Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension

C. Andrew Segall, *Member, IEEE*, and Gary J. Sullivan, *Fellow, IEEE*

(Invited Paper)

***Abstract*—A scalable extension to the H.264/AVC video coding standard has been developed within the Joint Video Team (JVT), a joint organization of the ITU-T Video Coding Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The extension allows multiple resolutions of an image sequence to be contained in a single bit stream. In this paper, we introduce the spatially scalable extension within the resulting Scalable Video Coding standard. The high-level design is described and individual coding tools are explained. Additionally, encoder issues are identified. Finally, the performance of the design is reported.**

***Index Terms*—H.264/AVC, Scalable Video Coding (SVC), spatial scalability.**

I. Introduction

WITH the expectation that future applications will support a diverse range of display resolutions and transmission channel capacities, the Joint Video Team (JVT) has developed a scalable extension [1], [2] to the state-of-the-art H.264/AVC video coding standard [3]–[6]. This extension is commonly known as Scalable Video Coding (SVC) and it provides support for multiple display resolutions within a single compressed bit stream (or in hierarchically related bit streams), which is referred to here as *spatial scalability*. Additionally, the SVC extensions support combinations of *temporal scalability* (frame rate enhancement) and *quality scalability* (fidelity enhancement for pictures of the same resolution) with the spatial scalability feature [2]. This is achieved while balancing both decoder complexity and coding efficiency.

The resolution diversity of current display devices motivates the need for spatial scalability. Specifically, larger format, high definition displays are becoming common in consumer applications, with displays containing over two million pixels readily available. By contrast, lower resolution displays with between ten thousand and one hundred thousand pixels are also popular in applications constrained by size, power and weight. Unfortunately, transmitting a single representation of a video sequence to the range of display resolutions available in the market is impractical. For example, it is rarely justifiable to design a device with low display resolution with the capacity for decoding and down-sampling high-resolution video material. Such a requirement could increase the cost and power of the device to the point of exceeding the very constraints that determined its display resolution. In addition, sending the high-resolution details that are ultimately not shown on the display for such a device is a waste of its receiving channel bit rate.

Diverse, limited, and time-varying channel capacity provides a second motivation for spatial scalability. Here, the concern is that channel capacity may preclude the reliable transmission of high-resolution video to specific devices or at specific time instances. Spatial scalability allows for the rapid bit rate adaptation that can be a necessity in such scenarios. This bit rate adaptation is achieved without transcoding operations or feedback to a complex real-time encoding process, both of which can introduce unacceptable complexity and delay.

The purpose of this paper is to discuss key concepts of spatial scalability within the SVC extension. This project is the fourth in a historical series of efforts to standardize spatially SVC schemes (after prior efforts in MPEG-2 [7], [8], H.263 Annex O [9], and MPEG-4 part 2 [10]), although the prior designs were basically not successful in terms of industry adoption. This paper points out several ways in which the new design addresses the problems of those prior approaches.

The rest of this paper is organized as follows. Section II provides an overview of H.264/AVC spatially scalable coding and compares it to alternative scalable approaches. In Section III, the specific coding tools within the spatial SVC design are described. In Section IV, encoder issues related to spatial SVC are considered. In Section V, the performance of the spatial SVC extension is presented. Finally, conclusions are provided in Section VI.

II. Overview

The SVC extension of H.264/AVC provides a mechanism for reusing an encoded lower resolution version of an image sequence for the coding of a corresponding higher resolution sequence. This is shown in Fig. 1, where a diagram of a hypothetical SVC encoder is provided. Subsequent sections discuss the specific tools introduced in the SVC extension. However, to better aid in the understanding of the SVC design, this section focuses on higher level concepts. We begin by identifying basic concepts and definitions necessary for discussion of the SVC design. Then, we consider the high level spatial relationship between resolutions in a bit stream. Finally, we summarize

Manuscript received January 7, 2007; revised July 24, 2007. This paper was recommended by Guest Editor T. Wiegand.

C. A. Segall is with Sharp Laboratories of America, Camas, WA 98607 USA (e-mail: asegall@sharplabs.com).

G. J. Sullivan is with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: garysull@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2007.906824

1051-8215/$25.00 © 2007 IEEE

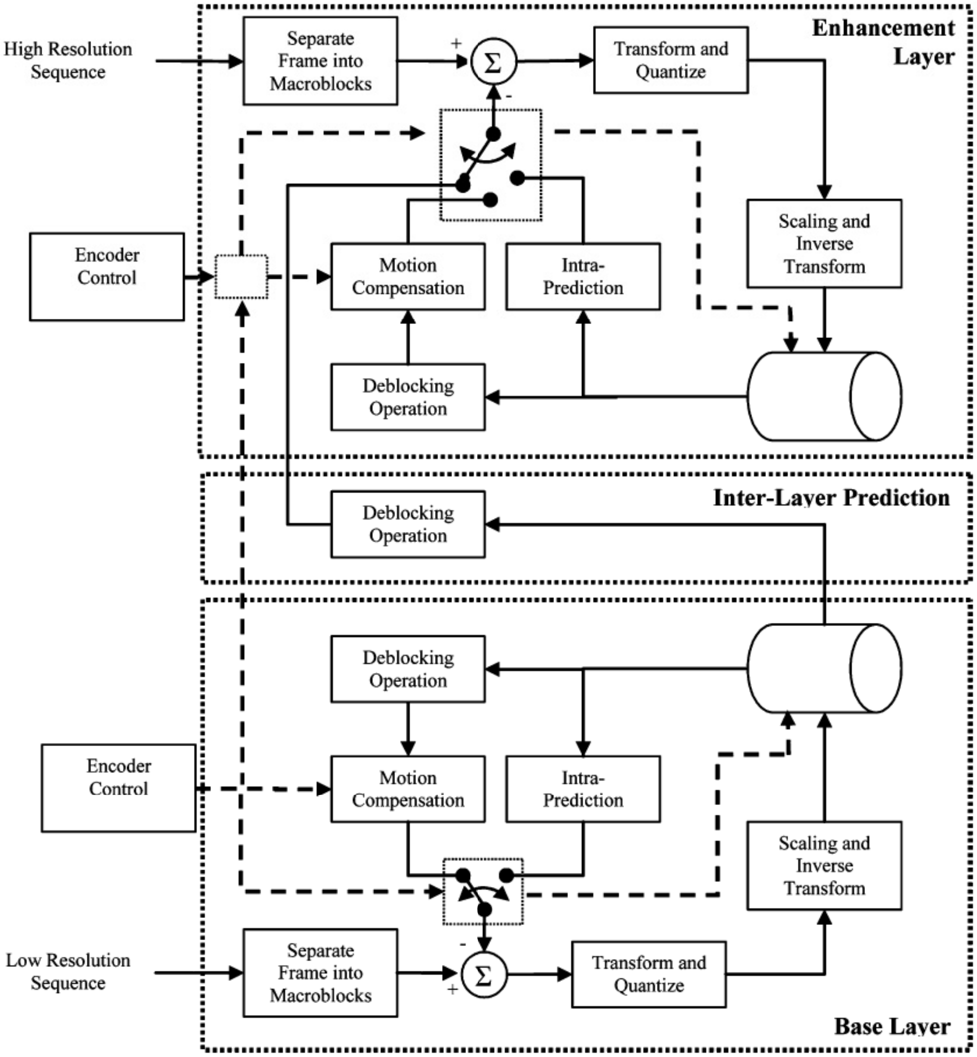


Fig. 1. High-level diagram of spatial scalability in the SVC design. The "base layer" encoder takes a lower resolution video sequence as input and encodes it with the H.264/AVC video coding standard while conforming to a legacy profile. The enhancement layer encoder takes a higher resolution sequence as input. The higher resolution sequence can be encoded with ordinary H.264/AVC technologies. Moreover, inter-layer prediction can be used to provide additional coding choices. For the case of intra-picture coded blocks in the base layer, reconstructed intensities provide a prediction for the enhancement layer. For the case of inter-picture coded blocks in the base layer, enhancement layer motion vectors and residual difference information can be predicted from the base layer. Further resolution layers can be added in an analogous fashion and can utilize either the base layer or previously transmitted enhancement layers for inter-layer prediction. Moreover, other forms of SVC (temporal or quality) enhancement may also be present.

two key design concepts in the SVC extension—image pyramids and single-loop decoding.

A. Basic Concepts

The basic mission of a scalable design is two-fold: 1) to minimize the coding efficiency loss relative to single-layer coding; and 2) to minimize the complexity increase (especially for decoders) relative to single-layer coding. By *single-layer coding*, we refer to the coding of a video sequence without providing the scalability functionality. Unless a result with coding efficiency significantly superior to a simulcast solution can be obtained, a scalable solution with any complexity penalty is useless. By *simulcast*, we refer to the coding of both source video sequences of a scalable scenario as entirely separate single-layer bit streams and transmitting them using the sum of the two bit rates. The challenge here is considerable—among the three typical basic forms of bit stream scalability, i.e., spatial, temporal, and quality, the spatial form seems to be the most difficult in which to achieve significant superiority to a simulcast solution. One dominant reason for this is the focus of the JVT on supporting lower resolution versions of image sequences with high visual quality, as opposed to lower resolution representations that provide high coding efficiency.

The lowest resolution video data in a spatially scalable system is sometimes referred to as the *base layer* (especially when it is decodable by an ordinary nonscalable single-layer decoder), and the higher resolution video data is often referred to as the *enhancement layer*. Processes that determine or predict the value of enhancement layer data from previously reconstructed data of a lower resolution layer at the same time instance are referred to as *inter-layer* prediction processes, and the source for the prediction is referred to as the *reference layer*. Other forms of prediction include *inter-picture* prediction, involving prediction operating temporally between different pictures of the same resolution layer, and *intra-picture* prediction, involving prediction operating spatially within the same picture of one particular resolution layer.

From a video coding specification perspective, the set of data comprising a SVC representation is treated as a single bit stream. However, from a systems multiplex or file storage perspective, the data might often be handled differently—as distinct hierarchically related streams of content that are coordinated using decoding timestamps or other such mechanisms. In this fashion, a system can ease the handling of the data, such as enabling channel bit rate adaptation or ensuring that legacy decoders that do not support scalability are presented with only the base layer for decoding.

Some degree of familiarity with the concepts of the original H.264/AVC standard is assumed in the presentation provided herein, such as the concepts of macroblocks, motion partitions, biprediction, inter-picture prediction using multiple reference pictures, and reference picture lists. Readers unfamiliar with this background information may benefit from referring to [3]–[6]. Moreover, one topic that is somewhat neglected in this presentation is that of interlaced-scan video content. Herein the principles of the spatial SVC design are explained under the assumption of frame-structured progressive-scan pictures, so that the concepts can be described without the need to consider the details of the handling of interlaced fields and frames. The application of these SVC concepts to interlaced video is straightforward for those familiar with interlaced video coding using H.264/AVC. For further information about interlaced video support in the SVC context, the reader is referred to [11]. The overview of SVC in general that is found in [2] will also be of interest to many readers.

An additional simplification used in much of the discussion for this overview paper is to primarily consider a bit stream containing only two layers—a lower resolution base layer and a higher resolution spatial scalability enhancement layer. In fact, the SVC design fully supports multilayer scenarios including multiple spatial scalability layers and the mixing of spatial scalability layers with other layers that provide temporal or quality scalability. Considerable flexibility is also provided in regard to the selection of the reference layer for each enhancement layer, such that a bit stream can contain branching dependency structures.

As with prior international standards for video coding (scalable and nonscalable), the scope of the standard is limited to specifying the decoding process and the format of the syntax. Encoder designers are free to use any encoding algorithms they wish, so long as the bit stream they produce conforms to the format specification. Any kind of preprocessing is also allowed prior to encoding, and decoding devices are allowed to contain any sort of post-processing, error and loss concealment techniques, and display-related customization.

B. Inter-Layer Spatial Relationships and Profile Constraints

An important feature of the SVC design, from a high-level functionality perspective, is the ability for the lower resolution and higher resolution pictures in a spatially scalable bit stream to represent different regions of a video scene. For example, a system may transmit standard definition television content, with a picture aspect ratio of 4:3, as a base layer and high definition television content, with a picture aspect ratio of 16:9, in a higher resolution enhancement layer. Such a use case requires cropping and offsetting the origins of the picture regions in addition to scaling, as the lower resolution layer signal may not represent the entire extent of the higher resolution sequence (and vice versa). The common "pan and scan" technique used on standard-definition DVDs for converting wide screen data for display on a 4:3 display is an example of a more limited form of such display adaptation. The SVC extension supports such capability in a flexible but straightforward manner. Relative positioning and windowing parameters are provided in picture-level syntax structures, so that flexible cropping, scaling, and alignment relationships can not only be supported but may be varied on a picture-by-picture basis.

However, such flexibility can be constrained to simplify the use cases for particular applications. In particular, the SVC extension includes the definition of three *profiles* of the design. These are the "Scalable Baseline" profile, the "Scalable High" profile, and the "Scalable High Intra" profile. While the latter two profiles support full spatial SVC flexibility, the Scalable Baseline profile imposes the following constraints to enable simplified application scenarios.

- The width and height of the scaled regions of lower resolution and higher resolution pictures must have the same scaling ratio, and this ratio can only have the value 1.5 or 2.
- The spatial offsets specifying the relative location of the upper left corner of the lower and higher resolution picture regions must be multiples of 16 both horizontally and vertically (i.e., they must be in units of macroblocks).

The case using a scaling ratio of 2 with spatial offset constraints as noted above is often referred to as *dyadic* spatial scalability, whereas the more general case is known as *extended* spatial scalability [12].

C. Image Pyramids and Related Coarse-to-Fine Hierarchies

Image pyramids describe a relationship between lower resolution and higher resolution versions of an image.[1] This relationship is found in a variety of image and video processing scenarios, and image pyramids have been incorporated into a variety of applications, e.g., [13]–[19], as well as previous scalable efforts in the video coding standards [7]–[10]. In the SVC extension, a coarse-to-fine hierarchy of images is also used for spatial scalability. The original high-resolution image sequence is converted to lower resolutions by filtering and decimating. Then, the sequence of pictures at the lowest of these resolutions is coded in a manner such that it can be decoded independently. Each higher resolution video sequence is coded relative to a decoded lower resolution sequence.

[1] The terms *picture* and *image* are used interchangeably herein.

The use of an image pyramid for video coding does not come without penalties. Specifically, an image pyramid is an overcomplete decomposition. In other words, the number of image samples in the entire pyramid structure is larger than the number of samples in an original high-resolution image. This is in contrast to embedded representations that use critically sampled decompositions. For example, wavelet decompositions are well known to provide inherent scalability and viable image coding designs [20]–[24]. In the development of the SVC extension, such critically sampled decompositions were also considered [25]–[28]. However, the aliasing introduced by these decompositions, while suitable for still image coding, were deemed problematic for video. Specifically, the aliasing can make effective motion compensated inter-picture prediction more difficult, as well as lead to objectionable temporal artifacts. Additionally, the wavelet design may be likely to require more computational resources than the traditional block-based coding approach.

The decision to use an image pyramid in the SVC project provides flexibility for encoder and application designers. The down-sampling operation is not defined in the standard, so that encoder designers are free to employ the down-sampler that they consider most suitable. For example, applications that are sensitive to encoder hardware costs would select a down-sampler with minimum complexity for the specific implementation architecture. Alternatively, in other applications, additional computational complexity may be acceptable in order to achieve a higher quality result. These applications would choose a more sophisticated, and likely more complex, down-sampling method.

D. Single-Loop Decoding Concept

The concept of image pyramids describes the relationship between images of different resolution. However, image pyramids do not capture the evolution of that relationship between compressed images through time in sequences of such images. To understand this relationship, we need to consider the concepts of *multiloop* and *single-loop* decoding. Single-loop decoding, also called *constrained inter-layer prediction* [29]–[32], is a fundamental property of the new SVC design, and it is described in the remainder of the section.

In the family of ITU-T and ISO/IEC video coding standards, which includes H.264/AVC, block-wise motion compensated inter-picture prediction plays a critical role in improving coding efficiency. This is accomplished by transmitting (or having the decoder infer) one or more motion vectors to predict a block in the current picture from the content of previously decoded reference pictures. Then, additional information about the residual difference between the prediction and the actual image data may be sent. For natural image sequences, which often contain slowly evolving features, the motion compensation process exploits the inherent characteristics of the image sequence.

In designing the SVC extension of the H.264/AVC standard, a fundamental question was how to use the motion compensation process within the context of spatial scalability. One potential approach (used in all previous standardized designs) would be to perform multiloop decoding. In this scenario, each low-resolution picture is completely decoded, including low-resolution motion-compensation prediction operations in particular. Then, the coarse-to-fine relationship of the image pyramid is used to predict the lower frequency components of a higher resolution enhancement-layer picture using up-sampling of the decoded lower resolution picture. Additionally, motion compensated inter-picture prediction is performed again at the enhancement layer. This predicts the high frequency components of the enhancement layer.

Using a decoder with multiple motion compensation loops does improve the coding efficiency of a scalable video codec, but the benefit in coding efficiency turns out to be minimal when all available coded data is used effectively in other ways [29]–[32]. Moreover, the multiloop decoding scheme increases decoding complexity. Motion compensation is performed at each resolution and the reconstructed pictures of all levels of the pyramid are stored for each time instant. This becomes problematic in practice, as motion compensation requires high memory bandwidths for many processing architectures [33], and the extra decoding processes involved in multiloop decoding add undesirable sequential dependencies to the decoding process as well as require extra encoder and decoder implementation and debugging efforts.

In the SVC design, a lower complexity approach is adopted. Motion compensation is performed only at the target decoded resolution (e.g., the displayed resolution). Thus, the decoding structure of the SVC design is referred to as a single-loop design, which simply means that only the operation of a single motion compensation loop is necessary to reconstruct the image sequence for any resolution layer. This provides an important feature, as it reduces the complexity of motion compensation to that of a single-layer decoder—eliminating the major source of complexity penalty in prior SVC designs. As will be seen in the next section, good coding efficiency can still be achieved without requiring multiloop decoding, by effectively propagating the information found in the coded motion vectors, mode information and residual difference data from each lower resolution layer to each next higher resolution layer. This propagation employs the previously described image pyramid concept.

To further ease implementation, the syntax of the SVC extension has been designed in a way that it allows the separate parsing of each layer of the syntax (without parsing other layers and without operating the decoding processes of lower layers) [34], [35]. Completing the full decoding process, of course, requires further processing of some parsed data of each layer up to the target decoded layer (but not full multilayer decoding, due to the single-loop nature of the design).

III. Coding Tools

SVC introduces several design features to enable spatial scalability. These tools include the calculation of corresponding positions in different resolution layers, methods for inter-layer prediction of various data such as macroblock prediction modes and motion vectors, an "I_BL" macroblock type that uses inter-layer up-sampled image prediction, and a residual difference signal prediction technique that uses inter-layer up-sampled residual difference prediction. These tools are provided in addition to the original single-layer coding tools, such as (spatial) intra-picture and (temporal) inter-picture coding techniques, and an encoder must determine when each tool is most appropriate. Describing the new tools and how they are combined effectively with the original H.264/AVC single-layer design features is the focus of this section.

A. *Calculation of Corresponding Spatial Positions*

The first design feature that we will discuss in detail is the calculation of corresponding positions in adjacent levels of the pyramid hierarchy. This concept is used in several ways in the spatial SVC extension.

Identifying sample locations in a lower resolution layer that correspond to sample locations in the enhancement layer is performed at fractional-sample accuracy. Specifically, sample positions are calculated to 1/16th sample position increments and derived using fixed-point operations as

$$\begin{aligned} B_x &= \text{Round}\left(\frac{E_x * D_x + R_x}{2^{S-4}}\right) \\ B_y &= \text{Round}\left(\frac{E_y * D_y + R_y}{2^{S-4}}\right) \end{aligned} \tag{1}$$

where B_x and B_y are, respectively, the horizontal and vertical sample coordinates in the lower resolution (e.g., base layer) picture array, E_x and E_y are horizontal and vertical sample coordinates in the high-resolution (enhancement-layer) picture array, and R_x and R_y are higher precision ($1/2^S$ sample position) reference offset locations for grid reference position alignment, and D_x and D_y are scaled inverses of the horizontal and vertical resampling ratios. D_x and D_y are specified as

$$\begin{aligned} D_x &= \text{Round}\left(\frac{2^S * \text{BaseWidth}}{\text{ScaledBaseWidth}}\right) \\ D_x &= \text{Round}\left(\frac{2^S * \text{BaseHeight}}{\text{ScaledBaseHeight}}\right) \end{aligned} \tag{2}$$

where ***BaseWidth*** and ***BaseHeight*** denote the width and height of the rectangular region of the lower resolution picture array to be up-sampled, respectively, and ***ScaledBaseWidth*** and ***ScaledBaseHeight*** denote the width and height of the corresponding region of the up-sampled lower resolution picture array, respectively. The precision control parameter S has been chosen to trade off between precision and ease of computation; S is specified to be 16 for most uses to enable the use of 16-bit word-length arithmetic, and to be a somewhat larger number optimized for 32-bit arithmetic for enhanced-capability decoders that support very large picture sizes. The basic design of these formulas was proposed in [36], and some later refinements were subsequently applied. The formulas are designed for computational simplicity as follows.

- The above formulas are specified for implementation using two's complement integer operations, most of which require at most 16 bits of dynamic range (for example, noting that ***BaseWidth*** is always less than ***ScaledBaseWidth***, D_x requires no more than S bits).
- Multiplication and division scale factors that are powers of two are specified to be performed using left and right binary arithmetic shifts.
- Rounding of a ratio is accomplished by adding half of the value of the denominator prior to right shifting.
- The D_x and D_y computations only need to be performed once, with the results reused repeatedly for computations of B_x and B_y for the entire image (or sequence of video images).
- When moving from position to position from right to left or top to bottom in computing B_x and B_y for a series of values of E_x and E_y, a multiplication operation can be converted to an addition so that computation of each B_x and B_y can be performed incrementally, requiring only one addition and one right shift operation to obtain the result of each formula.

This design supports essentially arbitrary resizing ratios (except in constrained applications using the Scalable Baseline profile), and the position calculation equations have low complexity regardless of the ratio, in contrast to some prior standardized designs in which only relatively simple rational ratios were practical due to the way the position calculations were specified.

B. *Coarse-to-Fine Projection of Macroblock Modes, Motion Partitioning, Reference Picture Indices, and Motion Vectors*

In the enhancement layer syntax for areas of the enhancement layer that correspond to areas within the lower resolution picture, a flag, called the base mode flag, can be sent for each nonskipped macroblock[2] to determine whether the macroblock mode, motion segmentation, reference picture indices, and motion vectors are to be inferred from the data at corresponding positions in the lower resolution layer. The basic concepts of this inference process were proposed for use with dyadic spatial scalability in [37] and were extended to arbitrary spatial scalability relationships in [38]–[40]. In some sense the projection consists of first projecting the sample grid of the finer level to the coarser level of the pyramid and then using this projection to propagate data from the coarser level to the finer level.

When the base mode flag is equal to 0, the macroblock prediction mode is sent within the enhancement layer macroblock-level syntax. Then, within each motion partition,[3] a flag can be sent for each reference picture list, called the motion prediction flag, to determine whether reference picture indexes will be sent in the enhancement layer or not and whether the motion vectors are to be predicted within the enhancement layer or using inter-layer prediction from the lower resolution layer motion data.

When the base mode flag is equal to 1, since the finest granularity of H.264/AVC coding decisions is at the 4 × 4 level, the inference process is performed based on 4 × 4 luma block structures. For each 4 × 4 luma block, the process begins by identifying a corresponding block in the lower resolution layer. Numbering the samples of the luma block from 0 to 3 both horizontally and vertically, the luma sample at position (1,1) is used to determine the block's associated data. A corresponding sample in the lower resolution layer for this sample is identified in a similar manner as described in Section III-A, but

[2] To save the need to repeatedly send the base mode flag in cases when an encoder will not vary its value in applicable macroblocks, a default value for the flag can alternatively be sent at the slice header level.

[3] To save the need to repeatedly send the motion prediction flag in cases when an encoder will not vary its value in applicable macroblocks, a default value for the flag can alternatively be sent at the slice header level.

with nearest sample precision instead of 1/16th sample precision. The prediction type (intra-picture, inter-picture predictive, or inter-picture bipredictive), reference picture indices, and motion vectors associated with the prediction block containing the corresponding lower resolution layer position are then assigned to the 4×4 enhancement layer block. Motion vectors are scaled by the resampling ratio and offset by any relative picture grid spatial offset so that they become relevant to the enhancement layer picture coordinates. Then a merging process takes place to determine the final mode and motion segmentation in the enhancement layer macroblock.

If all 4×4 luma blocks of the enhancement macroblock correspond to intra-picture coded lower resolution layer blocks, the inferred macroblock type is considered to be "I_BL," a macroblock type that is described in the following section; otherwise, motion segmentation, reference picture indices, and motion vectors then need to be inferred. (It should be noted that because the prediction mode is determined from only one position in each 4×4 block, it is possible that a few samples in enhancement layer I_BL macroblock may have corresponding locations in the lower resolution layer picture that lie in inter-picture predicted regions of the lower resolution layer.)

In H.264/AVC, reference picture indexes have an 8×8 luma granularity. To achieve this granularity, for each 8×8 luma region of the enhancement layer, the reference picture index is set to the minimum of the reference picture indexes inferred from the corresponding constituent 4×4 blocks when performing inter-layer motion prediction [38]–[40]. When some lower resolution layer blocks are in a B-slice, the minimum is computed separately for each of the two reference picture lists and biprediction is inferred if both lists were used in the set of 4×4 blocks. For 4×4 regions that did not use a selected reference picture index (or indices, in the case of biprediction), the motion vector is set to that of a neighboring block that did (so that some motion vector value is assigned that is relevant to the selected reference picture index).

Then the values of motion vectors are inspected to determine the final motion partitioning of the enhancement layer macroblock (4×4, 4×8, 8×8, 8×16, 16×18, or 16×16). Partitions with identical reference picture indexes and similar or identical motion vectors are merged to make the final predicted motion more coherent and reduce the complexity of the associated inter-picture prediction processing [41].

The result is predicted mode and motion data that fits with the same basic structure of ordinary single-layer H.264/AVC prediction.

C. I_BL Macroblock Type and Inter-Layer Texture Prediction

The "I_BL" macroblock type provides an additional prediction source for the scalable enhancement layer. When the use of I_BL is inferred as described above, the decoder performs the following steps. First, it decodes the identified 4×4 co-located blocks in the lower resolution layer and applies a deblocking filter. Next, it up-samples the decoded samples of the lower resolution layer to form a prediction. Then, when indicated by the encoder, it receives residual difference information in the enhancement bit stream and adds this residual difference data to the prediction. Finally, a deblocking filter is applied to the resulting picture.

It is important to understand that I_BL macroblocks cannot occur at arbitrary locations in the enhancement layer. Instead, the I_BL macroblock type is only available when the lower resolution layer is decodable without motion compensation.[4] This is due to the single-loop design of the SVC extension. For the case of *dyadic* scalability, where the lower resolution and higher resolution sequences differ by a factor of two in both dimensions, this is equivalent to requiring the 8×8 submacroblock region in the lower resolution picture that corresponds to the 16×16 macroblock region of the enhancement-layer to be in an intra-picture coded macroblock of the lower resolution layer. For nondyadic scalability though, it is possible for regions that correspond to the samples in the enhancement-layer macroblock to span multiple macroblocks in the lower resolution layer, and, as previously mentioned, to even contain a few samples that were coded using inter-picture prediction. In such a case, the I_BL macroblock type is allowed and the up-sampling of the decoded sample values of the lower resolution layer uses extrapolating repetition of the samples within the intra-picture coded region.

To allow the use of the I_BL macroblock type without requiring the decoding of inter-picture predicted regions of the lower resolution layer, the lower resolution layer must be coded using is the H.264/AVC feature known as ***constrained intra-picture prediction***, which makes the decoding process of the intra-picture coded regions independent of the content of neighboring inter-picture coded regions. (However, the decoding of some neighboring intra-picture coded regions of the lower resolution reference layer may be necessary in order to decode the blocks on which the enhancement-layer macroblock depends, as the corresponding lower resolution region may use intra-picture prediction from such neighboring regions. The number of neighboring intra-picture coded regions that are needed to reconstruct the enhancement-layer macroblock is restricted as a profile constraint.)

After the reference blocks in the lower resolution layer are identified, their sample intensities are reconstructed (including the application of a deblocking operation) and up-sampled to the higher resolution grid. In the SVC design, the up-sampling operation for the luma component consists of applying a separable four-tap poly-phase interpolation filter. The numerator tap values for the filter are provided in Table I, and a rounding right shift of five positions is performed to normalize the result. The magnitude response of the poly-phase filter is presented in Fig. 2. A significant amount of study was conducted to design the filter for good performance with minimal complexity [42] and to determine whether supporting additional alternative filters would be beneficial. However, no clear need for other such filters was identified [43]. The chroma component of the signal is also up-sampled but with a different (simpler) interpolating kernel. This filter is shown in Table II (normalized in the same

[4] It is also possible to restrict the location of I_BL macroblocks to be constrained by the slice boundaries of the reference layer. This restriction is indicated by syntax signaled by the encoder at the sequence level. This can enable better support of parallel processing for encoding and decoding and can also be useful for a distributed encoding functionality known as continuous-presence multipoint.

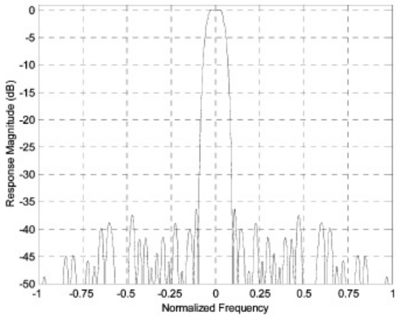


Fig. 2. Frequency response of the poly-phase filter for luma up-sampling. The filter is used for up-sampling reconstructed luma values from a reference layer to an enhancement layer for an I_BL macroblock.

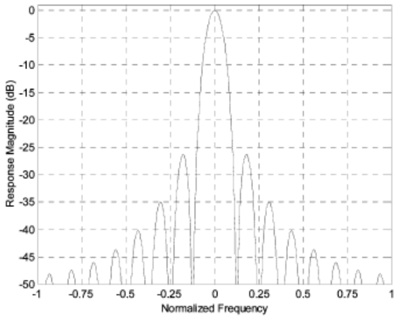


Fig. 3. Frequency response of the poly-phase filter for chroma upsampling. The filter is used for up-sampling reconstructed chroma values from a reference layer to an enhancement layer for an I_BL macroblock.

TABLE I
INTERPOLATION FILTER FOR LUMA UP-SAMPLING

phase index	interpolation filter coefficients			
	e[-1]	e[0]	e[1]	e[2]
0	0	32	0	0
1	-1	32	2	-1
2	-2	31	4	-1
3	-3	30	6	-1
4	-3	28	8	-1
5	-4	26	11	-1
6	-4	24	14	-2
7	-3	22	16	-3
8	-3	19	19	-3
9	-3	16	22	-3
10	-2	14	24	-4
11	-1	11	26	-4
12	-1	8	28	-3
13	-1	6	30	-3
14	-1	4	31	-2
15	-1	2	32	-1

The I_BL mode projects reconstructed sample values in the lower-resolution layer to a higher-resolution grid with a normative up-sampling process. The up-sampling filter is a separable poly-phase filter with four taps per phase horizontally and vertically.

TABLE II
INTERPOLATION FILTER FOR CHROMA UP-SAMPLING

phase index	interpolation filter coefficients			
	e[-1]	e[0]	e[1]	e[2]
0	0	32	0	0
1	0	30	2	0
2	0	28	4	0
3	0	26	6	0
4	0	24	8	0
5	0	22	10	0
6	0	20	12	0
7	0	18	14	0
8	0	16	16	0
9	0	14	18	0
10	0	12	20	0
11	0	10	22	0
12	0	8	24	0
13	0	6	26	0
14	0	4	28	0
15	0	2	30	0

The I_BL mode projects reconstructed chroma sample values in the lower-resolution layer to a higher-resolution grid with a normative up-sampling process. This up-sampling filter is bi-linear. This bi-linear filter is also used for up-sampling luma and chroma residual difference data (as described in section III.D). The use of different filters for luma texture, versus chroma texture and residual difference information was motivated by complexity considerations

manner as for Table I) and corresponds to bilinear interpolation. A frequency response plot is provided in Fig. 3. The use of different interpolation filters for luma and chroma is motivated by complexity considerations. In prior standardized designs, the up-sampling filtering quality was only bilinear for both luma and chroma, resulting in significantly lower luma prediction quality.

When applying the up-sampling operator, the decoder must select the appropriate phase from the poly-phase filter [44]. This requires a standardized procedure to ensure that the encoder decisions will properly affect the decoded picture results, and the method consists of applying the fixed point position calculation technique described in Section III-A above and then using the least-significant four bits (the fractional part of the result) to determine the phase selection index (left column of Tables I and II) while using the most-significant bits to determine which samples to filter (using the tap values in the row selected by the phase index).

One important issue to understand is that the up-sampling operation performed in the decoding process can have implications on the design of the down-sampler that is used for the generation of the source image pyramid by the encoder [44]. In particular, the phase characteristics of the encoder down-sampling filter operations should be designed to match the subsequent results of the decoding process. The phase of the low-pass filtering operation results in the creation of effective spatial locations for the down-sampled image samples, and then the phase of the up-sampling filter in the decoding process results in the creation of an effective spatial location for the up-sampled image samples. For a well-designed encoder, the picture sample predictions that result from that cascade of operations will be in

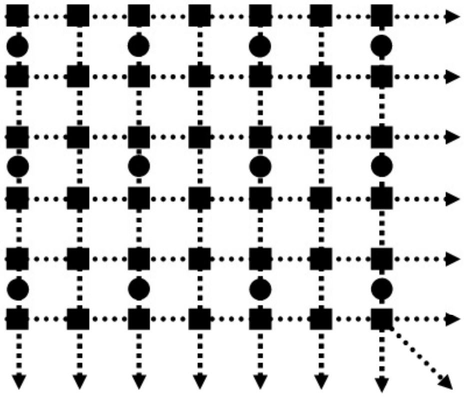


Fig. 4. Nominal locations of luma and chroma samples for an H.264/AVC frame picture. Luma sample positions are denoted with the symbol ■, and chroma sample positions are denoted with the symbol •. Note that the chroma samples are not vertically co-located with luma sample positions. Furthermore, H.264/AVC (Annex E) additionally supports the ability for encoders to indicate the use of a variety of alternative (different-than-nominal) alignments of luma and chroma 4:2:0 sampling grids. Handling the variety of possible relative alignments requires a flexible design for the texture and residual difference up-sampling operations in the decoding process.

the same positions as the original samples of the higher resolution picture. Using an inappropriate filter in the encoding process could result in reduced compression capability and visual artifacts.

The origin of the coordinate systems used in the up-sampling process for the luma samples is placed a half sample to the left of the left-most luma sample horizontally and a half sample above the top-most luma sample vertically. For the dyadic case, this convention results in having exactly the same area covered by a macroblock in an enhancement layer as is covered by an 8×8 submacroblock in the lower resolution layer [44]. This results in the up-sampling operation in the decoding process using alternating 1/4 and 3/4 phase offsets (phase indexes 4 and 12 in Tables I and II). For an encoder, this corresponds to using a half-phase filter [a symmetric finite impulse response (FIR) filter with an even number of taps] in the down-sampling process (both horizontally and vertically).

Relative chroma positioning requires further careful attention. The chroma components of a video picture sequence are typically represented at a resolution that is lower than that of the luma component. Practical applications typically down-sample the chroma planes by a factor of two in the horizontal direction and often also by a factor of two in the vertical direction. This process results in the well-known 4:2:2 and 4:2:0 formats, where 4:2:2 denotes a down-sampling factor of two horizontally, and 4:2:0 denotes a down-sampling factor of two in both dimensions.

The nominal H.264/AVC sample grid positions for the 4:2:0 chroma format are defined relative to the luma grid as shown in Fig. 4. Alternative sampling grid alignments can also be indicated by syntax supported in the standard. The positioning shown in Fig. 4 corresponds to generating the lower resolution chroma samples (starting with full-resolution 4:4:4 chroma sampling) by application of a zero-phase horizontal filter (a symmetric FIR filter with an odd number of taps) and a half-phase vertical filter (a symmetric FIR filter with an even number of taps). Maintaining the relative chroma positioning shown in Fig. 4 when constructing a lower resolution image of a dyadic image pyramid using a half-phase luma filter (as described above) requires using a quarter-phase chroma filter in the encoder horizontal down-sampling process and a half-phase chroma filter in the encoder vertical down-sampling process. Corresponding up-sampling processes alternate between the 1/4 and 3/4 phase positions both horizontally and vertically. Since alternative sampling grid alignments are also supported in the H.264/AVC standard, the SVC design provides syntax to allow an encoder to slightly shift the phase positioning of the chroma grid relative to the luma grid during the decoding process to make the design flexible and allow it to be customized to fit encoding characteristics.

Adjustments in the spatial correspondence formulas of Subsection III-A for chroma have also been specified to support the 4:4:4, 4:2:2, and 4:2:0 chroma sampling formats (with support of a variety of positioning alignments between the luma and chroma picture grid positions), although the current generation of defined SVC profiles supports only 4:2:0 sampling.

After up-sampling and (when indicated by the encoder) adding a additional inverse-transformed residual difference signal, a deblocking filter is applied in the I_BL decoding process that is similar to that of the ordinary H.264/AVC decoding process, but with altered boundary strength calculations. The modification of the filter strength is motivated by the fact that the lower resolution picture has also been deblocked prior to up-sampling [45].

To conclude this subsection, let us summarize the I_BL macroblock type. The mode operates by reconstructing lower resolution layer intensity values and then up-sampling the intensity information to predict high-resolution samples. The up-sampling operator employs a separable four-tap filter horizontally and vertically for luma and a separable two-tap filter for chroma, and it is designed for reduced complexity and proper handling of the chroma information. After up-sampling (when indicated by encoded syntax), the SVC decoder also receives additional residual difference information to refine the up-sampled prediction. This refinement process is identical to receiving residual difference data for other predicted modes. Finally, a deblocking filter is applied to the decoded result.

D. Inter-Layer Residual Prediction

The previous subsection describes the I_BL macroblock type, which uses prediction of higher resolution sample values from lower resolution sample values. This reflects a traditional image pyramid approach. However, as mentioned in the previous subsection, the single-loop design of the SVC extension restricts the I_BL macroblock type to be available only when the corresponding lower resolution layer area is intra-picture coded. For enhancement-layer regions that correspond to inter-picture coded regions in the reference layer, an alternative method of using the coded lower resolution signal is provided by the SVC design that does not require the reconstruction of the sample values of the lower resolution layer picture. This is the inter-layer *residual prediction* technique.

Residual prediction is activated by a flag, called the residual prediction flag, that can be sent in the macroblock level syntax[5] of the higher resolution layer for nonskipped macroblocks. Residual prediction can also be activated for applicable skipped macroblocks when adaptive changes of the base mode flag are disabled and the default value of the base mode flag is 1 (otherwise, skipped macroblocks of the enhancement layer will be decoded using ordinary skip-mode temporal inter-picture prediction within the enhancement layer). All combinations of the residual prediction flag with the base mode flag are allowed (although there are some restriction details—such as prohibiting residual prediction when the base mode flag is 1 and the inferred macroblock type does not use temporal inter-picture prediction). When residual prediction is performed, instead of fully decoding the picture samples of the lower resolution layer, the residual difference data of the lower resolution layer is decoded and up-sampled and added to the motion compensated prediction of the higher resolution layer (without using the inter-picture prediction signal of the lower resolution layer). Finally, additional residual difference data can be transmitted in the enhancement layer bit stream to refine the result.

The position calculations for the up-sampling process are computed as described in Section III-A, and the up-sampling of the residual difference signal is performed using a simple bilinear up-sampling filter as shown in Table II, as there was no need demonstrated for using a more complex filter in this case (unlike for the up-sampling of luma texture signals in the I_BL macroblock type). The bilinear up-sampling is applied only within each residual transform block of the base layer—such that up-sampled positions that lie between different residual difference transform blocks are generated by extrapolating repetition of the values of the residual difference signal samples along the block edge.

IV. Encoder Issues

The encoding strategy for spatial scalability is not defined by the standard. Nonetheless, it is a critical design problem and deserves discussion. Unintelligent down-sampling and coding of the lower resolution layer signals will affect the end-to-end coding efficiency of both layers of the video content (and particularly that of the enhancement layer). In this section, we comment on relevant encoder issues.

Reference software was developed by the JVT in parallel with the SVC extension. This software is publicly available as the Joint Scalable Video Model, or JSVM [46]. The JSVM uses a coarse-to-fine coding approach, where the lower resolution layer sequence is first generated from the higher resolution data and then coded without regard to the content of the higher resolution pictures. Subsequent encoding passes predict the higher resolution information from the lower resolution layer when it leads to improved coding efficiency. Further details of the JSVM encoding algorithms are provided in [47].

With a coarse-to-fine coding strategy, one significant concern is how to generate the lower resolution image sequence from the high-resolution input. In the development of the SVC design, primarily a fixed down-sampling operator was used. The down-sampler was designed as a windowed sinc function [48], with the following impulse response tap values:

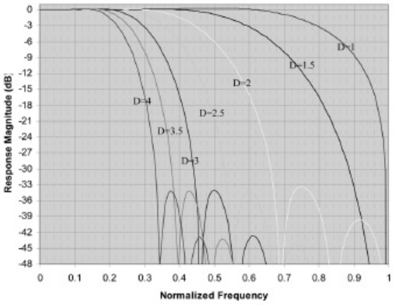


Fig. 5. Frequency response of cosine-windowed sinc function at phase position $p = 0.5$ with $N = 3$ and various D parameters to enable various cut-off frequencies [12], [48].

$$h(i) = f((i+p)/D) \tag{3}$$

where D is a bandwidth control parameter and p is a phase control parameter such that $0 \leq p < 1$ and

$$f(x) = \begin{cases} W(x/N) \cdot \dfrac{\sin(\pi x)}{(\pi x)}, & |x| < N \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $N-1$ is the number of side-lobes of the sinc function that are included on each side and the window function is the cosine window [48], given by

$$W(t) = \cos(\pi t/2). \tag{5}$$

As shown in Fig. 5, the D parameter can be used to control the cut-off frequency of the low-pass filter and thereby enable filter design for various downsampling ratios [48]. For the dyadic case, the filter was designed with half-sample phase, i.e., $p = 0.5$ (both horizontally and vertically, and applied to both luma and chroma).

Alternative down-samplers are anticipated in practice. For example, some encoder designers may prefer a specific down-sampler for its visual properties. A more sophisticated design might also take more care in achieving a particular relative positioning of luma and chroma as discussed in Section III-C. Optionally, some implementations may design an alternative down-sampler to better suit a particular architecture. No matter the motivation, the SVC design uses a particular prediction mechanism to predict high-resolution samples and residual difference from corresponding decoded reference layer data. In view of the design history, we suggest that using down-sampling filters that have a frequency response similar to those defined in [44] may provide reasonable coding efficiency.

More sophisticated down-sampling methods are also expected in the future. For example, the down-sampler may be content dependent. Alternatively, it may be designed to maximize coding efficiency for particular applications. One example of this type of filter is considered in [49], where the down-sampler design is posed as an optimization problem. The

[5] To save the need to repeatedly send the residual prediction flag in cases when an encoder will always apply residual prediction to applicable macroblocks, residual prediction can alternatively be enabled by default using syntax at the slice header level.

goal of the down-sampler is then to maximize the end-to-end coding efficiency while maintaining adequate lower resolution picture quality.

As a second direction for filtering improvement, it is also anticipated that a means for motion compensated temporal filtering [50], [51] can differentiate SVC devices. This is due to the single-loop decoding methodology, where the motion vectors, prediction modes, and residual difference data are projected to higher resolutions. By independently down-sampling each picture in the image sequence, temporal relationships between pictures are ignored. Incorporating temporal information into the down-sampling operation could increase the correlation between motion vectors in the lower and higher resolution grids and reduce the residual difference that remains to be coded after inter-layer prediction.

So far in this section, we have considered coarse-to-fine encoding techniques, where the lower resolution sequence is coded without regard to the content of the higher resolution data. This method is the most straightforward approach; however, alternatives are possible, and are highly advisable to consider. For example, a fine-to-coarse-to-fine strategy first encodes the high-resolution data, then encodes the lower resolution data with knowledge of the high-resolution bit stream, and then re-encodes the high-resolution sequence. The advantage of this approach is that the lower resolution can be coded with knowledge of the target, higher resolution data. This allows, for example, the motion vectors and transform coefficients of the lower resolution layer to be biased to provide better predictions for the high-resolution sequence.

Additional permutations to the encoder strategy are also conceivable, and the upper bound on performance would be achieved with a joint optimization of the encoding decisions across all resolutions. Preliminary results for such techniques were reported in [52] and [53], where Lagrange optimization techniques were applied to jointly optimize encoder decisions for multiple scalability layers. Conceptually, such an approach allows for a tradeoff between lower and higher resolution layer quality. In [53] it was reported that it was feasible to keep the quality of both layers within 10% of the coding efficiency of single-layer coding with such techniques.

V. Performance

The performance of the SVC extension requires careful analysis. In this section, we report the results of some coding efficiency experiments using the JSVM software that is maintained by the JVT [46], [47].[6] This software provides both an example implementation of an SVC encoder as well as an example implementation of a down-sampling operation for generating lower resolution image sequences. It is important for the reader to understand that the combination of example implementations is designed to maximize the quality and fidelity of the lower resolution sequences. Thus, the results in this section provide an indication of system performance when the lower resolution sequence is of primary importance and thus cannot be degraded relative to a single-layer encoding. However, the authors anticipate that many practical scenarios allow for balancing the quality of the lower resolution sequence with other performance parameters (such as end-to-end coding efficiency or encoder complexity). For such scenarios, the reported results provide only a lower bound on achievable performance.

In the rest of this section, we focus on a small number of application scenarios. All are derived from test conditions utilized by the JVT group during the development of the SVC extension [54]. Specifically, these test conditions consider: 1) a video streaming example with two resolutions and dyadic scalability; and 2) a surveillance, broadcast or storage example with three resolutions and dyadic scalability. Additionally, an extended spatial scalability example with two resolutions and nondyadic scalability was also considered by the JVT [55].

Experiments make use of image sequences that are common in the video coding community. Specifically, sequences consist of the Bus, Foreman, Football, Mobile, Crew, City, Harbor, and Soccer test data. The spatial resolution of the first four sequences is 352×288 luma samples per picture (common intermediate format, or CIF), while the remaining sequences have a spatial resolution of 704×576 luma samples per picture (4CIF). All image frames used 4:2:0 color sampling. Also, please note that the 704×576 (4CIF) data is derived from cropped 1280×720p material. It is therefore actually more representative of so called "high-definition" video content.

A. *Video Streaming*

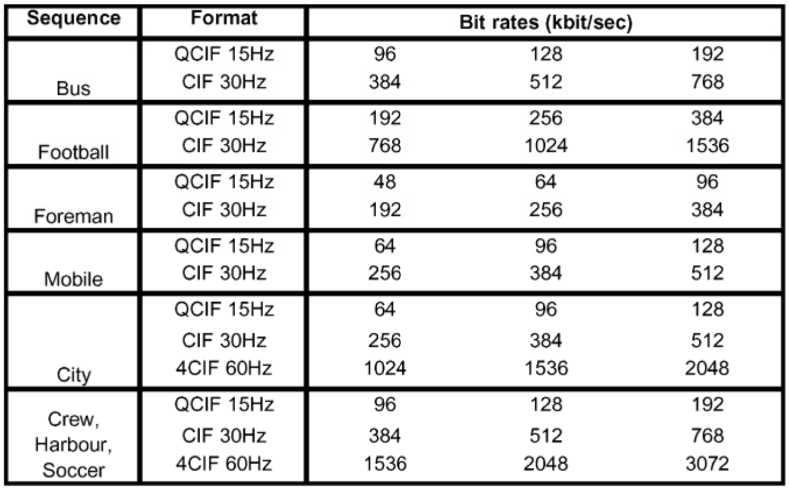
As a first scenario, the JVT defined test conditions consistent with a video streaming application. These tests made use of the Bus, Foreman, Football and Mobile sequences and represented the sequences in a two-layer bit stream. The base layer of the bit stream contained image data with a luma resolution of 176×144 luma samples per frame (QCIF) and a temporal rate of 15 fps. The enhancement layer of the bit stream contained image data with a luma resolution of CIF and a temporal rate of 30 fps. The down-sampling filter used to generate the base layer sequence from the enhancement layer sequence was the separable linear filter kernel with tap values $\{-8, 0, 24, 48, 48, 24, 0, -8\}$ (normalized by a 128 divisor with rounding).

Rate points for the test sequences were also defined and appear in Table III. The bit rates in the table were selected to correspond to applications of practical interest. To achieve the target bit rate, no rate control algorithm was used. Instead, the quantization parameter was varied to adjust the output bit rate. This required multiple encoding passes. Finally, while the specific configuration files are available [54], we mention that the image sequences were encoded with a hierarchical B-frame structure [56], [57] and with only one intra-picture coded frame (located at the beginning of the sequence). The interval between P-frames was 16 frames for Football and 32 frames for Bus, Mobile and Foreman. (The difference was due to the relatively high motion of the Football sequence.)

Representative results appear in Fig. 6, where rate-distortion plots for the Bus and Football sequences are illustrated. Distortion values are reported for the enhancement layer, while rate parameters represent the aggregate rate of the scalable bit stream. For comparison, Fig. 6 also contains the rate-distortion

[6] Specifically, we consider the JSVM_9_1 software version.

TABLE III
TEST CONDITIONS FOR DYADIC SCALABILITY



Sequence	Format	Bit rates (kbit/sec)		
Bus	QCIF 15Hz	96	128	192
	CIF 30Hz	384	512	768
Football	QCIF 15Hz	192	256	384
	CIF 30Hz	768	1024	1536
Foreman	QCIF 15Hz	48	64	96
	CIF 30Hz	192	256	384
Mobile	QCIF 15Hz	64	96	128
	CIF 30Hz	256	384	512
City	QCIF 15Hz	64	96	128
	CIF 30Hz	256	384	512
	4CIF 60Hz	1024	1536	2048
Crew, Harbour, Soccer	QCIF 15Hz	96	128	192
	CIF 30Hz	384	512	768
	4CIF 60Hz	1536	2048	3072

Bit rates as defined by the JVT when developing the SVC extensions [54]. Rates are for dyadic tests and include the scenarios identified as "video streaming" and "'broadcast" in this paper.

performance for a single layer H.264/AVC encoding as well as a simulcast scenario. The single-layer and simulcast results were generated using the same software implementation and encoding algorithms, though reconfigured to encode a single layer representation.

Visual evaluation of Fig. 6 provides insight into the system performance. Moreover, delta peak signal-to-noise ratio (PSNR) and bit rate measurements for all sequences are provided in Table IV. As can be seen from both the figure and table, the JSVM encoder outperforms the simulcast solution by an average of 9.6% in terms of bit rate. Compared to single layer coding, the JSVM encoder performs within 18% of the single layer codec. Of course, the SVC solution provides a lower resolution layer that can be easily extracted and decoded by legacy decoders. As noted above, the 18% penalty relative to single-layer coding may be partially attributed to the suboptimal "greedy" nature of the downsampling and encoding algorithms used in the test, which algorithmically optimizes only the lower resolution coding efficiency and thus represents only a lower bound on what is achievable.

B. *Surveillance, Broadcasting or Storage (Dyadic)*

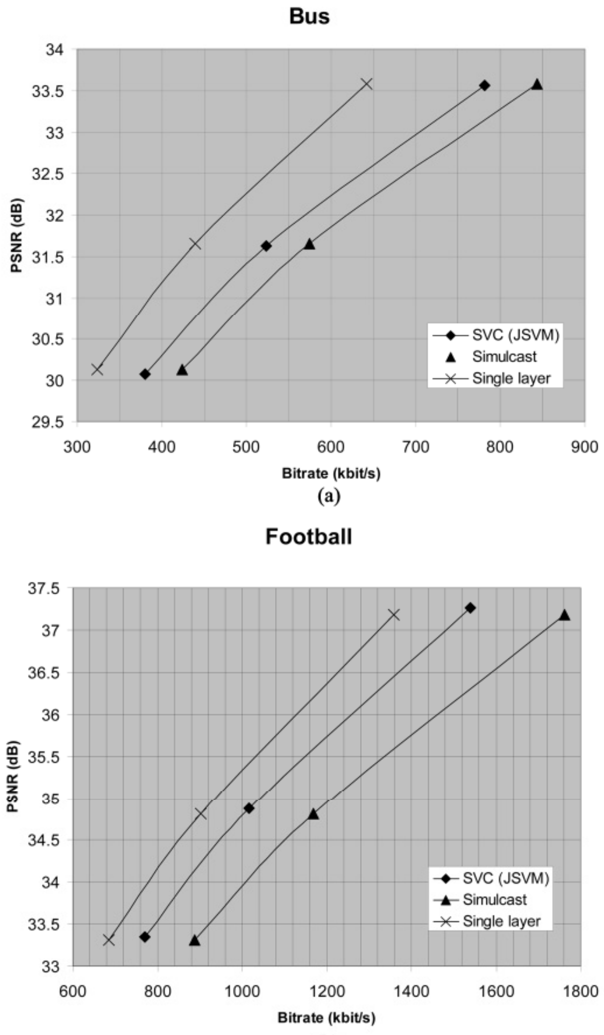
As a second scenario, the JVT defined test conditions consistent with a surveillance, broadcasting or storage application. These tests make use of the Crew, City, Harbour and Soccer test sequences in a three-layer configuration. The first two layers of the bit stream correspond to the two layers of the previous test. Specifically, the layers are QCIF and CIF resolutions, respectively, with frame rates corresponding to 15 and 30 fps. The third layer in the bit stream contains a 4CIF representation of the image sequence operating at 60 fps. Generation of the image sequences begins with the original 4CIF data. The CIF resolution is then constructed using the linear filter identified in the previous subsection. The QCIF resolution is subsequently created from the CIF data.

Rate points for the test sequences are also defined and previously provided in Table III. As before, no rate control algorithm is utilized to achieve the bit rate. Instead, the quantization parameter is varied to adjust the output bit rate. The image sequences are encoded with a hierarchical B-frame structure and with an intra frame inserted every 64 frames. (This corresponds roughly to a one second interval; it is also one of the notable differences between the video streaming and broadcasting scenarios.) The interval between P-frames is 16 frames for Crew, 32 frames for Soccer and 64 frames for City and Harbour. These intervals are derived heuristically.

Results appear in Fig. 7, where rate-distortion plots for the Crew and Harbour sequences are provided. Additionally, delta PSNR and bit rate measurements for all sequences are provided in Table V. As can bee seen from both the figure and table, the JSVM encoder outperforms the simulcast solution by an average of 10.5% in terms of bit rate. Compared to single layer coding, the JSVM encoder performs within 10.3% of the single layer codec on average. As before, the SVC solution provides a lower resolution layer that can be easily extracted and decoded by legacy hardware.

C. *Broadcasting (Nondyadic)*

The third scenario considered by the JVT is an application that uses a nondyadic relationship between layers. The primary difference between these nondyadic tests and previous dyadic experiments is the resolution relationships between base and enhancement layers. Specifically, the up-sampling ratios 4/3, 3/2, and 5/3 are considered. To be clear, these ratios denote the relationship between the horizontal (and vertical) dimension in the enhancement layer and the horizontal (and vertical) dimension in the base layer. For example, a scaling factor of 3/2 denotes that the base layer has a horizontal (and vertical) dimension equal to 2/3 of the corresponding enhancement layer dimension.



(a)

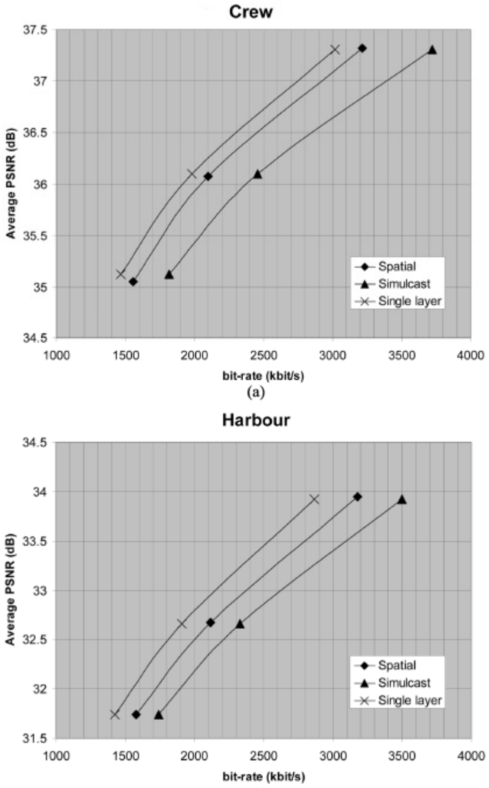
(b)

Fig. 6. Results for the (a) Bus and (b) Football sequences using the dyadic JVT test conditions and emulating a video streaming scenario. The SVC, simulcast and single layer solutions were all generated with the JSVM software maintained by the JVT.

TABLE IV
COMPARISON OF JSVM TO SIMULCAST AND SINGLE-LAYER SCENARIOS

	Simulcast		Single Layer	
Sequence	Delta Bitrate	Delta PSNR	Delta Bitrate	Delta PSNR
Bus	-8.9%	-0.03	19.4%	-0.03
Football	-13.0%	0.06	12.7%	0.06
Foreman	-10.5%	-0.05	15.8%	-0.05
Mobile	-5.9%	-0.02	24.3%	-0.02

Comparing the SVC extension to simulcast and single layer scenarios illustrates the performance of the SVC extension when base layer quality is uncompromised. Performance is sequence dependent but provides an average bit rate reduction of 9.6% compared to simulcast.



(a)

(b)

Fig. 7. Results for the (a) Crew and (b) Harbour sequences in the dyadic JVT test conditions that emulate a broadcasting scenario. The SVC, simulcast and single layer solutions were all generated with the JSVM software maintained by the JVT.

TABLE V
COMPARISON OF JSVM TO SIMULCAST AND SINGLE LAYER SCENARIOS

	Simulcast		Single Layer	
Sequence	Delta Bitrate	Delta PSNR	Delta Bitrate	Delta PSNR
City	-7.8%	0.00	14.2%	0.00
Crew	-14.2%	-0.03	6.2%	-0.03
Harbour	-9.2%	0.01	10.9%	0.01
Soccer	-10.7%	0.02	10.1%	0.02

Comparing the SVC extension to simulcast and single layer scenarios illustrates the performance of the SVC extension when base layer quality is uncompromised. Performance is sequence dependent but provides an average bit rate reduction of 10.5% compared to simulcast. Moreover, the scalable bitstream requires an average of 10.3% increase in bit rate compared to a single layer solution, while providing three decodable spatial resolutions.

The purpose of this third test is as much to stress the codec as to mimic specific application requirements. Nonetheless, in the spirit of comprehensive reporting, we provide results for these test conditions here. (For more additional results considering nondyadic scalability, please refer to [55], [58].) Experiments make use of the City, Crew, Harbour and Soccer image sequences. Again, no rate control algorithm is used to achieve the target bit rate. Instead, the quantization parameter is varied to adjust the output bit rate. Image sequences are encoded with a hierarchical B-frame structure and with an intra frame present every 32 frames. The interval between P-frames is 16 frames for all sequences.

Results for the nondyadic test are provided in Table VI. Unlike previous results though, the bit rates of the single layer and SVC encodings are matched and so the comparison is performed

TABLE VI
COMPARISON TO A NONDYADIC SCALABLE SYSTEM TO A SINGLE-LAYER APPROACH

Sequence	Scaling Ratio	Layer ID	Resolution	bitrate (Kbps)				PSNR (Y)		
				SVC	Single Layer	Delta		SVC	Single Layer	Delta (dB)
City	4/3	0	528x432	810.5	1028.1	0.60%		33.02	35.54	2.53
		1	4CIF	1034.2						
	3/2	0	448x384	715.1	1007.1	-1.53%		33.68	35.54	1.87
		1	672x576	991.7						
	5/3	0	384x336	609.1	972.3	0.23%		33.99	35.60	1.61
		1	640x560	974.5						
Crew	4/3	0	528x432	1192.0	1488.4	0.39%		35.29	36.52	1.22
		1	4CIF	1494.3						
	3/2	0	448x384	1041.2	1478.0	-1.28%		35.80	36.46	0.66
		1	672x576	1459.1						
	5/3	0	384x336	898.8	1417.7	1.84%		35.72	36.43	0.71
		1	640x560	1443.7						
Harbour	4/3	0	528x432	1180.4	1456.2	2.54%		30.76	32.56	1.81
		1	4CIF	1493.2						
	3/2	0	448x384	1045.5	1484.3	-1.38%		31.49	32.75	1.26
		1	672x576	1463.8						
	5/3	0	384x336	891.7	1442.8	-0.97%		31.58	32.93	1.35
		1	640x560	1428.8						
Soccer	4/3	0	528x432	1174.3	1511.1	-0.45%		34.64	36.62	1.98
		1	4CIF	1504.4						
	3/2	0	448x384	1051.5	1462.7	0.32%		35.38	36.62	1.24
		1	672x576	1467.3						
	5/3	0	384x336	893.5	1426.9	-0.59%		35.42	36.76	1.33
		1	640x560	1418.5						
Average						-0.02%				1.46

Using a multiple pass encoder, results are generated that satisfy the bit rate targets in [55]. Compared to the single layer system, the scalable approach decreases the PSNR of the output image by 1.46 dB on average while enabling additional, scalable functionality. Such non-dyadic uses also enable a higher-resolution (and thus higher perceptual quality) for the base layer than a dyadic case would. Furthermore, such cases may benefit from joint multi-layer encoder optimization techniques more than the dyadic case (although such optimization techniques were not used in this test).

in terms of PSNR instead of the delta bit rate. With this data, we can observe that the scalable scenario performs within 1.5 dB of the single layer bit stream on average while simultaneously supporting both a lower resolution and higher resolution display. The loss relative to a single-layer bitstream is relatively large in some cases in this test. However, we note the following aspects of these results.

- From a functionality perspective, these nondyadic ratios enable a higher resolution (and thus higher quality) base layer for a given enhancement layer resolution than a dyadic scalability scenario could deliver.
- Smaller up-sampling ratios more benefit more from joint inter-layer encoding optimization techniques due to the closer resolution correspondence between the layers. Such techniques were not applied here, and the result provides only a lower bound on what is achievable.

VI. CONCLUSION

The goal of this paper has been to introduce the spatial scalability part of the new standardized SVC design. This extension introduces scalability capability to the state-of-the-art H.264/AVC video coding standard. At the high level, the design is conceptually an image pyramid that uses a single-loop design methodology. This has the rather significant advantage of reducing decoder complexity, as an SVC decoder needs to perform only one motion compensation loop regardless of the enhancement layer being decoded. Even with the standardization of the SVC extension, there is significant opportunity for future work in the area of encoder design and resampling operators. This includes multilayer rate control and motion estimation algorithms, as well as improvements in down-sampling and post-processing. Advances in these aspects of an end-to-end SVC system will further improve applications, and it is expected to further improve coding efficiency.

ACKNOWLEDGMENT

The authors wish to acknowledge all of the companies and individuals that contributed to the spatial SVC extension of the H.264/AVC standard. Any meritorious material in this paper is more theirs than ours. They also wish to thank J. Zhao for her help in assembling the experimental results.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, Eds., "Amendment 3 to ITU-T Rec. H.264 (2005) | ISO/IEC 14496-10:2005," *Scalable Video Coding*, Jul. 2007.

[2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[3] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 | ISO/IEC 14496-10, Version 2: May 2004, Version 3: Mar.2005, Version 4: Sept. 2005, Version 5: June 2006, Version 7: Apr. 2007, Version 8 (with SVC extension) "Consented" July 2007, May 2003.

[4] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," *Proc. IEEE*, vol. 93, no. 1, pp. 18–31, Jan. 2005.

[6] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Commun. Mag.*, vol. 44, no. 8, pp. 134–144, Aug. 2006.

[7] *Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Part 2: Video (MPEG-2 Video)*, ITU-T Rec. H.262 | ISO/IEC 13818-2, 1994.

[8] B. Haskell, A. Puri, and A. N. Netravali, *Digital video: An introduction to MPEG-2*. New York: Chapman & Hall, 1997.

[9] *Video Coding for Low Bit Rate Communication*, ITU-T Rec. H.263, Feb. 1998.

[10] *ISO/IEC 14496-2 Information Technology-Coding of Audio-Visual Objects: Part 2—Visual (MPEG-4 Visual)* ver. 1, 2000.

[11] E. Francois, J. Viéron, and V. Bottreau, "Interlaced coding in SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1136–1148, Sep. 2007.

[12] E. Francois, J. Viéron, S. Sun, and G. J. Sullivan, "Extended spatial scalability: A generalization of spatial scalability for SVC extension of AVC/H.264," presented at the Picture Coding Symp., Beijing, China, Apr. 2006.

[13] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, Apr. 1983.

[14] J. M. Ogden, E. Adelson, J. Bergen, and P. Burt, "Pyramid-based computer graphics," *RCA Engineer*, vol. 30, pp. 4–15, 1985.

[15] P. Burt, "Smart sensing within a pyramid vision maching," *Proc. IEEE*, vol. 76, no. 8, pp. 1006–1015, Aug. 1988.

[16] A. Toet, "Hierarchical image fusion," *Mach. Vis. Appl.*, vol. 3, no. 1, pp. 1–11, Dec. 1990.

[17] M. Unser, "An improved least squares Laplacian pyramid for image compression," *Signal Process.: Image Commun.*, vol. 27, no. 2, pp. 187–203, May 1992.

[18] D. Heeger and J. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. SIGGRAPH*, 1995, pp. 229–238.

[19] J. Goutsias and H. J. A. M. Heijmans, "Nonlinear multiresolution signal decomposition schemes I: Morphological pyramids," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1862–1876, Nov. 2000.

[20] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding standard," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[21] D. Taubman and M. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2001.

[22] *Information Technology—JPEG 2000 Image Coding System: Core Coding System*, ITU-T Rec. T.800 | ISO/IEC 15444-1, 2004.

[23] S.-T. Hsiang, *Preliminary Results for Intra-frame Dyadic Spatial Scalable Coding Based on a Subband/Wavelet Filter Banks Framework*, Joint Video Team Doc. JVT-U133, Oct. 20–27, 2006.

[24] S.-T. Hsiang, *CE1: SVC Intra-frame AVC/H.264 Sub-Band Coding (SBC)*, Joint Video Team Doc. JVT-X059, Jun.–July 2007.

[25] W.-H. Peng, C.-Y. Tsai, T. Chiang, and H.-M. Hang, "Advances of MPEG scalable video coding standards," in *Knowledge-Based Intelligent Information and Engineering Systems*. Berlin, Germany: Springer, 1995, vol. 3684, pp. 889–895.

[26] R. Xiong, F. Wu, J. Xu, S. Li, and Y.-Q. Zhang, "Barbell lifting wavelet transform for highly scalable video coding," presented at the Picture Coding Symp., San Francisco, CA, Dec. 2004.

[27] R. Xiong *et al.*, "Barbell-lifting-based 3-D wavelet coding scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1256–1269, Sep. 2007.

[28] N. Adami *et al.*, "Overview of scalable video coding using wavelet-based approaches," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, Sep. 2007.

[29] H. Schwarz, T. Hinz, H. Kirchoffer, D. Marpe, and T. Wiegand, *Technical description of the HHI proposal for SVC CE1*, ISO/IEC JTC 1/SC 29/WG 11 (MPEG), Doc. M11244, Oct. 2004.

[30] H. Schwarz, T. Hinz, D. Marpe, and T. Wiegand, *Further Progress on Scalable Extension of H.264*, ITU-T SG 16/Q 6 (VCEG), Doc. VCEG-X08, Oct. 2004.

[31] H. Schwarz, D. Marpe, and T. Wiegand, *Further Results on Constrained Inter-Layer Prediction* Joint Video Team Doc. JVT-O074, Apr. 16–22, 2005.

[32] H. Schwarz, T. Hinz, D. Marpe, and T. Wiegand, "Constrained inter-layer prediction for single-loop decoding in spatial scalability," in *Proc. IEEE ICIP*, Genova, Italy, Sep. 2005, pp. 870–873.

[33] M. Horowitz, A. Joch, and F. Kossentini, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 704–716, Jul. 2003.

[34] H. Schwarz, D. Marpe, and T. Wiegand, *Independent Parsing of Spatial and CGS Layers*, Joint Video Team, Doc. JVT-S069, Mar.–Apr. 2006.

[35] H. Schwarz and T. Wiegand, *Updated Results for Independent Parsing of Spatial and CGS Layers*, Joint Video Team, Doc. JVT-T079, Jul. 15–21, 2006.

[36] G. J. Sullivan, *Position Calculation for SVC Upsampling*, Joint Video Team, Doc. JVT-R067, Jan. 2006.

[37] H. Schwarz, D. Marpe, and T. Wiegand, *SVC core Experiment 2.1: Inter-layer Prediction of Motion and Residual Data*, ISO/IEC JTC 1/SC 29/WG 11 (MPEG), Doc. M11043, Jul. 2004.

[38] E. Francois, J. Viéron, G. Marquant, N. Burdin, and P. Lopez, *Generic Extended Spatial Scalability*, Joint Video Team, Doc. JVT-O041, Apr. 2005.

[39] J. Viéron, E. Francois, and N. Burdin, *CE10: Unified Motion Upsampling in Extended Spatial Scalability*, Joint Video Team, Doc. JVT-P019, Jul. 2005.

[40] G. Marquant, E. Francoise, N. Burdin, P. Lopez, and J. Viéron, "Extended spatial scalability for nondyadic video formats: From SDTV to HDTV," presented at the SPIE VCIP, Beijing, China, Jul. 2005.

[41] X. Wang and J. Ridge, *Improvement of Macroblock Mode Prediction in ESS*, Joint Video Team, Doc. JVT-V108, Jan. 2007.

[42] S. Sun, *Upsampling Filter Design with Cubic Splines*, Joint Video Team, Doc. JVT-S016, Mar.–Apr. 2006.

[43] A. Segall and S. Regunathan, *Ad hoc Group Report on Spatial Scalability, Resampling, and Inter-Layer Prediction*, Joint Video Team, Doc. JVT-W007, Apr. 2007.

[44] S. Sun, J. Reichel, E. Francois, H. Schwarz, M. Wien, and G. J. Sullivan, *Unified Solution for Spatial Scalability*, Joint Video Team, Doc. JVT-R018, Jan. 2006.

[45] S. Sun, Deblocking Filter for I_BL Blocks in Spatial Scalable Video Coding-Response to CE2 Part 2, Joint Video Team, Doc. JVT-P013, Jul. 2005.

[46] J. Viéron, M. Wien, and H. Schwarz, *Joint Scalable Video Model (JSVM) 11 Software*, Joint Video Team, Doc. JVT-X203, Jun.–Jul. 2007.

[47] J. Reichel, H. Schwarz, and M. Wien, *Joint Scalable Video Model Algorithmic Text Description*, Joint Video Team, Doc. JVT-X202, Jul.–Jul. 2007.

[48] S. Sun and J. Reichel, *AHG Report: Spatial Scalability Resampling*, Joint Video Team, Doc. JVT-R006, Jan. 2006.

[49] A. Segall and A. Katsaggelos, "Resampling for spatial scalability," in *Proc. IEEE ICIP*, Atlanta, GA, Oct. 2006, pp. 181–184.

[50] H. Schwarz, D. Marpe, and T. Wiegand, *Comparison of MCTF and Closed-Loop Hierarchical B Pictures*, Joint Video Team, Doc. JVT-P059, Jul. 2005.

[51] A. Tabatabai, Z. Visharam, and T. Suzuki, *Study of Effect of Update Step in MCTF*, Joint Video Team, Doc. JVT-Q026, Oct. 2005.

[52] H. Schwarz and T. Wiegand, *Preliminary Results for a Rate-Distortion Multi-Loop SVC Encoder*, Joint Video Team, Doc. JVT-T080, Jul. 2006.

[53] H. Schwarz and T. Wiegand, *Further Results For an R-D Optimized Multi-Loop SVC Encoder*, Joint Video Team, Doc. JVT-W071, Apr. 2007.

[54] M. Wien and H. Schwarz, *Testing Conditions for SVC Coding Efficiency and JSVM Performance Evaluation*, Joint Video Team, Doc. JVT-Q205, Oct. 2005.

[55] E. Francois and J. Viéron, *Additional Results on ESS Evaluation*, Joint Video Team, Doc. JVT-Q013, Oct. 2005.

[56] H. Schwarz, D. Marpe, and T. Wiegand, *Hierarchical B Pictures*, Joint Video Team, Doc. JVT-P014, Jul. 2005.

[57] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE Conf. Multimedia Expo*, Toronto, ON, Canada, Jul. 2006, pp. 1929–1932.

[58] E. Francois and J. Viéron, "Extended spatial scalability: A generalization of spatial scalability for non dyadic configurations," in *Proc. IEEE ICIP*, Atlanta, GA, Oct. 2006, pp. 169–172.

C. Andrew Segall (S'00–M'05) received the B.S. and M.S. degrees in electrical engineering from Oklahoma State University, Stillwater, in 1995 and 1997, respectively, and the Ph.D. degree in electrical engineering from Northwestern University, Evanston, IL, in 2002.

He is a currently a Senior Engineer at Sharp Laboratories of America, Camas, WA, where he develops video coding and video processing algorithms for next generation display devices. From 2002 to 2004, he was a Senior Engineer at Pixcise, Inc., Palo Alto, CA, where he developed scalable compression methods for high definition video. His research interests are in image and video processing and include video coding, super resolution and scale space theory.

Gary J. Sullivan (S'83–M'91–SM'01–F'06) received the B.S. and M.Eng. degrees in electrical engineering from the University of Louisville J.B. Speed School of Engineering, Louisville, KY, in 1982 and 1983, respectively, and the Ph.D. and Engineer degrees in electrical engineering from the University of California, Los Angeles, in 1991.

He is the Rapporteur/Chairman of the ITU-T Video Coding Experts Group (VCEG), a Rapporteur/Co-Chairman of the ISO/IEC Moving Picture Experts Group (MPEG), and a Rapporteur/Co-Chairman of the Joint Video Team (JVT), which is a joint project between the VCEG and MPEG organizations. He is also the ITU-T video and image coding liaison representative to MPEG and JPEG/JBIG. Prior to joining Microsoft in 1999, he was the Manager of Communications Core Research at PictureTel Corporation (now Polycom), the world leader in videoconferencing communication at the time. He was previously a Howard Hughes Fellow and Member of the Technical Staff in the Advanced Systems Division of Hughes Aircraft Corporation and was a Terrain-Following Radar (TFR) System Software Engineer for Texas Instruments. He holds the position of Video Architect in the Core Media Processing Team of the Entertainment and Devices Division of Microsoft Corporation, Redmond, WA. At Microsoft he designed and remains lead engineer for the DirectX Video Acceleration (DXVA) API/DDI video decoding feature of the Microsoft Windows operating system. His research interests and areas of publication include image and video compression, rate-distortion optimization, motion estimation and compensation, scalar and vector quantization, and scalable and error/packet-loss resilient video coding.

Dr. Sullivan received the Technical Achievement award of the International Committee on Technology Standards (INCITS) in 2005 for his work on H.264/MPEG-4 AVC and other video standardization topics. He was a Guest Editor of the TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY for the Special Issue on the H.264/AVC Video Coding Standard in July 2003.