SEAMLESS SWITCHING IN MULTI-RATE VIDEO STREAMING SYSTEMS: A WAVELET-BASED SCHEME VERSUS THE SP-FRAME SCHEME

Wei Zhang and Bing Zeng

Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong, emails: {eezw,eezeng}@ust.hk

ABSTRACT - ***The multiple bit-rate (MBR) representation of video sequences offers an effective solution to video streaming services over the Internet. To facilitate such MBR-based streaming services, a mechanism is required to support seamless switching among multiple bit-streams when a bandwidth change is detected. The SP-frames developed in H.264 provide such a mechanism at each pre-selected switching point. In this paper, we propose a new switching scheme that is based on a wavelet-domain processing of the reconstructed frame at each switching point. We will compare our scheme with the SP-frame scheme in aspects such as the quality drop at each switching point and all subsequent frames, the count of overhead bits to support an arbitrary switching, and the computational complexity. The results indicate that our scheme can achieve the seamless switching with a better rate-distortion performance at cost of slightly increasing the computations at the decoder side*.**

1. INTRODUCTION

It is well-known that one of the most challenging requirements in real-time video streaming applications over the Internet is to maintain a robust service-of-quality (QoS) under various bandwidth variations.

One effective solution is to encode each source video into multiple bit-streams of different qualities, where each bit-stream is targeted at a pre-selected bit-rate. Then, the server can dynamically select the appropriate bit-stream according to the available bandwidth. In this scenario, the most important issue is: such a streaming system needs to be equipped with a mechanism that allows arbitrary switching among different bit-streams when a bandwidth change is detected during the streaming service.

Let us use $F(t)$ to represent the frame of a source video at frame number t, and $F_u(t)$ to denote the corresponding reconstructed frame at rate r_u, $u = 1,\cdots,N$. Suppose that a bandwidth change is detected at t_0 (corresponding to a P-frame) and a switching from $F_u(\cdot)$ to $F_v(\cdot)$ is needed right at t_0. The simplest and most straightforward way is to perform the so-called direct switching with the transmitted frames being arranged as: $\{...F_u(t_0-1), F_v(t_0), F_v(t_0+1)...\}$. However, since there exists mismatching between $F_u(t_0-1)$ and $F_v(t_0-1)$, errors will occur when $F_u(t_0-1)$ (instead of the correct prediction frame $F_v(t_0-1)$) is used to perform the motion compensation. More severely, such errors will propagate into all subsequent frames until the next I-frame is received - causing the so-called drifting errors that can often become too large to be accepted, especially in the low-to-high switching case.

A new frame type, the so-called S-frames, has been proposed in [1] for bit-stream switching and random access. More recently, another frame type - the so-called SP-frames - is developed in [2]-[4] and it has been included in the H.264 standards. The main feature of an SP-frame is that it can be reconstructed exactly by using different prediction frames. The SP-frame based seamless switching is rather straightforward: First, some switching points, $t_m\big|_{m=0,\cdots,M}$ (all at P-frames), are pre-selected. Then, further quantization is applied on the reconstructed frame $F_u(t_m)$ at each switching point so as to generate the so-called primary SP-frame $S_u(t_m)$. Because of the re-quantization, each primary SP-frame will experience certain quality drop and this drop will also influence the coding of all subsequent frames (up to the next I-frame). Meanwhile, a secondary SP-frame $\hat{S}_{u,v}(t_m)$ is generated to represent the difference between $S_u(t_m)$ and the reference frame $F_v(t_m-1)$ at rate r_v in order to support the seamless switching from r_v to r_u at t_m.

At each switching point, all primary and secondary SP-frames are generated via a DCT-domain processing. Thus, their sizes are fixed once the re-quantization factor is selected, which is not flexible in adapting the bandwidth change. Secondly, at each switching point, there are a total number of $N \times (N-1)$ secondary SP-frames that have to be prepared and stored in the server to support arbitrary switching among N bit-streams and each secondary SP-frame usually has a large size. Therefore, it not only requires a lot of computations but also is quite wasteful in terms of storage requirement. Overall, we feel that the rate-distortion (R-D) performance achieved in the SP-frame switching scheme is not very satisfactory. For instance, several tens of kilobits are usually needed for each secondary SP-frame of the QCIF format and the quality drop is controlled within about 0.5 dB [4].

In our paper, we attempt to develop a more effective mechanism for multiple bit-streams that can achieve the seamless switching at a better R-D performance and storage efficiency. The unique feature of our scheme is that the processing on $F_u(t_m)$ at each switching point is performed in a wavelet domain.

The rest of this paper is organized as follows: Section 2 explains how the reconstructed frame $F_u(t_m)$ at each pre-selected switching point is further processed in the wavelet domain. Here, we will focus on the optimal bit allocation and the impact on the coding of all subsequent frames in comparison with the SP-frame scheme. Then, a simple but quite novel switching scheme is presented in Section 3, again with a comparison with the SP-frame scheme. Some experimental results are given in Section 4. Finally, Section 5 presents the conclusions of this paper.

1-4244-0367-7/06/$20.00 ©2006 IEEE

2. Wavelet-domain Processing of Reconstructed Frames

The reason we choose to apply a wavelet-domain coding on the reconstructed frame at each switching point is two-fold: (1) a lot of previous studies proved that the wavelet coding is better than the DCT-based coding and (2) the wavelet coding can be made scalable easily, which is essential in our multiple bit-streams based streaming system to control the overhead budget that is needed at each switching point.

The wavelet coding we have chosen in this paper is the SPIHT [6] algorithm. SPIHT itself is simple and straightforward. The only critical issue here is how to allocate the given bit budget over individual hierarchical trees that are formed after the wavelet.

2.1 Optimal Bit Allocation

The simplest strategy is to average the total budget over all hierarchical trees. However, due to the spatial location and intrinsic characteristics of individual trees, they play a role with different importance among a whole frame. Therefore, a bit allocation optimization is necessary.

Following the SPIHT principle, we know that a number of hierarchical trees, denoted as $T(k)$, $k = 1,...,K$, are generated after the wavelet decomposition of the reconstructed frame at a switching point. Each tree can be represented into an embedded bit-stream that can be truncated at any position, n_k. The contribution of $T(k)$ after truncating at n_k toward the overall distortion is denoted as $D_k(n_k)$. The goal is to select the optimal truncation position in the embedded bit-stream of each hierarchical tree, i.e., $\{n_k \mid k = 1,2,...K\}$, so as to minimize the overall distortion $D = \sum D_k(n_k)$ subject to the total budget B, where $\sum n_k \le B$.

This is a typical Lagrangian-type problem. However, since we cannot derive the exact expression of $D_k(n_k)$ in terms of n_k, this problem is not solvable analytically. In our work, we develop the following method: Firstly, we do the lossless bit-plane coding of each hierarchical tree $T(k)$ that is generated after the wavelet transform to the reference frame $F_u(t_m)$, starting from the highest bit-plane, assuming that there are totally a number of L bit-planes in each tree. Before the coding of the l-th bit-plane, we use the already coded L-th to $l+1$-th bit-planes to produce a reconstructed frame. Then, the sum of square differences between this reconstructed frame and the original frame is computed and denoted as $SSD_{l+1}(k)$. Clearly, $SSD_{l+1}(k) \ge SSD_l(k)$, and $SSD_{L+1}(k)$ is the sum of square differences between the original frame and zero. Based on $SSD_l(k)$, we define the distortion reduction as $\Delta SSD_l(k) = SSD_{l+1}(k) - SSD_l(k)$, which is the contribution from the coding of the l-th bit-plane of $T(k)$. In the meantime, we also count the number of bits for the lossless coding of the l-th bit-plane of $T(k)$, denoted as $C_l(k)$. Finally, a unit coding contribution (UCC) is defined as the ratio of $\Delta SSD_l(k)$ and $C_l(k)$, denote as $UCC_l(k)$.

After computing $UCC_l(k)$ for all k and l, we rank them from the largest to the smallest. Then, the SPIHT coding always starts from the bit-plane with the largest UCC, continues on the second largest one, and so on. The left part of Fig.1 shows the coding sequence where 4 hierarchical trees are included and each tree has 3 bit-planes. It is seen from this figure that there are totally 7 bit-planes to be selected and the numbers are used to denote the coding order. However, it is easy to see that such arrangement will run into problem in practice. As the bit-plane $N-1$ of T_2 is not selected, all bits received for the bit-plane $N-2$ of T_2 are not decodable. Similarly, as the bit-plane $N-2$ is selected before the bit-plane N in T_3, all bits in the bit-plane $N-2$ of T_3 may become un-decodable if it happens that some bits in the bit-plane N of T_3 are not sent. Some adjustments are therefore necessary. For this example, the correct coding sequence after the adjustment is shown in the right part of Fig. 1.

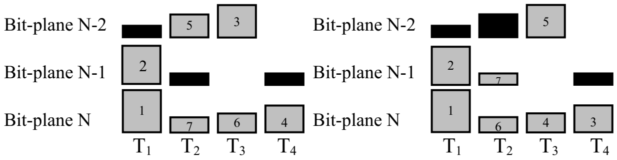

Fig. 1. Left: Coding sequence of one example with 4 trees. Right: Adjusted coding sequence of the same example

In practice, we need to compute $UCC_l(k)$ for each rate r_u, from the reconstructed frame $F_u(\cdot)$ at each switching point. Once the coding sequence is determined, we start the SPIHT coding until the given budget B is used up. In this way, B is un-evenly allocated over all hierarchical trees. The following matrix shows the actual bit allocation (with the total budget $B = 60$ kilobits) for the video sequence "Akiyo" (Y-component only, in CIF) at frame #15 (the source video is coded using H.264 with QP=32 and the 9/7 filter bank is used in the wavelet decomposition of 5 levels): it is seen that the allocation is very un-even:

$$[BAM] = \begin{bmatrix} 133 & 165 & 228 & 182 & 73 & 132 & 142 & 175 & 246 & 142 & 325 \\ 215 & 247 & 218 & 204 & 807 & 1103 & 857 & 498 & 621 & 247 & 392 \\ 181 & 342 & 170 & 223 & 911 & 1192 & 389 & 416 & 533 & 328 & 402 \\ 113 & 174 & 196 & 159 & 2141 & 1515 & 1247 & 890 & 117 & 118 & 397 \\ 257 & 361 & 293 & 316 & 1566 & 1521 & 1098 & 385 & 183 & 203 & 552 \\ 263 & 128 & 311 & 896 & 1508 & 1264 & 1423 & 1030 & 749 & 269 & 204 \\ 135 & 131 & 1303 & 177 & 618 & 1549 & 1057 & 851 & 476 & 940 & 411 \\ 240 & 598 & 762 & 383 & 347 & 1647 & 1153 & 1061 & 565 & 1483 & 246 \\ 150 & 810 & 584 & 290 & 219 & 1162 & 2186 & 1319 & 1603 & 942 & 316 \end{bmatrix}$$

with $\sum [BAM] = B$. Based on UCC, one bit allocation map $[BAM]_u$ can be derived for each r_u at a switching point. It is easy to see that about 1kbits (12 bits for each element) are needed to represent this map losslessly.

2.2 Influence on the Coding of Subsequent Frames

Same as in each SP-frame, the SPIHT processing of a reconstructed frame at each switching point will result in a different frame, and thus may cause some quality drop. More severely, this might influence the coding of all subsequent frames until the next I frame is transmitted. To evaluate how big this impact could be, we did many experiments, with some results presented in the following.

Figure 2 shows some results where there are 6 frames specifed as switching frames among 100 frames of the "Forman" and "Mobile" sequences (in CIF and at 30 frames/second), repectively. At each switching point, the reconstrued frame after the H.264 coding is further processed by SPIHT at $B = 70 + 10 + 10$ (for Y,U and V component, respectively) kbits for "Forman" and $B = 200 + 15 + 15$ kbits for "Mobile". The optimal bit alllocatin strategy developed above has been used in the SPIHT processing, and each SPIHT processed frame at a switching points is used in the coding of all subsequent frames. It is seen from these results that all quality curves after performing the SPIHT processing at each switching point do experience certain quality dorp, compared to the corresponding curves where all frames are coded as P-frames. However, it is well-controlled within 0.5 dB for both sequences. One important observation is that the coding quality drop at one swithing point does not seem to add up with others.

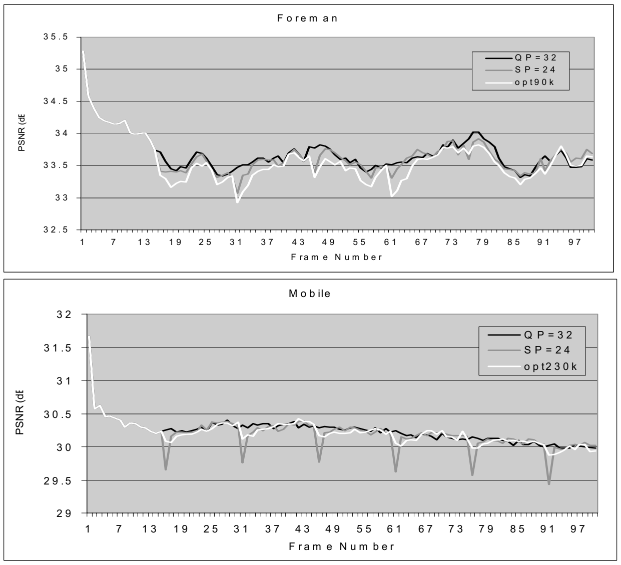

Fig. 2. Coding quality deviations after six reconstructed frames are further coded using SPIHT.

Figure 2 also includes the corresponding results of primary SP-frames genearted by H.264, where the re-quantization factore SPQP is set to be 24. It is clear that the SP-frame scheme yields slightly better results for the "Foreman" sequence and slightly worse results for the "Mobile" sequence. A comparision between the bit budget used in the SPIHT processing and the size of each secondary SP-frame generated in H.264 will be presented in the next section.

3. A New Switching Arrangement

After the switching frame $F_u(t_m)$ is further processed in the wavelet-domain for each rate r_u so as to obtain the modified version $\overline{F}_u(t_m)$, a simple but quite novel switching mechanism between two bit-streams can be arranged as in Fig. 3.

Suppose that the bit-stream at rate r_u is currently streamed and a switching to the rate r_v is needed right at the pre-selected point t_m. Then, the transmitted video frames around the switching point are arranged to be $\{F_u(t_m-1), \overline{F}_v(t_m), F_v(t_m+1)\}$. From our earlier analysis, we see that the number of bits used for representing $\overline{F}_v(t_m)$ is about 1kbits + B_v, where B_v is the total bit budget allowed at each switching point to SPIHT-code $F_v(t_m)$ into $\overline{F}_v(t_m)$, and about 1kbits are need to represent $[BAM]_v$ (as $F_v(t_m)$ is not available at the switching point, we need to know $[BAM]_v$ so that all received bits for representing $\overline{F}_v(t_m)$ can be correctly partitioned among hierarchical trees). On the other hand, $\{F_u(t_m-1), F_u(t_m)/\overline{F}_u(t_m), F_u(t_m+1)\}$ are sent if no switching happens. It is important to notice that the SPIHT processing on $F_u(t_m)$ so as to generate $\overline{F}_u(t_m)$ does not require any extra bits to be sent, because the same processing can be done at the receiver side.

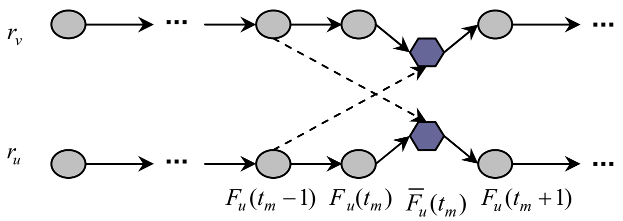

Fig. 3. A new switching arrangement between two bit-streams.

It is seen from Fig. 3 that, at the encoder side, we need to perform an SPIHT on each reconstructed frame at a switching point. On the other hand, the SP-frame scheme needs to perform a re-quantization to get one primary SP-frame and prepare all secondary SP-frames - the number may be large. We believe that the computational complexity involved in our scheme is much lower. However, we also need to do the SPIHT processing at the decoder side when a switching does not happen at a pre-selected point. This adds an extra complexity. Fortunately, switching points are not inserted very often and the SPIHT coding does not require too many computations as compared to other jobs that are needed to do at the decoder side (such as inverse VLC, de-quantization, IDCT, motion compensation, etc.).

We have run H.264 for both test sequences to generate all secondary SP-frames under the same configuration as used in Fig.2, and Table 1 presents the sizes of these secondary SP-frame at each pre-selected switching point for switching between QP=28 and QP=36, with SPQP=24. In fact, we have referred to the bit-counts listed in Table 1 to choose the budget B, so that it is always significantly (15% ~ 30%) smaller than the size of the corresponding secondary SP-frame.

It can be seen from Table 1 that the number of bits needed in the switching-up and switching-down cases are quite similar. We also find from our simulation results that the larger the difference between two QP values, the more extra bits will be needed to represent the secondary SP-frame. This result is acceptable in the switching-up case. However, it becomes rather absurd in the switching-down case: more extra bits have to be transmitted when the bandwidth already gets to be smaller! In principle, this problem can be avoided in our switching scheme by assigning a smaller budget B.

4. Experimental Results

In our simulations, 5 bit-streams are generated by using H.264 at different QP values: QP5=24, QP4=28, QP3=32, QP2=36 and

Table 1. Bit-counts of secondary SP-Frames.

Sequence	Switching	Frame #15	Frame #30	Frame #45	Frame #60	Frame #75	Frame #90
Foreman	QP:28→36	112600	111240	107496	113408	107248	117392
	QP:36→28	117448	115600	111632	117872	111776	123304
Mobile	QP:28→36	254584	251600	254784	257600	265888	272720
	QP:36→28	275456	272768	274432	280352	288888	296200

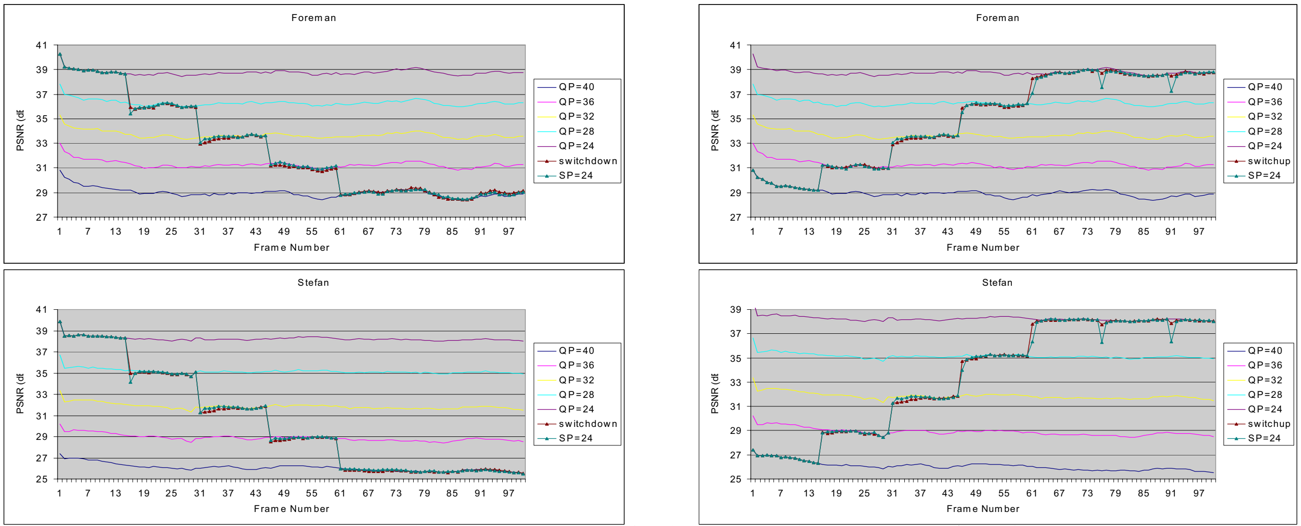

Fig. 4. Two switching scenarios among five bit-streams of "Foreman" and "Stefan".

Table 2. Budgets used in our simulations – same at all switching points.

Sequence	QP=40	QP=36	QP=32	QP=28	QP=24
Foreman	75+10+10	75+10+10	70+10+10	60+10+10	65+10+10
Mobile	170+10+10	160+10+10	120+10+10	120+10+10	130+10+10

QP1=40, respectively. Overall, 100 frames are encoded, with the first frame as I-frame and rest of them as P-frames. Then, six switching point are selected at #15, #30, #45, #60, #75 and #90. Figure 4 shows the results in terms of luminance PSNR for the "Foreman" and "Stefan" sequences, while Table 2 lists the bit budgets used to obtain theses results.

For each of these two sequences, the first plot shows the monotonic switching-down scenario and the second one shows the monotonic switching-up scenario. Five color curves without markers in Fig. 4 represent the H.264-coded results with all frames (except for the first one) coded as P-frames. Therefore, it is expected that the quality curve after inserting some switching points will always be (slightly) worse. However, it is seen from Fig. 4 that the results achieved in our switching scheme (the curves with small-triangle markers) are nearly perfect at all switching points for both sequences.

Figure 4 also presents the results obtained by using the SP-frame switching scheme (the curves with small-plus markers), and Table 3 summarizes the sizes of the corresponding secondary SP-frames that need to be sent at each switching point. It is seen that while the resulting quality curves are nearly the same as our results, the SP-frame switching scheme requires many more bits to be sent at each switching point.

5. Conclusions

MBR representation seems to be one good solution to the video streaming services over heterogeneous networks. In this paper, we developed an efficient method that allows seamless switching among different bit-streams in a multi-rate based streaming system when a channel bandwidth change is detected. The unique feature of our method is that, at a pre-selected switching point, the reconstructed frame at each rate undergoes through an independent SPIHT processing in the wavelet domain in which an optimal bit allocation over all hierarchical trees has been applied. Comparing with the SP-frame switching scheme, our method proves to be able to achieve the seamless switching at a better rate-distortion performance.

Acknowledgement: This work has been supported partly by DAG research grant from HKUST and a RGC research grant from HKSAR

References

[1] N. Färber and B. Girod, "Robust H.263 compatible video transmission for mobile access to video servers," in *Proc. ICIP'97*, Santa Barbara, CA, Oct. 1997.

[2] M. Karczewicz and R. Kurceren, "A proposal for SP-frames," ITU-T Video Coding Experts Group Meeting, Eibsee, Germany, Jan. 2001, Doc. VCEG-L-27.

[3] M. Karczewicz and R. Kurceren, "Improved SP-frame encoding," ITU-T Video Coding Experts Group Meeting, Austin, TX, April 2001, Doc. VCEG-M-73.

[4] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. CSVT*, vol. 13, pp. 637-644, July 2003.

[5] Z. X. Xiong, K. Ramchandran, M. T. Orchard, and Y. Q. Zhang, "A comparative study of DCT- and wavelet-based image coding", *IEEE Trans. CSVT*, vol. 9, no. 5, Aug 1999.

[6] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. CSVT*, vol. 6, pp. 243-250, June 1996.

Table 3. size of secondary SP-frame to be sent at each switching point (in bits)

sequence	scenario	#1	#2	#3	#4
Foreman	up	109792	101880	97664	96496
	down	95584	96920	98672	105992
Stefan	up	202272	185920	177936	156176
	down	159504	169104	185688	195208