FLEXIBLE P-PICTURE (FLEXP) CODING FOR THE EFFICIENT FINE-GRANULAR SCALABILITY (FGS)

You Zhou[1], Xiaoyan Sun[2], Feng Wu[2], Hong Bao[1], Shipeng Li[2]*

1) University of Science & Technology of Beijing, Beijing, 100080
2) Microsoft Research Asia, Beijing, 100080

ABSTRACT

This paper proposes a flexible P-picture (FlexP) coding technique, where P-pictures are selectively used as references for other pictures. For a given reference bit-rate, the proposed FlexP technique can improve the quality of references by both allocating more bits to the selected P-pictures and reducing the number of selected P-pictures. In case that the reference bits for each selected P-picture are fixed, the proposed technique is able to reduce the reference bit-rate by selecting fewer P-pictures as references, meanwhile limit drifting error at low bit-rates. Therefore, the proposed FlexP technique provides more flexibility on bandwidth adaptation, as well as achieves a good trade-off between high coding efficiency and low drifting error at the enhancement layer especially when the R-D rate allocation is enabled. The experimental results show that the fine-granular scalable video coding with the proposed technique can gain more than 1.0dB.

1. INTRODUCTION

It is well-known that MPEG-4 FGS provides flexible and precise adaptation on channel bandwidth variations and excellent robustness to packet losses and bit errors. These features are eagerly requested by streaming video over the Internet and wireless networks. The typical structure of MPEG-4 FGS is depicted in Figure 1 [1], where the coding of the enhancement layer contains no motion compensation. Just due to the open-loop structure at the enhancement layer, MPEG-4 FGS cannot achieve high coding efficiency at moderate and high bit-rates.

Some schemes, such as PFGS [2], MC-FGS [3] and RFGS [4], have been proposed to improve the coding efficiency of MPEG-4 FGS by introducing part of the enhancement layer into motion compensation loop. In general, they are called jointly as advanced FGS coding, as shown in Figure 2. The shadow regions at the enhancement layer indicate the bits that are used to reconstruct the high quality references. Obviously, the performances of advanced FGS coding are dependent on how many bits of the enhancement layer are introduced into motion compensation loop. The more reference bits there are at the enhancement layer, the better performance the advanced FGS coding can achieve at high bit-rates. However, in case that the reference bits cannot be completely transmitted to the client at low bit-rates, the performance of advanced FGS coding would be no better or even worse than that of FGS due to drifting error.

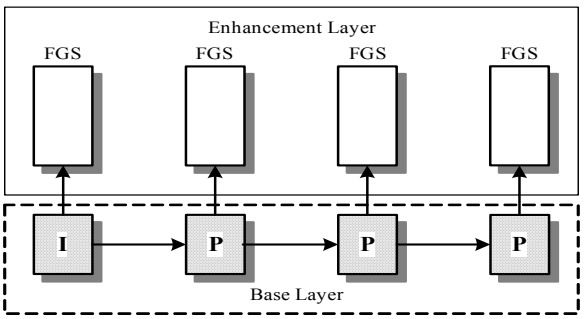


Figure 1: The typical structure of MPEG-4 FGS.

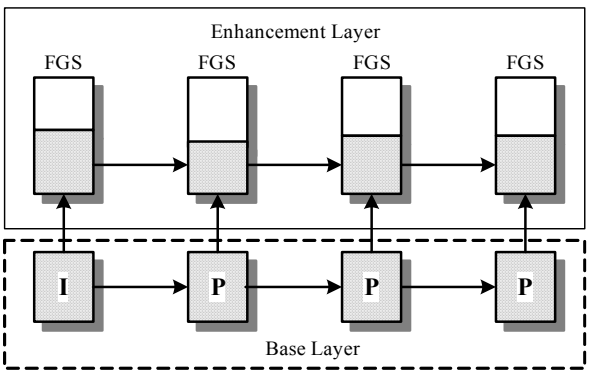


Figure 2: The generic structure of advanced FGS.

To relief the above dilemma, a flexible P-picture coding technique is proposed in this paper. As a matter of fact, some pictures in a video clip may have much stronger correlation. In this case, the coding performance of a picture may have no big difference when it is predicted from either the closest reference or one of other references. Therefore, the proposed FlexP coding suggests that P-

[1] This work has been done while the author is with Microsoft Research Asia.

pictures should be selectively used as references. The reference bit-rate at the enhancement layer can be reduced with fewer P-pictures selected, or the saved bits from non-selected P-pictures are borrowed to improve the reference quality of selected P-pictures without increasing the reference bit-rate. Although B-pictures can be also used for this purpose, it brings additional delay which is unexpected by many real-time and low-latency applications.

The rest of this paper is organized as follows. Section 2 introduces the basic idea of the proposed FlexP coding. The detailed encoding and decoding processes of FlexP are also described in Section 3. Section 4 gives the experimental results of several test sequences by applying the proposed FlexP into the PFGS encoder. Section 5 concludes this paper.

2. THE FLEXP CODING TECHNIQUE

The idea that P-pictures are predicted from other references instead of the closest one was once proposed in the temporal scalable coding [5] and the implementation of VCR functionalities [6]. In this paper, we apply the idea in the fine-granular scalable coding and propose the FlexP coding technique. The typically structure of the proposed FlexP is exemplified in Figure 3. Similar to the common FGS coding, there are two streams: base layer and enhancement layer. Assume that all pictures are coded as P-picture except for the first picture of every group of pictures (GOP).

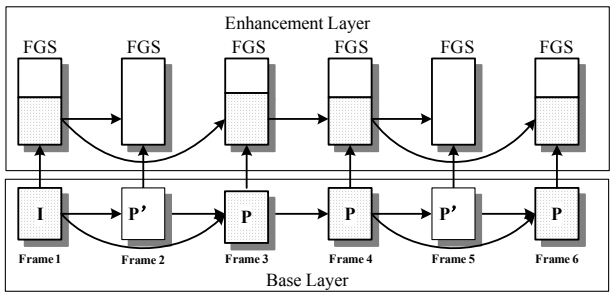


Figure 3: The structure of the proposed FlexP coding.

As shown in Figure 3, there are two types of P-pictures in the proposed FlexP scheme. The first type is normal P-picture (e.g. Frame 3, Frame 4, and Frame 6) denoted by P in Figure 3. The second one is non-selected P-picture (e.g. Frame 2 and Frame 5) denoted by P' in Figure 3. The two types of P-pictures are coded with the same method except that the non-selected P-pictures are not used as references at all. In other words, at least one future picture of the base layer and the enhancement layer is predicted from the normal P-pictures. On the contrary, no picture is predicted from the non-selected P-pictures. A flag is inserted into the coded stream to notify the decoder whether the received data is a normal P-picture or a non-selected P-picture.

For the example given in Figure 3, if the reference bits of each P-picture at the enhancement layer are fixed, the reference bit-rate is significantly decreased because Frame 2 and Frame 5 are not selected. Due to the diminished bit rate of the enhancement reference, the drifting error would be effectively reduced at low bit-rates. Additionally, if the saved bits from Frame 2 and Frame 5 are allocated to other normal P-pictures, the quality of reconstructed references in normal P-pictures will be enhanced, thereby improving the coding efficiency. Thus, the proposed FlexP technique provides more freedom and flexibility to optimize the fine-granular scalable coding.

In general, one coding scheme would suffer from the degradation on coding efficiency if the closest reference is not utilized in P-pictures coding (e.g. in [5] and [6]). It is because that normal P-pictures and non-selected P-pictures are regularly alternate in those schemes, regardless of the correlation of pictures. However in the proposed FlexP technique, only when the similar performance is achieved by predicting from either of previous two references, the closest reconstructed P-picture is destined as a non-selected P-picture. Therefore, the proposed technique has a limited side effect on the coding efficiency.

3. THE ENCODING AND DECODING PROCESS

The encoding and decoding processes of the proposed technique are described in this section.

Similar to the multi-hypothesis motion compensation [7][8], the encoder owns two buffers to save previous two reconstructed references. The encoding process is illustrated in Figure 4. Firstly, as shown in Figure 4 (a), Frame 1 is coded as I-picture and Frame 2 is coded as P-picture by predicting from the reconstructed I-picture. In this case, the encoder can not decide whether Frame 2 is a normal P-picture or not. Instead, both of reconstructions of Frame 1 and Frame 2 are saved in buffers. Secondly, as shown in Figure 3 (b), when Frame 3 is input in the encoder, there are two references ready to be used in coding the input picture. With the same prediction strategy, Frame 3 is predicted separately from the reconstructed references of Frame 1 and Frame 2. The result of motion prediction is evaluated by the following R-D metric

$$E = (SAD_b + \lambda_b R_b) + w(SAD_e + \lambda_e R_e), \quad (1)$$

in which SAD_b and SAD_e are the sum of absolute difference of the predictive residues at the base layer and the enhancement layer, respectively. R_b indicates the number of bits spend in coding the motion data (motion-compensation modes and motion vectors) at base layer, while R_e presents the number of bits used to code mode information at enhancement layer. λ_b and λ_e are the

Lagrange multipliers at the base layer and the enhancement layer. Obviously, both the base layer and the enhancement layer are taken into account in this metric as proposed in [9].

If the *E* of Frame 2 is close to that of Frame 1, i.e., Frame 2 and Frame 1 have a strong correlation, Frame 3 is predicted from the reconstruction of Frame 1; otherwise, Frame 3 is predicted from the reconstruction of Frame 2. As shown in Figure 3 (b), Frame 3 selects the reconstruction of Frame 1 as its reference.

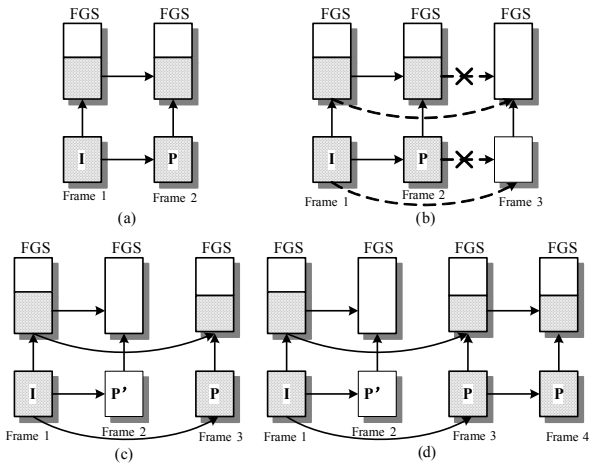


Figure 4: The encoding process of the proposed FlexP.

Clearly, the encoder can't decide the type of Frame 2 until Frame 3 selects its predicted reference. Normally, if Frame 1 is selected as the prediction reference of Frame 3 (as shown in Figure 4 (c)), Frame 2 will be treated as non-selected P-picture, that is, pictures after Frame 3 should use Frame 3 or Frame 1 as reference. In other case, if Frame 3 selects Frame 2 as reference, Frame 2 will be a normal P-picture.

Succeedingly, there are two different methods to code Frame 4. The first one is similar to Frame 3. The type of Frame 3 will be decided by the prediction of Frame 4. The input Frame 4 can be predicted from either Frame 1 or Frame 3, though Frame 1 may not be a good reference for Frame 4 in most cases because the long temporal distance between them. This method provides better flexibility to the advanced FGS coding methods because the encoder can make decision for more pictures on selection between normal P-picture and non-selected P-picture.

The second one is to force Frame 4 to be coded as a normal P-picture so that the decoder can discard the reconstruction of Frame 1. Thus the type of Frame 3 will always be a normal P-picture, due to which the flexibility of the proposed FlexP technique is constrained. In this paper we focus our attention on the former method. The following input pictures are dealt with the similar way as Frame 3 and Frame 4.

Only one buffer is required at decoder side in the proposed FlexP. When the stream of Frame 1 is input in the decoder, the reconstructed reference is saved in the buffer. The input stream of Frame 2 is first decoded and reconstructed as a normal P-picture. However, if the flag indicates it as a normal P-picture, the reconstructed reference is used to update the buffer; otherwise the reconstructed reference is discarded because no other pictures are predicted from it. The buffer still keeps the previous reference for the next picture. The next input frame, no matter it is a normal P-picture or a non-selected P-picture, is always reconstructed as a normal-picture from the reference saved in the buffer. However, the update of the reference buffer is dependent on the type flag.

4. EXPERIMENTAL RESULTS

The experiments have been performed in this section to verify the performance of the proposed FlexP scheme. To readily compare the proposed scheme with the advanced FGS scheme, a FlexP scheme has been implemented based on H.264-based PFGS. Three test sequences, Foreman, News and Carphone with QCIF format, are used in the experiments.

In the base layer, only the first frame was coded as I-picture, and the others were coded as P-picture. The value of *E* in equation (1) is used to decide which reference should be selected to predict the input picture. Let E_2 denotes the value of E of the closest reference frame (such as Frame 2 in Fig.4b) and E_1 denotes the value of the latest selected reference frame (such as Frame 1 in Fig.4b). If E_2 / E_1 is more than 0.91 (this is an empirical value), the input frame is predicted from the latest selected reference frame; otherwise the frame is predicted from the closest reference frame.

The base layer codec is H.264 JM 6.1e. Some main experimental conditions are given here. Quarter-pixel motion estimation is applied and the search range is ± 16 pixels. Hadamard transform, all coding modes and CABAC are enabled. Multiple references and RDO are disabled. The QP is set as 35 for all frames. The frame rate of the base layer and the enhancement layer is 30 Hz.

In the PFGS coding, 7200 bits are used to reconstruct the high quality reference. Uniform truncate is employed to allocate bits evenly to every frame. In the proposed FlexP coding, fixed quantization parameter, 35, is used in base layer coding. In enhancement layer, since the non-selected P pictures do not cause any drifting error, bits are first evenly allocated to I-picture and normal P-pictures at low bit-rates. And if the available bit-rate is more than the reference rate, the rest bits are evenly allocated to all pictures.

Three schemes, PFGS, FlexP with the fixed reference rate and FlexP with the fixed bits in each normal P-picture, are compared in the experiments. The fixed bits in each normal P-picture are 7200 and the fixed reference rate is 216 kbps. All PSNR verses rate curves are shown in Figure 5. For the FlexP scheme with the fixed bits in each normal P-picture, the FlexP curves show the small drifting error at low bit rate and good performance at moderate bit rates. However, the performance at high bit-rates is worse than that of PFGS. For the FlexP scheme with the fixed reference rate, the FlexP (constant ref. bit-rate) curves show the performance much better than that of PFGS. The coding efficiency gain is up to 1.0dB in Foreman, 2.0dB in News and 1.0 dB in Carphone. An interesting phenomenon is observed in Figure 5. The performance of this scheme is also better than that of PFGS at low bit-rates. The reason is that non-selected P-pictures do not bring any the accumulation of mismatches.

5. CONCLUSIONS

This paper proposes a FlexP coding technique to advanced FGS. It provides much flexibly and freedom to optimize the advanced FGS coding, thereby achieving a good trade-off between high coding efficiency and low drifting error. Compared with the case of all picture coded as normal P-picture, the proposed FlexP coding can improve coding efficiency more than 1.0dB. The proposed technique is still on development and needs more studies, such as how to re-allocate the reference bit-rate to every picture at the enhancement layer and how to optimally truncate the generated stream.

REFERENCES

[1] W. Li, Overview of Fine Granularity Scalability in MPEG-4 video standard, *IEEE trans. on CSVT*, vol. 11, no 3, 301-317, 2001.

[2] F. Wu, S. Li, Y.-Q. Zhang, A framework for efficient progressive fine granular scalable video coding, *IEEE trans. on CSVT*, vol. 11, no. 3, 332-344, 2001.

[3] M. Schaar, H. Radha, Adaptive motion-compensation fine-granular-scalability (AMC-FGS) for wireless video, *IEEE trans. on CSVT*, vol. 12, no. 6, 360-371, 2002.

[4] H. Huang, C. Wang, T. Chiang, A robust fine granularity scalability using trellis-based predictive leak, *IEEE trans. on CSVT*, vol. 12, no. 6, 372-385, 2002.

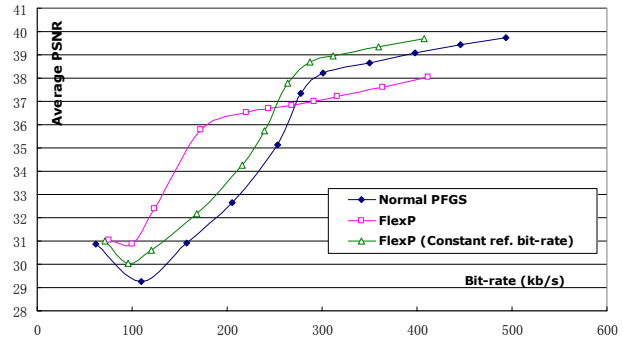
[5] S. Wenger, Temporal scalability using P-pictures for low-latency applications, *IEEE workshop on Multimedia Signal Processing*, 559-564, 1998.

[6] A. Tourapis, F. Wu, S. Li, Enabling VCR functionalities in streaming media, *International Conference on Information Technology: Research and Education*, 1-5, 2003.

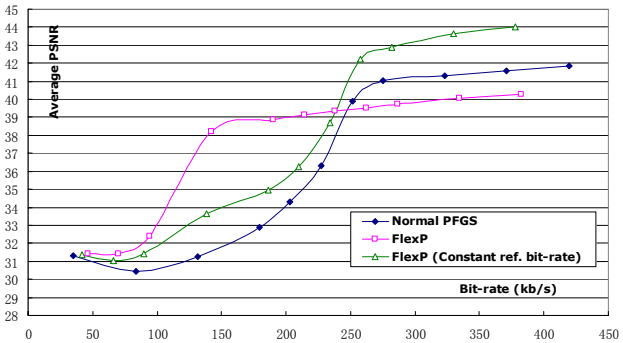
[7] T. Wiegand, X. Z. Zhang, B. Girod, Long-term memory motion-compensated prediction, *IEEE trans. on CSVT*, vol. 9, no. 1, 70-84, 1999.

[8] M. Flierl, T. Wiegand, B. Girod, A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction, *Data Compression Conference*, 239-248, 1998.

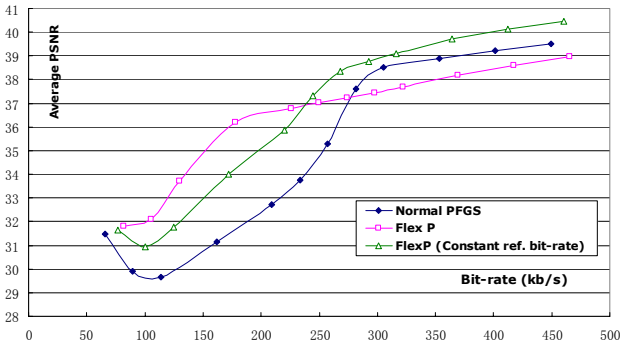
[9] Z. J. Yang, F. Wu, S. Li, Rate distortion optimization in the scalable video coding, *ISCAS 2003*, vol. 2, 884-887, 2003.



a) Foreman



b) News



c) Carphone

Figure 5: The PSNR verses rate curves of the PFGS scheme and the proposed Flex P schemes.