A Hybrid Temporal-SNR Fine-Granular Scalability for Internet Video

Mihaela van der Schaar, *Member, IEEE*, and Hayder Radha, *Member, IEEE*

***Abstract*—Transmission of video over bandwidth varying networks like the Internet requires a highly scalable solution capable of adapting to the network condition in real-time. To address this requirement, scalable video-coding schemes with multiple enhancement layers have been proposed. However, under this multiple-layer paradigm, the transmission bit-rate of each layer has to be predetermined at encoding time. Consequently, the range of bit-rates that can be covered with these compression schemes is limited and often lower than, or different from, the desired range required at transmission time. In this paper, a novel scalable video-coding framework and a corresponding compression method for Internet video streaming are introduced. Building upon the MPEG-4 SNR fine-granular-scalability (FGS) approach, the proposed framework provides a new level of abstraction between the encoding and transmission process by supporting both SNR and temporal scalability through a single enhancement layer. Therefore, our proposed approach enables streaming systems to support full SNR, full temporal, and hybrid temporal-SNR scalability in real-time depending on the available bandwidth, packet-loss patterns, user preferences, and/or receiver complexity. Moreover, our experiments revealed that the presented FGS temporal-SNR scalability has similar or better PSNR performance than the multilayer scalability schemes. Subsequently, an Internet video streaming system employing the proposed hybrid FGS-temporal scalability structure is introduced, together with a very simple, yet effective, rate-control that performs the tradeoffs between individual image quality (SNR) and motion-smoothness in real-time. The hybrid temporal-SNR scalability presented in this paper has been recently adopted in the MPEG-4 standard to support video-streaming applications.**

***Index Terms*—FGS, FGS-temporal scalability, internet video streaming, MPEG-4, scalable video-coding, temporal-SNR rate-control.**

I. Introduction

MULTIMEDIA distribution over the Internet is becoming increasingly popular. However, since the Internet was designed for computer data communication, satisfying the necessary requirements for the effective delivery of multimedia streams poses significant challenges. For example, the Internet is characterized by large bandwidth variations due to heterogeneous access-technologies of the receivers (e.g., analog modem, cable modem, xDSL, etc.) or due to dynamic changes in network conditions (e.g., congestion events). Moreover, the Internet experiences a relatively high percentage of packet-losses due to the underlying "best-effort" model of the Internet Protocol (IP) [1], [8].

To cope with the heterogeneous structure of the Internet and its lack of Quality-of-Service (QoS) guarantees, scalable coding schemes have been proposed for Internet video streaming. However, most of these schemes concentrate on SNR-scalability or combinations of SNR-scalability with spatial and/or temporal scalability using multiple layers [2], [3], where the transmission bit-rate of each layer is predetermined at encoding time. Consequently, the range of bit-rates that can be covered with these compression schemes is limited and often lower than, or different from, the desired range required at transmission time. Also, the tradeoffs between image quality (SNR) and temporal resolution (motion-smoothness) are made at encoding-time and do not take into account the actual (available) transmission bandwidth, packet-losses, viewing preferences of the users, or receiver complexity.

Recently, another class of scalable coding algorithms has been proposed for Internet video streaming: the 3-D wavelet/sub-band coding techniques [4]–[7]. While these methods are able to adapt in real-time (i.e., at transmission time) to the Internet bandwidth variations, their disadvantages are relatively high computational complexity and large memory requirements which make them unpopular for low-power devices like the mobile phones, for instance.

Alternatively, in the MPEG-4 standard, a fine-granular scalability (FGS) video-coding scheme [8]–[10] has recently been introduced that provides both bandwidth-scalability and packet-loss resilience [11] at a low-cost. However, a limitation of the original MPEG-4 FGS framework (see Fig. 1) is that only the image quality of the base-layer pictures can be enhanced (i.e., it provides only SNR scalability). However, if clients with very different connection capabilities need to access the same video sequence, tradeoffs should be made between the frame rate (motion smoothness) and image quality (SNR) of each individual frame. Therefore, the frame rate of the transmitted video sequence has to be enhanced in conjunction with the individual image quality.

In this paper, a novel scalable video-coding framework and a corresponding compression method for Internet video streaming is introduced. Building upon the MPEG-4 FGS approach, the proposed framework provides a new level of abstraction between the encoding and transmission process by supporting *both* SNR and temporal scalability through a *single* enhancement layer. This abstraction is important, since the transmission bandwidth is not known at encoding time and thus, the optimal tradeoffs cannot be made *a priori*. Moreover, depending on the individual user preference, which

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

The authors are with Philips Research Laboratories, Briarcliff Manor, NY 10510 USA (e-mail: Mihaela.vanderSchaar@philips.com; Hayder.Radha@philips.com).

Publisher Item Identifier S 1051-8215(01)01853-5.

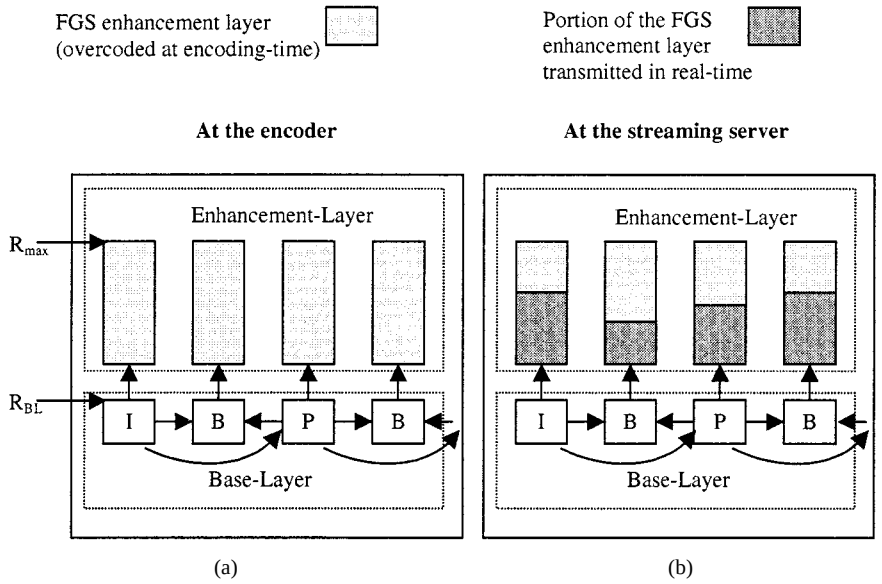


Fig. 1. FGS structure at the encoder and streaming server for a typical Internet streaming application.

cannot be anticipated at encoding time, the individual image quality or the motion smoothness can be enhanced. With the proposed solution, which employs a fine-granular single layer for both SNR and temporal scalability, these decisions can be easily performed at transmission time depending on the user, decoder or server requirements. Another advantage of the novel framework presented in this paper is its reduced decoder complexity, requiring minimal addition to the original MPEG-4 FGS (SNR) implementation.

The proposed hybrid temporal-SNR FGS scalable coding is based on a novel scalable technique for coding B-frames. This novel method is especially beneficial for devices with limited computational resources (e.g., mobile phones, wireless gadgets, etc.) that cannot guarantee the full decoding of nonscalable B-frames. With the proposed scalable algorithm, the video quality is enhanced even if the B-frames are only partially decoded due to the limited computational resources.

The paper is organized as follows. In Section II, the FGS SNR-scalability currently adopted in MPEG-4 is presented. Subsequently, in Section III-A, a new method for the fine-granular coding of B-frames is proposed. Section III-B presents the FGS temporal-SNR scalability structure and in Section III-C its performance is evaluated against that of a multilayer temporal-FGS scalability framework. The application of the proposed FGS temporal-SNR scalability to Internet video streaming is described in Section IV. Subsequently, a low-complexity rate-control is introduced that is able to perform the tradeoff between SNR and motion-smoothness in real-time, depending on the bit-rate availability of the various clients. Section V presents the conclusions.

II. FGS SNR-Scalability

FGS has recently been introduced [8], [9] to compensate for the unpredictability and variability in bandwidth between sender and receiver(s) over the Internet. FGS has also been adopted by MPEG-4 as the video-coding tool for streaming applications [10]. The scalability structure of the FGS method is portrayed in Fig. 1(a). In addition to the base layer, which is coded with an MPEG-4 compliant nonscalable coder, FGS consists of a single enhancement layer coded in a progressive (fine granular) manner. Under this framework, the scalable video content can be compressed over any desired bit-rate range $[R_{\min}, R_{\max}]$.[1]

The base layer is coded with a bit-rate R_{BL}, chosen so that the available bandwidth (over the time-varying network) is higher than R_{BL} at all times ($R_{BL} \leq R_{\min}$). Subsequently, the enhancement layer is over-coded at encoding time using a bit-rate $(R_{\max} - R_{BL})$, as portrayed in Fig. 1(a). The enhancement layer can be coded progressively (bit-plane by bit-plane[2]) by employing any embedded compression technique [12] (e.g., wavelet or embedded DCT coding scheme [13]). In the remainder of this paper, the effective and low-complexity bit-plane embedded-DCT algorithm adopted by the MPEG-4 standard has been employed for the FGS enhancement-layer coding [10], [13]. As can be seen from Fig. 1(a), the enhancement-layer frames are intra-coded, but the coding efficiency from temporal redundancy exploitation is partially retained because the MPEG-4 motion-compensated (MC) scheme is employed at the base layer. The block-diagram of the FGS encoder is portrayed in Fig. 2, clearly illustrating the small additional computational complexity introduced by the FGS coding of the enhancement layer. While the computational complexity is low, the memory requirements are more significant, since one additional frame memory is necessary in the enhancement layer for the bit-plane coding and the memory bandwidth increases since the data is scanned progressively (i.e., bit-plane by bit-plane).

At the streaming server, the enhancement layer improves upon the base-layer video, fully utilizing the bandwidth R

[1] $R_{\min}$ and $R_{\max}$ are the minimum and maximum bandwidth, respectively, available over the network at all times.

[2] In a progressive coder, the more significant bit-planes are transmitted prior to the less significant bit-planes.

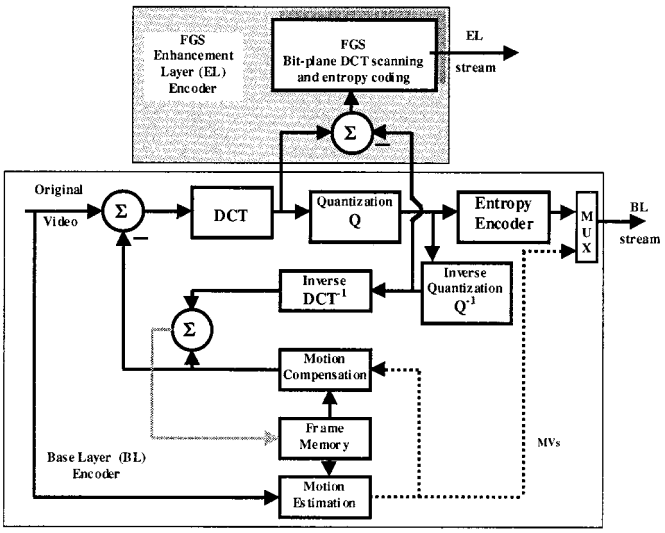


Fig. 2. Block-diagram of the SNR-FGS encoder for the base and enhancement layers.

available at transmission-time [see Fig. 1(b)]. The FGS scalability structure allows for resilient Internet video transmission [8], since the base-layer video can be reliably delivered using re-transmission or other packet-loss recovery methods [8], [11], while the enhancement layer can be left unprotected since the packet losses do not propagate. Then, at the decoder side, the base layer and the received portion of the enhancement-layer data are decompressed. Furthermore, the FGS enhancement-layer decoder is complexity-scalable, since the decompression process can be stopped whenever the processing power of the decoder is exceeded. This characteristic of FGS is very important, since the same content is accessed by various receivers (e.g., set-top-boxes, PCs, wireless telephones, etc.) with different computing power, memory, display resolutions etc. A more detailed description of the FGS technique can be found in [8] and [10].

It is interesting to note that the range $[R_{\min}, R_{\max}]$ can be determined off-line (e.g., for a particular set of Internet access technologies). For unicast streaming, an estimate for the available bandwidth R can be generated in real-time for a particular session. Based on this estimate, the server transmits the enhancement layer using a bit-rate $R_{EL} = \min(R_{\max} - R_{BL}, R - R_{BL})$. Due to the fine granularity of the enhancement layer, simultaneous real-time rate control on multiple streams can be implemented with minimal processing. This FGS property is very important since in the unicast (on-demand) case, the streaming video server may be required to serve thousands of clients simultaneously, and thus, only limited processing can be performed at the server side to prevent overloading.

For multicast streaming, a set of intermediate bit-rates, $R_1, R_2, \ldots, R_N$, can be used to partition the enhancement layer into sub-streams. In this case, N fine-granular streams are multicasted using the bit-rates: $R_{e1} = R_1 - R_{BL}, \ldots, R_{eN} = R_N - R_{N-1}$, with $R_{BL} < R_1 < \cdots < R_N \leq R_{\max}$.

Therefore, the fine granularity provided by FGS gives the server total flexibility in adapting to the network condition in both unicast and multicast scenarios without significant complexity [8]. Furthermore, when compared with other scalable approaches, FGS provides a good balance between coding-efficiency and scalability [14].

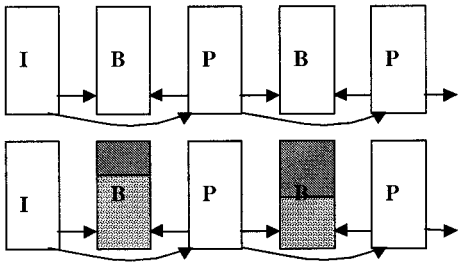


Fig. 3. Top: standard MPEG-4 nonscalable codec. Bottom: alternative codec using FGS for the coding of the B-frames texture.

III. Hybrid Temporal-SNR Scalability

A. Fine-Granular Coding of B-Frames

In this section, a novel scalable technique for coding B-frames is introduced that is based on the FGS bit-plane compression scheme previously described. The fine-granular coding of B-frames is interesting because it allows the partial decoding of B-frames depending on for example, the received bit-rate, or the available computational resources. This feature is especially beneficial for devices with limited computational resources, where the full decoding of the computationally expensive B-frames cannot always be guaranteed. A disadvantage of nonscalable B-frames is that partial decoding will lead to a very poor image quality because only a portion of the image is enhanced. Moreover, as will be demonstrated in this section, there is no performance penalty associated with the increased flexibility provided by the progressive coding of B-frames.

From the FGS encoder block-diagram depicted in Fig. 2, it becomes clear that the FGS residual signal (FGSR) and the motion-compensation residual signal (MCR) result from two different processes: quantization and MC prediction, respectively

$$\text{FGSR}(i) = F(i) - F_q(i) \quad \text{and} \quad \text{MCR}(i) = F(i) - F_q^{MC}(i-1),$$

where

$F(i)$	current frame to be coded;
$F_q(i)$	current frame reconstructed at the decoder side, after quantization and dequantization;
$F_q^{MC}(i-1)$	MC prediction based on the previous reconstructed frame at the decoder side.

Despite their different provenience, the FGS and the MCR signals have very similar statistical properties [12], [15], indicating that coding methods that prove to be efficient for one of the residual signals will also lead to good rate-distortion performances when applied to the other signal. In [12] and [14], it was shown that there is no performance penalty associated with the fine-granular compression of the SNR residual signal (FGSR) when compared with the standard quantization and variable-length coding methods employed in the MPEG-4 multilayer SNR scalability schemes. Consequently, FGS is a good candidate for the compression of the MCRs.

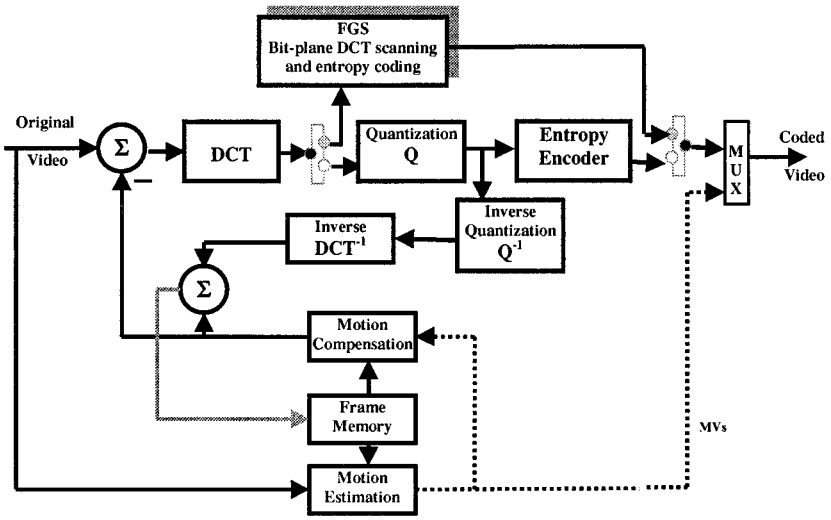


Fig. 4. Block-diagram of the novel nonscalable encoder with fine-granular B-frames.

To validate this assumption, the performance of the nonscalable MPEG-4 codec has been compared to that of a modified MPEG-4 coder where the B-frames were coded using FGS[3] (i.e., the scenarios depicted in Fig. 3). For the experiments, the TM-5 rate control has been used for the standard MPEG-4 nonscalable codec, while the alternative coder allocates the same number of bits to each frame for a fair comparison. The encoder of the alternative codec is shown in Fig. 4.

For the fine-granular coding of the B-frames, two different algorithms have been used: one based on the FGS scheme adopted in MPEG-4 [13], the other based on a novel algorithm, which replaces Huffman with an adaptive arithmetic coding (AC) without multiple contexts [20]. The comparison has been performed for the 100 kb/s–1 Mb/s range and it revealed a minor decrease in PSNR at lower bit-rates (0.1 dB) and a moderate increase in PSNR at higher bit-rates (0.5 dB), when Huffman coding has been used. However, if an adaptive arithmetic coding without multiple contexts [20] is employed for the entropy coding, the alternative coder always has the best performance.

Several results are given in Table I for the MPEG-4 video test sequence *Foreman* at CIF-resolution and 10 Hz, with a GOP of $N = 24$ frames and $M = 3$. These results indicate that the flexibility associated with the progressive coding of B-frames does not result in an image quality penalty. Another important conclusion that can be drawn from Table I is that as the bit-rate increases, the performance gain of the proposed FGS based codec compared with the nonscalable codec also increases. This is because at high bit-rates, the reference I- and P-frames are coded with a better quality that leads to a better temporal decorrelation for the B-frames. Hence, at high bit-rates, the MCR of the B-frames contains mostly high frequencies that are coded less efficiently by the standard MPEG-4 run-amplitude entropy codec than the FGS entropy coder.

[3]Here, FGS has been used only for the embedded coding of the B-frames texture. Motion-vectors still need to be computed and transmitted with the conventional MPEG-4 strategy, with the only difference being that they are now clustered in the beginning of the B-frames, leading to an inherent data-partitioning and consequently an improved error resilience.

TABLE I
RATE-DISTORTION PERFORMANCE OF THE VARIOUS B-FRAME IMPLEMENTATIONS

R (kb/s)	PSNR (Y/U/V) MPEG-4 Standard Codec	PSNR (Y/U/V) FGS coding of B-frames with MPEG-4 FGS-coding	PSNR (Y/U/V) FGS coding of B-frames with AC
100	29.20, 35.36, 35.97	29.09, 35.60, 35.83	29.34, 35.71, 36.00
400	35.62, 40.76, 42.02	35.52, 40.74, 41.96	35.62, 40.76, 42.00
1000	39.47, 43.51, 44.95	40.08, 43.84, 45.28	40.15, 43.88, 45.31

Consequently, coding schemes employing only I- and P-frames in the base layer (e.g., like in the H.261 standard or the Simple Profile in MPEG-4) can use fine granular coded B-frames to enhance their flexibility, coding performance and error resilience, while having no rate-distortion penalty compared with the conventional IPB-compression schemes.

Nevertheless, it is important to mention that even though there is no coding penalty associated with the FGS coding of B-frames, an encoder like the one depicted in Fig. 4 is more complex than a conventional nonscalable codec. The increased complexity is due to the different codecs used for the texture compression of I-/P- and B-frames, respectively. Hence, the architecture proposed in Fig. 4 becomes interesting only in a scalable coder, where FGS is used for the compression of the SNR residual signal. In this case, the FGS codec can be employed for the compression of both the motion-compensation and SNR residuals.

B. Introduction to Fine-Granular Temporal Scalability (FGST)

As mentioned earlier, a limitation of the current FGS implementation is that the frame rate is "locked" to the original base-layer frame rate, independent of the available bandwidth. In H.263, MPEG-2, and MPEG-4, the temporal scalability tool enables frame-rate variations of a video sequence by coding a base layer at a frame rate f_B frames per second (fps) and

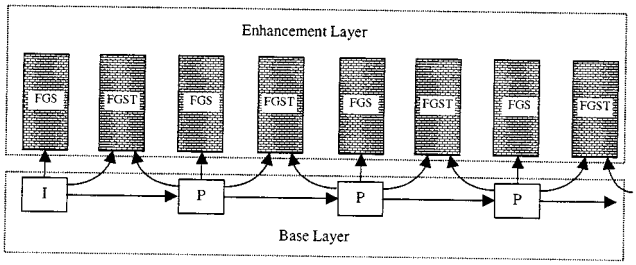


Fig. 5. Example of all-FGS hybrid temporal-SNR scalability.

coding additional frames at the enhancement layer with f_E. The video sequence can then be played depending on the available bandwidth and decoder capabilities at either a frame rate f_B or $f_B + f_E$. To enable optimal viewing conditions, depending on the available bandwidth or user preferences, the combination of FGS and temporal scalability is crucial for Internet video transmission. Furthermore, the experiments presented in Section III-A reveal that there is no penalty associated with the FGS coding of MC residuals. Therefore, we propose a single-layer FGS hybrid temporal-SNR scalability, which extends the flexibility specific to FGS to the hybrid temporal-SNR scalability scheme. The proposed hybrid temporal-SNR scalability provides total flexibility in supporting:

1) SNR scalability while maintaining the same frame rate;
2) temporal scalability by increasing only the frame rate;
3) both SNR and temporal scalabilities.

Fig. 5 shows the proposed hybrid scalability structure. In addition to the standard SNR FGS frames, this hybrid structure includes MC residual frames in the enhancement layer. We refer to these MC frames as the FGST pictures. As shown in the figure, each FGST picture is predicted from base-layer frames that do not coincide temporally with that FGST picture, and therefore, this leads to the desired temporal scalability feature. Moreover, the FGST residual signal is coded using the same fine-granular video coding method employed for compressing the standard SNR FGS frames.

Consequently, each FGST picture includes two types of information: 1) motion vectors (MVs), which are computed with respect to the temporally adjacent base-layer frames and 2) texture data, representing the fine-granular (i.e., bitplane-DCT) coded MC FGST residual. These two sets of information of an FGST picture are coded and transmitted using a data-partitioning strategy. Unlike the base layer, where the MVs and corresponding texture data of each macroblock are sent macroblock-by-macroblock, for FGST, all motion vectors are clustered and transmitted first. Subsequently, the coded representation of the DCT bitplanes' residual signal is sent. This data partitioning strategy provides a useful packet-loss resilience tool by enabling the transmission of the MV data in designated packets (i.e., separate from the residual-DCT signal packets of both SNR and temporal FGS frames). These MV-designated packets can then be provided with a higher level of protection than other packets, thereby reducing the negative impact of packet losses on the MC FGST frames.

Fig. 6(a) shows a functional architecture for the hybrid temporal-SNR FGS encoder. It is important to note that although the SNR FGS residual can be computed directly in the DCT domain, the FGST residual is computed in the pixel domain because the motion compensation takes place in the pixel-domain. Therefore, the FGST residual frames have to be DCT transformed prior to their bitplane-based entropy coding. In addition, and as mentioned above, the FGST residual is computed based on a motion-compensation approach from base-layer pictures.

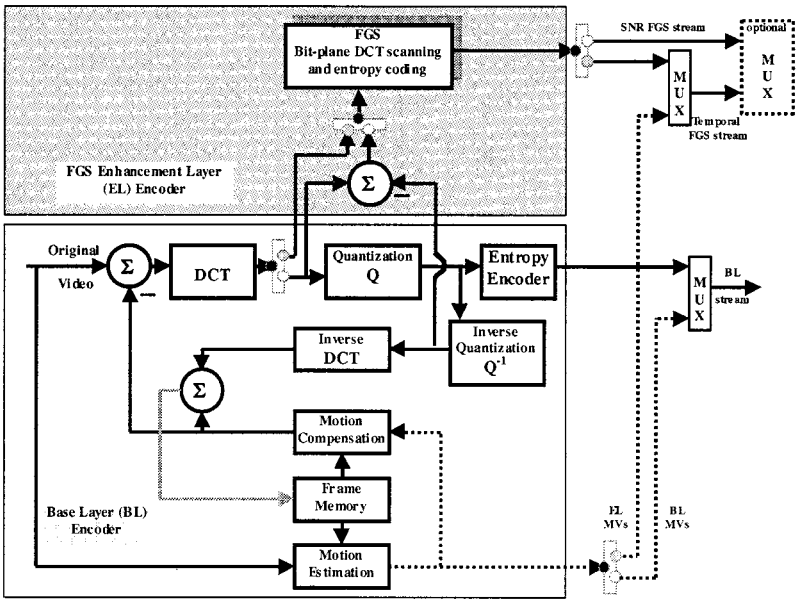
However, the additional computations needed for FGST frames' coding can be reduced to a minimum if the encoder and respectively decoder architectures allow the reusability of existing blocks. As shown in Fig. 6(a), the DCT, motion estimation, motion compensation, and frame memory[4] functional blocks from the base-layer encoder can be reused for computing the FGST DCT residual signal. This can be achieved through a novel (yet simple) data-flow control that takes advantage of the fact that the encoder never compresses a base-layer frame and an FGST frame at the same instance. Therefore, the functional blocks available for the encoding of the base layer can be reused for FGST compression. Similarly, the SNR FGS entropy-encoder can be shared between the SNR FGS and FGST frames, since both of these picture types are never compressed at the same instance of time.

As depicted in Fig. 6(a), the motion estimator outputs two sets of motion vectors: one set for the base-layer pictures and the other for the FGST frames. The MVs associated with FGST frames are multiplexed with the enhancement-layer bitstream using the data-partitioning strategy explained above. Moreover, the two FGS enhancement-layer streams can be either multiplexed to generate a single stream (which consists of both SNR and temporal FGS pictures) *or* stored/transmitted in two separate streams.

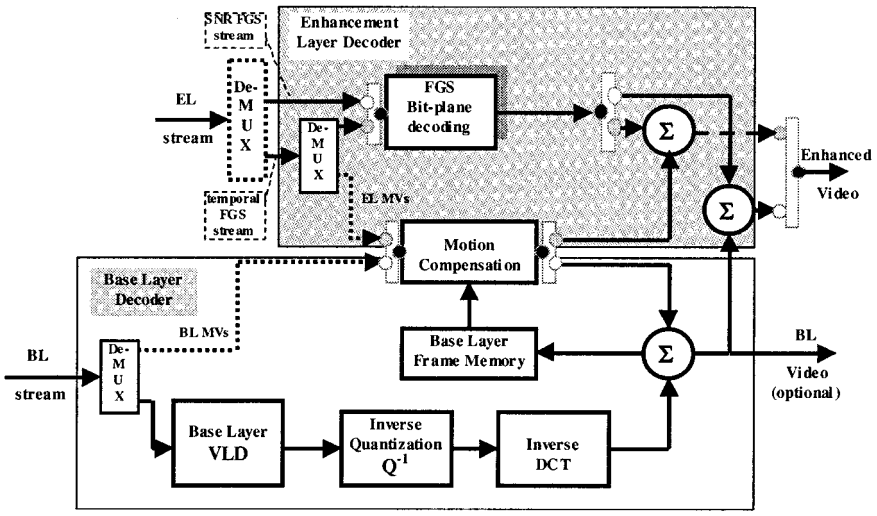
Fig. 6(b) illustrates the corresponding functional architecture for the hybrid temporal-SNR FGS decoder. Similar to the encoder architecture described above, the decoding of the FGST frames can be realized with minimal complexity overhead. This is accomplished by sharing the motion-compensation functional block with the base layer and sharing the standard SNR FGS decoding path. As shown in Fig. 6(b), the FGST compressed stream is de-multiplexed to separate the MV's data from the coded residual information. The FGST MVs are used by the motion-compensation block to compute the FGST predicted frame, while the compressed residual information is decoded and inversely transformed by the enhancement-layer decoder. The two signals are added together to generate the FGST frame which can be sent directly to the display device. For the SNR FGS compressed frames, the decoded signal has to be added to the corresponding base-layer frames before the display operation.

To limit the amount of memory required for the implementation of a codec supporting the FGST functionality, the enhancement-layer frames cannot be used as a reference for MC prediction, i.e., the FGST-frames are only predicted from base-layer frames. Hence, two reference frames memories are necessary for the implementation of the proposed FGST scheme. These

[4]Depending on whether the base layer allows the encoding/decoding of B-frames, a different number of frame-memories is necessary for the base-layer compression. If B-frames are never used in the base layer, an additional frame memory is necessary for the FGST compression, otherwise the existent frame-memories can be reused.



(a)



(b)

Fig. 6. Architecture of the all-FGS hybrid temporal-SNR scalability: (a) encoder and (b) decoder.

two frame memories can be shared by the motion-compensation unit of FGST and the base-layer B-frames.

C. Comparison with Multilayer FGS-Temporal Scalability

An alternative to the proposed FGS temporal-SNR scalability would be to separate the SNR and temporal-scalability layers, as depicted in Fig. 7. The FGS layer is coded then on top of both the base and temporal enhancement layers and therefore enhances the SNR quality of all frames. For simplicity, this alternative implementation is referred to in the remainder of the paper as multilayer FGS-temporal scalability, since one base-layer and two enhancement layers are employed for its realization.

In the multilayer FGS-temporal scalability implementation illustrated in Fig. 7, the bit-rate of the temporal scalability layer is predetermined at encoding time. Since the temporal enhancement layer is not fine-granular, it needs to be entirely decoded in order to improve the temporal resolution of the decoded sequence, requiring a discrete bit-rate of $R_{BL} + R_{EL}^{T}$. Another disadvantage of this solution resides in its increased implementation complexity, since two residuals need to be computed for the temporal-frames (MC and FGS-residuals) and two algorithms need to be employed for the decoding of the enhancement-layer texture (i.e., the traditional nonscalable decoding of

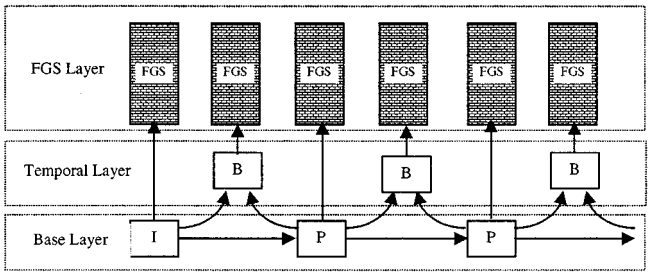


Fig. 7. Multilayer FGS-temporal scalability structure.

TABLE II
RATE-DISTORTION PERFORMANCE OF THE MULTILAYER FGS-TEMPORAL SCALABILITY AND FGS TEMPORAL-SNR SCALABILITY

R_{BL} (kb/s)	R_{EL}^{T} (kb/s)	R_{EL}^{FGS} (kb/s)	SNR (Y/U/V) Multi-layer FGS-temporal (Figure 7)	SNR (Y/U/V) All-FGS temporal-SNR (Figure 5)
100	40	0	30.57,37.07,37.68	30.54,37.02,37.59
100	40	100	32.94,38.28,39.27	32.95,38.28,39.27
100	40	200	34.73,39.57,40.73	34.74,39.57,40.74
100	40	300	36.01,40.41,41.40	36.02,40.41,41.41
100	40	400	37.54,41.32,42.62	37.55,41.32,42.63
100	40	500	38.75,42.29,43.60	38.75,42.29,43.61
100	40	600	39.53,42.99,43.99	39.54,43.00,44.00
100	40	700	40.44,43.67,44.55	40.45,43.67,44.57
100	40	800	41.51,44.30,45.24	41.53,44.31,46.26

the frames in the temporal layer followed by the FGS decoding of the SNR-residual).

For the implementation of the fine granularity, the embedded DCT bit-plane coding method currently adopted for MPEG-4 FGS [14] has been employed. In order to keep the implementation of the all-FGS hybrid temporal-SNR scalability at low complexity, the same VLC tables have been used for both FGS and FGST frames.[5]

The two hybrid FGS-temporal scalability structures have been implemented and their results are presented in Table II for the sequence *Foreman*. The experiments have been performed for a frame rate $f_{BL} = f_{EL} = 5$ fps. The base layer contains GOPs with only I and P-frames $(M = 1)$, which last for 2.4 s $(N = 12)$, and employs TM-5 for rate control. For the coding of the temporal layer B-frames in the multilayer implementation, a fixed $QP = 28$ has been employed. To provide a fair comparison, the temporal/FGS (FGST) frames in the all-FGS implementation have been decoded with the same amount of bits as employed for the B-frames in the temporal scalability layer of the multilayer scalability implementation. This bit-rate adjustment is easily performed due to the embedded-stream property of FGS. In Table II, R_{BL} represents the base-layer rate, R_{EL}^{T} represents the temporal-layer rate for the multilayer implementation, and R_{EL}^{FGS} is the SNR FGS layer bit-rate. As can be seen from Table II, the rate-distortion performance of the previously two described implementations of hybrid SNR/temporal scalability is very similar. In other words, there is no penalty associated with the proposed single-layer scalability solution.

IV. INTERNET VIDEO STREAMING AND RATE-CONTROL FOR HYBRID TEMPORAL-SNR SCALABILITY

A. Internet Video Streaming using FGS Hybrid Temporal-SNR Scalability

The structure of the FGS hybrid temporal-SNR scalability is very flexible and permits tradeoffs to be easily performed between improving the SNR and motion smoothness. In general, the proposed scalability structure supports variable frame-rate scenarios where the enhancement-layer frame rates may vary

[5]Alternatively, Huffman-coding can be replaced with adaptive arithmetic coding without multiple contexts, leading to a coding gain ranging between 0.1 and 0.7 dB.

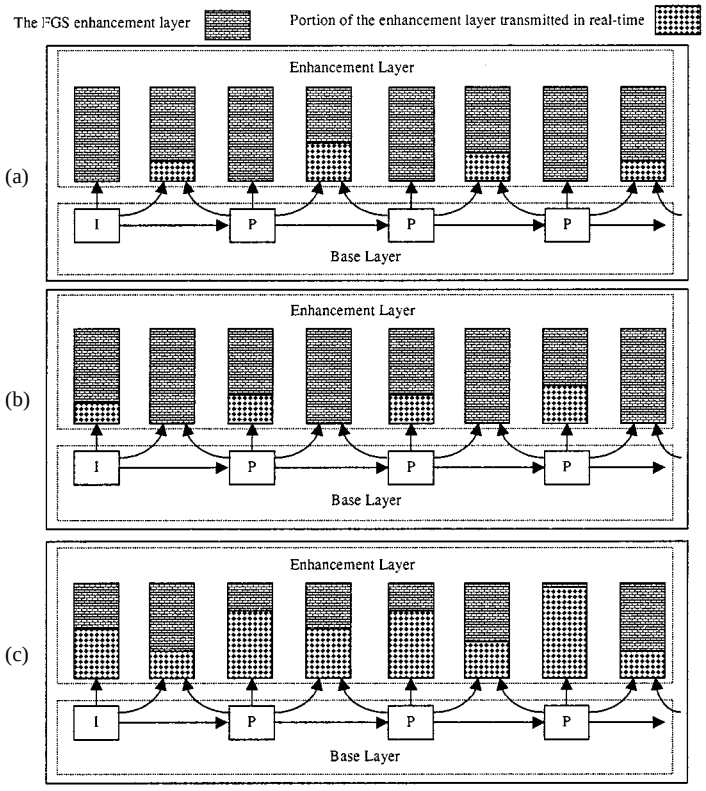


Fig. 8. Examples of using the new scalability structure in supporting temporal, SNR, and joint temporal-SNR scalability in a fine-granular way.

with time. Furthermore, it is important to note that while the generated compressed stream has a total frame rate f_T, the transmitted stream could have a different frame rate f_t. A similar observation is true for the SNR quality improvement: at transmission time, it can be decided to what extent to improve the image quality (SNR) of the individual frames depending on the available bandwidth.

For example, depending on the image content and available bandwidth, the server can decide to enhance only the image quality of the base layer at frame rate f_{BL} by sending first the FGS residual frames corresponding to the base-layer frames [Fig. 8(a)]. In this case, $f_t = f_{BL}$. An alternative is to enhance the motion-smoothness by sending the temporal enhancement layer (the FGST-frames) prior to any FGS enhancement layer [Fig. 8(b)]. The sequence can then be played at frame rate $f_t = f_T = f_{BL} + f_{EL}$. Then, if there is still available bandwidth, the SNR quality of all I-, P-, and B-frames of the base layer and temporal enhancement layers could be enhanced by transmitting their FGS enhancement layers [Fig. 8(c)]. Moreover, variations between the previously mentioned alternatives are also possible: SNR improvement of base layer at bit-rate $R1$ followed by temporal resolution improvement at bit-rate $R2$ $(R2 > R1)$ followed by SNR improvement of both base and temporal enhancement layer at bit-rate $R3$ $(R3 > R2)$. It is important to notice that this FGS hybrid temporal-SNR scheme permits tradeoffs between temporal-resolution and SNR improvements *at transmission* time, depending on the available bandwidth R and possible user preferences, and *not at encoding time*.

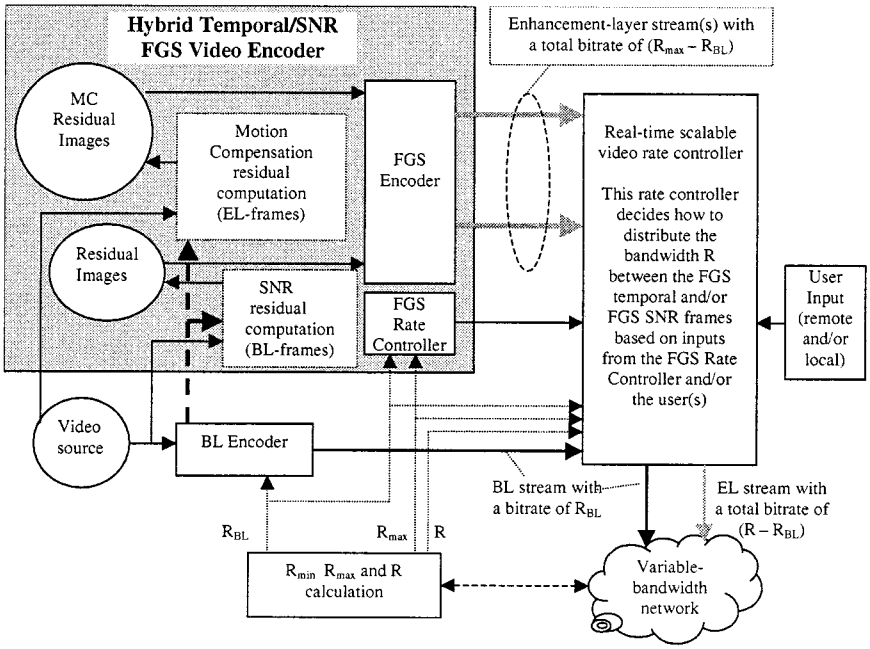


Fig. 9. Internet video streaming system using the proposed hybrid FGS-temporal scalability structure.

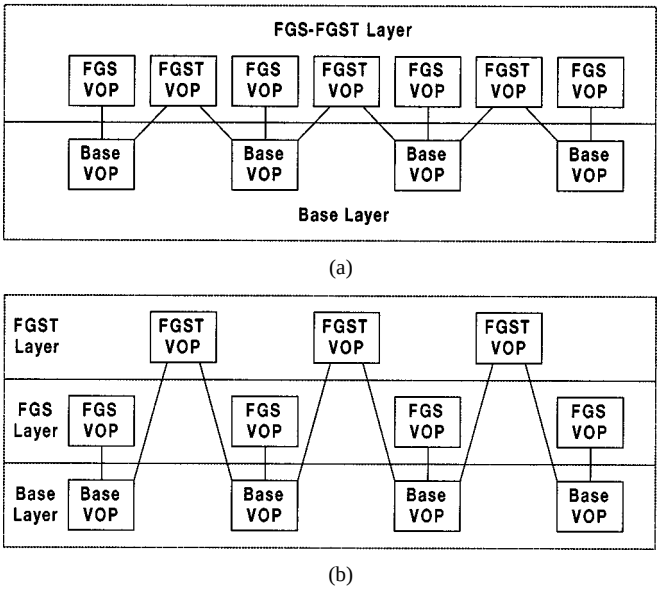


Fig. 10. Two alternatives of streaming the FGS temporal-SNR enhancement layers. (a) Single FGS stream. (b) Two FGS streams.

In Fig. 9, an example of an Internet video streaming system employing the proposed hybrid scalability structure is portrayed. The compressed FGS/FGST stream(s) can be stored or transmitted in real-time. In both scenarios, a rate controller can be employed to determine at transmission-time the bit-rates (i.e., $R_{EL} = R_{EL}^{\mathrm{FGST}} + R_{EL}^{\mathrm{FGS}} = R - R_{BL}$) that should be allocated for transmitting the hybrid FGST and/or SNR frames depending, among other things, on inputs from the FGS Rate Controller (see Section IV-B) and/or the user(s).

Furthermore, it is important to mention that the FGS and FGST streams can either be combined to generate a single FGS stream [see Fig. 10(a)] or separated as two (temporal and SNR) FGS streams [see Fig. 10(b)].[6] By coding the FGS and FGST streams in separate enhancement layers, the tradeoff between SNR and temporal improvements can easily be performed at the video object layer (VOL)[7] level by assigning different priorities to the two enhancement-layer streams. This becomes more difficult in the case where both FGS and FGST streams are coded in the same layer, since the header of each frame[8] (VOP) needs to be decoded to determine its type (i.e., an FGS-VOP or FGST-VOP). However, the advantage of having only one FGS-FGST enhancement layer is that only one stream needs to be transmitted, thereby reducing the system overhead (e.g., the management of multiple streams).

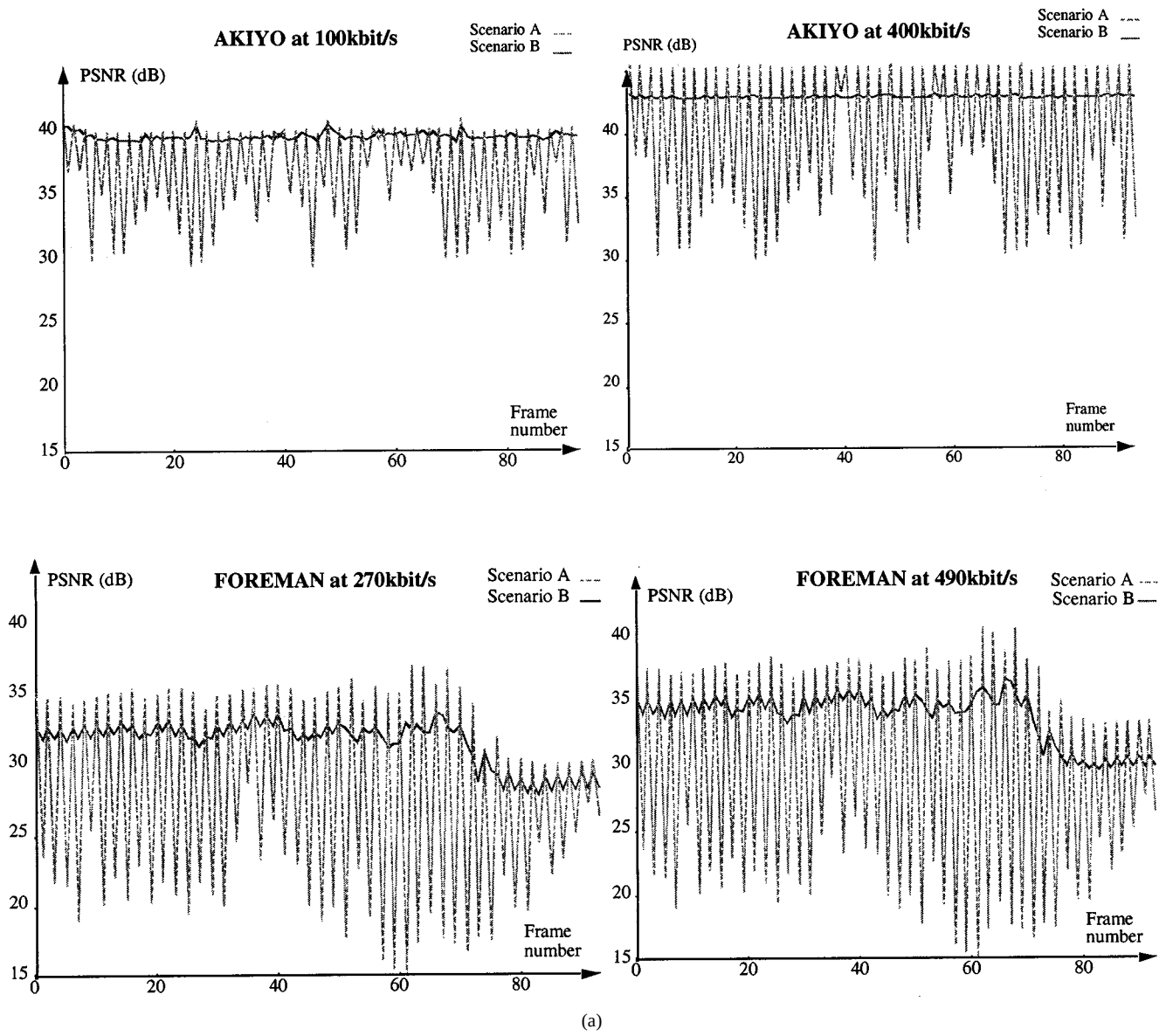
B. Temporal-SNR Tradeoffs for Improved Visual Quality

In order to perform the temporal/SNR rate-allocation in either real-time or off-line, a mechanism needs to be identified to determine the optimal visual quality. In [16], an efficient rate-control for temporal scalability has been proposed that is based on three parameters: 1) bit-rate adherence; 2) motion; and 3) frame separation. While this method proves to be very efficient in determining which frames should be transmitted in the temporal layer of an H.263–compliant codec for optimal perception, it does not provide any mechanism for performing the temporal-SNR tradeoffs. Such a mechanism is presented in [17], where a real-time frame-rate control for H.263+ video coding is proposed. Nevertheless, although the real-time rate-control presented in [17] is not very costly when applied to a single stream,

[6]Both methods are currently supported by the MPEG-4 standard.

[7]VOL and VOP are the MPEG-4 terminology for a video object layer and video object plane, respectively.

[8]In the current MPEG-4 FGS profile, each frame contains only one object, i.e., one VOP.



(a)

Fig. 11. (a) Performance of the FGS temporal-SNR scalability for the extreme scenarios described in Section IV-B, where no tradeoffs were made between SNR and motion-smoothness.

its complexity grows considerably if performed simultaneously on a large number of unicast streams. Moreover, in the H.263+ case, the tradeoff between the SNR and temporal enhancement layer cannot easily be made on a frame-by-frame basis, since the number of bits per enhancement-layer frame is different. Thus, a large number of frames (e.g., a GOP) needs to be employed for a robust rate-allocation mechanism that guarantees a fixed overall bit-rate. However, in the FGS temporal-SNR scalability proposed in this paper, performing such a tradeoff becomes easier due to the fine-granularity of the enhancement layer. Therefore, a simple and novel algorithm is introduced for performing the temporal-SNR tradeoff on multiple streams with low complexity.

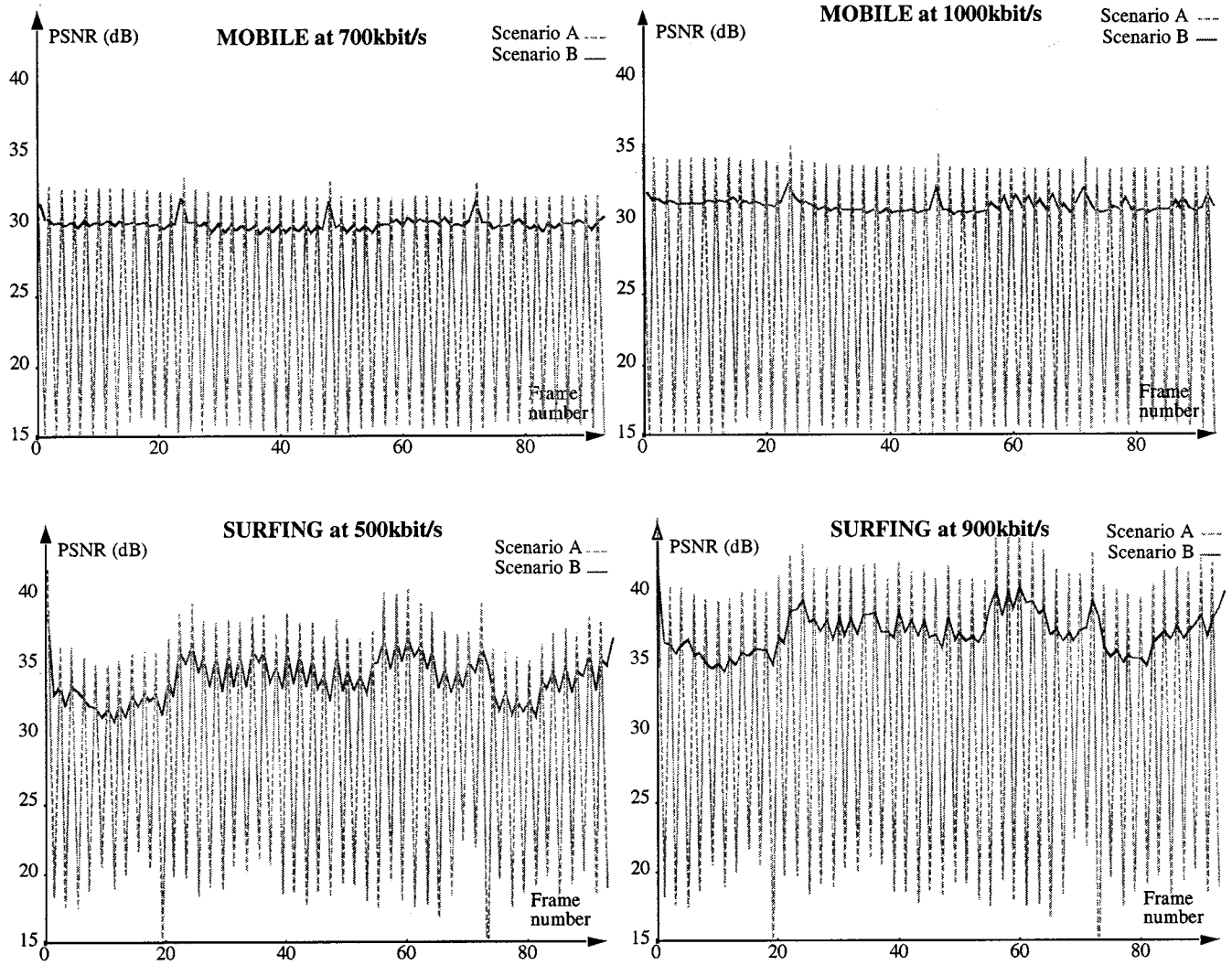
For the original FGS coder (see Fig. 1), a very simple rate control for the enhancement-layer frames showed good performance results: the available enhancement-layer bandwidth is distributed evenly between the FGS (i.e., SNR) frames. Thus

$$F_{e1} = F_{e2} = \cdots = F_{en} = \frac{R_{\text{enh}}}{f_{BL}} = \frac{R - R_{BL}}{f_{BL}}$$

where

- F_{ei} number of bits for the *transmitted* enhancement-frame i;
- R_{enh} bit-rate available for the enhancement layer at *transmission-time*;
- f_{BL} base layer and FGS (SNR) enhancement-layer frame rate;
- R total bandwidth available at transmission time;
- R_{BL} base-layer bit-rate.

Consequently, since each base-layer frame is enhanced by the same number of bits, the FGS quality "follows" the quality of the base layer. Visual evaluation of the FGS streams coded with this simple rate-control revealed a good performance [14]. Hence, a similar rate-control strategy could be adopted for the FGS temporal-SNR scalability introduced in this paper. In this case, the available bandwidth is split evenly between all temporal (FGST) and SNR (FGS) enhancement frames (see Fig. 8). If a single FGST frame is inserted between all base-layer frames



(b)

Fig. 11. *(Continued.)* (b) Performance of the FGS temporal-SNR scalability for the extreme scenarios described in Section IV-B, where no tradeoffs were made between SNR and motion smoothness.

(i.e., the odd frames are FGS frames, the even are FGST frames), the size of each enhancement-layer frame equals

$$F_{e1}^{\mathrm{FGS}} = F_{e2}^{\mathrm{FGST}} = F_{e3}^{\mathrm{FGS}} = F_{e4}^{\mathrm{FGST}} = \ldots = \frac{R_{\mathrm{enh}}}{f_t} = \frac{R - R_{BL}}{f_{BL} + f_{EL}}$$

where

- f_t total frame rate at *transmission-time*;
- f_{BL} base-layer and FGS (SNR) enhancement-layer frame rate;
- f_{EL} FGST enhancement-layer frame rate.

To evaluate whether this rate-control strategy gives a good tradeoff between motion smoothness and individual image quality, the PSNR performance of an entire set of sequences coded at different bit-rates has been analyzed under two extreme scenarios.

1) *Scenario A (Best Case for Individual Image Quality):* the enhancement bit-rate R_{enh} available at transmission-time is used solely for PSNR improvement (i.e., for sending only FGS residual frames at the frame rate f_{BL}). The number of bits for each FGS frame equals $F_{eA}^{\mathrm{FGS}} = R_{\mathrm{enh}}/f_t = (R - R_{BL})/f_{BL}$ and $F_{eA}^{\mathrm{FGST}} = 0$.
2) *Scenario B (Best Case for Motion Smoothness):* the enhancement bit-rate R_{enh} available at transmission time is used for sending all FGS and FGST frames, at the overall frame rate of $f_{BL}+f_{EL}$. The number of bits for each FGS and FGST frame, respectively, equals $F_{eB}^{\mathrm{FGS}} = F_{eB}^{\mathrm{FGST}} = R_{\mathrm{enh}}/f_t = (R - R_{BL})/(f_{BL} + f_{EL})$.

In our experiments, we considered $f_{BL} = f_{EL} = 5$ fps (i.e., one bi-directional predicted FGST frame is coded between two base-layer frames). The base layer is coded in both scenarios with the same bit-rate R_{BL} and contains only I- and P-frames (i.e., $M = 1$) and a GOP-size of 2.4 s (i.e., $N = 12$). For a fair comparison, the enhancement-layer rate is kept the same for both scenarios, and thus $F_{eB}^{\mathrm{FGS}} = F_{eB}^{\mathrm{FGST}} = R_{\mathrm{enh}}/2f_{BL} = F_{eA}^{\mathrm{FGS}}/2$. The sequences[9] adopted for the analysis contain various degrees of motion and textures and are representative of

[9]The sequences employed in our evaluation are well-known MPEG-4 test sequences (*Foreman*, *Akiyo*, and *Mobile*), with one exception, the *Surfing* sequence, which was chosen due to its fast-motion characteristics.

TABLE III
CHARACTERIZATION OF THE SEQUENCES EMPLOYED FOR THE RATE-CONTROL EVALUATION

Sequence	Motion-characteristics	Texture-characteristics
Akiyo	Static background, talking-head	Easy texture
Foreman	First-part of the sequence contains high-motion, the second part is almost static	Beginning of the sequence has a relatively easy texture, the second part has a relatively detailed-texture
Mobile	Slow, constant movement throughout the sequence	Very detailed texture
Surfing	Very high-motion throughout the sequence, lots of scene-changes	Relatively detailed, random texture

the content streamed over the Internet. Their characteristics are summarized in Table III and their performances at various bit rates are illustrated in Fig. 11.

In scenario A, the FGST frames are not transmitted, and thus each FGS frame [i.e., FGS(i)] is displayed twice[10] to produce a 10-Hz sequence. Thus, the PSNR value for the Not-Transmitted (NT) frame, denoted as PSNR$_A$(FGS$^{\mathrm{NT}}(i+1)$), is computed based on the previously decoded FGS-frame i. If the scene contains a large degree of motion, the difference between adjacent frames at 10 Hz is relatively high, and thus, PSNR values for the sent frames [i.e., PSNR$_A$(FGS(i))] will be considerably higher than for the not-transmitted frames [i.e., PSNR$_A$(FGS$^{\mathrm{NT}}(i+1)$) $\ll$ PSNR$_A$(FGS(i))]. This observation can be easily verified by evaluating the performance plots of the *Surfing* sequence, which has a high degree of motion. Alternatively, in the *Akiyo* sequence, where only low motion-activity exists, the PSNR swing between the transmitted and not-transmitted frames is considerably lower. Furthermore, it is important to notice that the PSNR swing in Scenario A increases with the bit rate, thereby indicating that the motion jerkiness becomes more visually disturbing as the individual image quality (SNR) improves.

In scenario B, all FGS and FGST frames are transmitted, and the PSNR of each FGS frame i and FGST frame $i+1$ is computed as PSNR$_B$(FGS(i)) and PSNR$_B$(FGST$(i+1)$), respectively. In this scenario, the motion portrayal of the sequences improves as indicated by the relatively small variations between the PSNR values of the various frames. However, the PSNR quality of the sequence decreases for scenario B when compared to scenario A, since only half the number of bits is allocated to each frame under this rate-allocation strategy. Depending on the transmission bit-rate and the sequence characteristics, the loss in individual image quality between the two scenarios can vary between less than 1dB up to several decibels for sequences with difficult to code textures at relatively high-transmission bit-rates (e.g., *Mobile* sequence).

From the performance plots portrayed in Fig. 11 and the visual evaluation of the resulting sequences, the following conclusions can be drawn.

1) The difference between PSNR$_A$(FGS(i)) and PSNR$_B$(FGS(i)), denoted ΔPSNR$^T(i)$, is a good quantification of the improvement in PSNR which can be obtained by applying scenario A (i.e., SNR enhancement only).
2) The difference between PSNR$_A$(FGS(i)) and PSNR$_A$(FGS$^{\mathrm{NT}}(i+1)$), denoted ΔPSNR$^{\mathrm{NT}}(i+1)$, is a good measure for the motion activity within (part of) a sequence: a large difference corresponds to a high degree of motion activity. For example, the ΔPSNR$^{\mathrm{NT}}$ for the *Akiyo* sequence, which is characterized by slow-motion, is considerably lower than the ΔPSNR$^{\mathrm{NT}}$ for the *Surfing* sequence, which has fast-motion. However, ΔPSNR$^{\mathrm{NT}}$ is also dependent on the level of detail of the MC residual signal texture (e.g., the ΔPSNR$^{\mathrm{NT}}$ of the *Mobile* sequence is relatively large despite its slow motion, due to the texture richness).
3) If the individual image quality of (part of) a sequence is very high, like in the *Akiyo* sequence at $R = 400$ kbits/s, there is no further need for individual image quality improvement, and thus all the available bandwidth should be spent for eliminating the motion jerkiness by coding FGST frames.
4) If the individual image quality of a (part of a) sequence is low, like in the *Mobile* sequence at $R = 700$ kbits/s, improving the overall image quality has the first priority and thus all the available bandwidth should be spent on the FGS frames coding.

In summary, if a sequence is characterized by high-motion (i.e., ΔPSNR$^{\mathrm{NT}} \gg \Delta$PSNR^T) or the overall image quality of the FGST frames is already very high, scenario B (i.e., transmission of both FGS and FGST frames) should be employed to obtain a good motion portrayal. Alternatively, if there is only slow motion within the sequence or the overall image quality is poor, scenario A (i.e., SNR-only) should be employed to obtain a good image quality.

This very simple heuristic rate-allocation algorithm was applied to the sequences listed in Table III. For the *Mobile* sequence at both 700 and 1000 kbits/s, scenario A has been employed since the individual image quality is relatively low even at these high transmission bit-rates. Scenario B is selected for the *Surfing* sequence at both 500 and 900 kbits/s due to the large motion activity within the sequence. For the *Akiyo* sequence, scenario B has also been employed despite the relatively slow motion within the sequence since the individual image quality is very high even at low transmission bit-rate (e.g., 100 kbits/s). The visual evaluations of the sequences under both rate-control scenarios were in-line with the automatic decision of the rate-allocation mechanism.

However, for the *Foreman* sequence at, e.g., 490 kbits/s, choosing a single scenario for the entire sequence is not optimal, since the amount of motion activity and image quality vary considerably on a scene basis. For example, the beginning of the sequence is characterized by relatively simple textures that are moving fast, while the end of the sequence is almost static but has very detailed textures. Hence, for an optimal tradeoff between the motion smoothness and SNR, the two scenarios should be alternately used, dependent on the values of ΔPSNR^T and ΔPSNR$^{\mathrm{NT}}$, as illustrated in Fig. 12. From the visual evaluation of this rate-controlled sequence, it can be seen that the overall image quality has improved compared to both

[10]For an improved motion portrayal, motion up-conversion could be applied at the receiver-side [19]. However, this would considerably increase the receivers' complexity.

previously mentioned extreme scenarios. The motion jerkiness that characterized the beginning of the sequence in the first scenario has considerably improved as a result of transmitting additional FGST frames. Furthermore, the image quality at the end of the sequence, which is almost static, has been improved with respect to the second scenario. Moreover, due to the employed rate-control, the quality fluctuations within the sequences (in both motion-smoothness and individual image quality) have been reduced, resulting in an overall more pleasing visual experience.

The computation of the PSNR values required for the rate-control algorithm can be performed at transmission time when the available bandwidth R is known. However, this would require the real-time decoding of each transmitted sequence and thus, add a large undesired complexity to the server. To reduce this complexity, the necessary PSNR-values and the corresponding choice of scenario A or B for the rate allocation of each pair of consecutive (FGS, FGST) frames can be determined at encoding-time (i.e., off-line) for a particular set of bit-rates R_k, $k = 1, \ldots, n$. Then, at transmission time, the server uses the pre-stored rate-allocation choices made for the bit-rate R_k, where $R_{k+1} > R \geq R_k$ depending on the available bit-rate R.

Another very important observation can be made from the plots portrayed in Fig. 11. The image quality of the individual frames improves considerably if a lower frame rate is employed for the coding of the base layer. This is very important, since the perceptual evaluation of the MPEG-4 streams coded at low bit-rates reveals that improving the SNR-quality has the highest priority for most sequences. Therefore, the proposed FGS temporal-SNR scalability scheme provides an indirect mechanism for enhancing the base-layer quality, since the encoder can choose to code the base layer at a lower frame rate, thereby providing the transmitted images with a higher individual image quality. For the clients with receiving bit-rates (much) higher than the base-layer rate, the motion-smoothness can be improved by sending additional frames in the FGST enhancement-layer. Such a tradeoff could not be performed with the original FGS scheme (without FGST), since the frame rate was fixed for all bit-rates (i.e., clients). In that scenario, choosing a frame rate of e.g., 5 fps for compressing the base layer with a better quality would have resulted in penalizing the clients with high connection bit-rates, which would have received a jerky motion video at 5 fps regardless of the available bandwidth.

C. Temporal-SNR Tradeoffs using Base-Layer Information

It is important to mention that the switching decision between FGS and FGST can be made more robust by employing base-layer information. For instance, the motion vectors determined at the base layer can be employed to determine the motion activity for a certain part of the sequence as described in [16]. However, the relatively complex method proposed in [16] can be replaced by a simpler method that relies on information already available to the base-layer encoder: the number of bits required for the compressed representation of the FGST frames motion vectors. The number of bits required for the motion-vectors is a good indication of the motion activity present within the sequence. The number of bits spent on the motion-vector transmission is plotted in Fig. 13 for the FGST frames of the four previously mentioned sequences. From Fig. 13, it can be concluded that the size of the compressed motion-vectors correlates well with the motion activity within the sequence as described in Table III.

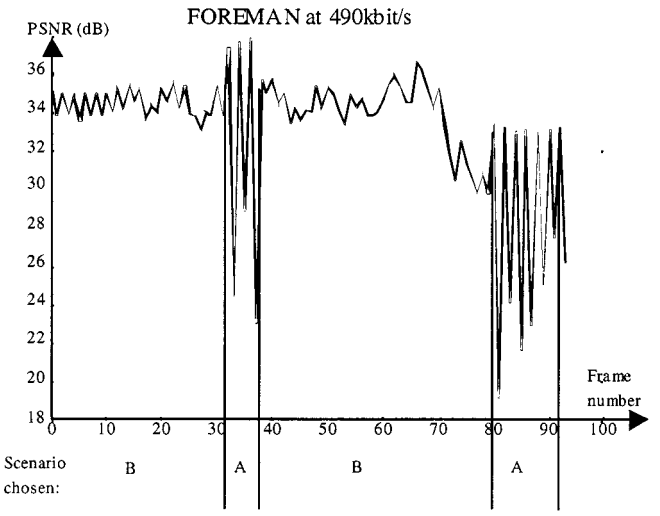


Fig. 12. Performance of the FGS temporal-SNR scalability after the proposed rate-control.

Then, based on the determined motion activity, Scenario A can be employed for sequences with low motion activity and Scenario B for sequences with high motion-activity. One disadvantage of this technique is that it does not take into account the quality of the sequence at the transmission bit-rate. For example, the *Akiyo* sequence will be coded using Scenario A based on this method. However, as previously established, *Akiyo* should be coded using Scenario B since the image quality of the sequence is already very high at the base-layer bit-rate R_{BL}. To improve the decision mechanism, another base-layer encoding parameter can be thus employed: the complexity measure of the I-frame. For example, the complexity measure used in the TM-5 rate control, which is known as Xi, can be employed for this purpose. This measure is determined as part of the TM-5 rate control performed at the base layer. Table IV gives the complexity measures of the intra-frames of the various sequences. A comparison of the determined Xi with the sequences' description in Table III reveals that the intra-frame complexity provides indeed a good indication of the texture characteristics.[11]

The combination of motion activity and the I-frame complexity measure forms a relatively robust mechanism for performing the SNR versus motion-smoothness tradeoffs at a relatively low cost (since these parameters are already computed at the base layer). However, the performance of this method is limited since the switch from Scenario B to Scenario A at higher transmission bit-rates cannot solely be based on base-layer information. The switching should depend on the quality of the decoded enhanced images at the specific transmission bit-rate. Nevertheless, determining the SNR versus motion-smoothness tradeoffs based on information already available from the encoding process forms an interesting topic for further research.

[11]The activity of an I-frame is a good description of the texture characteristics within the GOP only if I-frames are inserted for each scene change.

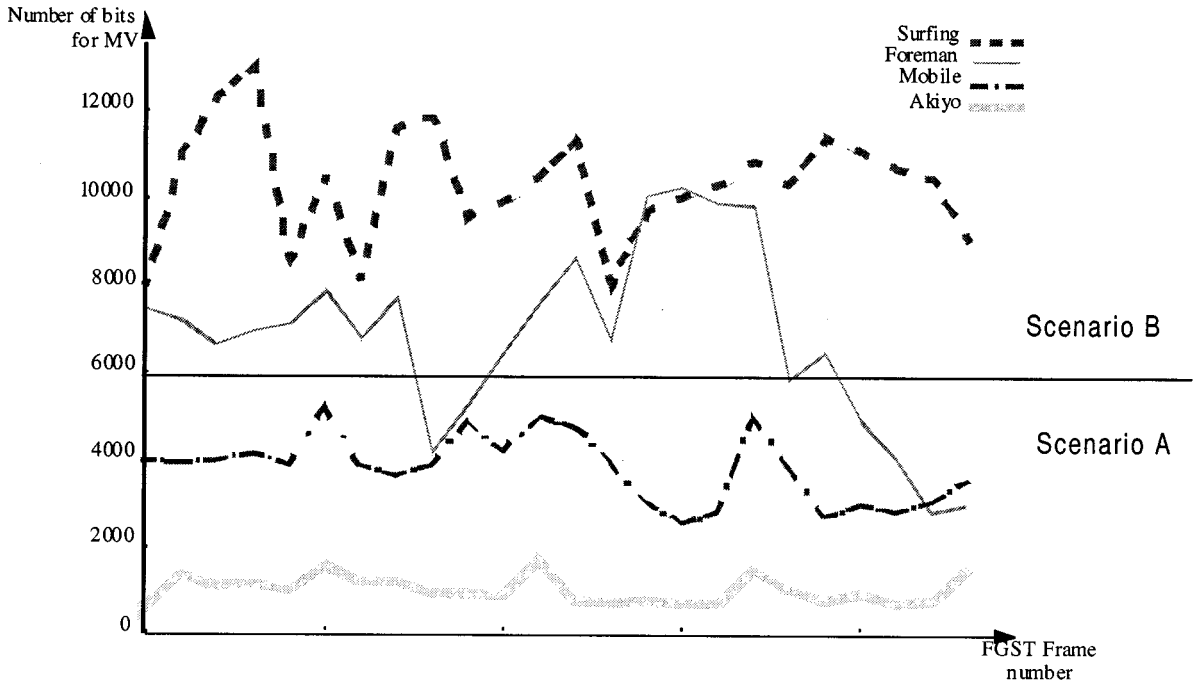


Fig. 13. Number of bits required for the compressed representation of the FGST-frames motion vectors as a function of the FGST frame-number.

TABLE IV
TM-5 COMPLEXITY MEASURES OF THE I-FRAMES (Xi) FOR THE DIFFERENT GOPS OF THE VARIOUS SEQUENCES

Sequence	Xi for GOP_1	Xi for GOP_2	Xi for GOP_3	Xi for GOP_4
Akiyo	172186	171130	165652	165660
Foreman	278505	299549	310928	333242
Mobile	1052634	1002563	1015317	1023496
Surfing	253383	245674	282424	257653

Based on the number of bits required for coding the motion-vectors of the FGST frames depicted in Fig. 13, it can be concluded that a considerable amount of the FGST frames' bit-budget is used for the transmission of the motion information. This is especially true for sequences with a large motion activity. To obtain a constant quality for both the FGS and FGST frames at low bit-rates, a larger number of bits should be allocated to the FGST frames than to the FGS frames to compensate for the bits spent on the motion-vectors transmission. However, this is not necessary for sequences with a low motion-activity (e.g., *Akiyo*, *Mobile*), since the compressed motion-vectors require only a limited number of bits for these sequences.

V. CONCLUSION

In this paper, a novel coding method for Internet video streaming has been presented that allows real-time tradeoffs between image quality (SNR) and motion-smoothness with just one enhancement layer. The most important characteristic of the proposed method is that the encoding and transmission process are separated, allowing the tradeoffs to be performed in real time, depending on the available bandwidth, packet losses, or user preference. The presented scheme is also very resilient to packet-losses, since unequal error-protection can be easily employed to provide enhanced protection of the base layer and limited or no protection to the hybrid temporal-SNR enhancement layer. Additionally, the proposed method can be easily implemented in conjunction with the MPEG-4 FGS (SNR only) scheme with negligible additional complexity. Our experiments also revealed that the presented FGS temporal-SNR scalability has similar or better PSNR performance than the multilayer scalability schemes.

Subsequently, an Internet video streaming system employing the proposed hybrid FGS-temporal scalability structure has been described in detail. This system performs the tradeoffs between SNR and motion-smoothness in real time, depending on the user preference or on a low-complexity rate-control mechanism. For this purpose, a very simple, yet effective, rate control has been introduced that ensures improved visual quality for each client, depending on its available bandwidth. Nevertheless, the proposed rate-control is very simplistic and our future research concentrates on further improvements, e.g., employing a larger number of frames in the decision process or adopting more sophisticated methods for establishing the amount of motion in the scene, while keeping the processing at the server side to a minimal level.

Finally, it is also important to mention that the hybrid temporal-SNR scalability proposed in this paper has been recently

proposed [18] and adopted in the MPEG-4 standard [10] to support video-streaming applications.

ACKNOWLEDGMENT

The authors would like to thank W. Li from Webcast Technologies and H. Jiang and G. Thayer from Intel for their useful comments and help in cross-checking this functionality for the MPEG-4 standardization. Also, the authors would like to thank L. Boland, K. Challapali, and J. Lan from Philips Research for their useful comments on a previous version of this manuscript. Last but not least, the authors' sincere thanks to the three anonymous reviewers who provided excellent and thorough feedback that improved the quality of this paper.

REFERENCES

[1] J. Lu, "Signal processing for internet video streaming: A review," *Proc. SPIE Image and Video Communications and Processing 2000*, vol. 2974, pp. 246–259, Jan. 2000.

[2] B. Girod, K. W. Stuhlmüller, M. Link, and U. Horn, "Packet loss resilient internet video streaming," in *Proc. SPIE VCIP'99*, vol. 3653, Jan. 1999, pp. 833–844.

[3] T. Gardos, "Efficient receiver-driven layered video multicast using H.263+ SNR scalability," in *Proc. ICIP '98*, vol. 3, Oct. 1998, pp. 32–35.

[4] Y. Q. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, Sept. 1992.

[5] S. J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. Image Processing*, pp. 155–167, Feb. 1999.

[6] B. J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," in *Proc. IEEE Data Compression Conf.*, Mar. 1997.

[7] W. Tan and A. Zakhor, "Real-time internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. Multimedia*, vol. 1, pp. 172–186, June 1999.

[8] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, "Scalable internet video using MPEG-4," *Signal Processing: Image Commun.*, no. 15, pp. 95–126, Sept. 1999.

[9] H. Radha and Y. Chen, "Fine-granular-scalable video for packet networks," in *Packet Video '99*. New York: Columbia Univ., Apr. 1999.

[10] *MPEG-4 Video* , Proposed Draft Amendment (PDAM), ISO/IEC FGS v. 4.0 14 496-2, Mar. 2000.

[11] M. van der Schaar, H. Radha, and C. Dufour, "Scalable MPEG-4 video coding with graceful packet-loss resilience over bandwidth-varying networks," in *Proc. ICME 2000*, New York, July 2000.

[12] M. van der Schaar, Y. Chen, and H. Radha, "Embedded DCT and wavelet methods for fine granular scalable video: Analysis and comparison," *Proc. SPIEImage and Video Communications and Processing 2000*, vol. 2974, pp. 643–653, Jan. 2000.

[13] W. Li, "Bit-plane coding of DCT coefficients for fine granularity scalability," in *Contribution to 45th MPEG Mtg.*, Atlantic City, NJ, Oct. 1998, m3989.

[14] W. Li *et al.*, "Experiment result on fine granular scalability," in *Contribution to 47th MPEG Meeting*, Seoul, Korea, March 1999.

[15] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Trans. Commun.*, vol. 38, pp. 477–486, Apr. 1990.

[16] F. Ishtiaq and A. K. Katsaggelos, "A rate control method for H.263 temporal scalability," in *Proc. ICIP '99*, Kobe, Japan, Oct. 1999.

[17] H. Song, J. Kim, and C. C. J. Kuo, "Real-time encoding frame rate control for H.263+ video over Internet," *Signal Processing: Image Commun.*, vol. 15, pp. 95–126, Sept. 1999.

[18] M. van der Schaar, H. Radha, and Y. Chen, "An all FGS solution for hybrid temporal-SNR scalability," in *Contribution to 50th MPEG Meeting*, 1999.

[19] D. Bagni, G. de Haan, and V. Riva, "Motion compensated post-processing for low bit rate videoconferencing on ISDN lines," in *Dig. ICCE*, Chicago, IL, June 1997, pp. 28–29.

[20] I. H. Witten, R. M. Neal, and J. G. Clearly, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.

Mihaela van der Schaar (M'98) graduated in electrical engineering from Eindhoven University of Technology, Eindhoven, the Netherlands, in April 1996.

In the same year, she joined Philips Research Laboratories, Eindhoven, the Netherlands, where she became a member of the TV Systems Department. From 1996 to 1998, she was involved in several projects which investigated low-cost very high-quality video compression techniques and their implementation for TV, computer, and camera systems. Since 1998, she has been an expatriate with the Video Communications Department, Philips Research Briarcliff, Briarcliff Manor, NY, where she is currently involved in the research of video-coding techniques for Internet video streaming and leads a team of researchers working on scalable video coding, networking, and streaming algorithms. Since 1999, she has been an active participant to the MPEG-4 video standard, contributing to the "Fine Granularity Scalability" tool. Her research interests include video and graphics coding and video streaming over unreliable channels, topics in which she has co-authored more than 20 conference and journal papers. Additionally, she holds several patents.

Hayder Radha (M'92) received the M.S. degree from Purdue University, West Lafayette, IN, in 1986, and the B.S. degree (Hons.) from Michigan State University, East Lansing, in 1984, both in electrical engineering. He received the Ph.D. and Ph.M. degrees from the Center for Telecommunications Research (CTR), Columbia University, New York, in 1993 and 1991, respectively.

He joined the faculty at Michigan State University in 2000 as an Associate Professor in the Department of Electrical and Computer Engineering. He was with Philips Research during 1996–2000, working in the area of video communications and high-definition television, and is a Philips Research Fellow. He initiated the Internet Video Research Program at Philips Research and led a team of researchers working on scalable video coding, networking, and streaming algorithms. Previously, he was a Distinguished Member of Technical Staff at Bell Laboratories, where he worked in the areas of digital communications, signal and image processing, and broadband multimedia communications. He served as a Co-Chair and an Editor of the ATM and LAN Video Coding Experts Group of the ITU-T between 1994–1996. He is also an Adjunct Professor at the City University of New York (CUNY). His research interests include image and video coding, multimedia communications and networking, and the transmission of multimedia data over wireless and packet networks. He has more than 20 patents in these areas (between granted and pending).