# Seamless Switching of Scalable Video Bitstreams for Efficient Streaming

Xiaoyan Sun, Feng Wu, *Member, IEEE*, Shipeng Li, *Member, IEEE*, Wen Gao, *Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*—Efficient adaptation to channel bandwidth is broadly required for effective streaming video over the Internet. To address this requirement, a novel seamless switching scheme among scalable video bitstreams is proposed in this paper. It can significantly improve the performance of video streaming over a broad range of bit rates by fully taking advantage of both the high coding efficiency of nonscalable bitstreams and the flexibility of scalable bitstreams, where small channel bandwidth fluctuations are accommodated by the scalability of a single scalable bitstream, whereas large channel bandwidth fluctuations are tolerated by flexible switching between different scalable bitstreams. Two main techniques for switching between video bitstreams are proposed in this paper. Firstly, a novel coding scheme is proposed to enable drift-free switching at any frame from the current scalable bitstream to one operated at lower rates without sending any overhead bits. Secondly, an switching-frame coding scheme is proposed to greatly re-duce the number of extra bits needed for switching from the current scalable bitstream to one operated at higher rates. Compared with existing approaches, such as switching between nonscalable bitstreams and streaming with a single scalable bitstream, our experimental results clearly show that the proposed scheme brings higher efficiency and more flexibility in video streaming.**

***Index Terms*—Bitstream switching, fine granularity scalable video coding, scalable video coding, SP frame, video streaming.**

## I. Introduction

DUE TO THE explosive growth and great success of the Internet, as well as the increasing demands on video services, streaming video over the Internet has drawn tremendous attention in both academia and industry [1]. In contrast to download mode video services, where bitstreams are first downloaded completely over the Internet and then played back at the client, video streaming enables users to experience a presentation on the fly while it is being downloaded through the Internet. In virtue of the streaming techniques, users no longer have to suffer from long and even unacceptable transfer time for full download [2]–[4]. However, the Internet, which was initially designed for data transmission and communication among computers, is inherently a heterogeneous and dynamic network. For example, the channel bandwidth may fluctuate in a wide range from below 64 kbps to well above 1 Mbps. This brings great challenges to the present video coding and streaming technologies in providing a smooth playback experience and best video quality available to the users. In response to such challenges, a variety of techniques on coding and/or streaming [3]–[7] have been proposed to enhance video streaming services. Two major approaches, namely, switching among multiple nonscalable bitstreams and streaming with a single scalable bitstream, have been extensively investigated in recent years.

### A. Switching Among Multiple Nonscalable Bitstreams

In this approach, a video sequence is compressed into several nonscalable bitstreams at different bit rates. Channel bandwidth variation is adapted by dynamically switching among these bitstreams. Because of the temporal prediction, switching at a predictive frame would result in different references at the encoder and the decoder, and such a mismatch would bring the so called *drifting error* which would propagate to subsequent frames until the prediction chain is cut, for example, by an I frame. In order to achieve drift-free switching, some special frames, known as *key frames*, are coded either without temporal prediction or with an extra switching bitstream [8]–[11]. They provide access points to accomplish the switching.

When nonpredictive frames, which are usually I frames, are periodically inserted into every nonscalable bitstream as key frames, switching is performed by properly selecting a nonscalable bitstream according to the available channel bandwidth and by delivering the key frame of the selected bitstream instead of the current one. Subsequently, the selected bitstream is transmitted to the client. Obviously, the larger the number of nonscalable bitstreams and key frames, the more flexible this approach in terms of bandwidth adaptation is. In general, increasing the number of nonscalable bitstreams is limited by the storage and management capabilities of the server. When a change in channel bandwidth is detected, the required switching cannot be accomplished until the arrival of a key frame to avoid frame-drops and to enable drift-free switching. However, frequently inserting key frames into nonscalable bitstreams would greatly degrade the coding efficiency since no temporal correlation is exploited in key-frame coding. In addition, since the bitstream size of a compressed key-frame is far larger than that of a normal predictive frame at the same decoded quality level, long transmission latency and even network congestion may arise especially when the channel bandwidth drops. In this case,

Manuscript received December 30, 2002; revised October 12, 2003. This work was supported by Microsoft Research Asia and by the National Science Foundation of China (60333020). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pascal Frossard.

X. Sun was with the Department of Computer Science, Harbin Institute of Technology, Harbin 150001, China. She is now with Microsoft Research Asia, Beijing 100080, China (e-mail: t-xysun@microsoft.com).

F. Wu, S. Li, and Y.-Q. Zhang are with Microsoft Research Asia, Beijing 100080, China (e-mail: fengwu@microsoft.com; spli@microsoft.com; yzhang@microsoft.com).

W. Gao is with the Institute of Computing Technology, Beijing 100080, China (e-mail: wgao@ict.ac.cn).

Digital Object Identifier 10.1109/TMM.2003.822818

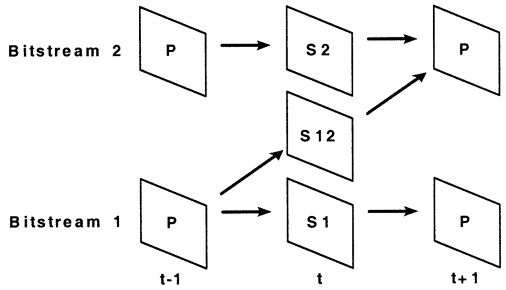



Fig. 1. Switching from Bitstream 1 to Bitstream 2 through SP frames.

sending more data would further deteriorate the network conditions. Thus, the approach of switching among nonscalable bitstreams, using key frames, only provides coarse and lagged capability in adapting to channel bandwidth variations.

Drift-free switching among nonscalable bitstreams through predictive frames has also been investigated in recent years. Since a predictive frame is always coded using the prediction from the previous reconstructed reference, without any special treatment, switching between bitstreams at such a frame would lead to drifting errors due to the mismatch of the reconstructed references. One way to solve such a problem is to losslessly compress the difference between two reconstructed references into an extra bitstream at the switching points [8]. Obviously, this would not normally affect the coding efficiency of a nonscalable bitstream even though an extra bitstream is associated with each predictive frame at the server. However, whenever switching is requested, the extra bitstream must be sent to the client and unfortunately, it also brings huge and often unacceptable additional overhead bits.

In addition, Farber *et al.* first proposed a special predictive frame called the *S frame* to achieve switching among nonscalable bitstreams at predictive frames in [9]. But it was shown that S-frames tend to drift, and in order to keep drifting error within a small scale, S-frame size has to be considerably large. On the other hand, a technique which supports drifting-free switching at predictive frames has also been proposed to the JVT standard in [10], [11] and accepted as a new picture type, the *SP frame* [12]. Like the normal predictive frame, an SP frame exploits temporal redundancy using motion compensated predictive coding while allowing identical reconstruction of the frame even when different reference frames are used. Fig. 1 illustrates the process of switching from one nonscalable bitstream to another through SP frames. Two bitstreams, Bitstream 1 and Bitstream 2, are generated at different bit rates. The frames at $t-1$ and $t+1$ are coded as normal predictive frames in both Bitstream 1 and Bitstream 2. $S_1$, $S_2$ and $S_{12}$ at time $t$ are coded as SP frames, where an access point is provided for switching from Bitstream 1 to Bitstream 2 and vice versa. Because of the temporal prediction, SP frames can provide more switching points than I frames with same or similar coding efficiency. However, the size of an S12 bitstream is still similar to that of an I frame. When switching to bitstreams at lower bit rates, which means the networks already get congested, sending such a great amount of overhead bits in S12 may further deteriorate the networks. Therefore, the SP frame only partially solves the problems that exist in the approach of switching among multiple nonscalable bitstreams. For simplicity, the SP method proposed in [10], [11] is denoted as Org SP in this paper hereafter.

### B. Streaming With a Single Scalable Bitstream

In this approach, a video sequence is compressed into a single scalable bitstream, which can be truncated flexibly to adapt to channel bandwidth variation. Among numerous scalable coding techniques, MPEG-4 fine granularity scalable (FGS) coding has become prominent due to its fine-grain scalability [5], [13], [14]. In MPEG-4 FGS, a video sequence is represented by two layers of bitstreams, namely, the base layer bitstream and the enhancement layer bitstream. The base layer bitstream is coded with nonscalable coding techniques, whereas the enhancement layer bitstream is generated by coding the difference between the original image and the reconstructed base layer image using the bit-plane coding technique in [15]. Since the enhancement layer bitstream can be truncated arbitrarily in any frame, MPEG-4 FGS provides a nice capability in readily and precisely adapting to channel bandwidth variation. However, its motion prediction is always based on the lowest quality base layer. As a result, low coding efficiency is a major disadvantage that prevents MPEG-4 FGS from being widely deployed in video streaming applications.

The progressive FGS (PFGS) coding scheme [16], [17] is a significant improvement over MPEG-4 FGS by introducing two prediction loops with different quality references. Unlike MPEG-4 FGS, the PFGS scheme can use a reconstruction at the enhancement layer as a reference. Because the quality of a reconstructed frame is higher at the enhancement layer than at the base layer, the PFGS scheme provides more accurate motion prediction than MPEG-4 FGS, thus improving the coding efficiency. Since the enhancement layer reference may not be always available at the decoder, this would inevitably cause some drifting errors at lower enhancement bit rates. To solve this problem, macroblock-based PFGS (MBPFGS) [18] provides a good tradeoff between coding efficiency improvement and drifting error reduction by optimally selecting the reference of the enhancement layer at macroblock level. The experimental results in [18] show that MBPFGS can outperform MPEG-4 FGS up to 2 dB, while the drifting errors at lower enhancement bit rates are still much limited. In addition, RFGS proposed in [19] also presents a similar performance with two-loop prediction and drifting attenuation techniques. However, since only one high-quality reference is used in the enhancement layer coding, most coding efficiency gain appears within a certain bit-rate range around the high-quality reference. Generally, with today's technologies, there is still a coding efficiency loss in scalable coding schemes as compared with the nonscalable schemes over a broad range of bit rates.

### C. Seamless Switching Scheme Among Scalable Bitstreams

This paper proposes a seamless switching scheme among scalable bitstreams to significantly improve the efficiency of video streaming over a broad range of bit rates. Here, the *seamless switching* implies three meanings.

1) The quality in each scalable bitstream is smooth without the so-called quality cliff-off phenomenon.

2) The switching among scalable bitstreams is drifting-free.
3) Immediately switching from the current scalable bitstream to one operated at lower rates is allowed without any delay.

In the proposed scheme, each scalable bitstream has a base layer encoded at a different bit rate and can better adapt to channel bandwidth variation within a certain range of bit rates. If the channel bandwidth is out of this range, the scalable bitstream can be seamlessly switched from one to another with higher coding efficiency. A real-life analogy of this approach is very similar to changing gears when driving a car at different speeds. We will refer to switching from a scalable bitstream operated at lower bit rates to one operated at higher bit rates as *switching up* and the reverse process as *switching down* hereafter in this paper.

The key issue in this paper is how to flexibly and efficiently perform switching up and down among scalable bitstreams. Due to different channel conditions, the requirements for switching up and switching down are also quite different. When channel bandwidth somehow drops, the server has to rapidly switch from one bitstream at high bit rate to another at low bit rate to reduce the packet loss ratio and to maintain smooth video playback. Therefore, the technique for switching down should satisfy two basic requirements: 1) the scalable bitstreams could be switched down at any frame and 2) overhead bits should be avoided during switching down because they will increase the network traffic and may further deteriorate network conditions. Therefore, a novel technique is proposed in this paper for flexibly and efficiently switching down between scalable bitstreams.

When channel bandwidth increases, a delay is usually needed for the server to make a reliable decision for switching up. Moreover, it is also acceptable to transmit additional bits in this case. Therefore, the method of associating an extra bitstream with every switching point is adopted in the proposed scheme for switching up. As we mentioned above, in order to prevent drifting errors, the extra bitstream has to be generated by losslessly coding the mismatch between the reconstructed frames with different bitstreams. However, such a method may suffer from a large amount of overhead bits whenever a switching up is requested. To greatly reduce the number of overhead bits, a new technique referred to as *switching frame* (SF) is proposed in this paper for switching up. The SF scheme is similar to but different from the Org SP scheme in JVT standard.

This paper is organized as follows. Section II gives an overview of the proposed seamless switching scheme. The techniques for switching down and switching up are discussed in details in Sections III and IV, respectively. Example encoder and decoder for the seamless switching scheme with two MBPFGS bitstreams are described in Section V. Experimental results are given in Section VI. Finally, Section VII concludes this paper.

## II. The Proposed Seamless Switching Scheme

To address the different requirements on switching up and switching down, the proposed scheme for seamless switching among scalable bitstreams is discussed in this section. For simplicity, the proposed scheme with two scalable bitstreams is depicted in Fig. 2. The upper one outlined by the dot-and-dash box is the scalable bitstream with higher bit rate $R_H$ base layer, referenced as *SB-H*; whereas the corresponding lower one is the scalable bitstream with lower bit rate $R_L$ base layer, referenced as *SB-L*. In general, the base layer bitstream has to be completely transmitted to the client. That is, the bit rate of the base layer is the lower bound of channel bandwidth covered by a scalable bitstream. The enhancement layer bitstream can be transmitted partially according to the available channel bandwidth, as depicted in Fig. 2, where the shadow regions at the enhancement layer indicate the actual transmitted parts.

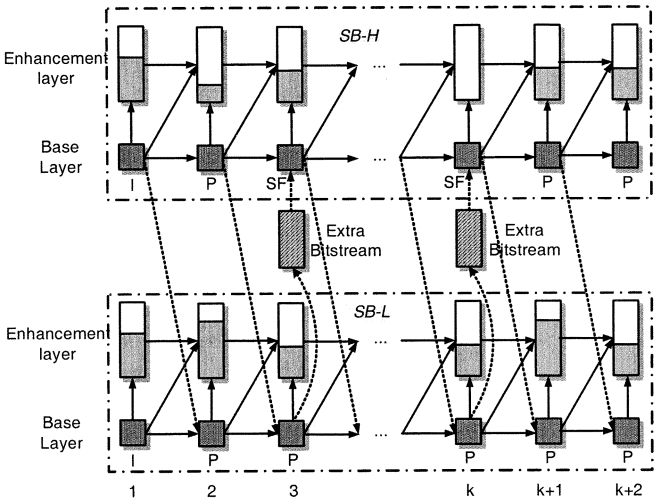



Fig. 2. Proposed scheme of seamless switching between scalable bitstreams.

Assume that channel bandwidth variation in a certain application is from $R_{\min}$ to $R_{\max}$. Obviously, $R_L$ should be no more than the minimum channel bandwidth, i.e., $R_L \leq R_{\min}$. $R_H$ is somewhere between $R_L$ and $R_{\max}$. It is recommended that $R_H$ is selected around $(R_{\max} + R_{\min})/2$ thus approximately covering the whole range of channel bandwidth with two scalable bitstreams. Although the bit rate of each enhancement layer bitstream can be truncated up to $R_{\max}$, it is not necessary in the proposed switching scheme. The maximum bit rate covered by *SB-L* is recommended to be slightly higher than $R_H$. With these two scalable bitstreams, any desired bit rate within the bandwidth range can be achieved by selecting the proper scalable bitstream and truncating the enhancement layer bitstream. To be more specific, if the available channel bandwidth is less than $R_H$, then *SB-L* is sent to the client; otherwise *SB-H* is sent.

The solid lines with solid arrows between two frames indicate motion prediction loops. In the advanced FGS coding technologies (e.g., PFGS and RFGS), the enhancement layer can be predicted from the reconstructed base layer, the reconstructed enhancement layer, or even the combination of them as proposed in [20], [21]. On the other hand, in the MPEG-4 FGS, the enhancement layer is predicted only from the base layer. Any corruptions at the enhancement layer bitstream, such as bitstream truncation, packet losses and transmitted errors, can be recovered automatically in the frames followed. This is an important feature provided by all FGS coding schemes. This is not the case for the base layer, however, where obviously any corruption or error could have a severe impact in the decoded video quality. Therefore, the key problem in the proposed seamless switching scheme is how to efficiently switch up and down among scalable bitstreams without introducing any drifting errors into the base layer.

As shown by the dashed lines with solid arrows from *SB-H* to *SB-L* in Fig. 2, the proposed scheme can switch down at any frame without additional bits. In other words, when a switching down is requested due to channel bandwidth drop, the proposed scheme does not have to know whether the current frame is an I frame or not and can immediately transmit another scalable bitstream in the next frame. It is one of the most desirable features in the video streaming applications. However, in order to eliminate drifting errors, the reconstructed base layer in *SB-L* has to be always identical regardless of switching down or not. The proposed scheme actually generates the two related base layers for this purpose. First, motion vectors are estimated in *SB-H* and are applied to both *SB-H* and *SB-L*. Second, the quantization information of *SB-L* is coded in *SB-H* bitstream. Finally, the video frames to be encoded at the base layer of *SB-L* are the reconstructed base layer frames from *SB-H* instead of the original video. Therefore, when decoding the *SB-H* base layer at the decoder, a "transcoding-like" process is going on simultaneously in the decoder to regenerate exactly the same *SB-L* base layer that has been encoded at the encoder with the quantization parameters and motion vectors transmitted with the *SB-H* bitstream. The identical reconstruction technique for switching down is discussed in details in the Section III.

Different from switching down, when switching up, the identical reconstruction of the base layer is guaranteed by using an additional bitstream. Since a delay is usually needed for the server to make a reliable decision for switching up, the proposed scheme may insert switching points periodically in the coded video sequence, such as at the third and $k$th frames indicated in Fig. 2. The reconstructed *SB-L* base layer together with the residue encoded in the extra bitstream can precisely recover the *SB-H* base layer.

Moreover, the extra bitstream has the "*additive*" property. This is different from Org SP frames where you can have either one lower bit-rate bitstream or one higher bit-rate bitstream. The additive property of the extra bitstream provides an "*embedded*" bitstream feature which could be very useful for some scenarios.

For example, while transmitting a *SB-L* bitstream, if the server decides to switch to *SB-H* right away, with the SF scheme, we could just send the extra bitstream to make the switching happen within this frame. This is in contrast to the Org SP scheme in which we have to wait until the next frame to send the switching bitstream to complete the switching. In addition, when the server tries to switching up and the client somehow loses the extra bitstream, it could just inform the server to stay in the *SB-L* bitstream without being interrupted by such a switching miss. However, with Org SP scheme, once the switching bitstream is corrupted or not received in the next frame, it is not possible for the client to fallback to the original bitstream without causing errors in the received video. Moreover, if we want to extend the SF scheme to more than two bitstreams, say $N$ bitstreams, it only requires $(N-1)$ extra bitstreams in order to achieve the arbitrary switching from one bitstream to any another one while it requires $N^*(N-1)$ bitstreams in Org SP scheme. All these are brought by the "additive" property of the extra bitstream.

The switching points in *SB-H* are *SF* frames instead of normal predictive frames so as to prevent drifting errors and reduce the size of the extra bitstream. Similar to a normal predictive frame, the SF frame uses the previous reconstructed reference for temporal prediction. Therefore, an SF frame is far better than an I frame in terms of coding efficiency and it is very close to a predictive (P) frame. The only difference between P frames and SF frames is that the reconstructed image in an SF frame is quantized twice. The extra bitstream is generated by first quantizing the reconstructed images from the SF frame and the *SB-L* base layer and then losslessly coding the residue between them. Normally, the size of the extra bitstream in the proposed scheme is still similar to that of an I frame, but it is only required in the case of switching up. The proposed SF frame coding technique is discussed in detail in Section IV.

## III. Switching Down Between Bitstreams

As mentioned above, the signal encoded at the *SB-L* base layer is the difference between the reconstructed *SB-H* base layer image and the *SB-L* base layer temporal prediction. Since the motion vectors and the quantization information are also coded in the *SB-H* base layer, the decoder can surely compute the *SB-L* base layer when the *SB-H* bitstream is being decoded. This is the key point to achieve switching down at any frame without having to transmit any additional bits. For the convenience of discussion hereafter, a lowercase letter denotes an image in pixel domain, and the corresponding uppercase letter denotes an image in the discete cosine transform (DCT) domain. The subscripts "b" and "e" indicate the base layer and the enhancement layer, respectively. The hat "$\sim$" denotes reconstructed image or DCT image.

Before a switching down is requested, the server is transmitting the *SB-H* bitstream to the client. Thus, the reconstructed *SB-H* base layer is available at both the encoder and decoder. When channel bandwidth drops below the effective bit-rate range of *SB-H*, the streaming server has to promptly switch from *SB-H* to *SB-L* to reduce packet loss ratio and maintain smooth playback. Considering that the network at such conditions can hardly afford to transmit more overhead bits, the proposed scheme prefers to calculate the reconstructed *SB-L* base layer directly from the *SB-H* base layer without sending any overhead bits, provided that the temporal prediction and quantization parameters of the *SB-L* base layer are available.

The quantization parameters of the *SB-L* base layer can be readily coded in the *SB-H* bitstream. As the quantization parameter (QP) is usually set in the range of $[1, 31]$ in MPEG-4, the number of overhead bits used in coding the QP at frame level is 5 bits. When the QPs are adjusted at macroblock (MB) level, the difference between two MB QPs, which is in the range of $[-2, +2]$, is coded with VLC. The average overhead bits should be less than $3 * \text{MB\_number}$. Therefore, together with the 5 bits for coding the quantization parameter of each frame, the number of overhead bits for QP's is $(3 * \text{MB-number}) + 5)$ per frame. For example, when video is coded in QCIF format, if the SB-H bitstream base layer is set to be 80 kbps at 10 fps, the percentage of the extra bits is around $(3*99+5)/8000 \approx 3.8\%$ Moreover, we only use frame-level QP for the results presented in Section VI so the overhead bits for carrying the quantization parameters are negligible.

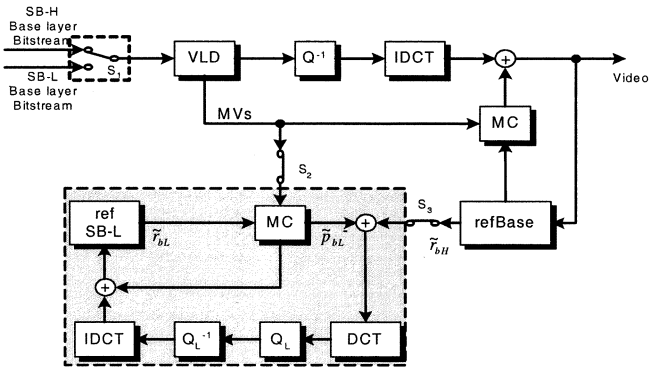



Fig. 3. Proposed base layer decoder with the switching down technique.

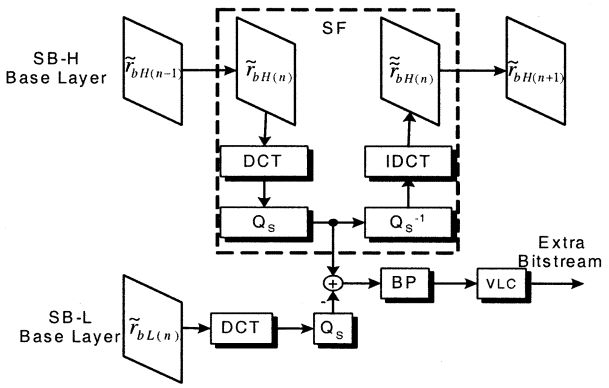



Fig. 4. Block diagram of the SF encoder.

With regard to the temporal prediction of the *SB-L* base layer, it will be always calculated in the *SB-H* base layer decoder. For the convenience of explaining this process, the modules outlined by the dashed line box in Fig. 3 illustrate the generation of the *SB-L* base layer when decoding the *SB-H* bitstream. In this case, while decoding the *SB-H* base layer, a simultaneous "transcoding-like" process also takes place to generate the *SB-L* base layer. This is the most important feature of the proposed scheme. On the other hand, when the *SB-L* bitstream is the input to the base layer decoder, the switches S2 and S3 in Fig. 3 are off and the dashed box part is not used. In this case, the decoder becomes a normal decoder.

When the base layer decoder starts to process the *SB-H* bitstream, it is reasonable to assume that the previous base layer reference $\tilde{r}_{bL}$ reconstructed from *SB-L* is already available and preserved in the *ref SB-L* module. There are two possible cases.

1) We just switched from the *SB-L* bitstream in the current frame, where the base layer reference $\tilde{r}_{bL}$ for *SB-L* is available from the *SB-L* bitstream decoding in the previous frame.
2) We are already decoding the *SB-H* bitstream, where the *SB-H* decoding process described as follows guarantees a *SB-L* base layer reference $\tilde{r}_{bL}$.

After the reconstruction of the current base layer $\tilde{r}_{bH}$ from *SB-H*, with the connection of the switches $S_2$ and $S_3$, $\tilde{r}_{bH}$ is input into the additional part before decoding the next frame. Because the same motion vectors are used in *SB-L* and *SB-H*, the temporal prediction $\tilde{p}_{bL}$ is obtained by performing similar motion prediction based on the previous reference $\tilde{r}_{bL}$. Subsequently, $\tilde{p}_{bL}$ is subtracted from $\tilde{r}_{bH}$ to obtain the same residue as that encoded in the *SB-L* base layer at the encoder. Since the *SB-L* quantization parameters are already coded in the *SB-H* bitstream, the *SB-L* base layer residue will go through the same process as in the *SB-L* base layer encoder, such as DCT, quantization and inverse quantization, IDCT, and motion compensation to reconstruct exactly the same base layer $\tilde{r}_{bL}$ as that at the *SB-L* base layer encoder. This process will be repeated while *SB-H* is being decoded. Since the reconstructed base layer $\tilde{r}_{bL}$ is always available at the decoder while decoding the *SB-H* bitstream, the proposed scheme can switch down freely at any frame.

Note that in the proposed scheme, the coding efficiency of the *SB-L* base layer would slightly drop if compared with an independently encoded *SB-L* base layer, since: 1) the reconstructed image of the *SB-H* base layer instead of the original video is used as the input of the *SB-L* base layer encoder, and the video quality has a slight loss in the beginning and 2) the same set of motion vectors obtained for the *SB-H* base layer is used to encode the *SB-L* base layer, and it may not be optimized for the *SB-L* base layer at a lower bit rate. Of course, the second problem can be solved if the *SB-H* bitstream also carries another set of motion vectors optimized for *SB-L* base layer.

## IV. Switching up Between Bitstreams

As depicted in Fig. 2, the proposed scheme switches up from the *SB-L* bitstream to the *SB-H* bitstream by attaching an extra bitstream to every switching point. Since the extra bitstream is usually generated by losslessly coding the mismatch information, this would require to transmit a considerable amount of overhead bits whenever a switching up is requested. To further improve the performance of the proposed scheme, the SF technique is discussed in this section which allows for a significant reduction on the number of the overhead bits required for switching up.

Fig. 4 illustrates the block diagram of the SF frame encoder. Assume that the $n$th frame is a switching point. In normal predictive frame coding, the reconstructed image $\tilde{r}_{bH(n)}$ from the *SB-H* base layer bitstream will be directly forwarded to the next frame as reference. However, in order to compress the difference between $\tilde{r}_{bH(n)}$ and $\tilde{r}_{bL(n)}$ into an acceptable size $\tilde{r}_{bH(n)}$ in the SF frame is transformed and quantized with the parameter $Q_s$ once more. The same DCT transform and quantization is also performed on $\tilde{r}_{bL(n)}$. As a result, the size of the extra bitstream is greatly reduced by coding the DCT residue between quantized $\tilde{r}_{bH(n)}$ and $\tilde{r}_{bL(n)}$.

Obviously, $\tilde{r}_{bL(n)}$ plus the DCT residue encoded in the extra bitstream can only recover the quantized $\tilde{r}_{bH(n)}$. In order to avoid the drifting errors caused by switching up, the SF frame uses $\tilde{\tilde{r}}_{bH(n)}$ instead of $\tilde{r}_{bH(n)}$ as the reference for the next frame coding. In other words, there are two paths to generate the identical $\tilde{\tilde{r}}_{bH(n)}$, one from the *SB-H* bitstream and another from the *SB-L* bitstream plus the extra bitstream. This feature is just what is desired by the switching process.

A larger $Q_s$ parameter means the smaller the size of the extra bitstream and the lower the quality of $\tilde{\tilde{r}}_{bH(n)}$. In fact, compared

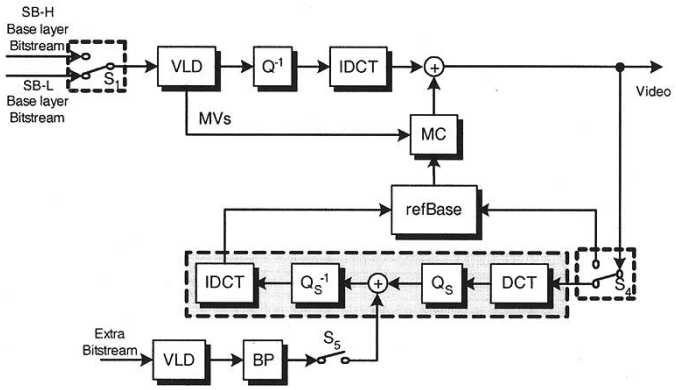



Fig. 5. Proposed base layer decoder with the switching up technique.

with switching through I frames and through lossless coding of the residue, SF frames can provide a better tradeoff between the coding efficiency of *SB-H* and the size of the extra bitstream by adjusting the parameter $Q_s$. Different entropy-coding techniques may be applied to generate the extra bitstream. For instance, the proposed scheme adopts the bit-plane coding technique as in MPEG-4 FGS.

Besides the size of the extra bitstream, another main concern in SF frame coding is the effect of the second quantization on the coding efficiency of two scalable bitstreams. The quantization on $\tilde{r}_{bL(n)}$ is only for the purpose of switching up. It does not involve the *SB-L* coding, thus the *SB-L* bitstream still maintains its original performance. Using $\tilde{\tilde{r}}_{bH(n)}$ as the reference will definitely affect the coding efficiency of *SB-H* to some extent. This is decided by the parameter $Q_s$ and the frequency of inserting SF frames. A smaller $Q_s$ parameter will have less influence on the performance of the *SB-H* bitstream but will increase the size of the extra bitstream. Fortunately, the SF frame usually appears at an interval of half a second or more in most applications. Hence, the coding efficiency loss caused by an SF frame should be a reasonable expense for switching up.

Accordingly, the base layer decoder of the proposed scheme for switching up is depicted in Fig. 5. The modules outlined by a dashed line box are the unique parts to the SF frame. When the *SB-H* base layer bitstream is input into the decoder, two cases are taken into account. If the current frame is not an SF frame, the decoder will act the same as a normal predictive decoder, that is, the refBase module is directly updated by the reconstructed image. If the current frame is an SF frame, the additional modules in Fig. 5 will take effect so that the reconstructed reference is quantized again before updating the refBase module. While a switching up takes place, the extra bitstream has to be transmitted and decoded. By adding the obtained residue to the quantized reference from the *SB-L* base layer, the quantized reference from the *SB-H* base layer is exactly recovered.

In fact, both the Org SP frame [10]–[12] and the SF frame employ similar methods for drift-free switching among bitstreams. However, the SF frame has several advantages. First, the SF frame only performs the second quantization on the reconstructed reference, whereas Org SP frame needs to quantize both the temporal prediction and the reconstructed reference. Thus, the SF frame is simpler especially in hardware implementation. Second, since an SF frame can output a higher quality reconstruction reference for display before the second quantization, it provides a better quality image for display than an Org SP frame. The SF frame coding technique has been proposed [22], [23] to the JVT standard, and the SF frame concept has been extended and integrated with Org SP scheme to form the final SP scheme in the latest JVT standard [24].

## V. Seamless Switching Encoder and Decoder With Two MBPFGS Bitstreams

To better understand the proposed seamless switching scheme, an example encoder and decoder are discussed in this section. Either MPEG-4 FGS or the advanced FGS coding technologies (e.g., PFGS and RFGS) can be adopted in the proposed scheme. In addition, one-loop MC-FGS [25] can also be applied to the proposed scheme. But since one-loop MC-FGS suffers from drifting errors at the base layer, the proposed scheme cannot guarantee the drift-free switching up and down in this case. For better coding performance, the MBPFGS [18] is chosen as the basic scalable video codec in this paper.

In the proposed seamless switching scheme, different decoding processes are used to decode the MBPFGS-H bitstream and the MBPFGS-L bitstream. The decoder for the MBPFGS-H bitstream indeed consists of two parallel decoders, one for the high-bit-rate reconstruction itself and the second one for the low-bit-rate reconstruction. On the other hand, the MBPFGS-L decoder only consists of a single decoder. When the MBPFGS-L bitstream is being decoded, the decoder is as same as a MBPFGS decoder if current frame is not a switching point. When a switching up is required, an additional quantization process with $Q_s$ is performed on the reconstructed reference of MBPFGS-L base layer to generate the quantized reference for switching. Meanwhile, the extra bitstream for switching up is decoded with BP decoder. Together with the quantized reference, the base layer of MBPFGS-H can be readily reconstructed. On the other hand, the decoder of the MBPFGS-H is more complicated than a MPFGS decoder. When the MBPFGS-H bitstream is decoded, one MBPFGS decoder is used to decoding the MBPFGS-H bitstream itself and an additional base layer decoding process is utilized to generate the MBPFGS-L base layer. The decoding process is discussed in details in the following paragraphs.

### A. Proposed Seamless Switching Encoder With Two MBPFGS Bitstreams

The proposed seamless switching encoder with two MBPFGS bitstreams is illustrated in Fig. 6. Motion estimation modules are omitted for simplicity. Two MBPFGS encoders are outlined by the dashed line boxes in Fig. 6. The upper one is denoted as MBPFGS-L because it generates a scalable bitstream with a lower bit —ate base layer, whereas the lower one is denoted as MBPFGS-H with a higher bit-rate base layer. The middle part between the two MBPFGS encoders is used to generate an extra bitstream for switching up.

The original video is first input to the MBPFGS-H encoder. Since the same motion vectors are used in both MBPFGS encoders, integer motion vectors are initially estimated by referencing the previous original video frame. Then, the fractional

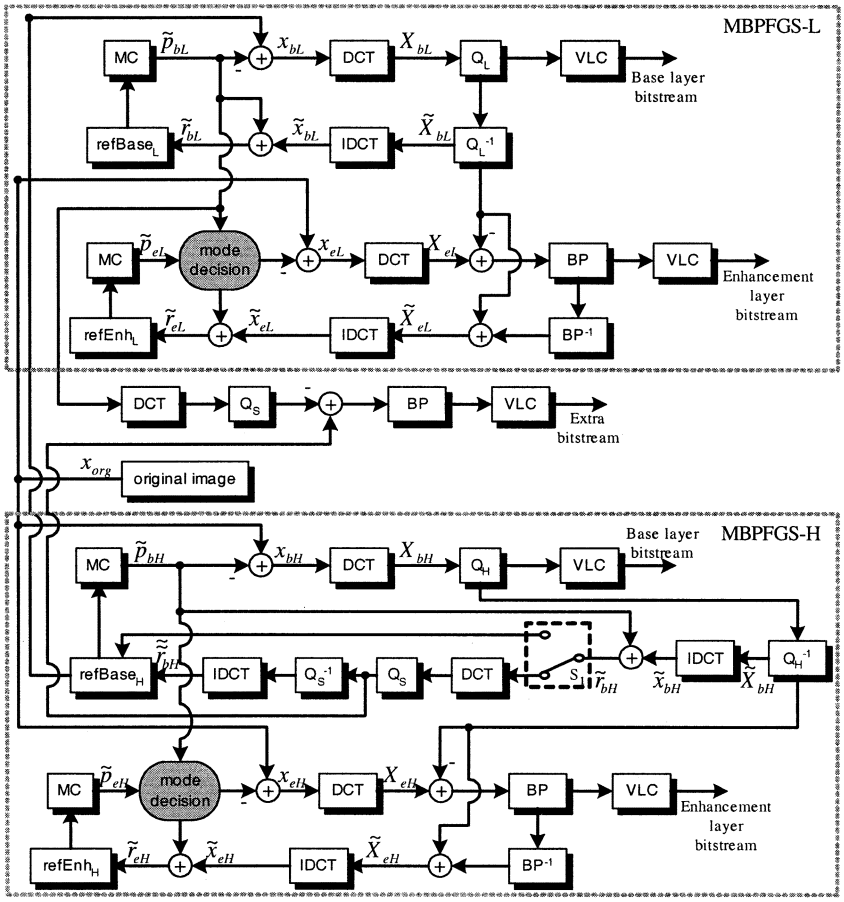



Fig. 6. Example encoder with two MBPFGS bitstreams.

pixel motion estimation is performed by referencing the base layer reconstructed from MBPFGS-H to maintain its coding efficiency. As shown in Fig. 6, each MBPFGS codec owns two different quality references because of the two-loop prediction technique. For instance, the low-quality reference $\tilde{r}_{bH}$ stored in the $\mathrm{refBase}_H$ module is reconstructed only from the base layer bitstream, whereas the high-quality reference $\tilde{r}_{eH}$ stored in the $\mathrm{refEnh}_H$ module is reconstructed from the base layer bitstream plus a part of the enhancement layer bitstream.

In MBPFGS coding, only the low-quality reference is allowed to be used in the base layer prediction and reconstruction, whereas the enhancement layer can select either the low-quality reference or the high-quality reference for a better prediction. When the high-quality reference is selected in the enhancement layer coding, the reference for reconstruction may be different from that for prediction. Because the low-quality reference is always available at both the encoder and decoder, selecting a low-quality reference would help MBPFGS to eliminate the drifting errors caused by the loss of the enhancement information at lower enhancement bit rates. Three modes for coding the enhancement layer macroblock are defined in MBPFGS to specify the actual references for each macroblock. The mode selection algorithm described in [18] is applied in the mode decision module.

There are two types of predictive frames, the SF frame and the normal predictive frame, in MBPFGS-H. The $S_1$ switch in Fig. 6 is under the control of the two predictive frame types. When the current frame is a normal predictive frame, the reconstructed reference $\tilde{r}_{bH}$ is directly stored into the $\mathrm{refBase}_H$ module. Otherwise, when the current frame is an SF frame, the $\mathrm{refBase}_H$ module is updated by the quantized $\tilde{r}_{bH}$.

MBPFGS-L obtains the motion vectors directly from MBPFGS-H without motion estimation. According to the proposed switching down technique, $\tilde{r}_{bH}$ or $\tilde{\tilde{r}}_{bH(n)}$ stored in the $\mathrm{refBase}_H$ module is forwarded to the MBPFGS-L base layer encoder as the input. Therefore, the predicted error $x$ encoded at the MBPFGS-L base layer is

$$x_{bL} = \tilde{r}_{bH} - \tilde{p}_{bL} \text{ or } x_{bL} = \tilde{\tilde{r}}_{bH} - \tilde{p}_{bL}$$

instead of

$$x_{bL} = x_{\mathrm{org}} - \tilde{p}_{bL}$$

where $x_{\mathrm{org}}$ is the original video and $\tilde{p}_{bL}$ is the temporal prediction in the MBPFGS-L base layer. However, the predicted error $x_{eL}$ is still calculated from the original video as usual

$$x_{eL} = x_{\mathrm{org}} - \tilde{p}_{eL}$$

so as to maintain the coding efficiency of the enhancement layer in MBPFGS-L.

The extra bitstream is generated only when the current frame is an SF frame in the MBPFGS-H base layer. The encoded signal is the difference between quantized $\tilde{r}_{bH}$ and $\tilde{r}_{bL}$ at the same time instant.

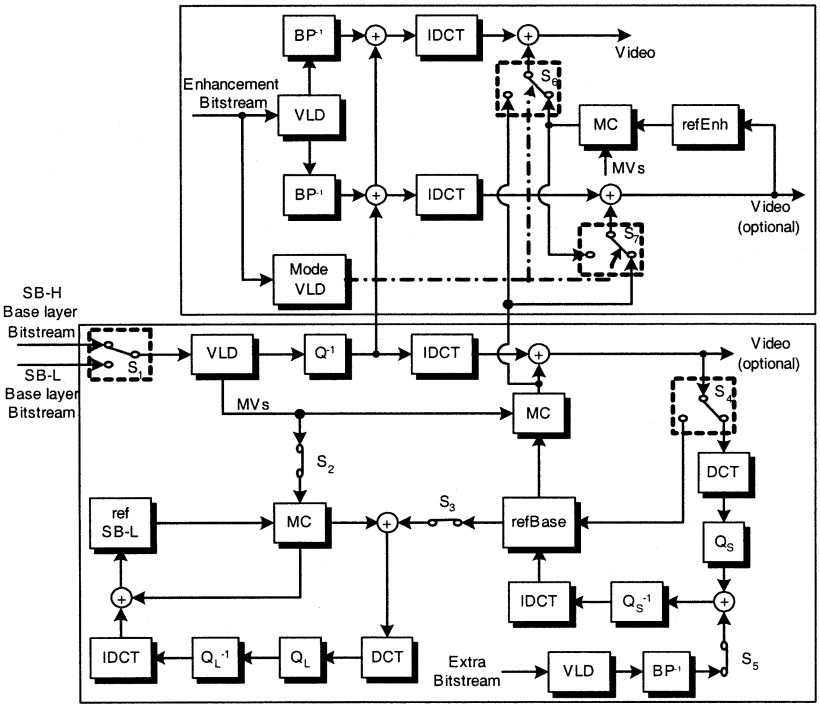



Fig. 7. Example seamless switching decoder.

### *B. The Proposed Seamless Switching Decoder With Two MBPFGS Bitstreams*

Fig. 7 illustrates the proposed seamless switching decoder, where a base layer bitstream and an optional enhancement layer bitstream are the normal inputs. In addition, the extra bitstream is input when a switching up takes place. The decoding processes of the base layer and the enhancement layer are outlined by the dashed line boxes. The upper part denotes the enhancement layer decoder, whereas the lower part is the base layer decoder.

The base layer decoder realizes the switching techniques discussed in Sections III and IV. When the MBPFGS-L bitstream is being decoded, switches $S_2$, $S_3$, and $S_5$ are off, while switch $S_4$ turns to the left to merely deliver the reconstructed base layer reference to the refBase module. If a switching up takes place at a certain access point, switch $S_4$ takes the right path and switch $S_5$ is on. Meanwhile, the reference stored in refBase is first copied to reference SB-L module for tracking the *SB-L* base layer decoding. The reconstructed base layer reference from MBPFGS-L is quantized with the parameter $Q_S$. This is added to the residue decoded from the extra bitstream which can identically reconstruct the desired base layer reference for decoding the MBPFGS-H bitstream.

When the MBPFGS-H bitstream is being input into the decoder, switches $S_2$ and $S_3$ are on so as to reconstruct the base layer reference of MBPFGS-L. Switch $S_5$ is off. If the current frame is an SF frame, switch $S_4$ will take the right path; otherwise, it will take the left path. Whenever a switching down is requested, the reference stored in the ref SB-L module is copied to the refBase module. In the next frame, the decoder starts to process the MBPFGS-L bitstream without drifting.

The MBPFGS-L and MBPFGS-H enhancement layer bitstream is decoded with the same block diagram. The Mode VLD module decodes the coding mode of each macroblock at the enhancement layer which is used to control the operations of the switches $S_6$ and $S_7$. In general, the bit rate of the high-quality reference is lower than the total bit rate of a scalable bitstream. Therefore, the enhancement layer decoder reconstructs the high-quality reference with part of the enhancement layer bitstream plus the base layer bitstream. However, the entire received bitstream is used to reconstruct a higher quality image for display.

## VI. Experimental Results and Analysis

Four different schemes, namely the proposed seamless switch scheme, MPEG-4 FGS, MBPFGS, and switching between nonscalable bitstreams through I frames (denoted as I-Switch), are compared in terms of both coding efficiency and channel bandwidth adaptation. All schemes are implemented based on MPEG-4. The QCIF sequences *News*, *Coastguard* and *Foreman* are used at 10 fps in this experiment. Except for the switching scheme through I frames, all other coding schemes in the experiment are coded with only one I frame at the beginning of the bitstream, whereas the rest of the frames are coded as P frames for better evaluating switching performance. The TM5 rate control is used in the base layer coding. The range of motion vectors in all schemes is limited to $\pm 15.5$ pixels with half pixel accuracy.

In the proposed scheme, the bit rate of the MBPFGS-L base layer is 32 kbps. The high-quality reference is reconstructed at 64 kbps (base layer plus 32 kbps enhancement layer), and the channel bandwidth covered by MBPFGS-L is from 32 to 128 kbps. The bit rate of the MBPFGS-H base layer is 80 kbps including the overhead bits for coding the quantization parameters of the MBPFGS-L base layer. The high-quality reference

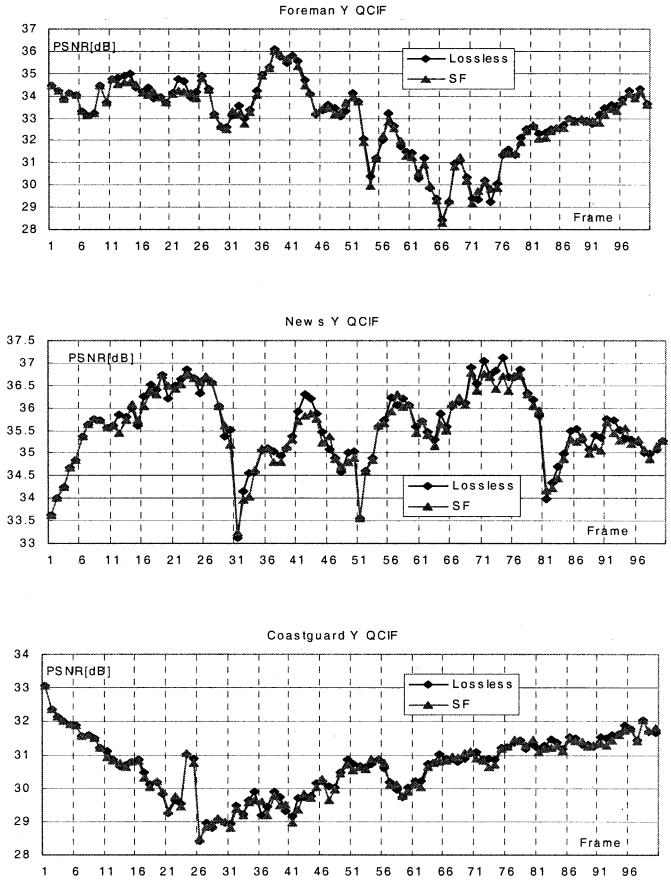



Fig. 8. Coding efficiency comparisons between SF with the quantization parameter $Q_S = 3$, and lossless residue coding without SF. Key frames for switching up are inserted with a 10-frame interval in the MBPFGS-H base layer.

is reconstructed at 112 kbps. The channel bandwidth range covered by the MBPFGS-H can be from 80 kbps up to the lossless rate. However, this experiment constrains the upper bound of the MBPFGS-H bitstream to 160 kbps.

Switching among nonscalable bitstreams is extensively used in many commercial streaming video systems. In this experiment, two MPEG-4 nonscalable bitstreams are generated with the same conditions as in the MBPFGS-L and MBPFGS-H base layers. However, an I frame is inserted at every 1-s interval for reducing the delay of switching between nonscalable bitstreams. In the single MPEG-4 FGS and MBPFGS schemes, the base layer bit rate is equal to that in MBPFGS-L for achieving the same lower bound on channel bandwidth. The higher quality reference in the single MBPFGS bitstream is reconstructed from several lower bit planes so that the total bit rate is initially more than 40 kbps.

The advantages and disadvantages of SF frames are first evaluated in the proposed scheme. Access points for switching up are inserted with 1s interval in the MBPFGS-H base layer. The negative effects of the SF frame on the coding efficiency are depicted in Fig. 8. The SF curve is obtained by inserting SF frames with the quantization parameter $Q_S = 3$ for all sequences. In the lossless case, the switching points are the normal predictive frames without the additional SF quantization. The experimental results show that the two PSNR curves are very close and

TABLE I
AVERAGE NUMBER OF BITS FOR EACH FRAME IN THE EXTRA BITSTREAM

| Sequence | SF | Lossless |
|---|---|---|
| News | 32959.11 | 147878.1 |
| Foreman | 34918.33 | 173569.3 |
| Coastguard | 39824.78 | 158157.7 |

the average PSNR loss caused by SF frames is below 0.1 dB. The average extra frame size listed in Table I is the average number of extra bits needed for each switching up. Compared with the lossless method, the SF frames can save up to 80% bits for switching up. In addition, the coding efficiency of the Org SP frame, SF frame, and I frame at the same baseline codec (i.e., JVT H.26L codec) have been evaluated in the JVT standard [11], [22]. The SF frame provides better coding efficiency.

The coding efficiency of these four schemes is compared in Fig. 9 using the curves of average PSNR versus bit rate. This is the static performance when no switching takes place. Switching between nonscalable bitstreams using I frames only provides two different quality levels, whereas the other three schemes can flexibly and precisely adapt to channel bandwidth variations and provide a smooth visual quality. Compared with the single MPEG-4 FGS and the single MBPFGS bitstream schemes, our proposed scheme can be up to 3.0 and 2.0 dB better at high bit rates, respectively.

As we mentioned before, since the same set of motion vectors obtained for the high bit-rate bitstream is used to encode the low bit-rate video in the proposed scheme, the coding efficiency of the low bit-rate bitstream is going to drop. Moreover, the reconstructed image of MBPFGS-H is used as the input video of the MBPFGS-L base layer encoder. It will also affect the coding performance of MBPFGS-L. As shown in Fig. 9, the coding efficiency of the MBPFGS-L loses on an average of 0.5 dB compared with the single MPFGS method with both base layers at bit-rate 32 kbps.

A dynamic channel is specified to verify the performance of the four different schemes in terms of bandwidth adaptation. The bit rate periodically switches from 72 to 152 kbps. For sequences *News* and *Coastguard*, each cycle starts at 72 kbps for 1 s and then switches to 152 kbps for 3 s. For the *Foreman* sequence, each cycle starts at 72 kbps for 2 s and then switches to 152 kbps for 3 s. The overhead bits for switching up are included in this simulation. The PSNR versus frame number curves are shown in Fig. 10. The proposed scheme switches up 3 times and switches down two times in order to adapt to the channel bandwidth variations. Apparently, the proposed scheme with the switching up and down techniques can always achieve the best performance among the four evaluated schemes at both low and high bit rates.

We note that from Table I the average frame size in the extra bitstream for switching up is quite large. Therefore, frequently switching ups and downs will definitely lower the efficiency of such a seamless switching scheme. For example, if for every second, the scalable bitstreams switch up and down once, the extra bitstream for switching up will cost 30–40 kbps bandwidth on average. This will bring a significant performance hit

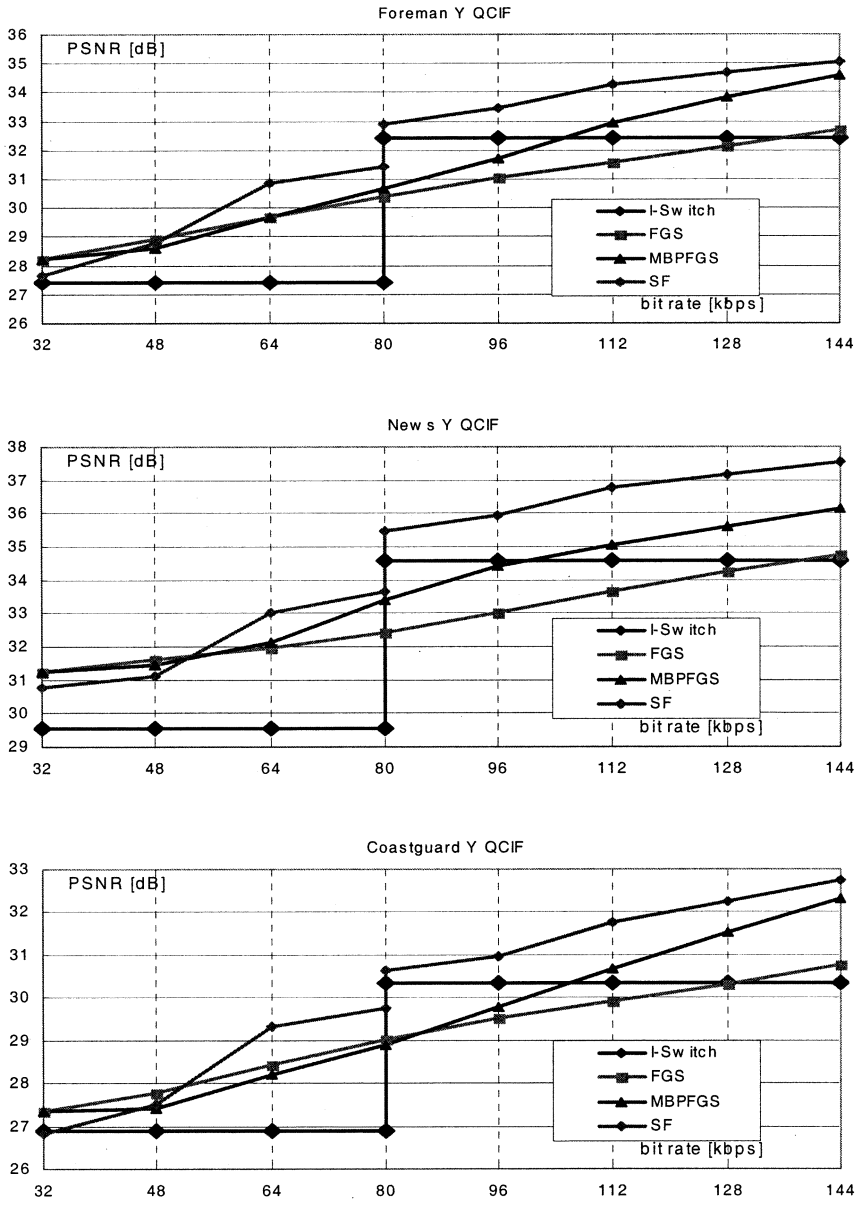



Fig. 9. Average PSNR versus bit-rate curves of four different schemes. The channel bandwidth varies from 32 to 144 kbps. A different bitstream is used for bit rates over 80 kbps in the I-Switch and SF schemes.

to the proposed approach. It could make the proposed scheme less efficient than just staying with the lower rate scalable bitstream. Fortunately, in practice, the streaming server has the ability to decide when to switch the bitstreams. An intelligent server should be able to consider the cost of such a switching and try to minimize the switching as much as possible. For example, if it detects that the channel is not stable enough with frequent bandwidth increases and drops, it might decide to just stay in the low rate scalable bitstream instead of switching up. Only if the channel bandwidth is steadily high for a cons-bit-rate scalable bitstream to avoid the excess overhead switching bitstream.

Although, for simplicity, this paper only discusses and evaluates the proposed switching scheme with only two scalable bitstreams, the proposed scheme can be extended to the case with multiple scalable bitstreams. In this case, the bitstreams can still switch from any one to another one same as with Org SP frame. For switching down, we keep all the lower bit-rate references (with increased complexity) during decoding a higher bit-rate bitstream. For switching up, we need only to transmit all the extra difference bitstreams between the current bitstream and the one we need to switch to, though according to the TCP-friendly protocol, switching up is a gradual process, and in most cases it is just one bitstream up at a time. One requirement here is that the $Q_s$ must be the same in generating all the extra bitstreams. As a minor advantage to some complexity-tolerable applications, in the SF frame case we only need $(N-1)$ extra bitstreams to accomplish the above arbitrary switching, whereas in the Org SP frame case, we would need $N*(N-1)$ extra bitstreams to achieve arbitrarily switching from one bitstream to another.

The major drawback of the proposed scheme is the high complexity. The complexity of the proposed scheme is much higher than a standard MPEG-4 decoder. When the lowest bit-rate bitstream is being decoded, the complexity is as same as

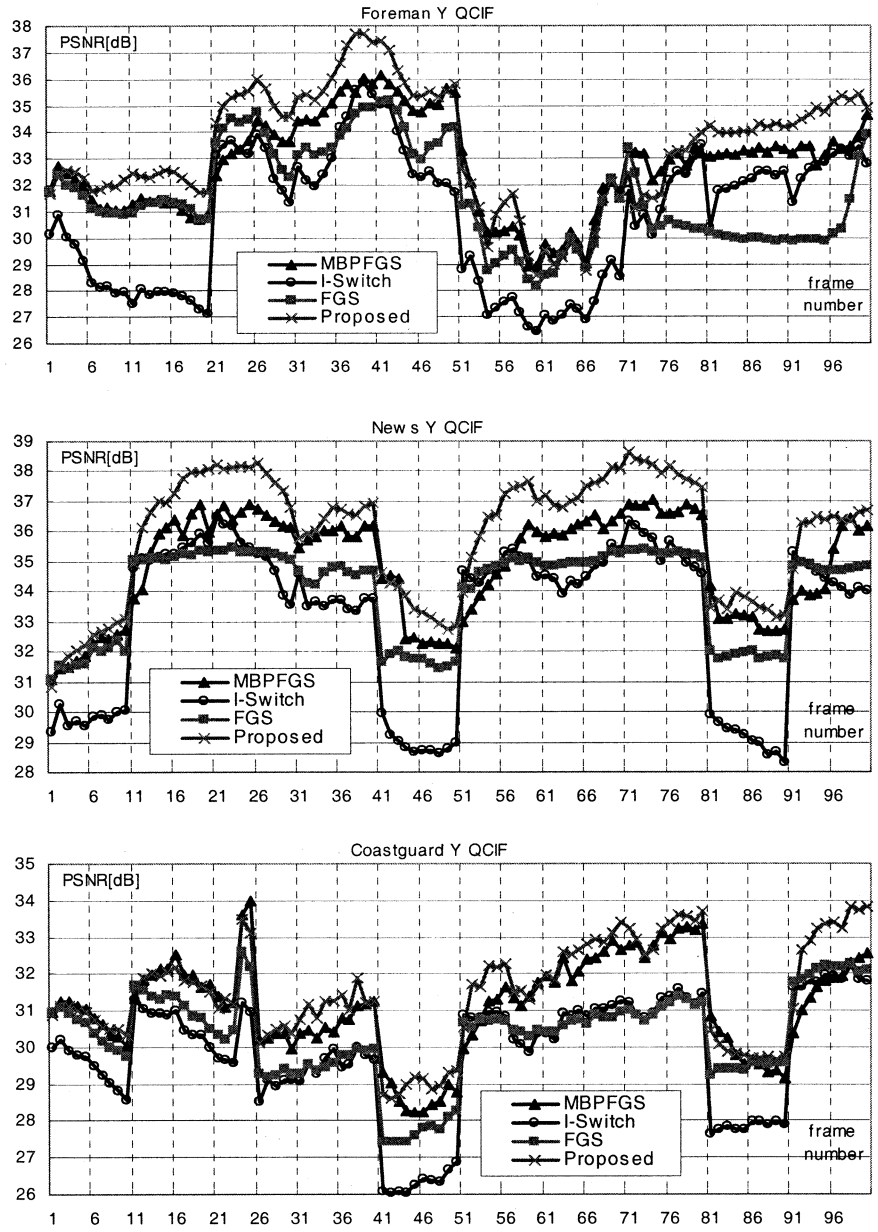



Fig. 10. PSNR versus frame number curves of four different schemes when channel bandwidth varies periodically between 72 and 152 kbps. For sequences *News* and *Coastguard*, each cycle starts at 72 kbps for 1 s and then switches to 152 kbps for 3 s. For the *Foreman sequence*, each cycle starts at 72 kbps for 2 s and then switches to 152 kbps for 3 s. The overhead bits for switching up are included in this simulation.

the PFGS decoder. However, when the high bit-rate bitstream is decoded, an additional set of DCT, IDCT, quantization, inverse quantization, and motion compensation modules are required as shown in Fig. 7. We have performed profiling of the cost of each module. As a rule of thumb, the PFGS decoder is about twice as complex as the nonscalable decoder (a standard MPEG-4 decoder) and the complexity of the proposed decoder increases in the order of a standard MPEG-4 decoder. Therefore, the complexity of the proposed decoder in decoding the high-bit-rate scalable bitstream is about three times of that of a standard MPEG-4 decoder. More, when additional higher bit-rate bitstreams are added, we need an extra set of DCT, IDCT, quantization, inverse quantization, and motion compensation modules for each additional bitstream, which is roughly the complexity of an extra MPEG-4 decoder. Therefore, the more the bitstreams that are supported, the more complex the decoder is if we want to decode the highest bit-rate bitstream. Besides the complexity, the large size of the overhead bits inside the highest bit-rate bitstream is another problem that the proposed scheme has to get over when supporting multiple bitstreams. Two bitstreams can provide good-quality service in most applications.

While the complexity may be very high in the proposed decoder, it also provides complexity scalability so as to be used on low-end and power-constrained video terminals. It can scale from the low bit-rate base layer decoding to low bit-rate scalable stream decoding, to high bit-rate base layer decoding to high-rate scalable bitstream decoding. This could be very useful to the power-constrained terminals. For such devices, device power status and network bandwidth can be jointly considered

to achieve optimal performance. The proposed scheme enables such a capability. When only the lowest bit-rate base layer bitstream is decoded in the terminal, the complexity of the proposed scheme is the same as a standard MPEG-4 decoder.

Besides the modules for generating the extra bitstream for switching up shown in Fig. 6, the encoder complexity increase for the proposed scheme is minimal. Of course, for each scalable bitstream, a scalable encoder has to be in place. This is different from the decoder, where the decoding of the enhancement layer bitstreams from different scalable bitstreams can share the same enhancement decoder.

## VII. Conclusions

This paper proposes a seamless switching scheme among scalable video bitstreams by fully taking advantage of both the high coding efficiency of nonscalable bitstreams and the flexibility of scalable bitstreams. In response to the different requirements on bitstream switching, two techniques are proposed in this paper for switching up and switching down, respectively. In a wide range of bit rates, the proposed scheme can improve coding efficiency up to 3.0 and 2.0 dB compared with MPEG-4 FGS and other advanced FGS technologies, respectively. Moreover, the proposed scheme allows switching down at any frame without having to transmit any additional bits, thus saving bandwidth when it is most needed. Furthermore, the proposed SF frame technique greatly reduces the overhead bits for switching up. The size of the extra bitstream is only about 20% of that when the difference is coded losslessly, while there is only about 0.1-dB coding efficiency loss on average. Dynamic tests under bandwidth fluctuations also show that the efficiency of proposed scheme is better than other schemes.

In addition, the proposed drift-free switching techniques only operate on the base layers of the scalable bitstreams. It can be also applied for switching among nonscalable bitstreams. However, since each scalable bitstream can cover a certain bit-rate range, the proposed switching scheme is most effective if applied in scalable bitstreams to cover broader bit-rate range efficiently.

Although we use two scalable bitstreams in our example encoder and decoder, the proposed scheme can be readily extended to support more scalable bitstreams with increased complexity.

The major drawback of the proposed approach is the high complexity. The proposed decoder to decode a high bit-rate scalable bitstream is about three times as complex as a standard MPEG-4 decoder. However, the complexity scalable feature of the proposed scheme on the other hand gives different devices the option to choose different levels of service according to their capabilities.

## Acknowledgment

The authors thank Dr. G. Shen and A. M. Tourapis for many valuable discussions and suggestions.

## References

[1] M. R. Civanlar, A. Luthra, S. Wenger, and W. W. Zhu, "Introduction to the special issue on streaming video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 265–268, Mar. 2001.

[2] J. Lu, "Signal processing for internet video streaming: A review," *Proc. SPIE, Image Video Commun. Process.*, vol. 3974, pp. 246–258, 2000.

[3] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the internet: Approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 282–300, Mar. 2001.

[4] G. Conklin, G. Greenbaum, K. Lillevold, A. Lippman, and Y. Reznik, "Video coding for streaming media delivery on the internet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 269–281, Mar. 2001.

[5] W. Li, "Streaming video profile in MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 301–317, Mar. 2001.

[6] M. Gallant and F. Kossentini, "Rate-distortion optimized layered coding with unequal error protection for robust internet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 357–372, Mar. 2001.

[7] W. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans Circuits Syst. Video Technol.*, vol. 11, pp. 373–386, Mar. 2001.

[8] B. Girod, N. Farber, and U. Horn, "Scalable codec architectures for internet video on demand," in *Proc. 1997 Asilomar Conf. Signals and Systems*, vol. 1, Nov. 1997, pp. 357–361.

[9] N. Farber and B. Girod, "Robust H.263 compatible video transmission for mobile access to video servers," in *Proc. Int. Conf. Image Processing*, vol. 2, Oct. 1997, pp. 73–76.

[10] M. Karczewisz and R. Kurceren, "A proposal for SP-frames," ITU-T Q.6/SG 16 VCEG-L27, 2001.

[11] R. Kurceren and M. Karczewisz, "Improved SP-frame encoding," ITU-T Q.6/SG 16 VCEG-M73, 2001.

[12] ITU-T Rec.H.264, ISO/IEC 14 496-10 AVC, "Joint Final Committee Draft (JFCD) of Joint Video Specification," JVT-D157, July 2002.

[13] W. Li, "Fine granularity scalability in MPEG-4 for streaming video," in *ISCAS 2000*, vol. 1, Geneva, Switzerland, May 2000, pp. 299–302.

[14] M. van der Schaar and H. Radha, "A hybrid temporal-SNR fine-granular scalability for internet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 318–331, Mar. 2001.

[15] F. Ling, W. Li, and H. Sun, "Bit-Plane coding of DCT coefficients for image and video compression," in *Proc. SPIE VCIP'99*, San Jose, CA, Jan. 1999.

[16] F. Wu, S. P. Li, and Y. Q. Zhang, "DCT-prediction based progressive fine granularity scalable coding," in *Proc. Int. Conf. Image Processing*, Vancouver, BC, Canada, 2000, pp. 566–569.

[17] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 332–344, Mar. 2001.

[18] X. Sun, F. Wu, S. Li, W. Gao, and Y.-Q. Zhang, "Macroblock-based progressive fine granularity scalable video coding," in *ICME 2001*, Tokyo, Japan, Aug. 2001.

[19] H. Huang, C. Wang, and T. Chiang, "A robust fine granularity scalability using trellis-based predictive leak," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 372–385, June 2002.

[20] A. Reibman and L. Bottou, "Managing drift in DCT-based scalable video coding," in *Data Compression Conf.*, 2001, pp. 351–360.

[21] W. S. Peng and Y. K. Chen, "Mode-adaptive fine granularity scalability," in *ICIP*, Thessaloniki, Greece, 2001, pp. 993–996.

[22] X. Sun, F. Wu, S. Li, W. Gao, and Y.-Q. Zhang, "Improved SP coding technique," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG*, Geneva, Switzerland, Jan. 2002, Doc. JVT-B097.

[23] X. Sun, F. Wu, S. Li, and R. Kurceren, "The improved JVT-B097 SP coding scheme," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG*, Fairfax, VA, May 2002, Doc. JVT-C114.

[24] "Joint final committee draft (JFCD) of joint video specification," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG*, Klagenfurt, The Netherlands, July 2002, Doc. JVT-D157.

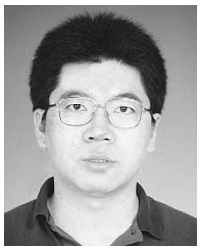
[25] M. Schaar and H. Radha, "Motion-compensation fine-granular-scalability (MC-FGS) for wireless multimedia," in *Proc. IEEE 4th Workshop on Multimedia Signal Processing*, Cannes, France, Oct. 2001, pp. 453–458.

**Xiaoyan Sun** received the B.S. and M.S. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 1997 and in 1999, respectively, where she is currently pursuing the Ph.D. degree.

Since 2000, she has been with Microsoft Research Asia, Beijing, China. Her research interests include video/image coding, video streaming, and multimedia processing.

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

He joined Microsoft Research Asia, Beijing, China, as an Associate Researcher in 1999 and was promoted to Researcher in 2001. He has played a major role in Internet Media Group to develop scalable video coding and streaming technologies. He has authored and co-authored over 60 papers in video compression and contributed some technologies to MPEG-4 and H.264. His research interests include video and audio compression, multimedia transmission, and video segmentation.



**Shipeng Li** (M'97) received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC), Hefei, 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, USTC, 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, Beijing, since May 1999. He has contributed some technologies in MPEG-4 and H.264. His research interests include image/video compression and communications, digital television, multimedia, and ireless communication.

**Wen Gao** (M'99) received the M.S. and Ph.D. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 1985 and in 1988, respectively, and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was a Research Fellow at Institute of Medical Electronics Engineering, University of Tokyo, Tokyo, Japan, in 1992, and a Visiting Professor at Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, in 1993. From 1994 to 1995, he was a Visiting Professor at the MIT AI Lab, Massachussetts Institute of Technology, Cambridge. Currently, he is the Vice President of the University of Science and Technology of China, the Deputy President of Graduate School of Chinese Academy of Sciences, Professor of computer science at Harbin Institute of Technology, and the Honor Professor in computer science at City University of Hong Kong. He is the head of Chinese National Delegation to the MPEG Working Group (ISO/SC29/WG11). He has published seven books and over 200 scientific papers. His research interests are in the areas of signal processing, image and video communication, computer vision, and artificial intelligence.

Dr. Gao is the Editor-in-Chief of the *Chinese Journal of Computers* and was the General Co-Chair of the IEEE International Conference on Multi-model Interface in 2002.

**Ya-Qin Zhang** (S'87–M'90–SM'93–F'98) received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 1983 and 1985, and the Ph.D. degree in electrical engineering from George Washington University, Washington, DC, in 1989.

He is currently the Corporate Vice President of Microsoft, responsible for mobility and embedded system products. He was the Managing Director, Microsoft Research Asia, Beijing, China, in 1999. Previously, he was the Director of the Multimedia Technology Laboratory, Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). Prior to that, he was with GTE Laboratories, Inc., Waltham, MA, from 1989 to 1994. He has been engaged in research and commercialization of MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He has authored and co-authored over 200 refereed papers in leading international conferences and journals, and has been granted over 40 U.S. patents 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 serves on the Board of Directors of five high-tech IT companies and has been a key contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia.

Dr. Zhang served as the Editor-In-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from July 1997 to July 1999. He was the Chairman of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems (CAS) Society. He serves on the editorial boards of seven other professional journals and over a dozen conference committees. He has received numerous awards, including several industry technical achievement awards and IEEE awards, such as the CAS Jubilee Golden Medal. He was named "Research Engineer of the Year" in 1998 by the Central Jersey Engineering Council for his "leadership and invention in communications technology, which has enabled dramatic advances in digital video compression and manipulation for broadcast and interactive television and networking applications." He received The Outstanding Young Electrical Engineer of 1998 award.