HTML Paper

MACROBLOCK-BASED PROGRESSIVE FINE GRANULARITY SCALABLE VIDEO CODING

Xiaoyan Sun[*1], *Feng Wu*[2], *Shipeng Li*[2], *Wen Gao*[1], *Ya-Qin Zhang*[2]

[1] Department of Computer Application, Harbin Institute of Technology, Harbin, 150001

[2]Microsoft Research China, Beijing, 100080

ABSTRACT

The Progressive Fine Granularity Scalable (PFGS) coding is a promising technique for streaming video applications. However, since the original PFGS only chooses its references as frame-based, it is very difficult to achieve a good trade-off between high coding efficiency and low drifting errors. In this paper, we present a flexible and effective scheme to control the PFGS coding at the macroblock level. Three INTER modes are first proposed for the enhancement macroblock coding. One of these modes provides a novel method to effectively reduce the drifting errors at low bit rates. Then, a decision-making mechanism based on temporal prediction is developed to choose the optimal coding mode for each enhancement macroblock, which offers a significant coding efficiency improvement for the fine granularity scalable coding scheme. Moreover, the proposed control mechanism can be easily implemented without any additional computation. The experimental results show that the proposed macroblock-based PFGS coding scheme can effectively reduce the drifting errors at low bit rates, while providing further coding efficiency improvement over the FGS and the original PFGS scheme at moderate or high bit rates.

1. INTRODUCTION

With the steady increase in the access bandwidth, more and more Internet applications start to use the streaming audio and video contents [1][2]. In response to the increasing demand on streaming video applications over the best-effort Internet, the coding objective for streaming video is changing to optimize the video quality for a wide range of bit rates. The Fine Granularity Scalable (FGS) video coding adopted in MPEG-4 standard is just such a technique [3][4]. Because of the fine granularity scalability provided by the bit-plane coding technique in the enhancement layer, the FGS scheme can easily adapt to the channel bandwidth fluctuations. However, since its motion prediction is always based on the lowest quality base layer, the coding efficiency of the FGS is not so good as, and sometimes worse than, the traditional SNR scalable coding [5]. Compared with the non-scalable video coding schemes, the PSNR of the FGS may drop 2.0dB or more at the same bit rate [3].

The Progressive Fine Granularity Scalable (PFGS) coding scheme proposed in [6] is an improvement over the FGS scheme. A typical architecture of the PFGS is depicted in Figure 1. The coding efficiency of the PFGS can be up to 1.0dB higher in average PSNR than that of the FGS at moderate or high bit rates because of the use of high quality references in the enhancement layer coding. In general, it is reasonable to assume that the base layer is always available in the decoder. However, when network bandwidth drops, the decoder may partially or completely lose the high quality references. The differences between the high quality references used in encoder and decoder will inevitably cause drifting problem. By alternatively reconstructing the high quality reference from the previous low quality reference and high quality reference, as suggested in Figure 1, the PFGS scheme can efficiently eliminate the error propagation. However, doing so also affects the coding efficiency, because the high quality reference does not always get the best quality it could get. Furthermore, it is very difficult for the PFGS scheme to provide a good trade-off between high coding efficiency and low drifting errors at the frame level.

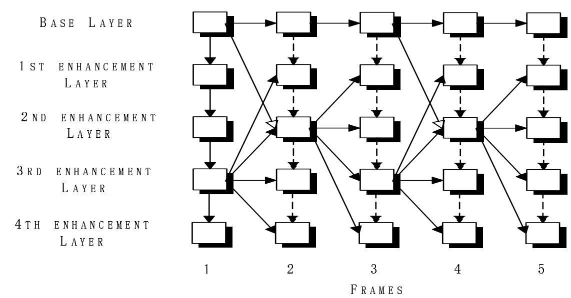

Figure 1 The architecture of the PFGS scheme (Solid arrows with solid lines are for prediction references, hollow arrows with solid lines are for reconstruction references from the previous base layer, and solid arrows with dashed lines are for prediction in DCT domain)

In this paper, a flexible macroblock-based PFGS coding is presented. First, three INTER modes are proposed for the enhancement macroblock coding according to the different references used for prediction and reconstruction. Then, a decision-making mechanism based on temporal prediction is proposed to choose an appropriate coding mode for each enhancement macroblock. With the proposed scheme, the PFGS can achieve a good trade-off between drifting error reduction and coding efficiency improvement. Moreover, no extra computation is needed for the mechanism.

The rest of this paper is organized as follows. Section 2 proposes three INTER modes for the enhancement MB coding. In Section 3, we present a decision-making mechanism that optimally chooses the coding mode of each MB. Some experimental results are given in Section 4. Finally, Section 5 concludes this paper.

[*] This work has been done while the author is with Microsoft Research China.

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

2. INTER MODES FOR ENHANCEMENT MACROBLOCK CODING

Figure 2 is a diagram of the PFGS encoder corresponding to the architecture shown in Figure 1. The motion estimation module is omitted in this diagram. Same motion vectors are used in both of the motion compensation modules in Figure 2. The PFGS encoder produces two sets of predicted DCT coefficients. (1) A set of predicted DCT coefficients X_b are prediction errors formed by referencing the previous low quality reference, and (2) a set of predicted DCT coefficients X_e are prediction errors formed by referencing the previous high quality reference. The DCT coefficients X_b are encoded in the base layer. The differences between X_e and the reconstructed $\tilde{X}_b$ form the enhancement bit-stream. Among all enhancement layers, only lower enhancement layers have a contribution to the high quality reference.

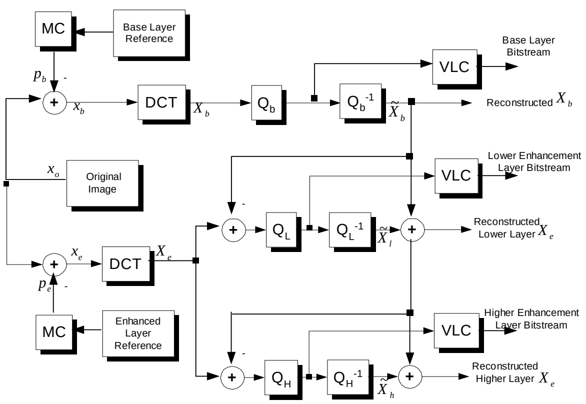

Figure 2 The diagram of a PFGS encoder.

In the original frame-based PFGS, the references are used for the whole frame in the enhancement layer coding. In fact, since there are two references in the PFGS scheme, a flexible method by choosing the references at each enhancement macroblock may achieve a better trade-off between drifting errors and coding efficiency. Derived from the FGS and the original PFGS scheme, three INTER coding modes for enhancement layer macroblock coding are proposed in Figure 3. Gray rectangular boxes in Figure 3 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.

In LPLR mode (mode 1), an enhancement macroblock is predicted and reconstructed both from the previous low quality reference. There is no drifting error in this mode. The PFGS scheme will be exactly the same as the FGS scheme if all enhancement macroblocks are encoded with this mode. The coding efficiency of this mode is low due to low quality temporal prediction.

In HPHR mode (mode 2), an enhancement macroblock is predicted and reconstructed both from the previous high quality reference. The PFGS scheme can provide the highest coding efficiency at high bit rates, if all enhancement macroblocks are encoded with this mode. But, if the previous high quality reference is not available due to network bandwidth or transmission errors, the decoder has to use the low quality reference instead. In this case, the reconstructed high quality reference for the next frame ($\hat{r}_e(n)$) in the decoder is

$$\hat{r}_e(n) = p_b(n) + f^{-1}(\tilde{X}_b(n) + \tilde{X}_l(n))\,. \tag{1}$$

Here $p_b(n)$ denotes the low quality prediction. $\tilde{X}_b(n)$ and $\tilde{X}_l(n)$ are DCT coefficients encoded in the base layer and low enhancement layer respectively. $f^{-1}(*)$ is the inverse DCT transform. However, the corresponding high quality reference for the next frame ($r_e(n)$) reconstructed in the encoder is

$$r_e(n) = p_e(n) + f^{-1}(\tilde{X}_b(n) + \tilde{X}_l(n))\,. \tag{2}$$

Here $p_e(n)$ denotes the high quality prediction. The differences between two reconstructed high quality references are

$$e(n) = p_e(n) - p_b(n)\,. \tag{3}$$

Because of the motion prediction/compensation loop in PFGS, the difference $e(n)$ not only could affect the decoded video quality of the current frame, but also could be propagated to the frames followed within the same Group of Picture (GOP). Hence, the quality of the decoded enhancement layer deteriorates rapidly while the frame number increases in the GOP.

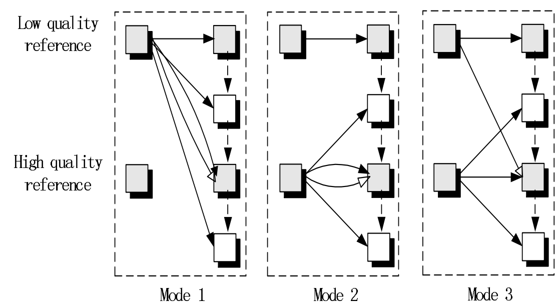

Figure 3 The coding modes for enhancement macroblock.

HPLR mode (mode 3) provides a novel method to reduce the accumulative errors. In this mode, an enhancement macroblock is predicted from the previous high quality reference, while reconstructed from the previous low quality reference at both encoder and decoder. Since the encoder and decoder can always obtain the same temporal prediction, the potential propagation of differences $e(n)$ are effectively eliminated. Although this idea has already been in the original frame-based PFGS to eliminate the drifting errors, this paper extends this technique to the macroblock level, which offers more flexibility for the PFGS scheme.

In HPLR mode, the difference that originally may occur between the encoder and decoder in HPHR mode is now transferred to the reconstructed high quality reference at the encoder. With the quality loss in the reconstructed high quality reference, the temporal prediction for the next frame also has the corresponding loss. Therefore, HPLR mode also affects the coding efficiency of the PFGS scheme. However, when the enhancement layer is predicted from the high quality reference, reasonably utilizing the mixture of HPHR mode and HPLR mode can provide a good trade-off between high coding efficiency and low drifting errors.

3. MODE DECISION FOR ENHANCEMENT MACROBLOCK CODING

In this section, a decision-making mechanism is proposed to choose the proper coding mode of each enhancement macroblock. Besides the INTER modes, INTRA mode is also allowed in the enhancement layer coding. INTRA mode or INTER mode are determined by the motion estimation module. If a macroblock is encoded with INTRA mode in the base layer, the corresponding enhancement macroblock is also encoded with INTRA mode. If a macroblock in the base layer is encoded with INTER mode, the proposed decision-making mechanism has to determine which INTER coding mode should be used in the corresponding enhancement macroblock.

The reference for prediction in LPLR mode is of low quality, but the references used in HPHR mode and HPLR mode are of high quality. Therefore, the criterion to distinguish LPLR mode from the other two INTER modes is

$$\min\left(\left\|X_b - \tilde{X}_b\right\|, \left\|X_e - \tilde{X}_b\right\|\right). \quad (4)$$

In other words, the DCT residues encoded in the enhancement layer are $X_b - \tilde{X}_b$ and $X_e - \tilde{X}_b$ respectively in LPLR mode and HPHR/HPLR. Thus, when the absolute mean of the former DCT residues is less than that of the latter one, the current macroblock is encoded using LPLR mode; otherwise the decision-making mechanism further determines the coding mode between HPHR mode and HPLR mode.

Both HPHR mode and HPLR mode are predicted from the high quality reference, the difference between them is the references used for reconstruction. Note that the more enhancement macroblocks coded with the HPHR mode, the higher coding efficiency can be achieved. So most of enhancement macroblocks should be encoded with HPHR mode. But, too many enhancement macroblocks encoded with HPHR mode also could cause drifting errors at lower enhancement bit rates. Therefore, the decision-making mechanism needs to estimate the potential drifting errors. If the estimated drifting error is larger than a given threshold, this macroblock should be encoded with HPLR mode; otherwise, HPHR mode should be used.

In order to accurately estimate the drifting error, the encoder should have to establish an additional set of motion compensation modules to calculate the effect of errors in the lower enhancement layers. This will increase the computational complexity of the encoder. In fact, accurately calculating the drifting error is not necessary in most applications, a rough estimation is normally good enough. Thus, a simplified decision-making mechanism is proposed to approximately estimate the drifting error.

According to Formula (3), the simplified method estimates the potential drifting error based on the previous low quality prediction and the previous high quality prediction. Obviously, the larger the difference between two temporal predictions is, the larger the quality loss will be caused when the previous high quality reference is not available. In order to control the possible quality loss, the decision-making mechanism defines a criterion as follows,

$$\left\|p_e(n) - p_b(n)\right\| > k \times \left\|x_o - r_e(n)\right\|. \quad (5)$$

Here x_o is the current original image, and *k* is the acceptable loss factor. The factor *k* is an adjustable parameter, which controls the performance of the PFGS scheme at low bit rates and high bit rates. $\left\|x_o - r_e(n)\right\|$ is the mean squared error of the reconstructed high quality reference. When the difference between two temporal predictions is larger than the right-hand side value of Formula (5), this macroblock should be encoded with HPLR mode, since HPHR mode may cause a significant drifting error in this case. But the coding mode of each macroblock is determined before coding process, $r_e(n)$ in Formula (5) is generally not available during the process of mode decision. Therefore, the following criterion is applied to approximate Formula (5)

$$\left\|p_e(n) - p_b(n)\right\| > k' \times \left\|x_o - p_e(n)\right\|. \quad (6)$$

Where $r_e(n)$ is replaced by $p_e(n)$. Since two temporal predictions in Formula (6) are already available, no additional computation is introduced to the PFGS encoder.

It is clear that the proposed decision-making mechanism can be easily implemented to control the MB-based PFGS scheme. The coding mode information of each macroblock should be included in the macroblock header. Since the base layer bit-stream already provides the information about the INTRA mode, only the INTER mode information needs to be encoded in the enhancement bit-stream. A simple VLC table is used to compress them into the enhancement bit-stream.

4. EXPERIMENTAL RESULTS

Extensive experiments have been performed to verify the proposed MB-based PFGS scheme. The coding efficiency of the proposed PFGS scheme is compared with that of the FGS scheme, the non-scalable coding scheme of MPEG-4 standard, and the original frame-based PFGS shown in Figure 1. Some major experimental conditions are shown in Table 1. Other experimental conditions are the same as those specified in the MPEG-4 core experiments[7]. Two sequences are used in this experiment: Foreman with CIF and QCIF format. Only the first frame is encoded as I frame, and the others are encoded as P frames. The encoding frame rate is 10Hz, with TM5 rate controller.

The curves of the average PSNR at each bit rate are given in Figure 4. Although using a high quality reference for prediction, the proposed MB-based PFGS still keeps almost the same performance as the FGS and the frame-based PFGS at low bit rates. At moderate or high bit rates, the quality of the MB-based PFGS is higher than that of the FGS and the frame-based PFGS up to 1.5dB and 1.1dB in average PSNR, respectively.

The curves on PSNR versus frame number are shown in Figure **5**. The FGS bit-stream and the PFGS bit-stream of the Foreman (CIF) are decoded at total bit rate 256kbits/s and 768kbits/s. At low bit rates, the PSNR of each frame of the MB-based PFGS is close to that of the FGS without drifting problem. At high bit rates, the PSNR of the MB-based PFGS is significantly higher than that of the FGS.

Table 1. The experiment conditions

Resolution	CIF	QCIF
Range of motion vectors (pixel)	±31.5	±15.5
Bits to generate the high quality reference	≥20000	≥5000
Acceptable loss factor	1.6	2.3

5. CONCLUSIONS

This paper proposed a flexible, effective and efficient macroblock-based control scheme for the PFGS coding, which can achieve a good trade-off between coding efficiency and drifting error. Three INTER modes are developed for the enhancement MB coding. A decision-making mechanism based on temporal prediction is proposed to properly select the coding mode for each enhancement macroblock. With the proposed techniques, the MB-based PFGS can obtain almost the same performance as the FGS and the frame-based PFGS at low bit rates. When the bit rate becomes moderate or high, the coding efficiency of the MB-based PFGS scheme is higher than that of the FGS and frame-based PFGS up to 1.5dB and 1.1dB, respectively. The coding efficiency gap between the fine granularity scalable and the non-scalable video coding is closing.

6. REFERENCES

[1] A. Luthra, "Need for simple streaming video profile", ISO/IEC JTC1/SC29/WG11, MPEG doc M5800, Noordwijkerhout, Netherlands, March 2000.

[2] J. Lu, "Signal processing for Internet video streaming: A review", SPIE in Image and Video Communication and Processing 2000, vol 3974, 246-258, 2000.

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

[4] W. Li, "Streaming video profile in MPEG-4", IEEE trans. Circuit and systems for video technology, special issue on streaming video (accepted).

[5] J. Arnold, M. Frater and Y. Wang, "Efficient drift-free signal-to-noise ratio scalability", IEEE trans. Circuit and systems for video technology, vol 10, no 1, 70-82, 2000

[6] F. Wu, S. Li and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding", IEEE trans. Circuit and systems for video technology, special issue on streaming video (accepted)

[7] "FGS core experiments", ISO/IEC JTC1/SC29/WG11, MPEG doc M3316, Noordwijkerhout, Netherlands, March 2000.

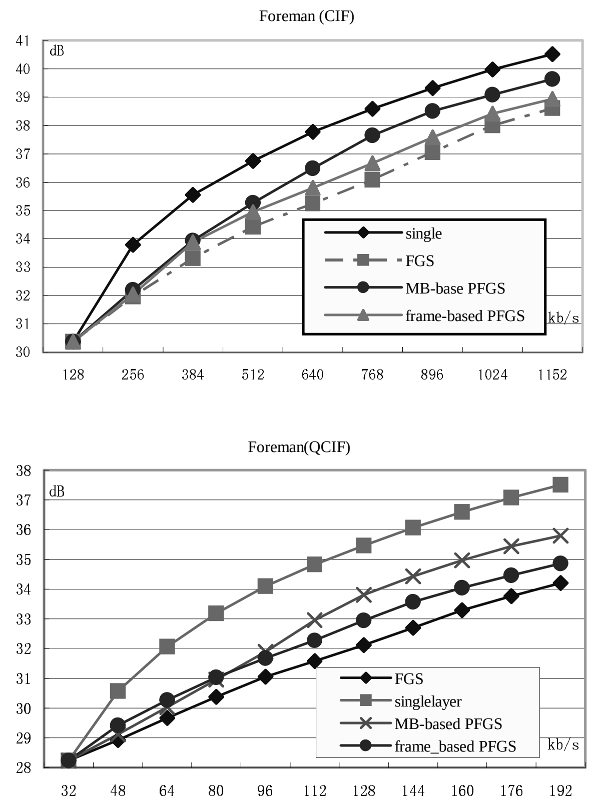

Figure 4. PSNR versus bit rate comparison among FGS, frame-based PFGS, MB-based PFGS and non-scalable coding for foreman (CIF) and foreman (QCIF), respectively.

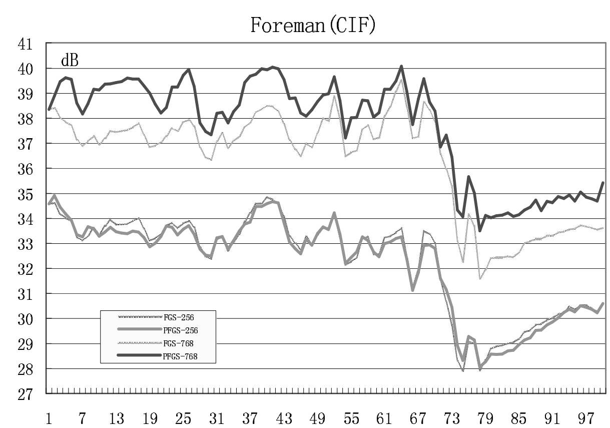

Figure 5. The curves of PSNR versus frame number.