D_i \le D_i^{\text{W}}$. We propose to use the weighted average of these two estimates to predict the actual mismatch error:

$$\widehat{D}_i = k_D D_i^{\text{B}} + (1-k_D) D_i^{\text{W}} \quad (4)$$

where $k_D \in [0,1]$. The selection of k_D is dependent on the distribution of decoder bandwidth.

In order to determine the coding mode of each MB so as to achieve good coding performance while keeping enough error robustness, a new index: "Coding gains Over Drifting Error" (CODE) is introduced:

$$\text{CODE}_i = G_i / \widehat{D}_i \quad (5)$$

where G_i and $\widehat{D}_i$ are obtained from (1) and (4), respectively. The index in (5) can be used to characterize the relative gain of coding performance improvement over the potential drifting error for a MB coded with fine prediction. A large CODE value of a MB implies a high possibility that using the fine-prediction to encode the MB can achieve high coding gain while the potential drift penalty is not serious. The mean and standard deviation of the CODE values, m_{CODE} and σ_{CODE}, in a frame are also calculated, respectively. The MBs are then classified into three groups which are encoded with distinct prediction mode (i.e., the ACP, AFP, and MP modes) as follows:

$$\text{MODE}_i = \begin{cases} \text{ACP} & \text{if } \text{CODE}_i < m_{\text{CODE}} - k_1\sigma_{\text{CODE}} \\ \text{AFP} & \text{if } \text{CODE}_i > m_{\text{CODE}} + k_2\sigma_{\text{CODE}} \\ \text{MP} & \text{otherwise} \end{cases} \quad (6)$$

The MBs with a CODE value larger than the upper threshold are encoded with the AFP mode since this tends to achieve significantly higher coding gain, while the risk of introducing drifting error is relatively low. On the contrary, the MBs with a CODE value smaller than the lower threshold are encoded with the ACP mode since they are more sensitive to drifting error. The remaining MBs are encoded with the MP mode to compromise on the coding gain and drifting error. The two parameters k_1 and k_2 can be adjusted according to the drift characteristics of MB. We apply our prediction mode decision method in (6) to all P-frames, while aggressively encoding MBs in B-frames with the AFP mode, since the mismatch error on B-frames will not propagate other frames.

2.3. Adaptive bit-allocation for EL bit-truncation

While streaming, the streaming server truncates each EL frame to an appropriate size to fit the channel bandwidth of the client terminal. If the fine prediction is used for encoding the BL and EL, the bit-allocation scheme for truncating the FGS EL frames can influence the performance largely [5]. For example, if reasonably more bits can be allocated to I/P-frames than B-frames, the decoder will be likely to receive more bit-planes of I/P-frames, leading to lower drifting error and higher video quality. In addition, B-frames can also reference to better-quality pictures for prediction at the encoder as well as for reconstruction at the decoder, should more EL bit-planes of the reference pictures used for prediction be received. We propose a new bit-allocation algorithm for truncating the EL bit-planes at the streaming server with three different cases of available bandwidths: low, medium, and high bit-rates. In the low bit-rate case, the available bandwidth is not sufficient to send all the EL bit-planes of I/P-frames used for the fine-predictions of both layers during the encoding process. Therefore, drifting error is inevitable when part of the EL data used for prediction is dropped in the truncation process. On the other hand, if the available bandwidth is sufficient for sending all the EL bits of I/P-frames used for fine prediction, but is less than PB_{EL}, the server starts to distribute the excessive bits to B-frames after the bit-allocations to I/P-frames can guarantee the bit-planes of I/P-frames used for fine-prediction be completely sent to the receiver. Moreover, if the channel condition is even better, the surplus of bits will also be allocated among I/P-frames while the related bits are reserved to avoid the drifting error. Such bit-rate adaptation by truncating the EL bit-planes can be performed at the server or routers. The truncation schemes for different cases are summarized below.

Table 1 Proposed EL bitstream truncation algorithm

Main:

if ($TB_{EL} \le B_{I\&P,EL}$) // perform low-rate bit truncation

$$TB^n_{I\&P,EL} = PB_{EL} \times PB^n_{I\&P,EL} / \sum_{n=1}^{N_{I\&P}} PB^n_{I\&P,EL}, \quad n = 1, 2, \ldots, N_{I\&P};$$

$$TB^m_{B,EL} = 0, \quad m = 1, 2, \ldots, N_B;$$

else if ($TB_{EL} \le PB_{EL}$) // perform medium-rate bit truncation

$$TB^n_{I\&P,EL} = PB^n_{I\&P,EL}, \quad n = 1, 2, \ldots, N_{I\&P};$$

$$TB^m_{B,EL} = PB_{B,EL} \times PB^m_{B,EL} / \sum_{m=0}^{N_B} PB^m_{B,EL}, \quad m = 1, 2, \ldots, N_B;$$

else // perform high-rate bit truncation

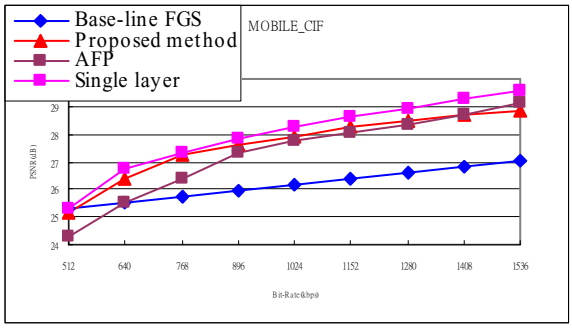
$$TB^n_{I\&P,EL} = PB^n_{I\&P,EL} + PB_{B,EL} \times PB^n_{I\&P,EL} / \left(\sum_{n=0}^{N_{I\&P}} PB^n_{I\&P,EL} + \sum_{m=0}^{N_B} PB^m_{B,EL} \right);$$

$$TB^m_{B,EL} = PB_{B,EL} \times PB^m_{B,EL} / \left(\sum_{n=0}^{N_{I\&P}} PB^n_{I\&P,EL} + \sum_{m=0}^{N_B} PB^m_{B,EL} \right);$$

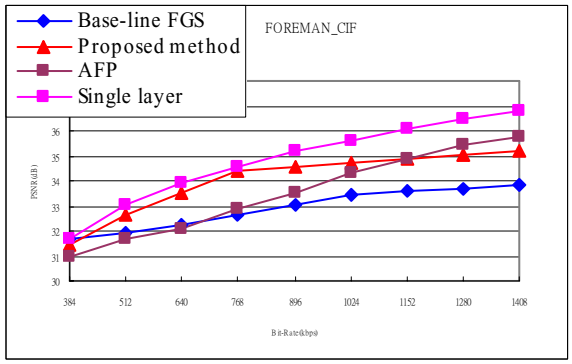
endif

where, N_B and $N_{I\&P}$ are the numbers of I/P-frames and B-frames in a GOP, respectively; N_{BP} is the number of bit-planes used for fine predictions; PB_{EL}, $PB_{I\&P,EL}$, and $PB^n_{I\&P,EL}$ are the numbers of EL bits in a GOP, all I/P-frames, the nth I/P-frame in a GOP used for fine predictions; $PB_{B,EL}$ and $PB^m_{B,EL}$ are the bit-counts of N_{BP} EL MSB bitplanes of all B-frames and of the mth B-frame in a GOP, respectively; TB_{EL}, $TB^n_{I\&P,EL}$, and $TB^m_{B,EL}$ are the bit-allocations of truncation for the EL, the nth I/P-frames of EL, and the mth B-frame of EL in a GOP.

3. EXPERIMENTAL RESULTS



(a) Mobile



(b) Foreman

Fig. 3. Frame-by-frame PNSR performance comparison using four coding methods for two test sequences.

Two CIF (352×288) test sequences, "Mobile" and "Foreman," are used in our experiments. The sequences are encoded with the (30,2) GOP structure at 30 fps. The BL is encoded at 512 and 384 kbps using the TM5 rate control scheme, respectively. Two EL bit-planes are used in the fine prediction (i.e., the AFP and MP modes). Fig. 3 compares the average PSNR performance of the proposed method with those of three other methods: the baseline FGS [1], all-fine prediction (AFP), and the single-layer codec at different bit-rates for the two test sequences. The experimental results show that the proposed method outperforms the other three in a wide range of bit-rates. The AFP and the baseline FGS schemes represent two different performance bounds at the highest and lowest bit-rate ranges, respectively. The proposed method achieves a good tradeoff between the two methods at a wide bit-rate range by adaptively introducing a predefined number of bit-planes into the MCP loops of BL and EL in a nice manner. The proposed method is significantly more robust than the AFP scheme. Only slight quality degradation due to the drifting error is observed at a rather small range of low bit-rates with the proposed method. The average PSNR improvement over the baseline FGS is 1.43 and 1.15 dB for the "Mobile" and "Foreman" sequences, respectively. The maximum PSNR improvement is up to 1.7~1.8 dB.

5. CONCLUSION

In this work, we proposed a three-mode FGS codec architecture accompanied with a novel two-pass adaptive inter-layer prediction scheme to improve the coding performance. After collecting coding statistics of MBs in a frame in the first-pass encoding, the proposed method estimates the potential coding gain and drifting error with the fine prediction then choose the coding mode accordingly in the second-pass encoding. We have also proposed a new bit-allocation method for EL frame truncation according to the channel bandwidth. With our method, the quality degradation due to the drifting error with the fine prediction at low channel rates can be reduced significantly, while the coding gain achieved for high bit-rates is up to 1.8 dB compared to the baseline FGS.

REFERENCES

[1] *Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile*, ISO/IEC 14496-2/FPDAM4, July 2000.

[2] W. Li, "Overview of fine granularity in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp.301-317, Mar. 2001.

[3] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.* vol.11, no. 3, pp. 332 - 344, Mar. 2001.

[4] M. van der Schaar and H. Radha, **"**Adaptive motion-compensation fine-granular-scalability (AMC-FGS) for wireless video," *IEEE Trans. Circuits Syst. Video Technol.* vol.12, no. 6, pp. 360-371, Jun. 2002.

[5] H.-C. Huang, C.-N. Wang, and T. Chiang, "A robust fine granularity scalability using trellis-based predictive leak," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 372-385, vol. 12, no. 6, Jun. 2002.

[6] A. R. Reibman, L. Bottou, and A. Basso, "Scalable coding with managed drift," *IEEE Trans. Circuits Syst. Video Technol.* vol.13, no. 2, pp. 131 -140, Feb. 2003.