The SP- and SI-Frames Design for H.264/AVC

Marta Karczewicz and Ragip Kurceren, *Member, IEEE*

***Abstract*—This paper discusses two new frame types, SP-frames and SI-frames, defined in the emerging video coding standard, known as ITU-T Rec. H.264 or ISO/IEC MPEG-4/Part 10-AVC. The main feature of SP-frames is that identical SP-frames can be reconstructed even when different reference frames are used for their prediction. This property allows them to replace I-frames in applications such as splicing, random access, and error recovery/resilience. We also include a description of SI-frames, which are used in conjunction with SP-frames. Finally, simulation results illustrating the coding efficiency of SP-frames are provided. It is shown that SP-frames have significantly better coding efficiency than I-frames while providing similar functionalities.**

***Index Terms*—AVC, bitstream switching, error recovery, error resiliency, H.264, JVT, MPEG-4, random access, SI-frames, SP-frames, splicing.**

I. Introduction

TO INCREASE compression efficiency and network friendliness, the emerging ITU-T Recommendation H.264, ISO/IEC MPEG-4/Part 10-AVC [1] video compression standard introduces an extensive set of new features. In this paper, we describe in detail two of these features, specifically, new frame types referred to as SP-frames [1]–[3] and SI-frames [1], [4].

SP-frames make use of motion compensated predictive coding to exploit temporal redundancy in the sequence similar to P-frames. The difference between SP- and P-frames is that SP-frames allow identical frames to be reconstructed even when they are predicted using different reference frames. Due to this property, SP-frames can be used instead of I-frames in such applications as bitstream switching, splicing, random access, fast forward, fast backward, and error resilience/recovery. At the same time, since SP-frames unlike I-frames are utilizing motion-compensated predictive coding, they require significantly fewer bits than I-frames to achieve similar quality. In some of the mentioned applications, SI-frames are used in conjunction with SP-frames. An SI-frame uses only spatial prediction as an I-frame and still reconstructs identically the corresponding SP-frame, which uses motion-compensated prediction.

The remainder of the paper is organized as follows. In Section II, a review of frame types being used in the existing standards is given. In Section III, we discuss how features of SP-frames can be exploited in several example applications. Section IV provides a description of SP- and SI-frame decoding and encoding processes. Section V includes details of experiments and a relative performance improvement of the SP-frames. Finally, Section V offers a summary and conclusions.

Manuscript received December 13, 2001; revised May 9, 2003.

The authors are with Nokia Research Center, Nokia Inc., Irving, TX 75039 USA (e-mail: ragip.kurceren@nokia.com and marta.karczewicz@nokia.com).

Digital Object Identifier 10.1109/TCSVT.2003.814969

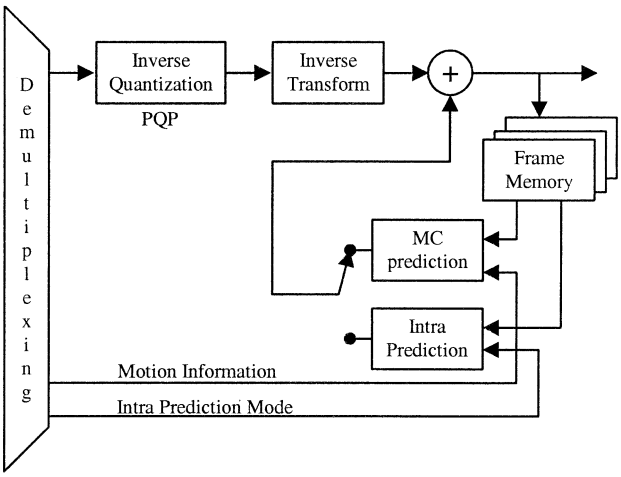

Fig. 1. Generic block diagram of decoding process.

II. Motivation

In the existing video coding standards, such as MPEG-2, H.263 and MPEG-4, three main types of frames are defined. Each frame type exploits a different type of redundancy existing in video sequences and consequently results in a different amount of compression efficiency and different functionality that it can provide.

An intra-frame (or I-frame) is a frame that is coded exploiting only the spatial correlation of the pixels within the frame without using any information from other frames. I-frames are utilized as a basis for decoding/decompression of other frames and provide access points to the coded sequence where decoding can begin.

A predictive-frame (or P-frame) is coded/compressed using motion prediction from a so-called reference frame, i.e., a past I- or P-frame available in an encoder and decoder buffer. Fig. 1 illustrates a generic decoding process for P- and I-frames. Finally, a bidirectional-frame (or B-frame) is coded/compressed using a prediction derived from an I-frame (P-frame) in its past or an I-frame (P-frame) in its future or a combination of both. B-frames are not used as a reference for prediction of other frames.

Since, in a typical video sequence, adjacent frames are highly correlated, higher compression efficiency is achieved when using B- or P-frames instead of I-frames. On the other hand, temporal predictive coding employed in P- and B-frames introduces temporal correlation within the coded bitstream, i.e., B- or P-frames cannot be decoded without correctly decoding their reference frames in the future and/or past. In cases when a reference frame used in an encoder and a reference frame used in a decoder are not identical either due to errors during transport or due to some intentional action on the server side,

the reconstructed values of the subsequent frames predicted from such a reference frame are different in the encoder than in the decoder. This mismatch would not only be confined to a single frame but would further propagate in time due to the motion-compensated coding.

In H.264/AVC, I-, P-, and B-frames have been extended with new coding features, which lead to a significant increase in coding efficiency. For example, H.264/AVC allows using more than one prior coded frame as a reference for P- and B-frames. Furthermore, in H.264/AVC, P-frames and B-frames can use prediction from subsequent frames [5]. These new features are described in detail elsewhere in this Special Issue.

Additionally, two new types of frames have been defined, namely, SP-frames [2], [3] and SI-frames [4]. The method of coding as defined for SP- and SI-frames allows obtaining frames having identical reconstructed values even when different reference frames are used for their prediction.

In the following, we describe on how these features of SP-frames and SI-frames can be exploited in specific applications [6].

A. Bitstream Switching

Video streaming has emerged as one of the essential applications over the fixed internet and in the near future over 3G wireless networks. The best-effort nature of today's networks causes variations of the effective bandwidth available to a user due to the changing network conditions. The server should then scale the bit rate of the compressed video, transmitted to the receiver, to accommodate these variations. In case of conversational services that are characterized by real-time encoding and point-to-point delivery, this can be achieved by adjusting, on the fly, source encoding parameters, such as a quantization parameter or a frame rate, based on the network feedback. In typical streaming scenarios when an already encoded video bitstream is to be sent to a client, the above solution cannot be applied.

The simplest way of achieving bandwidth scalability in the case of pre-encoded sequences is by representing each sequence using multiple and independent streams of different bandwidth and quality. The server then dynamically switches between the streams to accommodate the variations of the bandwidth available to the client.

Assume that we have multiple bitstreams generated independently with different encoding parameters, corresponding to the same video sequence. Let $\{P_{1,n-1}, P_{1,n}, P_{1,n+1}\}$ and $\{P_{2,n-1}, P_{2,n}, P_{2,n+1}\}$ denote the sequence of the decoded frames from bitstreams 1 and 2, respectively. Since the encoding parameters are different for each bitstream, the reconstructed frames from different bitstreams at the same time instant, for example, frames $P_{1,n-1}$ and $P_{2,n-1}$, will not be identical. Now let us assume that the server initially sends bitstream 1 up to time n after which it starts sending bitstream 2, i.e., the decoder would have received $\{P_{1,n-2}, P_{1,n-1}, P_{2,n}, P_{2,n+1}, P_{2,n+2}\}$. In this case, the frame $P_{2,n}$ can not be correctly decoded since the reference frame $P_{2,n-1}$ used to obtain its prediction is not received whereas the frame $P_{1,n-1}$, which is received instead of $P_{2,n-1}$, is not identical to $P_{2,n-1}$. Therefore, switching between bitstreams at arbitrary locations can lead to visual artifacts due to the mismatch between the reference frames.

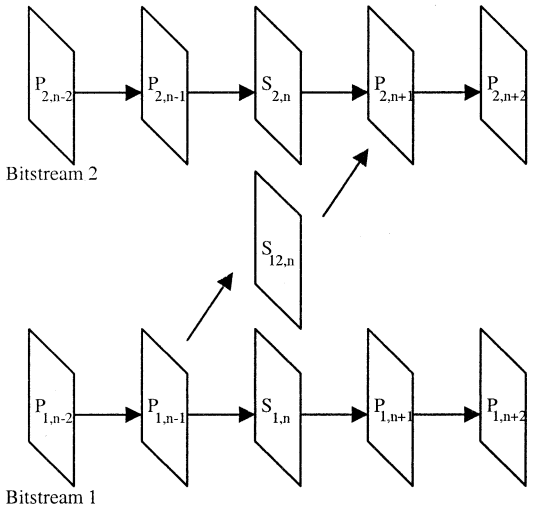

Fig. 2. Switching between bitstreams using SP-frames.

Furthermore, the visual artifacts will not only be confined to the frame $P_{2,n}$ but will further propagate in time due to motion-compensated coding.

In the prior video encoding standards, perfect (mismatch-free) switching between bitstreams is possible only at frames, which do not use any information prior to their location, i.e., at I-frames. Furthermore, by placing I-frames at fixed (e.g., 1 s) intervals, VCR functionalities, such as random access or "Fast Forward" and "Fast Backward" (increased playback rate) when streaming a video content, are achieved. The user may skip a portion of a video sequence and restart playing at any I-frame location. Similarly, the increased playback rate can be achieved by transmitting only I-frames. The drawback of using I-frames in these applications is that, since I-frames do not exploit any temporal redundancy, they require much larger number of bits than P-frames at the same quality.

From the properties of SP-frames, note that identical SP-frames can be obtained even when they are predicted using different reference frames. This feature can be exploited in bitstream switching as follows. Fig. 2 depicts an example how to utilize SP frames to switch between different bitstreams. Again assume that there are two bitstreams corresponding to the same sequence encoded at different bit rates and/or at different temporal resolutions. Within each encoded bitstream, SP-frames are placed at the locations at which switching from one bitstream to another will be allowed (frames $S_{1,n}$ and $S_{2,n}$ in Fig. 2). These SP-frames shall be referred to as primary SP-frames in what follows. Furthermore, for each primary SP-frame, a corresponding secondary SP-frame is generated, which has the same identical reconstructed values as the primary SP-frame. Such a secondary SP-frame is sent only during bitstream switching. In Fig. 2, the SP-frame $S_{12,n}$ is the secondary representation of $S_{2,n}$, which will be transmitted only when switching from bitstream 1 to bitstream 2. $S_{2,n}$ uses the previously reconstructed frames from bitstream 2 as the reference frames, while $S_{12,n}$ uses the previously reconstructed frames from bitstream 1 as the reference frames. However, due to the special encoding of the secondary SP-frame, described later, the reconstructed values of these frames are identical.

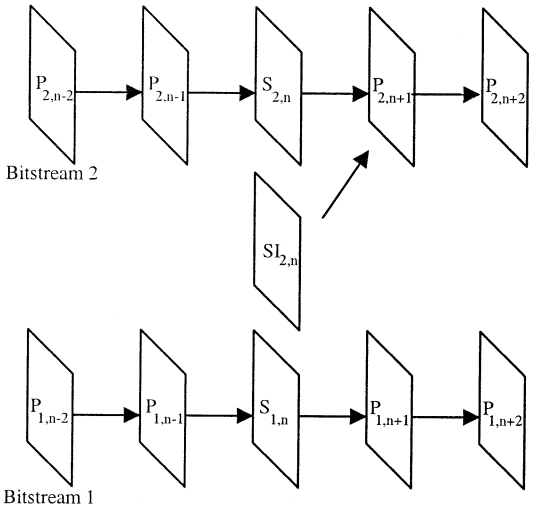

Fig. 3. Splicing, random access using SI-frames.

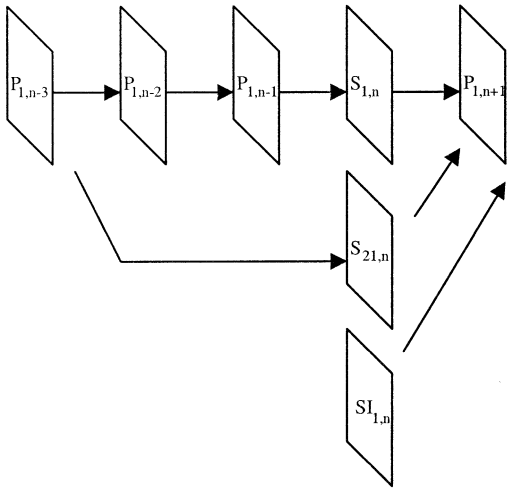

Fig. 4. SP-frames in error resiliency/recovery.

If one of the bitstreams has a lower temporal resolution, e.g., 1 fps, then this bitstream can be used to achieve fast-forward functionality. Specifically, decoding from the bitstream with the lower temporal resolution and then switching to the bitstream with the normal frame rate would provide such functionality.

B. Splicing and Random Access

The bitstream-switching example discussed earlier considers bitstreams representing the same sequence of images. However, this is not necessarily the case for other applications where bitstream switching is needed. Examples include

- switching between bitstreams arriving from different cameras capturing the same event but from different perspectives, or cameras placed around a building for surveillance;
- switching to local/national programming or insertion of commercials in TV broadcast, video bridging, etc.

Splicing refers to the process of concatenating encoded bitstreams and includes the examples discussed earlier.

When switching occurs between bitstreams representing different sequences, that affects encoding of secondary frames, i.e., encoding of $S_{12,n}$ in Fig. 2. Specifically, motion-compensated prediction of frames from one bitstream using reference frames from another bitstream when these bitstreams represent different sequences will not be as effective as when both bitstreams correspond to the same sequence. In such cases, using spatial prediction for the secondary frames could be more efficient. This is illustrated in Fig. 3, where the secondary frame is denoted as $\mathrm{SI}_{2,n}$ to indicate that this is an SI-frame encoded, as described later, using spatial prediction and having identical reconstructed values as the corresponding SP-frame $S_{2,n}$. SI-frames can provide also random access points to the bitstream and have further implications in error recovery and resiliency, which will be described in Sections II-C and D.

C. Error Recovery

Multiple representations of a single frame in the form of SP-frames predicted from different reference frames, e.g., the immediate previously reconstructed frame and a reconstructed frame further back in time, as illustrated in Fig. 4, can be used to increase error resilience and/or error recovery. Consider the case when an already encoded bitstream is being streamed and there has been a packet loss leading to a frame or slice loss. The client signals the lost frame to the server which then responds by sending one of the secondary representations of the next SP-frame. This secondary representation, e.g., $S_{21,n}$ in Fig. 4, uses the reference frames that have been correctly received by the client.

Similarly, as argued earlier in the discussion of splicing, another representation of the SP-frame can be generated without using any reference frames, i.e., $\mathrm{SI}_{1,n}$ in Fig. 4. In this case, the server can send the SI-frame representation, i.e., $\mathrm{SI}_{1,n}$ instead of $S_{1,n}$ to stop error propagation. For slice-based packetization and delivery, the server could further estimate the slices that would be affected by such a slice/frame loss and update only those slices in the next SP-frame with their secondary representations.

D. Error Resiliency

For lossy transport networks, intra macroblock refresh strategy has been shown to provide significant increase in error resiliency/recovery performance [7]–[10]. Furthermore, it has been illustrated in [7]–[10] that intra macroblock refresh rate, i.e., frequency at which a macroblock is intra-encoded, should depend on transport channel conditions, e.g., packet loss and/or a bit error rate. In interactive client/server scenarios, the encoder on the server side decides to encode the slices/macroblocks in the intra mode either based on: the specific feedback received from the client, or the expected network conditions calculated through negotiation, or the measured network conditions. However, when already encoded bitstreams are sent, which is the case in typical streaming applications, the above strategy cannot be applied directly. Either the sequence needs to be encoded with the worst-case expected network conditions or additional error resiliency/recovery mechanisms are required.

From the earlier discussion on SP-frame usage in error recovery and splicing applications, SP-frames or slices can be represented as SI-frames/slices that do not use any reference frames. This feature can be exploited in the adaptive intra refresh mechanism discussed above. First, a sequence is encoded

with some predefined ratio of SP-slices. Then during transport, instead of some of the SP-slices their secondary representation, that is SI-slices, is sent. The number of SI-slices that should be sent can be calculated similarly as in the real-time encoding/delivery approach.

E. Video Redundancy Coding

SP-frames have other uses in applications in which they do not act as replacements of I-frames. Video redundancy coding (VRC) can be given as an example. "The principle of the VRC method is to divide the sequence of pictures into two or more threads in such a way that all camera pictures are assigned to one of the threads in a round-robin fashion. Each thread is coded independently. In regular intervals, all threads converge into a so-called sync frame. From this sync frame, a new thread series is started. If one of these threads is damaged because of a packet loss, the remaining threads stay intact and can be used to predict the next sync frame. It is possible to continue the decoding of the damaged thread, which leads to slight picture degradation, or to stop its decoding which leads to a drop of the frame rate. Sync frames are always predicted out of one of the undamaged threads. This means that the number of transmitted I-frames can be kept small, because there is no need for complete re-synchronization." [1] For the sync frame, more than one representation (P-frame) is sent, each one using a reference frame from a different thread. Due to the usage of P-frames these representations are not identical. Therefore a mismatch is introduced when some of the representations cannot be decoded and their counterparts are used when decoding the following threads. The use of SP-frames as sync frames eliminates this problem.

III. Decoding and Encoding Processes for SP- and SI-Frames

In this section, we provide a detailed description of decoding and encoding processes for nonintra blocks in SP- and SI-frames. For intra blocks in SP- and SI-frames, the process identical to that of I-frames is applied [1]. As noted earlier, SP-frames can be further classified as secondary SP-frames, e.g., $S_{12,n}$ in Fig. 2, and primary SP-frames, e.g., $S_{1,n}$ and $S_{2,n}$ in Fig. 2. We first describe the decoding process for the secondary SP-frames and SI-frames [2] to illustrate the basic principle. Then the description of the improved decoder [3], which is used for decoding primary SP-frames, is provided. Finally, an example of an SP- and SI-frame encoder is given.

A. Decoding Process for Secondary SP-Frames and SI-Frames

Fig. 5 illustrates a general schematic block diagram of decoding process for secondary SP-frames and SI-frames. As can be observed from Figs. 1 and 5, SP- and SI-frames make use of the existing coding modules for P- and I- frames. First, a predicted block $P(x, y)$ is formed. For SP-frames, $P(x, y)$ is formed by motion-compensated prediction from the already decoded frames using the motion vectors and the reference frame number information that are received from the encoder. For SI-frames, it is formed by spatial prediction from the already decoded neighboring pixels within the same frame

[1]The description of VRC is copied from [14].

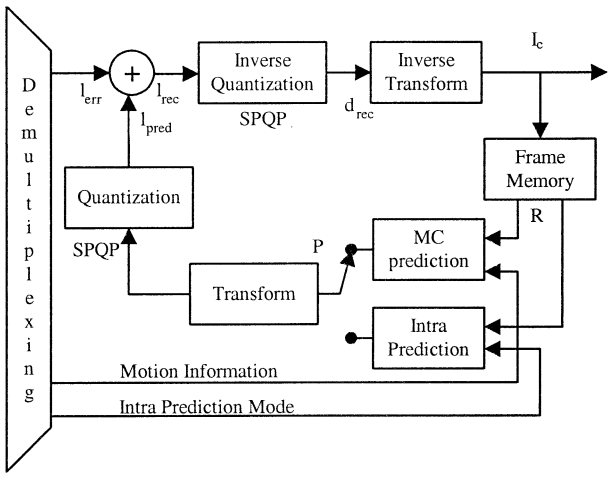

Fig. 5. Generic block diagram of decoding process for secondary SP- and SI-frames.

using the intra prediction mode information received from the encoder. Then, forward transform is applied to the predicted block $P(x, y)$ and the obtained coefficients are quantized. The quantized predicted block coefficients denoted by l_{pred} are added to the received quantized prediction error coefficients l_{err} to calculate the quantized reconstruction coefficients l_{rec}. The image is reconstructed by inverse transform of d_{rec}, which are found by dequantizing l_{rec}.

Since during SP-frame (SI-frame) decoding, unlike during P-frame (I-frame) decoding (compare Figs. 1 and 5), transform is applied to predicted blocks and resulting coefficients are quantized, coding efficiency of the SP-frames (SI-frames) is expected to be worse than that of P-frames (I-frames), as will be illustrated in Section IV. The quantization applied to the predicted block coefficients and the prediction error coefficients in the secondary SP- and SI-frame decoding scheme described above has to be the same. More specifically, both should use the same quantization parameter. To improve coding efficiency, the following improved decoding structure [3] is defined, which gives the flexibility to use different quantization parameters for the predicted block coefficients than for the prediction error coefficients.

B. Decoding Process for Primary SP-Frames

Fig. 6 illustrates a block diagram of the improved SP-frame decoder used for primary SP-frames. Similar to the earlier case, a predicted block $P(x, y)$ is formed and forward transform is applied. The obtained transform coefficients are denoted by c_{pred}. Then the quantized prediction error coefficients l_{err} that are received from the encoder are dequantized using a quantization parameter PQP and added to the predicted block transform coefficients c_{pred}. The sum is denoted by c_{rec}. Note that in the earlier case, the quantized coefficients were added whereas in this case the summation of coefficients is performed. The reconstruction coefficients c_{rec} are quantized and dequantized using a quantization parameter SPQP and inverse transform is applied to the resulting coefficients d_{rec}, similar to the earlier scheme. The quantization parameter used for reconstruction coefficients and in turn for the predicted block coefficients, namely SPQP, is

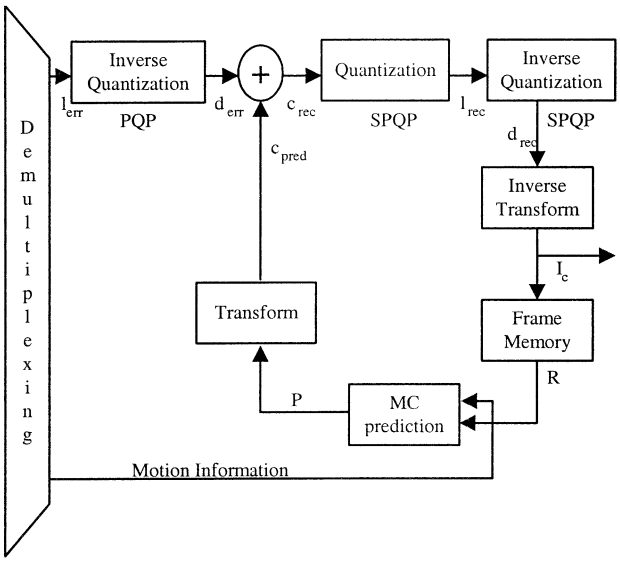

Fig. 6. Generic block diagram of decoding process for primary SP-frames.

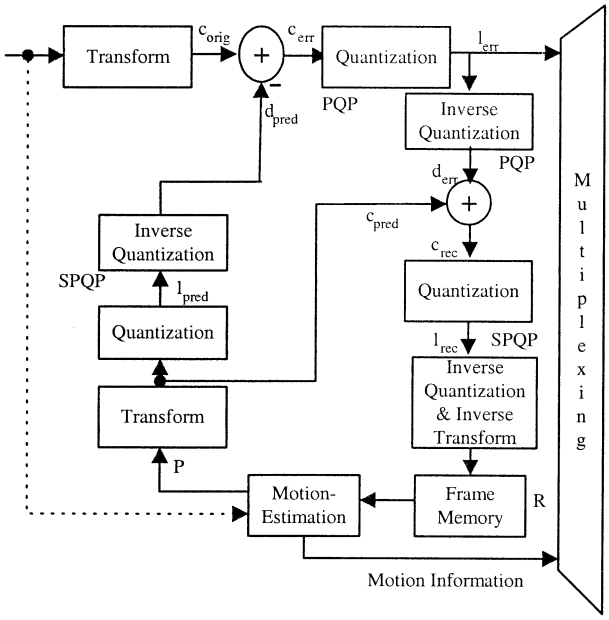

Fig. 7. Generic block diagram of encoding process for nonintra blocks in SP-frames.

not necessarily the same as the quantization parameter PQP used for the prediction error coefficients. Therefore, in this case, a finer quantization parameter, introducing smaller distortion, can be used for the predicted block coefficients than for the prediction error coefficients, which will result in smaller reconstruction error.

The SP- and SI-frame decoders discussed earlier in this section are general decoders and can easily be incorporated into other coding standards. The specific details as to how they are implemented in H.264/AVC can be found in [1].

C. SP-Frame and SI-Frame Encoder

In this section, we first present the encoding process for primary SP-frames and later for secondary SP- and SI-frames. The following applies to the encoding of nonintra blocks in SP- and SI-frames. For intra blocks in SP- and SI-frames, the identical process to that used for I-frames is applied [1].

Fig. 7 illustrates a general schematic block diagram of an example encoder corresponding to the primary SP-frames. First, a predicted block $P(x,y)$ is formed by motion-compensated prediction using the original image and the previously reconstructed frames. Then forward transform is applied to both the predicted block $P(x,y)$ and the corresponding block in the original image. The transform coefficients of $P(x,y)$ are quantised and dequantized using the quantization parameter SPQP. The obtained coefficients d_{pred} are then subtracted from the transform coefficients of the original image. The results of the subtraction represent the prediction error coefficients c_{err}. The prediction error coefficients are quantized using PQP and the resulting coefficients l_{err} are sent to the multiplexer together with motion vector information. The decoding process follows the identical steps as described earlier.

In the following, we illustrate with an example, how SP-frames provide the functionality mentioned earlier, i.e., identical frames are reconstructed even when different reference frames are used for their prediction. Let us denote by $I_c(x,y)$ the reconstructed values of the primary SP-frame encoded using the predicted frame $P_1(x,y)$ and obtained by inverse transform of the quantized reconstructed coefficients l_{rec} (see Fig. 6). Now assume that we would like to generate secondary representation of this primary SP-frame having the identical reconstructed values $I_c(x,y)$ and encoded using a different predicted frame $P_2(x,y)$. The problem becomes finding new prediction error coefficients $l_{\text{err},2}$ that would identically reconstruct the frame $I_c(x,y)$ using $P_2(x,y)$ instead of $P_1(x,y)$. The quantized transform coefficients $l_{\text{pred},2}$ are calculated for the predicted frame $P_2(x,y)$ using the quantization parameter SPQP. Then the prediction error coefficients for the secondary SP-frame are simply computed as $l_{\text{err},2} = l_{\text{rec}} - l_{\text{pred},2}$. On the decoder side, according to the decoding process for secondary SP-frames, as illustrated in Fig. 5, the decoder forms $P_2(x,y)$ and then computes $l_{\text{pred},2}$ by first applying transform to $P_2(x,y)$ and then quantizing obtained coefficients using SPQP. The coefficients $l_{\text{pred},2}$, identical to the ones on the encoder side, are added to the received prediction error coefficients $l_{\text{err},2}$. The resulting sum is equal to $l_{\text{err},2} + l_{\text{pred},2} = l_{\text{rec}} - l_{\text{pred},2} + l_{\text{pred},2} = l_{\text{rec}}$. This example illustrates that with the special encoding of secondary SP-frames, identical reconstructed values are obtained to a corresponding primary SP-frames.

When $P_2(x,y)$ is formed by intra-prediction, this case becomes converting a primary SP-frame with motion-compensated prediction into a secondary SI-frame with only spatial prediction. As shown earlier, this property has major implications in random access, and error recovery/resilience.

To achieve identical reconstruction, the quantization parameter used for a secondary SP-frame should be equal to the quantization parameter SPQP used for the predicted frame in a primary SP-frame. That means that using a finer quantization parameter value for SPQP although improves the coding efficiency of the primary SP-frames placed within the bitstream might result in larger frame sizes for the secondary SP-frames. Since the secondary representations are sent only during switching

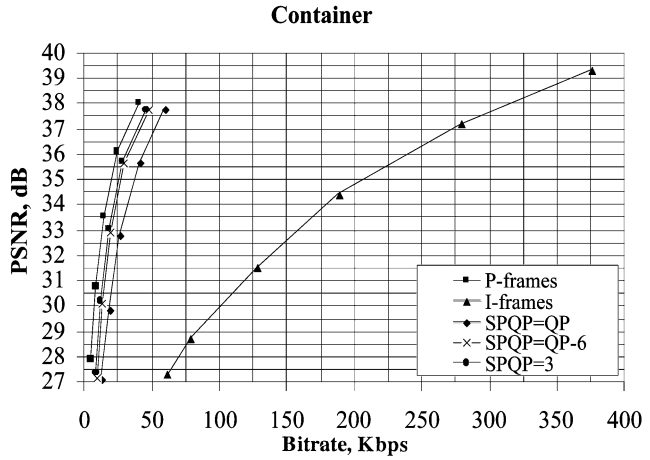

Fig. 8. Illustration of coding efficiencies SP frames using different SPQP values, also included are I- and P-frame performances.

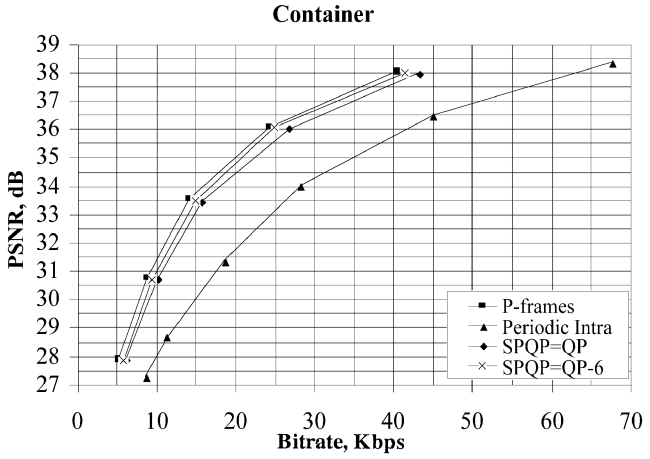

Fig. 9. Illustration of coding efficiencies SP frames using different SPQP values when inserted periodically, also included is the periodic-intra coding approach.

or random access, the choice of the SPQP value is application dependent. For example when SP-frames are used to facilitate random access one can expect that the SP-frames placed within a single bitstream will have the major influence on compression efficiency and therefore the SPQP value should be small. On the other hand, when SP-frames are used for streaming rate control, the SPQP value should be kept close to PQP since the secondary SP-frames sent during switching from one bitstream to another will have large share of the overall bandwidth.

IV. Results

In this section, we provide simulation results to illustrate the coding efficiency of SP-frames. First, we compare the coding efficiency of SP-frames with I- and P-frames. The results are obtained using TML 8.7 software [13] with five reference frames in UVLC mode with rate-distortion optimization option enabled. Later, a comparison of SP-frames with S-frames [11] is provided; these results are repeated from [12].

The results reported here are for some of the standard sequences used in JVT contributions with QCIF resolution and encoded at 10 fps. Similar results are observed for other sequences and further results can be found in [3].

1) Coding Efficiency of SP-Frames: Fig. 8 gives the comparison of coding efficiency of I-, P-, and SP- frames, in terms of their PSNR as a function of bit rate. These results are generated by encoding each frame of a sequence as either an I-, P-, or SP-frame, with the exception of the first frame, which is always an I-frame. We also include in Fig. 8 the results when SP-frames are coded using different values of SPQP. In the first case, SPQP is the same as PQP, then SPQP is equal to 3 and in the last case, SPQP is equal to PQP-6. It can be observed in Fig. 8, that SP-frames have lower coding efficiency than P-frames and significantly higher coding efficiency than I-frames. Note, however, that SP-frames provide functionalities that are usually achieved only with I-frames. As expected, the SP-frames performance improves with decreasing SPQP versus PQP. The SP-frames coding efficiency for $\mathrm{SPQP} = 3$ becomes very close, slightly worse at higher rates, to the P-frame coding efficiency. Further results can be found in [3].

2) Performance Improvement When SP-Frames are Inserted Periodically: In the following, we present simulation results when SP- and I-frames are introduced at fixed intervals, as it will be the case when enabling bitstream-switching or random-access functionality. Fig. 9 illustrates the results obtained under the following conditions: the first frame is encoded as an I-frame and at fixed intervals, in this case 1 s, the frames are encoded as I- or SP-frames while the remaining frames are encoded as P-frames. Also included in Fig. 9 is the performance achieved when all the frames are encoded as P-frames. Note that in this case, none of the functionalities mentioned earlier can be obtained while it provides a benchmark for comparison with both SP- and I-frame cases. As can be seen in Fig. 9, SP-frames while providing the same functionalities as I-frames have significantly higher coding efficiency. Furthermore, the performance of SP-frames improves and approaches the P-frame performance with decreasing SPQP. Further results can be found in [3].

3) Comparison With S-Frames: Farber *et al.* [11] introduced a specially encoded P-frame, called an S-frame, which is sent only when switching from one bitstream to another, similar to the secondary SP-frame $S_{12,n}$ in Fig. 2. More specifically, to enable switching from bitstream 1 to bitstream 2 at nth frame, an S-frame is generated by encoding the nth reconstructed frame in bitstream 2 as a P-type frame predicted from previously reconstructed frames from bitstream 1. Unlike SP-frames, S-frames introduce mismatch while switching between bitstreams.

To minimize drift, the quantization parameter QP used for S-frames should be kept small. In the following, we use QP equal to 3 to ensure that the initial PSNR difference is below 0.2 dB, as is recommended in [11].

In Fig. 10, we present examples illustrating switching between bitstreams using S-frames. Fig. 10 illustrates the PSNR profiles of bitstreams encoded with $\mathrm{QP} = 19$ and with $\mathrm{QP} = 13$ and two additional bitstreams created by switching between them. In each case, we switch from the bitstream encoded with $\mathrm{QP} = 19$ to the bitstream encoded with $\mathrm{QP} = 13$ but at different frames, namely at frames number 10 and 20.

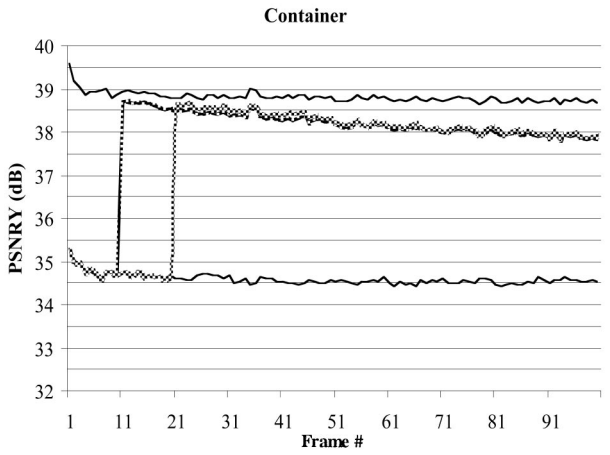

Fig. 10. Illustration of switching between bitstream encoded by $\mathrm{QP} = 19$ and 13 with S-frames, QP (S-frame) is equal to 3.

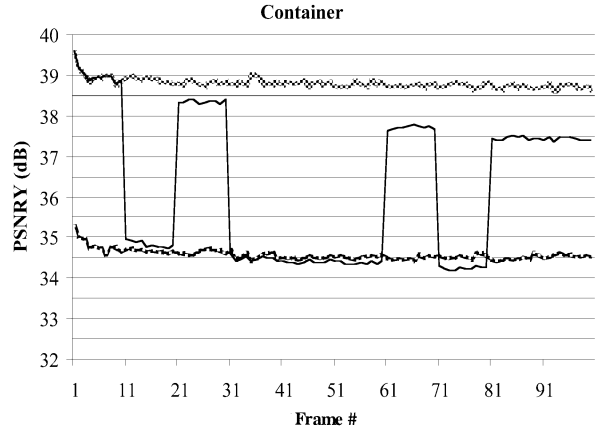

Fig. 11. Multiple switching between bitstreams encoded with $\mathrm{QP} = 19$ and 13 when using S-frames. QP for S-frame is equal to 3.

As can be seen from Fig. 10, the PSNR values of the reconstructed frames after the switch diverge from the values of the frames from the "target" bitstream—the bitstream that we are switching to. Moreover, the drift becomes more pronounced when multiple switches occur as illustrated in Fig. 11. In Fig. 11, the drift becomes larger than 1.5 dB after the last switch between bitstreams. Using even smaller quantization parameter for S-frames could reduce the drift; however, that would increase S-frame sizes which, as will be shown below, are already substantial.

In this section, we present a brief comparison of SP- and S- frames. Tables I and II list the average frame sizes of S- and SP-frames. Here, SP-frames refer to secondary SP-frames that would be used during switching, e.g., $S_{12,n}$ in Fig. 2. The average is taken over 10 S-frames (SP-frames) used to switch between two bitstreams at different locations. We switch from the bitstream encoded with QP_1 to the bitstream encoded with QP_2 where the values of QP_1 and QP_2 are given next to the sequence name in the first column of Tables I and II. Notice that in Table I, the switch always takes place from the lower to higher quality bitstream and in Table II from the higher to lower quality bitstream. In both cases, the sizes of the S-frames are considerably larger than that of the SP-frames. The differences

TABLE I
COMPARISON OF AVERAGE I-, S-, AND SP-FRAME SIZES THAT WOULD BE USED DURING SWITCHING FROM BITSTREAM ENCODED WITH QP_1 TO THE BITSTREAM ENCODED WITH QP_2

Sequence (QP_1-QP_2)	S-frame size	SP-frame Size, S_{12}
Foreman (22-16)	68938	28304
Hall (19-13)	58290	26329
Container (19-13)	60808	28430
News (25-19)	69921	20610

TABLE II
COMPARISON OF AVERAGE I-, S-, AND SP-FRAME SIZES THAT WOULD BE USED DURING SWITCHING FROM BITSTREAM ENCODED WITH QP_1 TO THE BITSTREAM ENCODED WITH QP_2

Sequence (QP_1-QP_2)	S-frame size	SP-frame Size, S_{12}
Foreman (16-22)	61950	15563
Hall (13-19)	55409	13694
Container (13-19)	52973	14485
News (19-25)	66566	10766

TABLE III
PSNR AND TOTAL BITS OVER 100 FRAMES FOR THE MULTIPLE SWITCHING EXAMPLE FOR S-, I-, AND SP-FRAME APPROACHES

Sequence (QP_1-QP_2)	Periodic I-frame		S-frames		SP-frames		% Differ. Btw.SP and I-frames	% Differ. Btw.SP and S-frames
	PSNR	Total Bits	PSNR	Total Bits	PSNR	Total Bits		
Container (25-19)	33.17	294561	32.84	412124	32.87	229422	28	80
News (25-19)	33.48	466260	33.28	586703	33.11	389091	20	51
Foreman (22-16)	35.27	894110	35.08	1012345	35.13	857023	4	18
Hall (19-13)	38.20	740551	37.91	721157	38.08	594827	24	21

are larger when the switch occurs from higher to lower quality bitstream. For example, for the "news" sequence, the average size of the S-frame is 3.4 times larger than that of the SP-frame when $QP_1 = 25$ and $QP_2 = 19$ (Table I) and 6.2 times larger when $\mathrm{QP}_1 = 19$ and $\mathrm{QP}_2 = 25$ (Table II). Similar results can be observed for other test conditions [12].

In the following, we measure PSNR and bit rate for a number of bitstreams, each one generated by switching four times between two different quality bitstreams representing the same sequence. Switching occurs every other second and the first one takes place from the higher to lower quality bitstream. S-, I-, and SP-frames are used to facilitate the switching. In the case of the SP-frame (I-frame) approach, one frame every second is encoded as an SP-frame (I-frame). From the results in Table III, it can be noted that the total number of bits when using SP-frames is considerably smaller than for the S-frame and I-frame approaches. For example, for the "container" sequence approximately 1.8 times more bits are required when using S-frames while the PSNR of the SP-frame approach is still slightly higher.

TABLE IV
PSNR AND TOTAL BITS OVER 100 FRAMES WHEN THERE IS NO SWITCHING FOR I-, S-, AND SP-FRAMES APPROACHES

Sequence (QP)	Periodic I-frames		S-frames		SP-frames		% Differ. Btw.SP and I-frames	% Differ. Btw.SP and S-frames
	PSNR	Total Bits	PSNR	Total Bits	PSNR	Total Bits		
Container (25)	30.82	183651	30.59	91655	30.41	109566	67	-16
Container (19)	34.78	371338	34.57	208643	34.53	232508	60	-10
News (25)	30.88	305617	30.75	199771	30.47	217500	41	-8
News (19)	35.23	578068	35.14	408082	34.9	432388	34	-6
Foreman (22)	32.73	562057	32.74	476294	32.54	507101	11	-6
Foreman (16)	36.96	1132016	36.94	973322	36.85	1020080	11	-5
Hall (19)	35.65	436376	35.69	264648	35.5	289364	51	-8
Hall (13)	39.91	953771	39.96	676407	39.91	711762	34	-5

Similarly, the I-frame method requires 1.3 times more bits than the SP-frame one.

In Table IV, we further illustrate the performance of each scheme when there is no switching between bitstreams. It can be seen that in this case the performance of the I-frame approach is significantly lower than that of the S- and SP-frame methods. The S-frame approach has the best coding efficiency that is equal to the P-frames performance. The difference between the S-frame and SP-frame coding efficiency is however quite small. Furthermore, the SP-frame efficiency can be improved by using smaller values of SPQP, as discussed earlier, but only at the expense of increasing the secondary SP-frame sizes, which would approach the S-frame sizes. Nevertheless, even in this case, SP-frames will provide drift-free switching.

V. SUMMARY

We have described two new frame types defined in H.264/AVC. These frame types, called SP- and SI-frames, can be used to provide functionalities such as bitstream switching, splicing, random access, error recovery, and error resiliency.

We have presented the decoding process for primary SP-frames (frames placed within a single bitstream) and secondary SP- and SI-frames (frames used when switching from one bitstream to another). Usage of different quantization parameter values for the predicted block coefficients (SPQP) and for the prediction error coefficients (PQP) allows to introduce a tradeoff between coding efficiency of primary and secondary SP-frames. The lower the value of SPQP with respect to PQP, the higher the coding efficiency of a primary SP-frame, while on the other hand, the larger number of bits is required when switching to this frame.

Later, we showed how secondary SP- and SI-frames can be encoded such that identical frames can be reconstructed even when different reference frames are used for their prediction.

We have compared coding efficiency of SP-frames against I- and P-frames. SP-frames have significantly better coding efficiency than I-frames while providing similar functionalities. Finally, we have shown the resulting performances of different schemes that are being used for switching between bitstreams. The S-frame approach introduces drift, which becomes significant when there are multiple switches between bitstreams. SP-frames, on the other hand, provide drift-free switching between bitstreams and their sizes are considerably smaller than that of S-frames. We have also included results of the periodic intra-coding approach. It is again noted that SP-frames provide better PSNR versus bit-rate performance.

REFERENCES

[1] T. Wiegand and G. Sullivan, "Study of final committee draft of joint video specification (ITU-T rec. H.264/ISO/IEC 14 496-10 AVC)," in 6th Meeting, Awaji, JP, Island, Dec. 5–13, 2002, Doc. JVT-G050d2, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6).

[2] M. Karczewicz and R. Kurceren, "A Proposal for SP-Frames," in ITU-T Video Coding Experts Group Meeting, Eibsee, Germany, Jan. 09–12, 2001, Doc. VCEG-L-27.

[3] ——, "Improved SP-Frame Encoding," in ITU-T Video Coding Experts Group Meeting, Austin, TX, Apr. 02–04, 2001, Doc. VCEG-M-73.

[4] R. Kurceren and M. Karczewicz, "New Macroblock Modes for SP-Frames," in *ITU-T Video Coding Experts Group Meeting*, Pattaya, Thailand, Dec. 4–6, 2001, Doc. VCEG-O-47.

[5] M. Hannuksela, "Prediction From Temporally Subsequent Pictures," in *ITU-T Video Coding Experts Group Meeting*, Portland, OR, Aug. 22–25, 2000, Doc. VCEG-K-38.

[6] R. Kurceren and M. Karczewicz, "SP-Frame demonstrations," in *ITU-T Video Coding Experts Group Meeting*, Santa Barbara, CA, Sept. 24–27, 2001, Doc. VCEG-N-42.

[7] S. Wenger and G. Côté, Intra-macroblock refresh in packet (picture) lossy scenarios, Whistler, BC, Canada, July 21–24, 1998, Doc. Q15-E-15.

[8] G. Côté, S. Wenger, and M. Gallant, Intra-macroblock refresh in packet (picture) lossy scenarios, Whistler, BC, Canada, July 21–24, 1998, Doc. Q15-E-37.

[9] S. Wenger, H.26L error resilience experiments: First results, Osaka, Japan, May 16–18, 2000, Doc. Q15-J-53.

[10] T. Stockhammer, G. Liebl, T. Oelbaum, T. Wiegand, and D. Marpe, "H.26L Simulation Results for Common Test Conditions for RTP/IP Over 3GPP/3GPP2," in *ITU-T Video Coding Experts Group Meeting*, Santa Barbara, CA, Sept. 24–27, 2001, Doc. VCEG-N-38.

[11] N. Farber and B. Girod, "Robust H.263 compatible video transmission for mobile access to video servers," in *Proc. Int. Conf. Image Processing, ICIP'97*, Santa Barbara, CA, Oct. 1997.

[12] R. Kurceren and M. Karczewicz, "Further Results for SP-Frames," in ITU-T Video Coding Experts Group Meeting, Austin, TX, Apr. 02–04, 2001, Doc. VCEG-M-38.

[13] TML 8.7 Software [Online]. Available: ftp://standard.pictel.com/video-site/h26l/.

[14] S. Wenger, "Simulation Results for $H.263+$ Error Resilience Modes K, R, N on the Internet," ITU-T, SG16, Question 15, doc. Q15-D-17, Apr. 7, 1998.

Marta Karczewicz received the M.S. degree in electrical engineering in 1994 and Dr.Technol. degree in 1997 from Tampere University of Technology (TUT), Tampere, Finland.

During 1994–1996, she was a Researcher in the Signal Processing Laboratory of TUT. Since 1996, she has been with the Visual Communication Laboratory, Nokia Research Center, Irving, TX, where she is currently a Senior Research Manager. Her research interests include image compression, communication, and computer graphics.

Ragip Kurceren (S'98–M'01) received the M.S. and Ph.D. degrees in electrical, computer, and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1996 and 2001, respectively.

From 1995 to 2000, he was a Research Assistant with the Center for Image Processing Research, Rensselaer Polytechnic Institute. Since July 2000, he has been with Nokia Research Center, Irving, TX. His research interests include digital image and video processing, including compression and transmission, and multimedia adaptation.

Dr. Kurceren is a co-recipient of the Best Paper Award from the IEEE Packet Video Workshop 2000.