DCT-PREDICTION BASED PROGRESSIVE FINE GRANULARITY SCALABLE CODING

Feng Wu, Shipeng Li, Ya-Qin Zhang
Microsoft Research China

ABSTRACT

In this paper, we propose a novel architecture for scalable video coding, namely, Progressive Fine Granularity Scalable (PFGS) coding, which can provide high coding efficiency along with good bandwidth adaptation and error recovery properties. Unlike the Fine Granularity Scalable (FGS) coding in MPEG-4 proposal, some of the enhancement layers in a current frame are predicted from a high quality enhancement layer in a reference frame, rather than always from the base layer. Using a high quality enhancement layer as the reference makes the motion prediction more accurate to improve the coding efficiency. On the other hand, use of multiple layers of different quality references may also result in increases and fluctuations of the prediction residues to be coded when switching the references, which may limit the coding efficiency improvement. A multiple-layer conditional replenishment approach is used to eliminate this kind of fluctuation. Experimental results show that our coding scheme can improve coding efficiency up to 0.5dB compared with fine granularity scalability coding.

1. INTRODUCTION

Transporting digital video over Internet or wireless channels has to deal with two major problems: bandwidth fluctuation and packet-losses and/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 solution to these problems is to compress and transmit a video sequence with scalability. Block DCT and wavelets have been two dominant transform techniques in existing video coding schemes. Although the use of Discrete Wavelet Transform (DWT) in scalable video coding has received much attention for its inherent spatial, temporal and rate scalability feature in recent years [1]~[3], block DCT transform based coders enjoyed the success due to their low complexity in implementation and their reasonably good performance. Therefore many good techniques have been proposed for scalable video coding based on block DCT transform [4]~[7].

One of the very promising techniques is the fine granularity scalable (FGS) coding scheme first proposed by Li et al [6]. In fine granularity scalable video coding scheme, the base layer video will 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 the minimum channel bandwidth. All the enhancement layers in the prediction frames are encoded based on the prediction from the base layer in the reference frames. Thus errors in the enhancement layers don’t cause any drifting problem in the prediction frames followed and furthermore the embedded enhancement layer bitstream can adapt to any channel bandwidth condition. However, since the prediction is always based on lowest quality base layer, the coding efficiency of FGS scheme is not as good as, sometimes much worse than, that of traditional SNR scalability schemes, such as in [7]. On the other hand, the traditional SNR scalability schemes use more accurate prediction from the same layer in reference frames, they normally provide better coding efficiency and are suitable for simulcast in stable channels. But if there is an error or packet loss in the a enhancement layer, it would propagate to the end of a GOP and would cause serious drifting problem in the same and higher layers in the prediction frames followed. Even though there might be sufficient bandwidth available later on, the decoder could not recover to the higher quality until another GOP start.

In this paper, we proposed a novel scalable coding scheme, namely, Progressive Fine Granularity Scalable (PFGS) coding, which can provide high coding efficiency along with good bandwidth adaptation and error recovery properties. Similarly to FGS, the PFGS coding scheme also encodes video frames into multiple layers, including a base layer of relatively low quality video and embedded multiple enhancement layers of increasingly higher quality video. However, differently from the FGS scheme, the PFGS coding scheme tries to use several high quality enhancement layer references for prediction in the enhancement layer encoding rather than always using base layer. Using high quality references makes the motion prediction more accurate to improve the coding efficiency. When and how to replace the low quality reference is actually a very artistic problem, because such improvements should not cause drifting problem nor affect the excellent properties of scalability, such as bandwidth adaptation and error recovery.

The rest of this paper is organized as follows. Section 2 introduces the basic ideas to build the PFGS architecture with high coding efficiency. Section 3 discusses in detail how to implement a PFGS encoder. Section 4 gives the experimental results of encoding several test sequences using the PFGS encoder. Section 5 concludes this paper.

2. THE ARCHITECTURE OF PFGS

Our proposed PFGS solution keeps the advantages of FGS coding such as, fine granularity scalability, channel adaptation and error recovery, while trying to use better predictions to improve the coding efficiency. There are two key points here in designing the architecture of PFGS. The first one is to use as many predictions from the same layer as possible, rather than always using the base layer (for coding efficiency). The second one is to keep a prediction path from the lowest layer to the highest layer across several frames (for error recovery and channel bandwidth adaptation). The first point makes the motion prediction as accurate as possible for that video layer to improve coding efficiency. The second point makes sure that there is no drifting problem in case of channel congestion, packet-loss or errors. With such a coding architecture, there is never a need to retransmit lost/error packets since gradually the higher video layer can be automatically reconstructed progressively over a few frames.

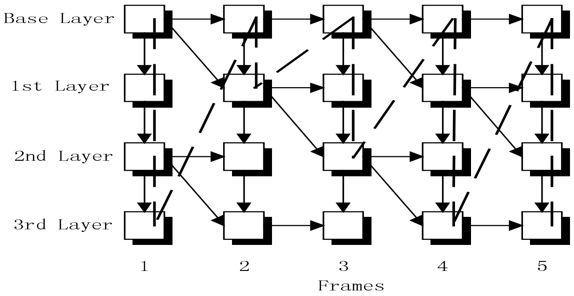

Figure 1: The proposed framework of PFGS.

Figure 1 conceptually illustrates an example PFGS architecture for efficient progressive video coding without drifting problem. In the illustrated architecture, frame 2 is predicted from the base layer and even enhancement layer of frame 1 (i.e., 2nd layer). Frame 3 is predicted from the base layer and odd enhancement layers of frame 2 (i.e., 1st and 3rd layer). Frame 4 is again predicted from the base layer and even enhancement layers of frame 3, and so on. Since the quality of an enhancement layer is higher than that of the base layer, such a PFGS coding scheme provides more accurate motion prediction to improve the coding efficiency.

The advantages of the proposed coding scheme are obvious when applied to video transmission over Internet or wireless channels. First, the encoded bitstream can adapt to the available bandwidth of the channel without drifting problem. Figure 1 shows an example of this bandwidth adaptation property. The thick 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 higher layer bits. However after frame 2, the available bandwidth increases, and the transmitter simply transmits more layers of video bits. After two frames (at frame 4), the decoder side can obtain up to the highest quality layer again. All these operations do not require any re-encoding and re-transmission of the video bitstream. All layers of video are efficiently encoded and embedded in a single bitstream.

This exemplifies a case where the group depth is 2. Group depth defines how many layers may refer back to a common reference layer. This depth can be changed. If the depth is 1, the scheme essentially becomes traditional SNR scalability schemes as in [7]. If the depth is equal to the total number of layers, the scheme essentially becomes the FGS scheme [6]. Moreover, the above description is only a special case of a more general architecture where in each frame the prediction layer used can be randomly assigned as long as a prediction path from lowest layer to highest layer is maintained across several frames.

3. IMPLEMENTATION OF PFGS

In the previous section, we described how to build a novel architecture for SNR scalable video coding. It’s clear that in PFGS architecture several different quality references are employed to predict the current frame. For a prediction frame, its base layer is always predicted from the reconstructed base layer in the previous reference frame. Lower enhancement layers may be predicted from either the reconstructed base layer or a reconstructed low quality enhancement layer in the reference frame. But higher enhancement layers are bound to use a high quality reference for motion compensation. We can partition all layers in the current frame into several groups depending on their references. The layers in a group share a common reference and are encoded using the same encoding process.

To illustrate how these layers are processed and compressed, consider the simple case using three references in the PFGS architecture as shown in Figure 2. In the illustrated encoding diagram, for simplicity, the inverse DCT and motion estimation modules are omitted. The three references with increasing quality are saved in Frame buffer 0, Frame buffer 1 and Frame buffer 2 separately. Assume these frame buffers get updated with the corresponding reconstructed layers of the current frame. Each frame buffer, motion compensation, DCT and quantizer form one complete encoding process outlined by a bold dash-line box in Fig. 2.

The first encoding process deals with the first group of layers, that is, the base layer and lower enhancement layers. Since the base layer and the lower enhancement layers use the reconstructed base layer as the prediction reference, the coding of these layers is exactly the same as the coding in FGS [6]. In this encoding process, original image and reconstructed based layer from previous reference frame (in frame buffer 0) are the inputs. After motion estimation and DCT transform, we obtain a set of DCT coefficients *DCT1* of the prediction error. Note a lowest quality reference is used in this motion compensation (i.e. the reconstructed base layer). The coefficients *DCT1* are quantized by scalar quantizer and then compressed by VLC into the base layer bitstream. Generally, the step Q_b of scalar quantizer is very large to generate a relatively short bitstream.

All enhancement layers in the first group encode the quantization error (residue) of the base layer. The bit-plane coding technique is used in the enhancement layers to provide embedded bitstream and good scalability. Thus the quantization steps of the enhancement layers are a series of factors 2^i, from the maximum number of bitplane of the residue to the desired bitplane in the first group

The encoding processes for the rest of groups of layers are exactly same. Every encoding process takes three inputs: original image, reconstructed enhancement reference layer (stored in enhancement frame buffer 1 or 2) and reconstructed DCT coefficients derived from the last encoding process. For example, in the encoding process of the middle group of enhancement layers, a new set of DCT coefficients *DCT2* of the prediction error can be obtained by motion compensation using reconstructed enhancement layer reference stored in frame buffer 1 and the DCT transform. Note that here a high quality reference is used for this motion compensation. We expect that the coefficients *DCT2* are statistically lower than *DCT1* because the reference of the latter is of high quality and hence closer to the original image. The residues between *DCT2* and reconstructed *DCT1* from the first group are again encoded using a bit-plane entropy coder up to the desired bit-plane level to generate a middle enhancement layer bitstream. Similarly, the residues between DCT3 and reconstructed DCT2 from the second group generate a high enhancement layer bitstream. The encoding process of each group of layers forms a part of the enhancement bitstream, they can merge a single bitstream for the delivery purpose.

Although the prediction error coefficients *DCT2* are statistically lower than *DCT1* in the above example, the dynamic range of the residues between *DCT2* and the reconstructed *DCT1* is not necessarily always less than that of the residues between *DCT1* and reconstructed *DCT1*. For some instances, the magnitudes of some individual residues may actually increase due to the non-ideal motion estimation and compensation. Sometimes the signs of the residues between *DCT2* and reconstructed *DCT1* may be reversed too. It means that switching the references may cause undesirable fluctuations and increases in the coefficients to be encoded. .

A conditional replenishment described in [8] is used to eliminate this kind of fluctuation when switching references between two layers. In the conditional replenishment, not all the prediction error coefficients *DCT1* are replaced by the coefficients *DCT2*. The lower layer prediction error coefficients *DCT1* are conditionally replaced by the higher layer prediction error coefficient *DCT2* depending on the values of the reconstructed *DCT1*. If the reconstructed coefficient is zero then the corresponding coefficient in *DCT1* is replaced with that in *DCT2*. If the reconstructed coefficient is not zero, then no replacement is done. For a general *n*-layer case, the conditional replenishment scheme needs to be slightly modified by using the sum of all reconstructed coefficients in lower layers as the input to the conditional replenishment rather than just the values from the immediate lower layer.

4. EXPERIMENTAL RESULTS

Simulations have been performed to test the performance of the proposed PFGS scheme. PFGS architecture used in these experiments is shown in Figure 1. The number of enhancement layers is not fixed; instead it is based on the number of bit planes needed to represent the residues in binary format. The MPEG-4 test sequences of Akiyo, Foreman and Coastguard (CIF format) are used in these experiments. The first frames of these sequences are encoded as I frames and other frames are encoded as P frames. The encoder of the base layer that includes motion compensation and DCT transformation can be compatible to H.263 or MPEG-4. Here we choose MPEG-4 coding scheme for the base layer. A simple half-pixel motion estimation scheme using linear interpolation is implemented between two adjacent original images. Limitations on the range of motion vectors are set at ±31.5 pixels. The same motion vectors are applied to motion compensation of all layers, which in turn produces prediction error images. The bit rate of base layer is 128kbit/s with TM5 rate control scheme and encoding frame rate is 10Hz. The bit rate of enhancement layers is not constrained. The streaming server can truncate it to fit in the capacity of the network. The truncating process can be independent of the encoder. In our simulation, the enhancement bitstreams are truncated at 64kbit/s, 128kbit/s, …, until 384kbit/s with intervals of 64kbit/s. The average PSNR of the whole sequence for PFGS and FGS are listed in Table 1.

From the results presented, we can clearly see that the proposed (PFGS) scheme can achieve up to 0.5dB PSNR gain on average over the FGS scheme while keeping the fine granularity scalability, channel adaptation and error recovery properties. Moreover, the PFGS shorten the possible drifting path to the number of frames equal to the number of groups of scalable layers instead of the size of a GOP in the traditional PSNR layered coding schemes.

Table 1: Simulation results of PFGS and FGS schemes

Enh layer Bit Rate (kbit/s)	Akiyo PSNR(dB)		Foreman PSNR(dB)		Coastguard PSNR(dB)	
	PFGS	FGS	PFGS	FGS	PFGS	FGS
64	42.10	42.10	31.32	31.17	27.86	27.75
128	42.76	42.75	32.14	31.89	28.72	28.52
192	43.20	43.13	32.85	32.51	29.54	29.27
256	43.59	43.47	33.60	33.14	30.16	29.79
320	44.02	43.84	34.27	33.70	30.62	30.19
384	44.48	44.27	34.76	34.17	31.10	30.62

5. CONCLUSIONS AND FUTURE WORKS

In summary, this paper exhibits a novel idea to build the architecture for scalable video coding. The proposed (PFGS) scheme can achieve consistently better coding efficiency than the FGS scheme while keeping the fine granularity scalability, channel adaptation and error recovery properties.

However, there are still many future research topics. The example architecture we discussed is far from mature. Firstly, it needs too many extra frame buffers to save the reconstructed layers as references. However, not every reference makes the same contribution to improve the coding efficiency. Only some of the references make great contributions to improve the coding efficiency, while others have little effect. How to choose minimal number of reference layers that can greatly improve the coding efficiency remains a question to be answered. Another problem is the fluctuations during switching references. The conditional replenishment can solve this problem, while it partly loses the advantages of high quality reference. Because the non-zero reconstructed coefficients in lower layers are essential condition to cause fluctuation, but sufficient condition. In many cases of the non-zero reconstructed coefficients, the residues can still decrease by replacing the prediction error coefficients in lower layers. A more efficient approach that can take full advantage of a high quality reference without causing any fluctuations should be investigated to further improve the coding efficiency.

6. REFERENCES

[1] K.Shen, E.J.Delp, “Wavelet base rate scalable video compression,” IEEE trans. on CSVT, vol 9, no 1, pp.109-121, 1999.

[2] J.M.Shapiro, “Embedded image coding using zerotree of wavelet coefficients,” IEEE trans. on SP, 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. on CSVT, vol 9 no 2, 1999.

[4] S.McCanne and V.Jacobson, “Receiver-driven layered multicast,” Proc. ACM SIGCOM’96, August, 1996, Stanford, CA, USA.

[5] J.Hartung, A.Jacquin J.Pawlyk and K.L.Shipley, “A real-time scalable software video codec for collaborative applications over packed networks,” Proc ACM Multimedia’98, Bristol, UK.

[6] Weiping Li, “Fine Granularity Scalability Using Bit-Plane Coding of DCT Coefficients,” MPEG98/m4204.

[7] James Macnicol, Michael Frater and John Arnold, “Results on Fine Granularity Scalability,” MPEG99/m5122.

[8] T. K. Tan, K. K. Pang, K. N. Ngan, “A Frequency Scalable Coding Scheme Emplying Pyramid and Subband Techniques,” IEEE Trans. Circuits and Systems for Video Technology, Vol. 4, No. 2, April 1994, pp 203-207.

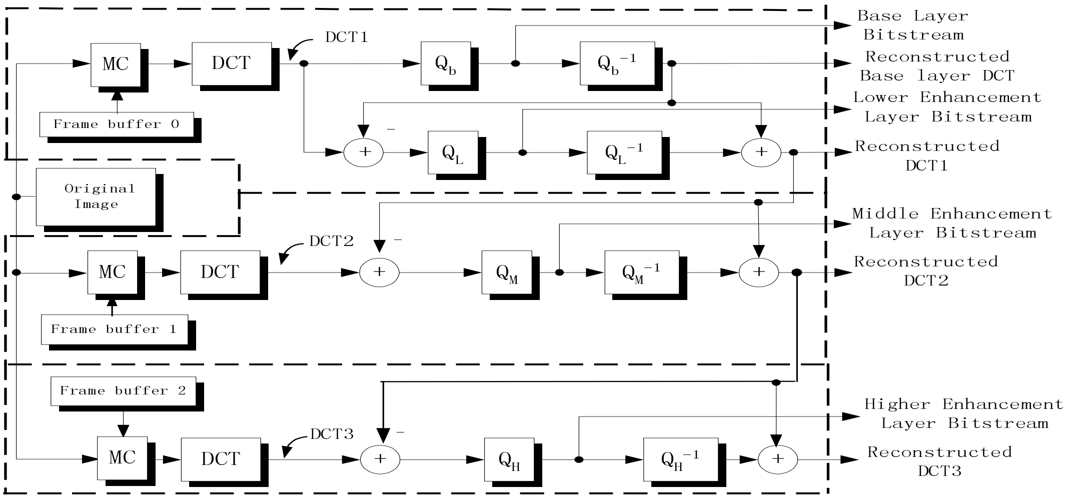

Figure 2: The encoding diagram of PFGS with three references.