DRIFTING REDUCTION IN PROGRESSIVE FINE GRANULARITY SCALABLE VIDEO CODING

Feng Wu[1], Shipeng Li[1], Bing Zeng[2], Ya-Qin Zhang[1]

[1] Microsoft Research China, Beijing, 100080
[2] The Hong Kong University of Science and Technology, Hong Kong

ABSTRACT

This paper investigates the drifting problem in Progressive Fine Granularity Scalable (PFGS) video coding, which arises from the fact that high quality references are used to predict the enhancement layer. Due to the unique property of the PFGS scheme, this paper first analyzes the occurrence and propagation of the drifting errors in details. As a result, an iterative model is established to simulate the drifting errors at the encoder. With the proposed drifting model, a decision-making mechanism can help the PFGS scheme to choose the coding mode of each macroblock in the enhancement layer. The experimental results show that the PFGS scheme incorporating the proposed technique can effectively reduce the drifting errors at low bit rates. Meanwhile, the PFGS scheme also improves the coding efficiency up to 1.8dB compared with the FGS scheme at moderate or high bit rates.

1. INTRODUCTION

Progressive Fine Granularity Scalable (PFGS) coding [1],[2] is a fine-grain PSNR scalable video coding scheme that offers high coding efficiency, good bandwidth adaptation, and error recovery capability. It represents an improvement over the Fine Granularity Scalable (FGS) coding scheme, which has been accepted by the MPEG-4 standard as a promising technique [3] for streaming video applications. Both PFGS and FGS are based on bit-plane coding of predicted DCT residues. However, by using high quality reference for prediction of the enhancement layer, PFGS outperforms FGS by as much as 1.0 dB in average PSNR [2].

In general, the bit rate of the base layer is low enough to fit in the minimum network bandwidth. Therefore, we assume that the base layer is always available at the decoder. However, since the high quality references in PFGS comprise part of the DCT coefficients encoded in the enhancement layer, more bandwidth is needed to transmit them to the decoder. When channel bandwidth somehow drops, the decoder may partially or completely lose the high quality reference. In this case, the decoder has to use the low quality reference instead, which would inevitably cause the drifting error in the enhancement layer.

In fact, the error drifting phenomena often occur in the layered scalable video coding and in scenarios such as transmitting video streams over the error-prone channel when previous reference frames are not fully reconstructed. Some approaches for error reduction and elimination in [4]~[6] aren't suitable for the PFGS scheme due to the fine granularity scalability property. The PFGS encoder has to choose different modes at the enhancement macroblock level to effectively reduce the drifting error at any bit rates[7]. Three inter coding modes are proposed in Figure 1. Gray rectangular boxes in Figure 1 denote those layers to be reconstructed as references. Solid arrows with solid lines are for temporal predictions, hollow arrows with solid lines are for reconstruction of high quality references, and solid arrows with dashed lines are for predictions in DCT domain.

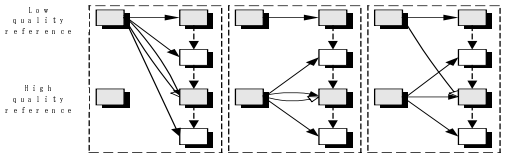

Figure 1: The coding modes for the enhancement macroblock.

In LPLR mode, all layers are predicted and reconstructed from the previous low quality reference. If all enhancement macroblocks are encoded with this mode, the PFGS scheme is exactly the same as the FGS scheme. In HPHR mode, the enhancement layers are predicted and reconstructed from the previous high quality reference. This mode can offer high coding efficiency. However, it will also result in drifting errors when the previous high quality reference is not available at the decoder. In HPLR mode, the enhancement layers are predicted from the previous high quality reference. However, the current high quality reference is reconstructed from the previous low quality reference. This mode can effectively eliminate the error propagation. However, since the reconstructed high quality reference may not have the best quality it could get, the HPLR mode also has negative effect in coding efficiency.

How to choose the coding mode of each macroblock is an encoding optimization issue, which depends on how the encoder estimates the drifting errors. This paper tries to establish a drifting model to estimate and reduce the drifting error in PFGS scheme. Section 2 first focuses on analyzing the occurrence and propagation of drifting errors when all enhancement macroblocks are encoded with the HPHR mode. It then establishes an iterative model for the drifting error at the encoder. Section 3 proposes a decision-making mechanism that can be used to optimally choose the coding mode of each enhancement macroblock based on the drifting model. The experimental results are given in Section 4. Section 5 concludes this paper.

2. DRIFTING ERROR ANALYSIS AND MODELING

Figure 2 is a block diagram of the PFGS encoder, in which all enhancement macroblocks are encoded with the HPHR mode. For simplicity, the motion estimation module is omitted. The motion estimation is performed between the current and previous original images. The motion vectors

obtained are used in both of the motion compensation modules in Figure 2. Since there are two references at the PFGS encoder, it produces two sets of predicted DCT coefficients. (1) A first set of predicted DCT coefficients X_b is the prediction error formed by referencing the previous low quality reference, and (2) a second set of predicted DCT coefficients X_e is the prediction error formed by referencing the previous high quality reference. X_b is encoded in the base layer, and the DCT residues between the reconstructed $\tilde{X}_b$ and X_e generate the enhancement bit-stream with the bit plane coding technique. Only the lower enhancement layer is used to reconstruct the high quality reference.

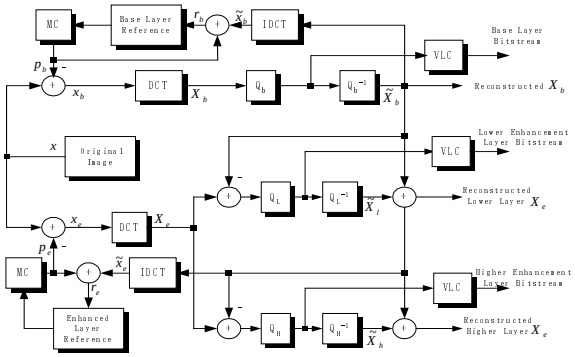

Figure 2: The block diagram of a PFGS encoder.

Different from the traditional layered scalable coding schemes where the enhancement bitstream must be either completely delivered or not, the fine grain scalability in PFGS facilitates the PFGS enhancement bitstream to be decoded at any length. When network bandwidth somehow decreases, the lower enhancement layers of one or multiple frames may be partially or even completely dropped. Once the network bandwidth gets back, the client side may obtain the enhancement layer again, which in turn causes the drifting error since the required previous reference is not available. In the sections followed, we will give a detailed analysis on the drifting errors when single/multiple lower enhancement layers get corrupted.

2.1 Lower enhancement layer in a single frame corrupted

The first frame of each group of picture (GOP) is encoded as an I frame. With the IDCT transform, the joint reconstructed DCT coefficients of the base layer and lower enhancement layer generate the high quality reference r_e. If part of the lower enhancement layer is corrupted during transmission, the difference between the high quality references used in the decoder and the encoder is

$$e(n) = DCT^{-1}(\tilde{X}_l(n) - \hat{X}_l(n)). \quad (1)$$

Here the hat "^" denotes the decoded DCT coefficients and images, and the hat "~" denotes the encoded DCT coefficients and images. n is the counter of frame. $\tilde{X}_l$ and $\hat{X}_l$ denote the encoded and decoded the residues of DCT coefficients in the lower enhancement layer, respectively. If the lower enhancement layer is correctly transmitted to the decoder, $e(n)$ should be zero. In the worst case, $e(n)$ is equal to the maximum value $DCT^{-1}(\tilde{X}_l(n))$ when the whole lower enhancement layer is dropped.

Similarly, for a predicted frame, the high quality references are reconstructed at the decoder and the encoder with the following formula

$$\begin{cases} \hat{r}_e(n) = \hat{p}_e(n) + DCT^{-1}(\hat{X}_b(n) + \hat{X}_l(n)) \\ r_e(n) = p_e(n) + DCT^{-1}(\tilde{X}_b(n) + \tilde{X}_l(n)) \end{cases}. \quad (2)$$

$p_e(n)$ is the motion prediction image from previous reference, X_b is the reconstructed base layer DCT coefficients. Since only the current lower enhancement layer is corrupted in this GOP, $\hat{p}_e(n)$ is always equal to $p_e(n)$, and $\hat{X}_b(n)$ is also equal to $\tilde{X}_b(n)$. By subtracting between the two formulas in (2), we can get the same result as the formula (1) when the current frame suffers from the transmitted errors in the lower enhancement layer.

Because of the motion prediction loop in the PFGS scheme, the high quality reference forms the next frame prediction through motion compensation, i.e.,

$$\begin{cases} \hat{p}_e(n+1) = f_{n+1}(\hat{r}_e(n)) \\ p_e(n+1) = f_{n+1}(r_e(n)) \end{cases} \quad (3)$$

Here $f()$ denotes the motion compensation. The subscript "n+1" indicates that the motion vectors of the $(n+1)^{th}$ frame are used in this motion compensation. Even though the lower enhancement layer in the $(n+1)^{th}$ frame is correctly delivered to the client, the decoded high quality reference in the $(n+1)^{th}$ frame still has quality loss due to the corrupted prediction from previous frame. The error propagated from the n^{th} frame can be represented as

$$\begin{aligned} y(n+1) &= p_e(n+1) - \hat{p}_e(n+1) \\ &= f_{n+1}(p_e(n) + DCT^{-1}(\tilde{X}_b(n) + \tilde{X}_l(n))) \\ &\quad - f_{n+1}(\hat{p}_e(n) + DCT^{-1}(\hat{X}_b(n) + \hat{X}_l(n))). \end{aligned} \quad (4)$$

Since the same motion vectors are used in both of the encoder and the decoder, the sum of each signal after motion compensation is equal to the motion compensation of the sum of two signals. With the linear DCT transform, the formula (4) can be simplified as

$$y(n+1) = f_{n+1}(DCT^{-1}(\tilde{X}_l(n) - \hat{X}_l(n))). \quad (5)$$

$y(n+1)$ is the error propagation from the n^{th} frame to the $(n+1)^{th}$ frame. Moreover, this error is also propagated to all frames followed in the GOP through motion compensation. With extension of the formula (5), the error propagation from the n^{th} frame to the $(n+k)^{th}$ is described as

$$y_n(n+k) = f_{n+k}(...(f_{n+1}(DCT^{-1}(\tilde{X}_l(n) - \hat{X}_l(n))))). \quad (6)$$

The above formula exhibits the occurrence and propagation of the drifting error when a single lower enhancement layer is corrupted during transmission.

2.2 Lower enhancement layers in multiple frames corrupted

When streaming over the Internet, the decrease of channel bandwidth usually corrupts the lower enhancement layers of multiple frames. The general case is that the lower enhancement layer of each frame may be corrupted during transmission. From the formula (6) we know that these errors can be propagated within this GOP. Therefore, the drifting error appeared in the n^{th} frame is the accumulated propagation from errors in all the previous frames, i.e.,

$$y(n) = \sum_{k=1}^{n-1} y_k(n). \quad (7)$$

Here $y_k(n)$ denotes the error propagation from the k[th] frame to the current frame. Incorporate (6) and (1) to (7), we can get the following formula

$$y(n) = \sum_{k=1}^{n-1} f_n(...(\ f_{k+1}(e(k)))) \quad . \tag{8}$$

Here $e(k)$ denotes the error occurred in the lower enhancement layer in the k[th] frame. This clearly explains why the drifting error can rapidly deteriorate the decoded video quality. If the formula (8) is used to calculate the drifting error at the encoder, the computational intensity is too high.

In the formula (8), the error occurred in each of the previous frames is first warped to the current frame through one or multiple motion compensation operations and is then accumulated. In other words, the same motion compensation is performed multiple times to propagate each error from one frame to the next. Since the same motion vectors are used in this process, all errors from the previous frames can be first added together and then propagated to the next frame through only one motion compensation. Therefore, an iterative model is derived from the formula (8)

$$y(n) = \begin{cases} 0 & n = 1 \\ f_n(y(n-1) + e(n-1)) & N \geq n > 1 \end{cases} . \tag{9}$$

Where N is the total number of frames in this GOP. $y(n-1)$ is the accumulative error propagated to the (n-1)[th] frame, and $e(n-1)$ is the new error occurred in the lower enhancement layer in the (n-1)[th] frame. With the motion compensation, their sum forms the next drifting error in the n[th] frame. Utilizing the formula (9) to model the drifting error at the encoder is very simple. The additional complexity is one motion compensation operation and one inverse DCT transform in each frame. Moreover, the extra computation is only needed at the encoder side for encoding optimization.

If the encoder can get feedback about the corrupted DCT coefficients in the low enhancement layer from the client decoder, the iterative model can accurately reproduce the drifting error originally occurred at the decoder. However, in the PFGS scheme, the iterative model is used to estimate the drifting error in the worst case, i.e., all lower enhancement layers in the first (n-1) frames are completely dropped, the decoder obtains the enhancement layer in the n[th] frame again.

3. DECISION-MAKING MECHANISM BASED ON THE DRIFTING MODEL

As shown in Figure 3, the HPHR and LPLR curves denote all enhancement macroblocks encoded with the HPHR mode and the LPLR mode, respectively. At low bit rates, the decoded quality indicated by the HPHR curve is far lower than that indicated by the LPLR curve because of the drifting errors. While channel bandwidth increases, the decoded quality of the HPHR curve also increases rapidly. At high bit rates, the HPHR curve is close to that of the single layer video coding. The decision-making mechanism is expected to reduce the drifting error at low bit rates and maintain the high coding efficiency at high bit rates by optimally choosing the inter coding mode of each enhancement macroblock.

The decision-making mechanism first determines whether the current macroblock should be encoded with the LPLR mode or not. The reference for prediction in the LPLR mode is of low quality, but the references used in the HPHR and HPLR mode are of high quality. Therefore, the criterion to distinguish the LPLR mode from other two modes is

$$\min(\ \left\| X_b - \tilde{X}_b \right\|, \left\| X_e - \tilde{X}_b \right\|) \ . \tag{10}$$

When the enhancement layer is predicted from the low quality reference, the DCT residues encoded in the enhancement layer are $X_b - \tilde{X}_b$, otherwise the encoded DCT residues are $X_e - \tilde{X}_b$. If the absolute mean of the first item in (10) is less than that of the second item, the current macroblock is encoded with the LPLR mode.

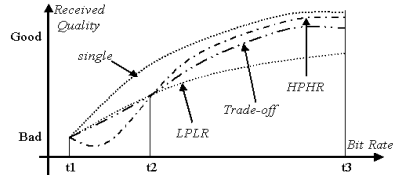

Figure 3: The PFGS scheme with different coding controls.

The key point in the decision-making mechanism is how to determine the coding mode between the HPHR mode and the HPLR mode. In fact, the decision-making mechanism is expected to achieve the performance of the Trade-off curve as shown in Figure 3. The LPLR curve is used as reference in the decision-making mechanism. The criterion for choosing the coding mode between the HPHR mode and the HPLR mode is defined as

$$\max_{mode\in\{HPHR,HPLR\}} \left(\int_{t2}^{t3} \left\| s(t) - s_{LPLR}(t) \right\|^2 dt - \lambda \int_{t1}^{t2} \left\| s_{LPLR}(t) - s(t) \right\|^2 dt \right) \tag{11}$$

$s(t)$ is the SNR curve obtained by optimal mode decision, and $s_{LPLR}(t)$ is the corresponding SNR curve when all enhancement macroblock are encoded with the LPLR mode. λ is the weighting factor. The first item in (11) denotes the coding efficiency gains at high bit rates, and the second item denotes the quality losses at low bit rates.

Obviously, the criterion (11) will result in rather intensive computation in real application. Thus, the proposed decision-making mechanism simplifies the criterion to only eliminate the drifting errors at low bit rates. Moreover, in order to further reduce the computation, the maximum quality loss at low bit rate is used to replace the integral. Since the drifting model discussed in the above section provides the maximum drifting estimation in the worst case, the new criterion is defined as

$$\left\| y(n) \right\|^2 \geq k \left\| x(n) - r_{LPLR}(n) \right\|^2 . \tag{12}$$

The left side of the inequation (12) is the drifting error propagated from the previous frames in the worst case, which will cause the maximum quality loss in the current frame. The right side of the inequation (12) is the given threshold for the quality loss at low bit rates. $r_{LPLR}(n)$ is the reconstructed high quality reference with the LPLR method, where the DCT coefficients encoded in the low enhancement layer are bit-plane quantized $\left\| X_b - \tilde{X}_b \right\|$. The factor k is an adjustable parameter to provide the balance between the drifting error and the coding efficiency. In other words, if the drifting error at the current macroblock is larger than the allowable quality loss, this macroblock is encoded with the HPLR mode to reduce the drifting error; otherwise this macroblock is encoded with the HPHR mode to get high coding efficiency.

4. EXPERIMENTS

The first experiment is to verify the performance of the proposed scheme in drifting reduction. In this experiment, only the first frame is encoded as an I frame, and other frames are encoded as P frames. Three fine granularity scalable coding schemes are compared in this experiment. In the PFGS scheme, the lower enhancement layer includes two or three bit planes depending on bits generated. The bit rate of the high quality reference is larger than 256kbits/s. In the decoder, the given bit rate for decoding the enhancement layer is 128kbits/s. Thus the lower enhancement layer in the PFGS scheme is only partially transmitted to the decoder.

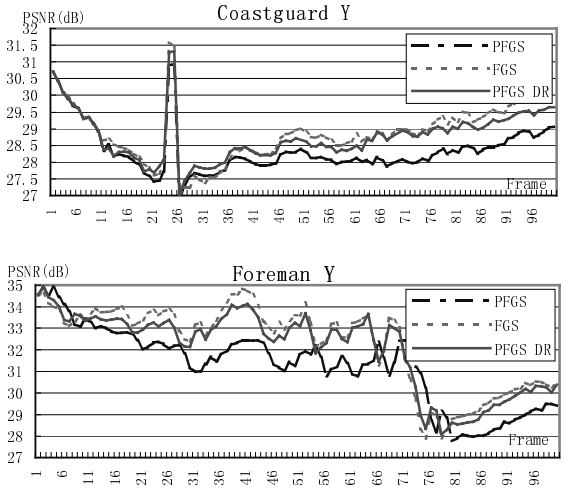

Figure 4: PSNR versus frame number at enhancement bit rate 128kbits/s.

The experimental results are depicted in Figure 4, in which PFGS denotes the PFGS scheme with all enhancement macroblock encoded with the HPHR mode, and PFGS DR denotes the PFGS scheme with the proposed mode decision scheme. Without drifting reduction, the maximum quality loss of the PFGS scheme reaches as high as 2.0 dB in Foreman and also exceeds 1.0 dB in Coastguard. We can see that the PSNR curves are clearly drifting toward the low end as the frame number increases. We also see that the proposed drifting reduction technique can greatly reduce the drifting error. Compared with the FGS scheme, the maximum quality loss of PFGS DR is below 0.5dB in Coastguard and below 1.0 dB in Foreman. The average quality loss is less than 0.1dB. Moreover, there is no obvious drifting phenomenon visually in the PFGS scheme with drifting reduction.

The second experiment is to examine the coding efficiency of the PFGS scheme with drifting reduction, i.e., the PFGS scheme is compared with FGS and non-scalable video coding schemes. The FGS codec and non-scalable video codec are from MPEG-4 VM. The PFGS codec is implemented based on baseline FGS with the drifting model and the proposed decision-making mechanism. In the non-scalable video coding, the output rate control is TM5. The experimental results are given in Figure 5. The PSNR gap between the FGS and the single layer non-scalable video coding scheme is larger than 2.0 dB. At the low enhancement layer bit rates, the PFGS scheme with drifting reduction has almost the same performance as FGS. When the total bit rate is larger than 512 kbits/s, the PFGS scheme with drifting reduction can significantly improve the coding efficiency over the FGS scheme. The improved PSNR is up to 1.8dB in Foreman and 1.5dB in Coastguard. The PSNR gap between PFGS DR and non-scalable video coding is now less than 1.0dB when the bit rate becomes moderate or high.

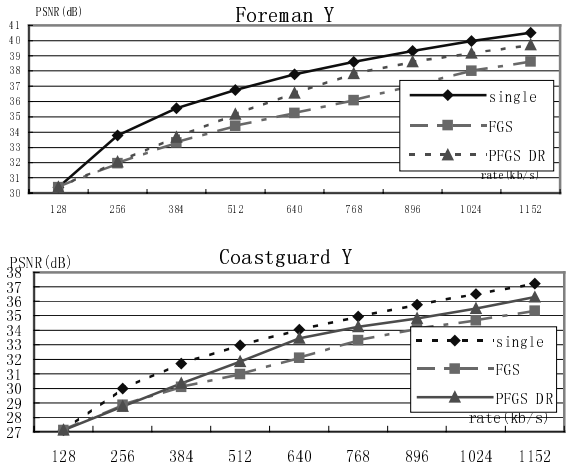

Figure 5: PSNR versus bit rate comparison among FGS, PFGS, and non-scalable coding.

5. CONCLUSION

This paper presents an iterative error drifting model for PFGS video coding. Based on this model, a mode decision scheme for PFGS inter-frame coding is proposed to reduce the drifting error at lower bit rates while significantly improving the coding efficiency at moderate or high bit rates. Experimental results verified the efficiency of the proposed scheme.

6. REFERENCES

[1] F. Wu, S. Li and Y.-Q. Zhang, "DCT-prediction based progressive fine granularity scalability coding", *ICIP 2000*, Vancouver, Canada, vol 3, pp556-559, Sep 10-13, 2000.

[2] F. Wu, S. Li and Y.-Q. Zhang, "A framework for efficient progressive fine granular scalable video coding", IEEE trans. CSVT, vol 11, no 3, 2001.

[3] W. Li, "Fine granularity scalability in MPEG-4 for streaming video", ISCAS 2000, Geneva, Switzerland, pp299-302, May 28-31, 2000.

[4] J. Arnold, M. Frater and Y. Wang, "Efficient drift-free signal-to-noise ratio scalability", IEEE trans. CSVT, vol 10, no 1, pp70-82, 2000.

[5] T. Lee, J. Wang and P. Chang, "Precise error tracking for terminating error propagation effects in H.263 video transmission", SPIE proceeding vol 3561, Electronic imaging and multimedia system, pp265-273, 1998.

[6] R. Mathew and J. Arnold, "Layer coding using bitstream decomposition with drift correction", IEEE trans. CSVT, vol 7, no 6, pp882-891, 1997.

[7] F. Wu, S. Li, X. Y. Sun, R. Yan and Y.-Q. Zhang, "Macroblock-based progressive fine granularity scalable video coding", ISO/IEC JTC1/SC29/WG11, m6779, Pisa, Italy, Jan, 2001.