A Framework for Efficient Progressive Fine Granularity Scalable Video Coding

Feng Wu, *Member, IEEE*, Shipeng Li, *Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*—In this paper, a basic framework for efficient scalable video coding, namely progressive fine granularity scalable (PFGS) video coding is proposed. Similar to the fine granularity scalable (FGS) video coding in MPEG-4, the PFGS framework has all the features of FGS, such as fine granularity bit-rate scalability, channel adaptation, and error recovery. On the other hand, different from the FGS coding, the PFGS framework uses multiple layers of references with increasing quality to make motion prediction more accurate for improved video-coding efficiency. However, using multiple layers of references with different quality also introduces several issues. First, extra frame buffers are needed for storing the multiple reconstructed reference layers. This would increase the memory cost and computational complexity of the PFGS scheme. Based on the basic framework, a simplified and efficient PFGS framework is further proposed. The simplified PFGS framework needs only one extra frame buffer with almost the same coding efficiency as in the original framework. Second, there might be undesirable increase and fluctuation of the coefficients to be coded when switching from a low-quality reference to a high-quality one, which could partially offset the advantage of using a high-quality reference. A further improved PFGS scheme can eliminate the fluctuation of enhancement-layer coefficients when switching references by always using only one high-quality prediction reference for all enhancement layers. Experimental results show that the PFGS framework can improve the coding efficiency up to more than 1 dB over the FGS scheme in terms of average PSNR, yet still keeps all the original properties, such as fine granularity, bandwidth adaptation, and error recovery. A simple simulation of transmitting the PFGS video over a wireless channel further confirms the error robustness of the PFGS scheme, although the advantages of PFGS have not been fully exploited.**

***Index Terms*—Bandwidth adaptation, bit-plane coding, error recovery, fine granularity scalability, layer-based video coding.**

I. INTRODUCTION

TRANSMITTING digital video over the Internet or wireless channels encounters two major problems: bandwidth fluctuation and packet-losses or errors. It is very desirable to have a video-coding scheme that can adapt to the channel conditions and recover gracefully from packet losses or errors. One good solution to these problems is to compress and transmit a video sequence with scalability. In this paper, we will focus on discussing scalable compression. Block discrete cosing transform (DCT) and wavelet have been two dominant transform techniques in the existing video-coding schemes. Although the use of discrete wavelet transform (DWT) in scalable coding has received much attention for its inherent multi-resolution and progressive characteristics in recent years [1]–[4], block DCT transform coders enjoyed the success due to their low complexity in implementation and their reasonably good performance. Therefore, a number of DCT-based techniques have been proposed for implementing scalable coding within the framework of the current coding standards, such as MPEG-2, and H.263 plus [5], [6]. When MPEG-4 issued a call for proposals for achieving fine granularity scalability (FGS) in video coding, three types of techniques were submitted, namely, bit-plane coding of the predicted DCT residue [7], [8], wavelet coding of image residue [9]–[11], and matching-pursuit coding of the predicted DCT residue [12], [13]. After several core experiments, the bit-plane coding of the DCT residue was accepted by MPEG-4 standard as a promising FGS coding scheme for the streaming video profile.

In the current FGS coding scheme, an encoder using the motion-compensated DCT transform coding which can be compatible to other standards, such as MPEG-2, MPEG-4, and H.263, etc., generates a base-layer video as the lowest quality layer. Generally, the base-layer video can be transmitted in a well-controlled channel to minimize errors or packet-losses, or in other words, the base layer can be encoded in a way to fit in the minimum channel bandwidth. The residues between original DCT coefficients and dequantized DCT coefficients of the base layer form the enhancement bitstream with the bit-plane coding technology, which can provide an embedded bitstream and fine granularity scalability. Fine granularity means that the enhancement bitstream can be decoded at any length. The number of enhancement layers produced by the FGS coding scheme is not fixed, but based on the number of bit planes needed to represent the residues in binary format. Each enhancement layer contains increasingly more detailed video data to enhance the base layer. The quality of video thereby improves with each enhancement layer.

One major feature of FGS coding scheme is that the base layer and all enhancement layers in a predicted frame are always predicted from the reconstructed version of the base layer in the reference frame. Therefore, the FGS coding scheme provides excellent error recovery from occasional data losses or errors in enhancement layers. By predicting all enhancement layers from the base layer, losses or corruptions of one or more enhancement layers during transmission have no effect to the frames followed. However, since the prediction is always based on the lowest quality base layer, the coding efficiency of the FGS scheme is not as good as, and sometimes much worse than, traditional SNR scalability schemes, such as in [15]. On the other hand, in

Manuscript received June 15, 2000; revised December 7, 2000. This paper was recommended by Guest Editor A. Luthra.

The authors are with Microsoft Research China, Haidian District, Beijing 100080, China (e-mail: fengwu@microsoft.com; spli@microsoft.com; yzhang@microsoft.com).

Publisher Item Identifier S 1051-8215(01)01856-0.

the traditional SNR scalability schemes, same layer references are used to provide better predictions, which in turn normally provides better coding efficiency. But once there is an error or packet loss in the enhancement layers, it would propagate to the end of a group of picture (GOP) and would cause a serious drifting problem in the higher layers of the following predicted frames. Even though there may be sufficient bandwidth available later on, the decoder could not recover to the highest quality until another GOP starts. Therefore, the traditional SNR scalability schemes are normally only suitable for simulcast in stable channels.

In order to improve the coding efficiency of FGS, a basic framework for more efficient scalable video coding was first proposed to MPEG-4 in [16]. We call this basic framework as progressive fine granularity scalable (PFGS) video coding. Similar to FGS, PFGS coding scheme also encodes video frames into multiple layers, including a base layer of relatively lower quality video and multiple enhancement layers of increasingly higher quality video. (Some also refer to all the enhancement layers as a single enhancement layer with multiple bit planes, but we prefer to refer to them as multiple layers.) However, in the PFGS framework we try to use several high-quality references for the predictions in the enhancement-layer encoding rather than always use the base layer. Using high-quality references would make motion predictions more accurate, thus it could improve the coding efficiency. Our experimental results show that the PFGS scheme can achieve consistently better coding efficiency than the FGS scheme while keeping all the properties of FGS, such as fine granularity scalability, channel adaptation, and error recovery.

There are still several issues to be addressed in the basic PFGS framework proposed in [16]. First, it needs multiple extra frame buffers to save multiple reconstructed layers as references, which increases the memory cost and computational complexity of the PFGS encoder and decoder. Fortunately, not every reference layer makes the same contribution to the improvement of coding efficiency. Only a few among the reference layers make significant contributions to improving the coding efficiency, while others have little effect. How to choose minimal number of reference layers to achieve high coding efficiency remains an open problem. Another problem is the fluctuation and increase of DCT coefficients when switching from a low-quality reference to a high-quality one. An efficient approach that can take full advantage of a high-quality reference without causing any fluctuation should be investigated to further improve the coding efficiency of the basic PFGS framework.

The rest of this paper is arranged as follows. Section II briefly introduces the basic idea to build the PFGS framework. In order to reduce the memory cost and computational complexity of the PFGS framework, a simplified framework with one extra frame buffer is proposed. Section III analyzes the reasons why multiple different quality references might cause the fluctuation and increase in residue coefficients. An improved technique integrating the high-quality reference into low enhancement-layer coding is proposed in this section. The encoder and decoder of the PFGS framework are illustrated and discussed in Section IV. Experimental results presented in Section V demonstrate the advantages of the proposed PFGS framework. A simple simulation of transmitting the PFGS video over a wireless channel further confirms the error robustness of the PFGS scheme in Section VI. Finally, Section VII concludes this paper.

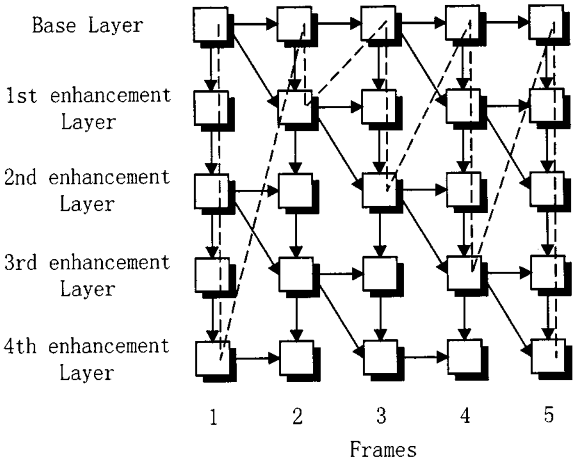


Fig. 1. Proposed PFGS framework.

II. BASIC PFGS FRAMEWORK

A. Basic Ideas to Build the PFGS Framework

As discussed in the previous section, the FGS video-coding scheme provides very good bandwidth adaptation and error-recovery properties, but it sacrifices coding efficiency. On the other hand, the SNR scalability traditional schemes have good coding efficiency, but they lose the bandwidth adaptation and error recovery properties. Is there a new framework that could provide a good balance between coding efficiency and all the scalability properties? In this paper, we try to find such a general framework that can keep the properties of FGS coding scheme, such as fine granularity scalability, channel adaptation, and error recovery, while using as many predictions from the same reference layer as possible as in the traditional SNR scalability schemes.

There are two key points in designing such a framework. The first point is to use as many predictions from the enhancement reference layers as possible (for coding efficiency), instead of always using base layer as in the FGS scheme. Since the quality of an enhancement layer is better than that of the base layer, such a framework makes motion compensation as accurate as possible for any given video layer to maintain coding efficiency. The second point is to keep a prediction path from the base layer to the highest quality layers across several frames (for error recovery and channel adaptation). This will make sure that the coding schemes can gracefully recover from losses or errors. Lost or erroneous higher quality enhancement layers may be automatically reconstructed from lower layers gradually over a few frames with such a prediction path.

Fig. 1 conceptually illustrates such an exemplary framework for efficient video coding with no drifting problem. In the illustrated framework, frame 2 is predicted from the base layer and even enhancement layers of frame 1 (i.e., the second and fourth enhancement layers). Frame 3 is predicted from the base layer and odd enhancement layers of frame 2 (i.e., the first and third enhancement layers). Frame 4 is again predicted from the base layer and even enhancement layers of frame 3, and so on. It is obvious that three highest enhancement layers in frame 2 are predicted from high-quality reference layers in frame 1, rather than from the base layer in frame 1. Since the quality of an enhancement layer is higher than that of the base layer, the framework shown in Fig. 1 provides more accurate motion prediction to improve the coding efficiency. On the other hand, many prediction paths from the lowest quality layer to the highest quality layer are preserved. For instance, the base layer of frame 1, the first enhancement layer of frame 2, the second enhancement layer of frame 3, the third enhancement layer of frame 4, and the fourth enhancement layer of frame 5 constitute such a complete path.

The advantages of the proposed framework are obvious when it is applied to video transmission over the Internet or wireless channels. The encoded bitstream can adapt to the available bandwidth of the channel without any drifting problem. Fig. 1 shows an example of this bandwidth adaptation process. The dashed line traces the transmitted video layers. Note that at frame 2, there is a reduction in bandwidth. At this frame, the transmitter (server) simply drops the bits of higher layers (from the second to the fourth enhancement layers). However, after frame 2, as the bandwidth increases, the transmitter simply transmits more layers of video bits. After three frames (at frame 5), the decoder side can obtain up to the highest quality video layer again. Note that in all these operations, no re-encoding or retransmission of the video bitstream is required. Similarly, when several enhancement layers in one or more frames have packet losses or errors, the recovery process is the same as that of bandwidth adaptation case. We can see that the processes of bandwidth adaptation and error recovery are a graceful and gradual one, which progressively recover across several frames. Therefore, this framework is called PFGS video coding.

Fig. 1 exemplifies a case where the group depth is 2. Group depth defines how many layers may refer back to a common reference layer. The group depth can be changed in each frame. If the group depth is 1, the proposed framework essentially becomes the traditional SNR scalability schemes as in [15]. If the group depth is equal to the total number of layers, the proposed framework essentially represents FGS in [7], [8]. In fact, the above description is just a special case of a more general case where in each frame the reference layers used for prediction can be randomly assigned as long as a prediction path from lowest layer to highest layer is maintained across several frames.

B. The Simplified PFGS Framework

It is very clear that compared to FGS, the implementation of such a PFGS framework as shown in Fig. 1 needs several extra frame buffers to save the reconstructed enhancement layers as references. In Fig. 1, there are two extra buffers needed for encoding each frame. For example, the second and fourth enhancement layers in frame 1 are used as references for the enhancement-layer coding in frame 2, and the first and third enhancement layers in frame 2 are also used as references for the enhancement-layer coding in frame 3, and so on. In fact, the number of extra frame buffers increases as the number of enhancement layers increases. If we could reduce the number of extra frame buffers to a minimum number while still maintaining almost the same coding efficiency, it will be a very significant complexity reduction for the PFGS implementation, especially for the hardware implementation.

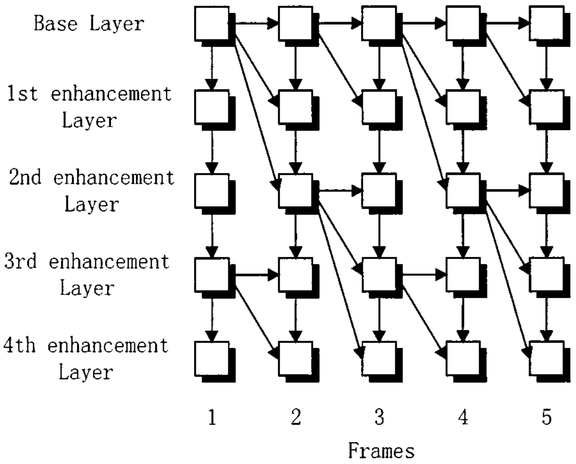


Fig. 2. Simplified PFGS framework with only two buffers.

Although the PFGS framework could use as many additional reference buffers as possible to achieve greater coding efficiency improvement, we still prefer a much simplified PFGS framework that can provide a good tradeoff between coding efficiency versus the memory cost and computational complexity. Fortunately, not all enhancement layers are well suited to be used as references. Only a few enhancement layers as references can make a significant contribution to improving the coding efficiency, whereas others have little effect. Generally, the lower enhancement layers are not good references. Since these layers may comprise errors with large magnitudes caused by motion across frames, the correlations of lower enhancement layers between adjacent frames are weak. Here, the correlation between two same layers in adjacent frames is defined as the absolute sum of binary difference of two bit planes. The larger the absolute value is, the weaker the correlation between them. The higher enhancement layers are not good references either. Firstly, the bit rate of higher enhancement layers is too high for most applications. The gain by using a high-quality reference will appear only at very high bit rate. Secondly, the small magnitude errors encoded in these layers may be just produced by noise. Therefore, the correlation of higher enhancement layers between adjacent frames are weak too. Only by using the middle enhancement layers as references can we achieve the most coding efficiency improvement, since the DCT coefficients in the middle enhancement layers show strong correlations between adjacent frames.

A simplified PFGS framework with only two frame buffers could offer a good tradeoff between coding efficiency versus the extra memory cost and computational complexity. In the simplified PFGS framework, the first frame buffer is used to save the reconstructed base layer in a previous frame as a reference for the base layer and the lower quality enhancement layers in a predicted frame. The second frame buffer is used to save a reconstructed enhancement layer in a previous frame as a reference for the higher quality enhancement layers. An exemplary framework is shown in Fig. 2. The base layer and the first two en-

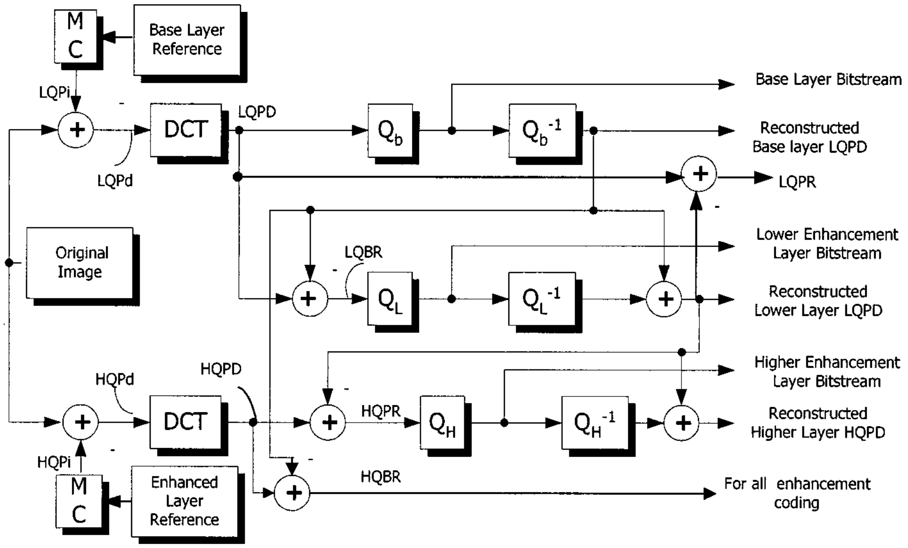


Fig. 3. Illustration of the relationships of all the differences in the PFGS framework.

hancement layers of frame 2 are predicted from the base layer of frame 1, and other higher quality enhancement layers of frame 2 are predicted from the third enhancement layer of frame 1. Instead of using a fixed enhancement layer, the reference alternates between two different enhancement layers to form a complete prediction path from the base layer to the highest enhancement layer for error recovery and channel adaptation. Which enhancement layer to use as the high-quality reference depends on the contents of the video sequence and the bit rate of the base layer. How to choose such an enhancement layer is an optimization issue in the encoding process. The index of the enhancement layer selected as the reference can be encoded as part of the video bitstream.

Since the framework in Fig. 2 uses two reference layers for the prediction, it produces two sets of predicted DCT coefficients: 1) a first set of predicted DCT coefficients are prediction errors formed by referencing the base layer, which is a low-quality reference layer and 2) a second set of predicted DCT coefficients are prediction errors formed by referencing a higher quality enhancement layer. The first set of predicted DCT coefficients are encoded in the base layer and lower enhancement layers, and the difference between the second set of predicted DCT coefficients and those reconstructed from the base layer and low enhancement layers is encoded to form higher enhancement layers.

III. IMPROVEMENTS TO THE PFGS FRAMEWORK

A. Potential Coding Inefficiency Due to Predictions from Two References

The goal of using the second high-quality reference is to reduce the bit-rate of the higher enhancement layers. As shown in Fig. 2, the second set of predicted DCT coefficients will have statistically lower magnitude compared with the first set of predicted DCT coefficients because its reference is of higher quality and hence closer to the original image. Theoretically, we also expect that the differences between the second set of predicted DCT coefficients and those reconstructed from the base layer and lower enhancement layers are smaller than the residues of the first set of predicted DCT coefficients after the base-layer and lower enhancement layers encoding. Lower DCT differences translate into fewer coding layers, therefore, resulting in better coding efficiency. However, the expectation is only valid statistically.

The displaced frame difference (DFD) image is defined as the difference between the original image and the motion predicted image. With the linear DCT transforms, the DCT coefficients of the DFD image are equal to the difference between the DCT coefficients of the original image and that of the predicted image. There are two kinds of differential operations in PFGS coding schemes for improving the coding efficiency. The first kind of differential operations are essentially the normal motion compensation operations that are used to generate the DFD in order to reduce the temporal redundancy. These subtractive processes are denoted with horizontal solid-line arrows between adjacent frames in Fig. 2. The second kind of differential operations are performed on the predicted DCT coefficients within one frame after switching from a lower quality reference to a higher quality one to reduce DCT coefficient redundancy. Normally, the results of the second kind of differential operations are called residues in this paper. For convenience, we define the following terminology for referencing these differences:

LQPi	low-quality predicted image that is generated by motion compensation from the lower quality reference;
LQPd	difference between LQPi and the original image;
LQPD	DCT coefficients of LQPd;
LQBR	DCT residues that are produced by subtracting the already encoded (quantized and then dequantized) DCT coefficients in the base layer from LQPD coefficients;
LQPR	DCT residues that are produced by subtracting the already encoded DCT coefficients in the previous layers from the LQPD coefficients;
HQPi	high-quality predicted image that is generated by motion compensation from the higher quality reference;
HQPd	differences between HQPi and the original image;

HQPD	DCT coefficients of HQPd;
HQPR	DCT residues that are produced by subtracting the already encoded DCT coefficients in the previous layers from the HQPD coefficients;
HQBR	DCT residues that are produced by subtracting the already encoded DCT coefficients in the base layer from the HQPD coefficients.

Specifically, Fig. 3 illustrates the relationships among the above terminology. It is obvious that the HQPi will produce lower DFD DCT coefficients compared with the LQPi because the reference is of higher quality. However, the dynamic range of the HQPD coefficients is not necessary always less than that of the LQPD coefficients. For some instances, the magnitude of individual HQPD coefficients may actually increase compared with that of the LQPD coefficients due to the nonideal motion compensation. Moreover, in order to reduce the redundancy between LQPD coefficients and HQPD coefficients and to further improve the coding efficiency, normally only the difference between the HQPD and the reconstructed low-layer LQPD; that is, the residue HQPR is coded in higher enhancement layers. Although doing so generally will reduce the energy of the coefficients to be coded, the dynamic range of the difference actually would increase (causing more fluctuation) and an additional sign map would be required to code the difference HQPR.

Both the undesired fluctuation in magnitude and the additional sign bit are particularly inefficient for bit-plane coding. First, if the undesired fluctuation and increase in magnitude exceed the range represented by residual bit planes, it would seriously affect the coding efficiency of bit-plane coding. For example, assume that three bit planes are used to encode the LQPR in higher enhancement layers when referencing a low-quality layer. However, after switching to a higher quality reference, if the absolute value of an individual HQPR coefficient exceeds 7, one or multiple additional bit planes have to be inserted between lower and higher enhancement layers to represent the excess range. Secondly, the sign of HQPR may be different from that of LQBR. Since the sign bit of every coefficient is encoded after the MSB of that coefficient, for those coefficients whose MSB's have been encoded in base layer and lower enhancement layers, new signs have to be encoded again.

B. A More Efficient PFGS Framework

Ideally, in order to avoid the excess fluctuation and increase in magnitude of the prediction residues mentioned above, the bit planes encoded in lower and higher enhancement layers should be from one set of prediction residues. In addition, the extra sign layer is also completely avoided by doing so. In the baseline FGS case, the bit planes of all enhancement layers are indeed always from the same set of residues LQBR. However, since it always uses base-layer video as references, the overall coding efficiency is limited, especially for higher enhancement layers. On the other hand, in the PFGS framework shown in Fig. 2, only the lower enhancement layers include bit planes from the LQBR, while the higher enhancement layers encode bit planes from HQPR as shown in Fig. 3, causing exactly the same problem.

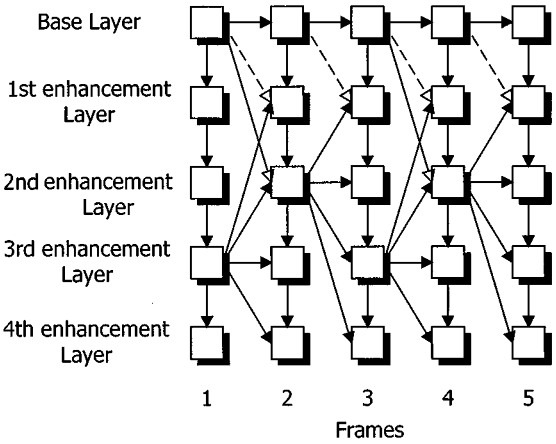


Fig. 4. Improved PFGS framework. Solid arrows are for prediction references and hollow arrows with solid lines for reconstruction references, hollow arrows with dashed lines are for reconstruction of lower layers when the previous enhancement reference layer is not available.

A conditional replenishment method described in [17] can be used to eliminate this kind of fluctuation when switching from a lower quality reference to a higher quality one. In the conditional replenishment scheme, not all the LQPR coefficients are replaced by the HQPR coefficients. The lower layer prediction coefficients LQPR are conditionally replaced by the higher layer prediction coefficients HQPR depending on the values of the reconstructed lower layer LQPD. If the reconstructed lower layer coefficient LQPD is zero then the corresponding coefficient in LQPR is replaced with that in HQPR. If the reconstructed coefficient is not zero, then no replacement is done. Though the conditional replenishment can solve the fluctuation problem, it partially loses the advantages of using a high-quality reference. Non-zeros in the reconstructed lower layer coefficients LQPD are an essential condition to cause fluctuation, but not a sufficient one. In many cases, even though there are nonzeros in the reconstructed lower layer coefficients, the residues can still be reduced by replacing the prediction coefficients LQPR with HQPR. On the other hand, the conditional replenishment method essentially needs a new reference mixed with the lower quality reference and the higher quality reference in DCT domain. For the decoder, since only two predicted image LQPi and HQPi are available, two extra DCT transforms will be needed in order to get the corresponding LQPD and HQPD, thus the computational complexity of a decoder will be increased. Therefore, a more efficient and simple approach that can take full advantage of a high-quality reference without causing any fluctuation should be investigated to further improve the coding efficiency.

We propose to solve this problem within the PFGS framework. The improved PFGS framework based on a new structure is given in Fig. 4. In this improved framework, while the base layer encoding is still the same as that in the baseline FGS and PFGS framework in Fig. 2, all enhancement layers encode differences between the HQPD and the dequantized LQPD from base layer, i.e., HQBR. Note that now the lower enhancement layers contain the first few most significant bits in the HQBR coefficients instead of that of the LQBR coefficients as in the baseline FGS and the PFGS framework in Fig. 2. Since all enhancement layers always encode the same set of coefficients

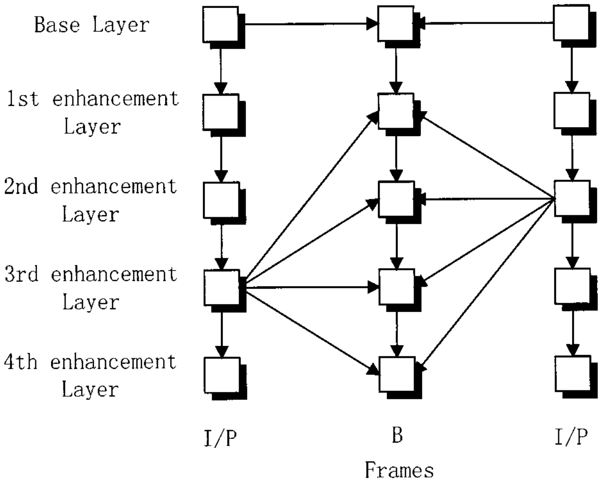


Fig. 5. B frames encoding in the improved PFGS framework.

HQBR in the improved PFGS framework, there is neither sign change nor fluctuation among any enhancement layers.

The lower enhancement layers, which also encode the differences between HQPD and reconstructed base layer LQPD, would introduce several problems. It seems that this would cause drifting errors if the HQPi reference were not available at the decoder. Moreover, this seems to destroy the channel adaptation and error recovery properties with FGS/PFGS. Indeed, when the HQPi reference is not available at the decoder, we have to use the LQPi reference instead; this would introduce some errors in lower enhancement layers due to the different references used in the encoder and the decoder. In fact, the overall quality loss in the lower enhancement layers is very small, because a better prediction compensates most of the loss caused by using different references.

How to minimize the quality loss in the lower enhancement layers is not important. It is much more important to find a way to prevent these errors in one frame from propagating to other frames. Fortunately, the framework shown in Fig. 4 suggests a scheme to solve the potential error-drifting problem completely. The key here is to make sure that the encoder and decoder have the same reconstructed references for any future frame prediction although the reconstructed reference may not have the best quality it could get if reconstructed using a high-quality reference. We will show this through an example. As in FGS, we still assume there is no error in the base layer. In the decoder end, if there are packet losses or errors in the third enhancement layer in frame 1 which is used in the encoder end to get the HQBR coefficients, all the enhancement layers in frame 2 will have to use the base layer in frame 1 as a reference. Of course, there would be some quality loss by doing so. However, as long as in both the encoder end and the decoder end the reconstruction of the second enhancement layer of frame 2 always uses the base layer of frame 1 as the reference, then the errors in the reconstruction of second enhancement layer could not further propagate to any frames followed. The unique feature of this improved framework is that in a prediction frame, the reference used for prediction could be different from the reference used for reconstruction. This feature prevents the error drifting and preserves all the bandwidth adaptation and error recovery features of PFGS. Moreover, it brings more coding efficiency gain in the higher enhancement layers. In Fig. 4, for example, we notice that the second enhancement layer in frame 2 can be always reconstructed by referencing the base layer of frame 1 to prevent the error drifting into future frames. However, if the third enhancement layer of frame 1 is available, a better quality second enhancement layer of frame 2 can still be reconstructed with it for display purpose only. The reconstruction of display image can be different from that of reference image is yet another feature of the improved framework.

The architecture of encoding B frames using the improved framework is shown in Fig. 5. The bi-directional motion estimation determines the type of motion compensation and motion vectors by referencing the original image of a previous intra (I) or prediction (P) frame and the original image of the next I or P frame. The LQPD coefficients of the B frame are formed by referencing the base layer in a forward reference frame and/or the base layer in a backward reference frame. At the same time, the HQPD coefficients of the B frame are formed by an enhancement layer in a forward reference frame and/or the enhancement reference in a backward reference frame. Similar to P frames, all the enhancement layers in a B frame are encoded with the difference between HQPD and dequantized LQPD from the base layer. Because none of the enhancement layers in a B frame is used as a reference for other frames, the errors in lower enhancement layers of B frames have no effect to any other frames. However, when there are enhancement reference layers available, they can be certainly used to produce better quality B frames.

IV. IMPLEMENTATION OF PFGS ENCODER AND DECODER

In the previous section, we discussed an improved PFGS framework. How to implement the encoder and decoder based on the improved PFGS framework will be the focus of this section. Firstly, an encoder with two reference buffers for video prediction is given in Fig. 6. Frame Buffer 0 is used to save the reconstructed base layer in a previous frame as a reference for the base layer coding. Frame Buffer 1 is used to save a reconstructed enhancement layer in a previous frame as a reference for coding all enhancement layers.

The base layer encoding is the same as that of base-line FGS, which can be compatible with other standards, such as MPEG-2, MPEG-4 and H.263. Motion estimation (ME) module gets the motion vectors between two adjacent original frames and outputs its results to two motion compensators (MCs). The first motion compensator predicts the image by referencing the reconstructed base layer in Frame Buffer 0. The second motion compensator predicts image by referencing a reconstructed enhancement layer in Frame Buffer 1. After the first motion compensation and DCT transform of the DFD image, we obtain LQPD coefficients in base layer encoder as shown in Fig. 6. The LQPD coefficients are quantized by scalar quantization and compressed by VLC into the base layer bitstream. Generally, the step size of scalar quantizer is large in order to generate a relatively short bitstream.

On the other hand, the second motion compensator and the second DCT module generate the HQPD, which are the DCT

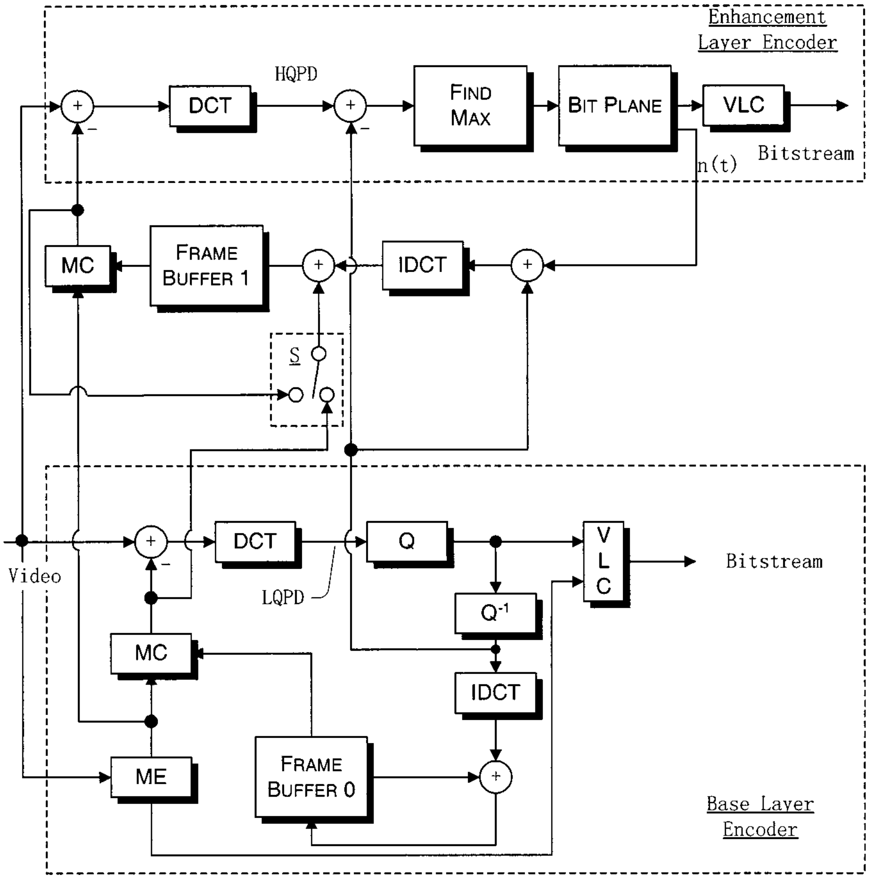


Fig. 6. Encoding diagram of the improved PFGS framework.

transforms of the DFD image with a high-quality reference. The differences (residues) between HQPD and the reconstructed LQPD from the base layer are encoded in all the enhancement layers. The bit plane coding technique is used in the enhancement layers to provide an embedded bitstream and fine granularity scalability. For the improved PFGS framework, the difference now is that all the enhancement layers always use an enhancement layer in the previous frame as a reference as opposed in the base-line FGS case that all enhancement layers always use the base layer in the previous frame as a reference. The maximum absolute value of the differences determines the maximum number of bit planes in a frame. The 64 absolute values in a 8×8 difference block are arranged in a zigzag order into an array. A bit plane is defined as an array of 64 bits which are taken from each of the 64 absolute values at the same significant bit position. This implies that the quantization steps of the enhancement layers are a series of factor 2^i, where i is from the maximum number of bit planes to 0. For the lowest enhancement layer, i equals to the maximum number of bit planes and for the highest enhancement layer i equals to 0. For each bit plane of each block, (RUN, EOP) symbols are formed and encoded using variable length codes to produce the enhancement bitstream for that bit plane. EOP stands for the "End Of Plane." The sign bit of each difference coefficient is encoded with one bit following the most significant bit (MSB) of that coefficient as the coefficient's MSB gets encoded. The binary "0" denotes a positive difference and binary "1" denotes a negative one.

The enhancement layer in the current frame used for the prediction of enhancement layers in the next frame is reconstructed using the first $n(t)$ bit planes in the bitstream and the reference of either the base layer or the enhancement reference layer in the previous frame which is controlled by the switch $\underline{S}$. If the enhancement reference layer for the next frame is a higher layer than the enhancement reference layer for the current frame, i.e., $n(t) > n(t-1)$, then the enhancement reference layer in the previous frame will be used to reconstruct the enhancement reference layer for the next frame. Otherwise, the base layer in the previous reference frame will be used to reconstruct the enhancement reference layer for the next frame.

Fig. 7 gives a diagram of a PFGS video decoder with two reference buffers. The two-buffer configuration offers a good tradeoff between coding efficiency and extra cost in memory and computational complexity. The first frame buffer is used to save the reconstructed base layer in a previous frame as the reference for the base layer. It can also be used as the reconstruction reference for some lower quality enhancement layers to generate a display image when the higher quality reference in a previous frame is not available due to errors or packet losses. A second frame buffer is used to save the reconstructed enhancement layer in a previous frame as the prediction reference for all enhancement layers. The decoder is very similar to a base-line FGS decoder but with additional modules to reconstruct and save a middle enhancement layer as a second reference. The switch $\underline{S}$ in the decoder is used to control which buffer to be

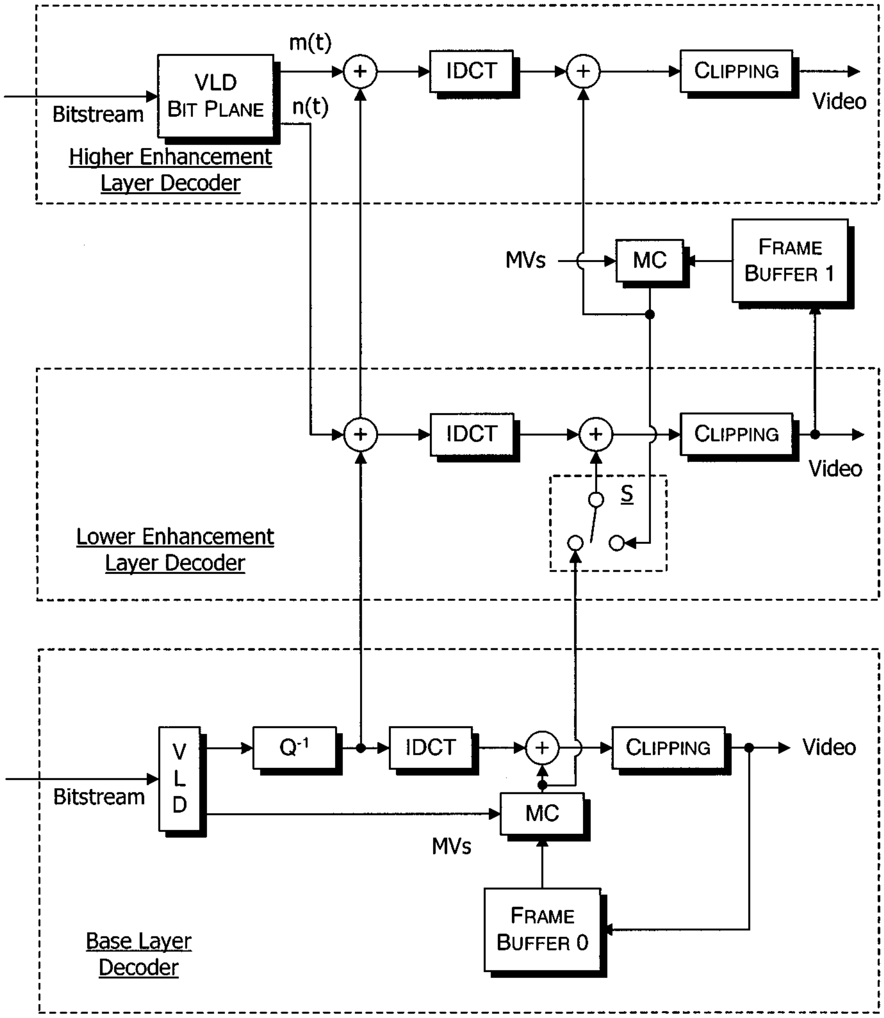


Fig. 7. Decoding diagram of the improved PFGS framework.

used to form the next enhancement layer reference. Compared with FGS, an additional buffer, an additional MC module and an additional IDCT are needed in the improved PFGS framework. In Fig. 7, $n(t)$ denotes the number of bit-plane layers needed to reconstruct the next enhancement reference layer. $m(t)$ denotes the number of bit-plane layers used to reconstruct the current display frame.

V. EXPERIMENTAL RESULTS AND ANALYSES

Extensive experiments and simulations have been performed to test the performance of the proposed PFGS framework.

First, a simple simulation experiment is designed to demonstrate that with only one extra frame buffer the simplified PFGS framework can still provide most of the coding efficiency gain obtained in the original PFGS framework with multiple buffers. The PFGS framework with multiple references is shown in Fig. 1 and the simplified PFGS framework with only two references is shown in Fig. 2. These two frameworks are almost the same except for the number of the references. At the same time, this experiment also gives the FGS results at the same testing conditions.

All test conditions are the same as those specified in the MPEG-4 core experiments. The sequences Akiyo, Foreman, and Coastguard (CIF format) are used in this experiment. Only the first frame is encoded as an I frame and all the other frames are encoded as P frames. The base layer encoder is a predictive encoder that includes motion compensation and DCT transformation modules, and it could be compatible to other standards, such as H.263, MPEG-2, or MPEG-4. In this experiment, we used MPEG-4 baseline video coder for the base layer coding. A simple half-pixel motion estimation scheme using linear interpolation was implemented to extract the motion vectors between video frames. The range of the motion vector is set to ±31.5 pixels. The same motion vectors are applied to all MCs, which in turn produce the predicted images.

The bit rate of the base layer is 128 kbits/s with TM5 rate control, and the encoding frame rate is 10 Hz. The bit rate of the enhancement layers is not constrained. Since the enhancement layers produce an embedded bitstream, the streaming server can truncate it at any place to match the channel bandwidth. The truncating procedure can be independent of a decoder. In our simulation experiments, the enhancement layer bitstream is truncated at 64 kbits/s, 128 kbits/s,… until 384 kbits/s, with an interval of 64 kbits/s. In order to reduce fluctuation when

TABLE I
PSNR Versus Bit-rate Comparison Between FGS, PFGSM, and PFGST for the Akiyo, Coastguard, and Foreman Sequences. PFGSM Denotes PFGS Framework with Multiple Buffers and PFGST Denotes PFGS Framework with Two Buffers

Bit rate	Akiyo			Coastguard			Foreman		
(kbit/s)	FGS	PFGSM	PFGST	FGS	PFGSM	PFGST	FGS	PFGSM	PFGST
192	42.10	42.12	42.10	27.75	27.86	27.84	31.18	31.32	31.30
256	42.75	42.77	42.76	28.52	28.72	28.70	31.89	32.14	32.12
320	43.13	43.21	43.20	29.27	29.54	29.52	32.51	32.86	32.83
384	43.47	43.60	43.59	29.79	30.16	30.14	33.15	33.60	33.54
448	43.84	44.03	44.02	30.19	30.62	30.60	33.70	34.27	34.15
512	44.27	44.50	44.48	30.62	31.10	31.08	34.17	34.76	34.61

switching between references with different quality, the conditional replenishment scheme is used in the two PFGS frameworks.

We define the least significant bit plane as index “0,” the next least significant bit plane as index “1,” and so on. For the PFGS framework with multiple frame buffers, even frames are predicted from the even layers of the previous frame, and the odd frames are predicted from the odd layers of the previous frame. This alternating structure repeats throughout the encoding of the whole video sequence. For the simplified PFGS framework with only two frame buffers, we choose the most efficient enhancement layers as references through experiments. For Akiyo sequence, the second reference alternates between the second bit plane and the third bit plane. For the Foreman sequence, the second reference alternates between the third bit plane and the fourth bit plane. For the coastguard sequence, the second reference alternates between the fourth bit plane and the fifth bit plane.

Table I gives some experimental results of coding these video sequences using the methods described above. For the Akiyo and Coastguard sequences, the luminance PSNR in the PFGS framework with two frame buffers is almost the same as that with multiple frame buffers. For the Foreman sequence, the luminance PSNR in PFGS framework with two frame buffers is about 0.15 dB less than that with multiple frame buffers. It is clearly shown that by only using one additional middle enhancement layer as the reference, we can achieve almost the same coding efficiency as using multiple references. On the other hand, the conditional replenishment approach used in the PFGS architecture with multiple references discards most part of the higher quality references, since HQPR coefficients do not replace LQPR coefficients when the MSB’s of the DCT coefficients have been encoded in previous enhancement layers. Meanwhile, from the results presented in Table I, we can see that the two PFGS frameworks can only achieve up to 0.5-dB PSNR gain on average. The main reason is that conditional replenishment scheme is not efficient although it can eliminate the fluctuation.

Extensive simulations have been conducted to test the performance of the improved PFGS framework proposed in this paper. There are two frame buffers in the improved PFGS framework. One is used to save the reconstructed base layer, and another is used to save the reconstructed enhancement reference layer. The base layer is always predicted from the first buffer and all enhancement layers are predicted from the second buffer. The results of the improved PFGS framework are compared with that of FGS. In addition, the results of the nonscalable video-coding scheme are also presented to show the cost of implementing the fine granularity functionality. The nonscalable video scheme is exactly same as the base layer of the FGS scheme, where TM5 controls the output bit rate.

In this experiment, each I frame is followed by 59 predicted frames including P frames and B frames. There are three B frames between every two P frames. The range of the motion vector is set to ±31.5 pixels. The same motion vectors are applied to the two motion compensators, which in turn produce the predicted images. The bit rate of base layer is 256 kbits/s with TM5 rate control, and encoding frame rate is 30 Hz. The enhancement bit streams are truncated at 128 kbits/s, 256 kbits/s, …, until 2048 kbits/s, with an interval of 128 kbits/s. In the PFGS scheme, to get a constant video quality, the enhancement layer bits cannot be equally allocated to each frame. The reason is very simple. For instance, in Fig. 4, frame 2 is predicted from the third enhancement layer of frame 1. Since the reference is of higher quality and is closer to the original image, a shorter bit stream will be generated due to the smaller prediction errors. However, frame 3 is predicted from the second enhancement layer with lower quality. Larger prediction errors would result in a longer bit stream in frame 3. Therefore, to get more or less the same quality across each frame, bits spent for encoding odd frames should be more than that for encoding even frames. On the other hand, if allocating a fixed bit rate for all frames, for example, at a bit rate where the third enhancement layer in an even frame is partially decoded, the previous odd frame may be only decoded up to the second enhancement layer due to the fact that its bit stream is longer. In turn, the quality of even frames may be limited since its higher layer reference (third enhancement layer) in odd frames is not available. This will not only cause the fluctuation of PSNR in adjacent frames, but also affect the overall coding efficiency of the video sequence. Fortunately, the PFGS scheme provides an embedded and fully scalable bit-stream, so that the rate control does not have to occur at the encoding time. A simple rate allocation or truncation module can be used in the streaming server to obtain the optimized quality given a bit-rate constraint.

Some experimental results of the improved PFGS framework are given in Figs. 8–10 for the Akiyo, Coastguard, and Foreman sequences, respectively. As we expected, generally, there is not too much performance gain at lower bit-rates compared with that of FGS. Sometimes there is even quality loss since the reconstruction reference in the decoder may be different from that in the encoder for the lowest few enhancement layers. For the

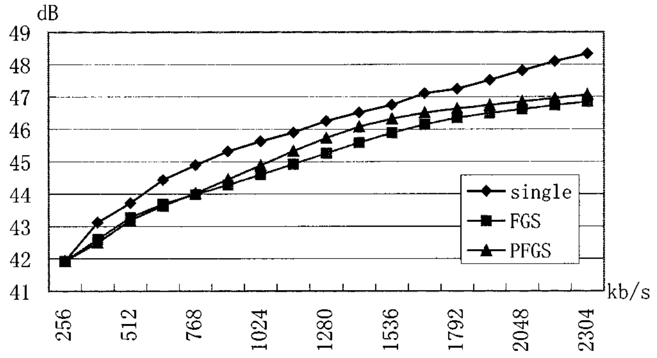


Fig. 8. PSNR versus bit-rate comparison between FGS and PFGS for the Akiyo Y component.

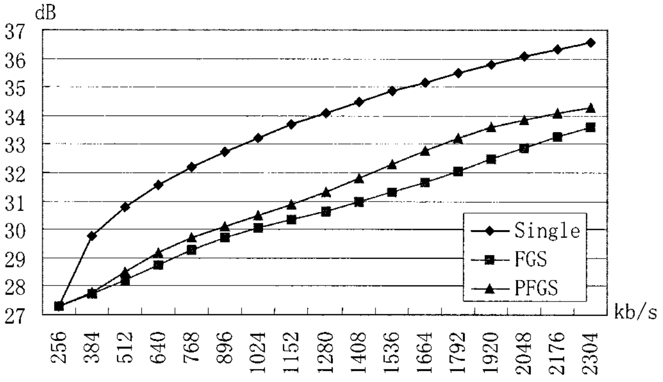


Fig. 9. PSNR versus bit-rate comparison between FGS and PFGS for the Coastguard Y component.

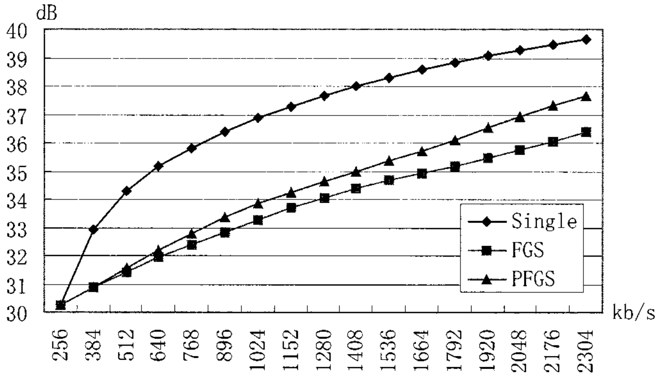


Fig. 10. PSNR versus bit-rate comparison between FGS and PFGS for the Foreman Y component.

Akiyo sequence, there is only some small quality loss at lower bit rates. This is because in lower enhancement layers, we have to use LQPi instead of HQPi to reconstruct the display image since the HQPi is not available yet at lower bit rates. For the Coastguard and Foreman sequences, we cannot even see the quality loss, because a better prediction compensates most of the loss caused by using different references. However, as the bit-rate increases, we can see consistently significant coding efficiency gain. For the Coastguard sequence, the gain in Y component can be up to 1 dB. For the Foreman sequence, the gain in Y component can be up to 1.3 dB. The reason why the Akiyo sequence does not show significant gain is that the base layer bit rate is already very high for this sequence. The quality is already very good at that bit rate.

The experimental results also show that the coding efficiency gap between the FGS scheme and the nonscalable video coding exceeds 3.0 dB. Although, the PFGS scheme has significantly improved the coding efficiency of FGS, the gap between the PFGS scheme and the nonscalable video coding is still large. How to further improve the coding efficiency of the PFGS scheme is still an open question.

VI. SIMULATION OF STREAMING PFGS VIDEO OVER WIRELESS CHANNELS

For most applications over the Internet, the fundamental technical problem is that the network bandwidth varies in a wide range from one user to another (user variations) and from time to time (temporal variations). The major advantage of the PFGS scheme is its bandwidth adaptation capability that provides a good solution for bandwidth fluctuation. With the rapid development of wireless communications, the wireless channel becomes an increasingly popular and convenient means to access the Internet. As we know, the wireless channels have different characteristics than the wired internet channels. They are typically noisy and suffer from a number of channel degradations, such as random bit errors and burst errors due to fading and multiple path reflections. When compressed video data is sent over these channels, the effect of channel errors on the 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 video quality. To deal with this problem, some error resilience and concealment methods are introduced in [18], [19].

For applications involving transmitting video over error-prone channels, such as the wireless channels, the PFGS framework provides a more robust solution. First, PFGS provides an inherent error recovery feature that can gracefully recover from any enhancement layer errors. Second, the PFGS framework 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 PFGS framework: the base layer bitstream and the enhancement layer bitstream. The base layer bitstream is very sensitive to channel errors. Any random errors or burst errors may cause the decoder to lose synchronization, and the decoded errors will propagate to the start of the next GOP. However, the enhancement layers can tolerate the channel errors. When there are errors in the 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 PFGS scheme. Generally, since the bit rate of the base layer is very low, channel coding

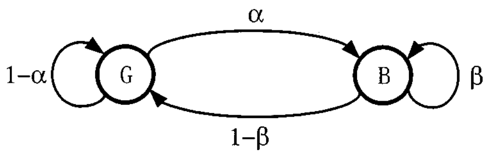


Fig. 11. Gilbert model for simulating wireless channel.

with the same number of bits can provide better protection to base layer than to the overall bitstream. On the other hand, if the channel is a time-varying channel, as in many wireless communication situations, it is very feasible to adaptively adjust the rate allocation between the source and channel coding operations seamlessly with the PFGS scheme. Therefore, the quality of base layer can be maintained at a stable level.

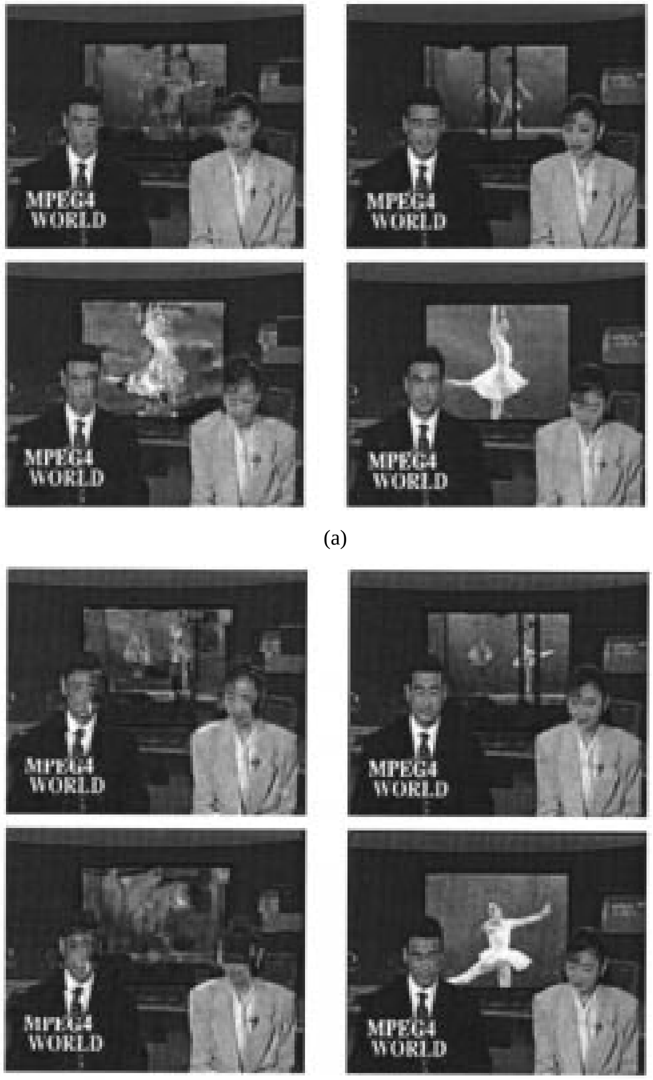
In this section, we simulate the delivery of two kinds of video bitstreams over wireless channels. The first kind of bitstreams is MPEG-4 single-layer bitstreams. Some error-resilience tools are applied from the source coding viewpoint, such as Resynchronization Marker, Data Partitioning, and Header Extension Code (HEC). The second kind of bitstreams are the PFGS bitstreams. The same error-resilience tools are applied to the base layer, and the enhancement layers are encoded without any error protection and concealment. In the PFGS coding scheme, both the overall source bit-rate and bit allocation between source and channel coding can be dynamically adjusted depending on the channel feedback. However, it is very difficult for the single-layer case to adapt to bandwidth fluctuation for bit re-allocation. Therefore, the channel bandwidth and bit allocation between source and channel coding are fixed in our simulation.

We use a two-state Markov model to simulate the wireless channel as in Fig. 11 [20]. This model can characterize the error sequences generated by data transmission channels. In the good state (G) errors occur with low probability α, while in bad state (B), they occur with high probability β. The errors occur in cluster or bursts with relatively long error free intervals (gaps) between them. The state transitions are shown in Fig. 11 and summarized by its transition probability matrix

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

The model can be used to generate sequences of symbol error. In this case, it is common to set $\alpha \approx 0$ and $\beta \approx 0.5$. However, in situations where a Reed–Solomon (RS) code over $GF(2^m)$ is to be used, it is more appropriate for the model to generate m-bit symbol errors. If $\beta = 0$, this model will simulate a random error, where α is the bit error rate.

The QCIF News sequence is used in our simulation, and the encoding frame rate is 10 Hz. Only the first frame is encoded as an I frame, and other frames are encoded as P frames. The total channel bandwidth is 64 and 8 kbits/s of it is used for channel coding for each bitstream. For the MPEG-4 single-layer bitstream, the bit rate of source coding is 56 kbits/s. For the PFGS bitstream, 32 kbits/s is used for base layer source coding. The enhancement layer bitstream can be truncated to 24 kbits/s to fit in the channel bandwidth. It is obvious that the PFGS bitstream can be decoded from 32 to 56 kbits/s. The channel parameter α randomly varies from 0.04 to 0.1, and parameter β also randomly varies from 0.3 to 0.6 during simulation. The channel coding is implemented with a general RS code. An RS (64, 56, 8) code is applied to the MPEG-4 single-layer bitstream (a block of 64 bytes with 56 bytes of data and 8 bytes of protection). And an RS (40, 32, 8) code is applied to the PFGS base layer bitstream (a block of 40 bytes with 32 bytes of data and 8 bytes of protection). Both decoders have included some basic error-concealment tools such as copying from motion-compensated areas in the previous frame, etc. The experimental results are shown in Fig. 12. Clearly, after going through the simulated wireless channel, there are too many visual artifacts in the decoded images from the received MPEG-4 single-layer bitstream which contains errors now, while the results using the improved PFGS framework still provide very good image quality. The reason is that the PFGS framework provides a robust enhancement layer bitstream. Since only the sensitive base layer bitstream needs the protection of channel coding, the 8 kbits/s channel coding can provide stronger protection for the base layer bitstream.



(a)

(b)

Fig. 12. Simulation results of delivering different video bitstreams over wireless channels. (a) Images decoded from the received MPEG-4 single-layer bitstream. (b) Images decoded from the received PFGS bitstream.

On the other hand, for the MPEG-4 single-layer bitstream, the whole bitstream is very sensitive to channel errors. The 8 kbits/s channel coding has to provide protection for every bit in that bitstream. Therefore, the channel coding protection becomes relatively weak and more channel errors could not be corrected under the same channel condition.

VII. CONCLUSION AND FUTURE WORK

In this paper, a highly efficient scalable video-coding framework is first proposed. Compared with the FGS scheme in MPEG-4, the proposed PFGS framework tries to use some higher quality references to improve the coding efficiency, since higher quality references make the motion predictions more accurate. In order to keep the scalability properties, such as fine granularity scalability, bandwidth adaptation, and error recovery, the basic PFGS framework preserves a complete prediction path from the lowest layer to the highest layer. However, multiple extra frame buffers are needed to save the reconstructed reference layers in the basic PFGS framework, which increases the memory cost and computational complexity. We simplify the basic PFGS framework from multiple extra frame buffers to one extra frame buffer, while still keeping almost the same coding efficiency. Moreover, an improved PFGS framework is proposed to further improve the coding efficiency by effectively eliminating the fluctuation at enhancement layers when switching references. While the original PFGS framework provides about 0.5-dB coding efficiency gain over the FGS in MPEG-4, the improved PFGS framework provides about 1-dB coding efficiency gain. The coding efficiency gap between scalable video coding and the nonscalable video coding is closing.

The advantages of the PFGS scheme, such as layered coding and error recovery are further demonstrated through a wireless transmission simulation. The simulation results clearly show that under the same conditions in an error-prone channel, the PFGS scheme can produce much better decoded images than the single-layer nonscalable MPEG-4 scheme.

In the future, more studies need to be done on how to further improve the coding efficiency and robustness of the PFGS framework, and on how to optimally transport it through internet and wireless channels. The coding efficiency gap between the nonscalable video coding and the PFGS video coding is still big although it is closing. How can we design an optimized PFGS encoder with improved coding efficiency? How can we optimally allocate (truncate) bits among different frames under an overall bit-rate constraint? Sometimes, the base layer bitstream may be still too long for certain applications, so how shall we add spatial scalability in the PFGS framework? How to address the real-time video communication issues? What are the best network protocols for streaming the PFGS video over the internet? These are all open questions that need to be answered.

Some simple error-detection and resynchronization tools should be added to the PFGS enhancement-layer bitstream to improve its robustness and efficiency. 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, we 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, how to allocate bit rate between source and channel coding dynamically according to channel conditions is another topic that needs further study.

ACKNOWLEDGMENT

The authors would like to thank Dr. W. Zhu, Dr. Q. Zhang, and G. Wang for many valuable suggestions in the wireless transmission, and especially for providing the wireless channel simulator and RS channel coding for the simulations. The authors also thank Dr. W. Li for many beneficial discussions, encouragements, and help on the PFGS framework.

REFERENCES

[1] K. Shen and E. J. Delp, "Wavelet base rate scalable video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 109–121, 1999.
[2] J. M. Shapiro, "Embedded image coding using zerotree of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, 1993.
[3] I. Sodagar, H.-J. Lee, P. Hatrack, and Y.-Q. Zhang, "Scalable wavelet coding for synthetic/natural hybrid images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 244–254, 1999.
[4] J.-Y. Tham, S. Ranganath, and A. A. Kassim, "Highly scalable wavelet-based video codec for very low bit-rate environment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, pp. 12–27, 1998.
[5] S. McCanne and V. Jacobson, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM'96*, Stanford, CA, Aug. 1996, pp. 117–130.
[6] J. Hartung, A. Jacquin, J. Pawlyk, and K. L. Shipley, "A real-time scalable software video codec for collaborative applications over packed networks," in *Proc ACM Multimedia'98*, Bristol, U.K., Sept. 1998, pp. 419–426.
[7] F. Ling, W. Li, and H.-Q. Sun, "Bitplane coding of DCT coefficients for image and video compression," in *Proc. SPIE-VCIP'99*, San Jose, CA, Jan. 25–27, 1999, pp. 500–508.
[8] W. Li, "Bitplane coding of DCT coefficients for fine granularity scalability,", ISO/IEC JTC1/SC29/WG11, MPEG98/M3989, Oct. 1998.
[9] B. Schuster, "Fine granular scalability with wavelets coding,", ISO/IEC JTC1/SC29/WG11, MPEG98/M4021, Oct. 1998.
[10] Y.-W. Chen, H. Radha, and R. A. Cohen, "Results of experiment of fine granular scalability with wavelet encoding of residuals,", ISO/IEC JTC1/SC29/WG11, MPEG98/M3988, Oct. 1998.
[11] J. Liang, J. Yu, Y. Wang, M. Srinath, and M.-H. Zhou, "Fine granularity scalable video coding using combination of MPEG4 video objects and still texture objects,", ISO/IEC JTC1/SC29/WG11, MPEG98/M4025, Oct. 1998.
[12] S. C. S. Cheung and A. Zakhor, "Matching pursuit coding for fine granular video scalability,", ISO/IEC JTC1/SC29/WG11, MPEG98/M3991, Oct. 1998.
[13] M. Benetiere and C. Dufour, "Matching pursuits residual coding for video fine granular scalability,", ISO/IEC JTC1/SC29/WG11, MPEG98/M4008, Oct. 1998.
[14] W. Li, "Fine granularity scalability in MPEG-4 for streaming video," in *Proc. ISCAS 2000*, vol. 1, Geneva, Switzerland, May 28–31, 2000, pp. 299–302.
[15] J. Macnicol, M. Frater, and J. Arnold, "Results on fine granularity scalability,", Melbourne, Australia, ISO/IEC JTC1/SC29/WG11, MPEG99/m5122, Oct. 1999.
[16] S. Li, F. Wu, and Y. Q. Zhang, "Study of a new approach to improve FGS video coding efficiency,", ISO/IEC JTC1/SC29/WG11, MPEG98/M5583, Dec. 1999.
[17] T. K. Tan, K. K. Pang, and K. N. Ngan, "A frequency scalable coding scheme emplying pyramid and subband techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 203–207, Apr. 1994.
[18] Y. Wang and Q. F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.
[19] R. Talluri, "Error-resilent video coding in the ISO MPEG-4 standard," *IEEE Commun. Mag.*, vol. 36, pp. 112–119, 1998.
[20] J. R. Yee and E. J. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," *IEEE Trans. Commun.*, vol. 43, pp. 2316–2323, Aug. 1995.

Feng Wu (M'00) received the B.S. degree in electrical engineering from the University of Xi'an Electrical Science and Technology in 1992. He received the M.S. and Ph.D. degrees in computer science from the Institute of Harbin Technology, Harbin, China, in 1996 and 1999, respectively.

He has been an Associate Research with Microsoft Research China, Beijing, since September 1999. His research interests include video and audio compression, video and audio transmission, and wireless communication.

Shipeng Li (M'97) received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC) in 1988 and 1991, respectively, and the Ph.D. degree from Lehigh University, Bethlehem, PA, in 1996, all in electrical engineering.

He was with Electrical Engineering Department, University of Science and Technology of China, Hefei, China, during 1991–1992. He was a Member of Technical Staff at Sarnoff Corporation, Princeton, NJ, during 1996–1999. He has been a Researcher with Microsoft Research China, Beining, since May 1999. His research interests include image/video compression and communications, digital television, multimedia, wireless communication. He has contributed several technologies in MPEG-4 international standard.

Ya-Qin Zhang (F'98) joined Microsoft Research China, Beijing, in January 1999 as the Assistant Managing Director. He was previously the Director of Multimedia Technology Laboratory at Sarnoff Corporation, Princeton, NJ. He has authored over 150 refereed papers and has 30 U.S. patents granted or pending in digital video, Internet, multimedia, wireless, and satellite communications. Many of the technologies he and his team developed have become the basis for start-up ventures, commercial products, and international standards. He has been an active contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia.

Dr. Wang was the Editor-In-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from July 1997 to July 1999. He serves on the Editorial Boards of seven other professional journals and over a dozen conference committees.