Standardized Extensions of High Efficiency Video Coding (HEVC)

Gary J. Sullivan, *Fellow, IEEE*, Jill M. Boyce, *Senior Member, IEEE*, Ying Chen, *Senior Member, IEEE*, Jens-Rainer Ohm, *Member, IEEE*, C. Andrew Segall, *Member, IEEE*, and Anthony Vetro, *Fellow, IEEE*

***Abstract*—This paper describes extensions to the High Efficiency Video Coding (HEVC) standard that are active areas of current development in the relevant international standardization committees. While the first version of HEVC is sufficient to cover a wide range of applications, needs for enhancing the standard in several ways have been identified, including work on range extensions for color format and bit depth enhancement, embedded-bitstream scalability, and 3D video. The standardization of extensions in each of these areas will be completed in 2014, and further work is also planned. The design for these extensions represents the latest state of the art for video coding and its applications.**

***Index Terms*—HEVC, JCT-VC, JCT-3V, MPEG, multiview video coding, range extensions, scalable video coding, standards development, VCEG, video compression, 3D video coding.**

I. Introduction

SINCE the recent completion of the first edition of the High Efficiency Video Coding (HEVC) standard [1], [2], now approved as ITU-T H.265 and ISO/IEC 23008-2, the relevant international standardization committees have shifted their focus toward the development of several key extensions of its capabilities to address the needs of an even broader range of applications. Although the first version of the HEVC standard already has a very broad scope, there are several key technical features that were left out of its first version in order to allow the development work to focus on the most "core" necessary elements of its design.

The extensions under current development, as of the time of preparation of this paper (reflecting the current status as of the Vienna meetings of July/August 2013), primarily fall into three areas: 1) the range extensions, which expand the range of bit depths and color sampling formats supported by the standard, and include an increased emphasis on high-quality coding, lossless coding, and screen-content coding; 2) the scalability extensions, which enable the use of embedded bitstream subsets as reduced-bit-rate representations of the video content; and 3) the 3D video extensions, which enable stereoscopic and multiview representations and consider newer 3D capabilities such as the use of depth maps and view-synthesis techniques.

The committees jointly responsible for HEVC are the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). For development of the HEVC standard, they formed the Joint Collaborative Team on Video Coding (JCT-VC) in January 2001, and have tasked it with the first two of the above-described extensions; and for work on 3D video topics for multiple standards including 3D video extensions for HEVC in particular, they formed a second (closely coordinated) team known as the Joint Collaborative Team on 3D Video (JCT-3V) in July 2012.

The rest of this paper is organized as follows. In the next section, a brief overview of the main features and coding tools supported in the HEVC first edition specification are summarized. Section III outlines the capabilities that will be provided by the range extensions and additional technology under consideration. In extending the HEVC design to accommodate scalable layers and multiple views, there is also a need to extend the high-level syntax of the standard; the functionality and key aspects of this design are reviewed in Section IV. In Sections V and VI, the scalability and 3D video extensions are presented, respectively. The design is also planned to support hybrid architectures, which provide a way to enhance legacy services with scalability layers or additional views; such architectures are described in Section VII. Conclusions and outlook are given in Section VIII.

II. Overview of HEVC First Edition Specification

HEVC defines a high-level syntax that supports network interfacing and other systems implementation aspects, and a video coding layer that carries the compressed picture data.

Many of the high-level syntax features of HEVC have been retained or extended from the H.264/MPEG-4 Advanced Video Coding (AVC) standard [3]. Parameter sets contain information that can be shared for the decoding of several pictures or sequences of pictures in the video bitstream. The parameter set structure provides a robust mechanism for conveying data that are essential to the decoding process by separating out this top-level header information to enable it to be repeated or reliably conveyed "out of band" as appropriate for the application. Each syntax structure is placed into a logical data packet called

Manuscript received June 06, 2013; revised August 24, 2013; accepted August 28, 2013. Date of publication October 11, 2013; date of current version November 18, 2013. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Yun He.

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

J. M. Boyce is with Vidyo, Inc., Hackensack, NJ 07601 USA (e-mail: jill@vidyo.com).

Y. Chen is with Multimedia R&D and Standards Group, Qualcomm Technologies, Inc., San Diego, CA 92121 USA (e-mail: cheny@qti.qualcomm.com).

J.-R. Ohm is with the Institute of Communications Engineering, RWTH Aachen University, 52056 Aachen, Germany (e-mail: ohm@ient.rwth-aachen.de).

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

A. Vetro is with Mitsubishi Electric Research Labs, Cambridge, MA 02139 USA (e-mail: avetro@merl.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/JSTSP.2013.2283657

1932-4553 © 2013 IEEE

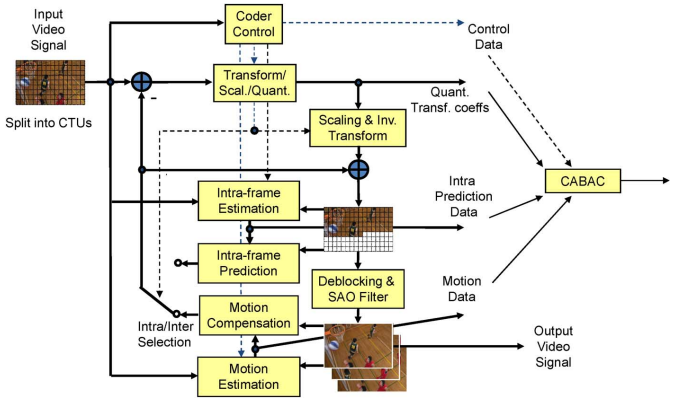


Fig. 1. Hybrid video encoder for HEVC.

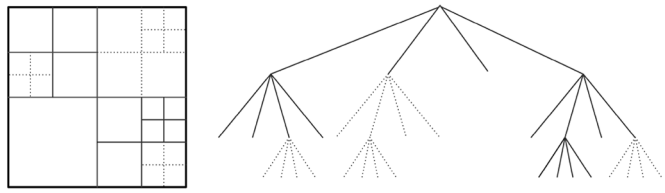


Fig. 2. Subdivision of a 64 × 64 luma CTB into CBs and TBs. Solid lines indicate CB boundaries and dotted lines indicate TB boundaries. Left: the CTB with its partitioning, right: the corresponding quadtree. In this example, the smallest leaf nodes are each 8 × 8 in size—although, in general, a TB can actually be as small as 4 × 4.

a *network abstraction layer* (NAL) unit. Depending on the content of a two-byte NAL unit header, it is possible to readily identify the purpose of the associated payload data, e.g., parameter sets, data for decoding random-accessible pictures, etc. A total of 31 NAL unit types are defined in the first edition (although the number can be increased, as a 6-bit code is used for NAL unit type signaling).

The high-level syntax of version 1 has been designed to make it extensible in a compatible way, particularly for cases where a legacy decoder needs to interpret a part of the bitstream. For this purpose, a new type of parameter set called the *video parameter set* (VPS) was defined in addition to the *sequence parameter set* (SPS) and *picture parameter set* (PPS) that were both already used in AVC. Furthermore, the NAL unit concept was also constructed in a way that enables more flexible random access, trick play, and partial sequence access (such as extraction of lower frame-rate temporal subsets). Additional NAL unit types are provided in HEVC to support various random access behaviors for video systems. In addition, layer identification and temporal sub-layer identification are enabled in the NAL unit header for generic support of multi-layer extensions, including scalable and 3D extensions.

The video coding layer of HEVC employs essentially the same block-based "hybrid" approach (inter-/intra-picture prediction and 2D transform coding) used in all video compression standards since H.261. Fig. 1 depicts the block diagram of a hybrid video encoder that could create a bitstream that conforms to the HEVC standard. A block-wise prediction residual is computed from corresponding regions of previously decoded pictures (inter-picture motion compensated prediction) or neighboring previously decoded samples from the same picture (intra-picture spatial prediction). The residual is then processed by a block transform, and the transform coefficients are quantized and entropy coded. Side information data such as motion vectors and mode switching parameters are also encoded and transmitted. Some key elements that enable the enhanced compression capability of HEVC are discussed below. A more detailed description of the key technical features can be found in [2].

Coding Tree Units and Coding Tree Block Structure: In contrast to the *macroblock* of previous standards (consisting of a 16 × 16 block of luma samples and two corresponding blocks of chroma samples), the analogous structure in HEVC is the *coding tree unit* (CTU). Each picture is split into CTUs of equal size. The CTU consists of a square *coding tree block* (CTB) for luma and corresponding CTBs for chroma. However, the specific size $L \times L$ of a luma CTB can be chosen by the encoder using $L = 16$, 32, or 64, and the larger sizes tend to provide better compression. In version 1, only 4:2:0 color sampling is supported, such that the corresponding chroma structures always have half the luma array size both horizontally and vertically. Each picture is segmented into sequences of CTUs in raster scan order, and each such sequence of CTUs is referred to as a *slice*. Each slice has a header that enables it to be decoded independently of all other slices in the picture. The CTBs of each CTU are partitioned into *coding blocks* (CBs), as indicated by a quadtree structure (Fig. 2). When a luma CTB is split by the quadtree, the luma and chroma CBs are split together, and a luma CB can be as small as 8 × 8 (accompanied by two 4 × 4 chroma CBs). One luma CB together with the two corresponding chroma CBs and associated syntax elements is referred to as a *coding unit* (CU).

Below the CU level, additional partitioning is performed into *prediction units* (PUs) and *transform units* (TUs). The decision whether to encode a picture area by *inter*-picture (motion compensated) or *intra*-picture (spatially extrapolated) prediction is made at the CU level. CBs have always square shapes. The luma and chroma *prediction blocks* (PBs) within a PU are also always square in the case of intra-picture prediction; for inter-picture prediction several non-square rectangular block shapes can also be chosen.

Transform Units and Transform Blocks: The prediction residual difference signal is coded using block transforms. A *transform unit* (TU) tree structure has its root at the CU level, where the CBs may be further split into smaller *transform blocks* (TBs). Integer basis functions approximating the *discrete cosine transform* (DCT) are defined for dyadic TB sizes from 4 × 4 to 32 × 32. For the 4 × 4 transform of intra-picture prediction residuals, an integer approximation of the *discrete sine transform* (DST) is used instead. The quantization of transform coefficients is controlled by a quantization parameter (QP) value which maps logarithmically to the quantizer step size (doubling each time the QP value increases by 6). Frequency-dependent quantization step size variation (based on transform coefficient position) is also supported. Coding and decoding of non-zero quantized coefficients is performed by grouping them into 4 × 4 coefficient sub-blocks and scanning the coefficients in each sub-block using a scanning order that is usually diagonal, but becomes horizontal or vertical for small TBs (8 × 8 and smaller) with particular directional modes

of intra-picture prediction. The position of the last non-zero coefficient in the scanning order is encoded first, followed by a "significance map" to identify which other preceding coefficients have non-zero values, and then the signs and magnitudes of the significant coefficients are coded.

Motion Compensation: Luma motion compensation uses quarter-sample precision, where 7-tap or 8-tap separable filters are applied in the horizontal and vertical dimensions for interpolation of fractional positions, with the specific filter type depending on the required fractional-sample position. Chroma motion compensation uses eighth-sample precision and 4-tap separable interpolation filters. Similar to AVC, multiple reference pictures are used. Per PB, either one or two motion vectors (MVs) can be applied, resulting in uni-predictive or bi-predictive coding, respectively, where bi-predictive coding uses an averaged result of two predictions to form the final prediction signal. Reference picture signaling is implemented using two reference picture lists (RPLs), called list 0 and list 1, where a picture from only one of these lists is used in the case of uni-prediction and pictures from both lists are used for bi-prediction. The reference picture index pointing into each respective list is part of the motion information. As in AVC, weighted prediction can be employed in either the uni-predictive or bi-predictive cases. Advanced motion vector prediction (AMVP) coding is used, including rules for deriving two MV prediction candidates, depending on availability, from MV data of adjacent PBs and a co-located position in the reference picture (the latter being referred to as temporal motion vector prediction, TMVP). The encoder signals the selected candidate MV predictor and sends a difference between the MV prediction value and the actual MV. A new "merge" mode for MV coding is also defined, signaling the inheritance of MVs from one of five candidates which are typically inferred from MVs of the neighboring PBs within the same CTU or MVs of a corresponding position in a reference picture. In merge mode, it is signaled which of the candidates is selected. Further, "skip" and "direct" motion inference is also specified—and in these cases no selection is signaled and the motion vector and reference picture index of the most probable candidate are used without modification. In any of the modes, candidate motion vectors are scaled according to the temporal distance from the actual reference picture, unless a reference picture is marked as a "long term reference."

Intra-Picture Prediction: Decoded boundary samples from adjacent blocks are used as prediction reference data for intra-picture spatial prediction in a PB. Intra-picture prediction can use 33 directional modes (compared to 8 such modes in AVC), plus DC (flat overall averaging) and planar (surface fitting) prediction modes. Chroma prediction is similar, but uses a simplified selection between fewer modes (horizontal, vertical, planar, DC, the same mode used for luma, or left-downward diagonal). The different intra-picture prediction modes are encoded by deriving most probable modes (e.g., the prediction directions) based on those of previously decoded neighboring PBs.

Entropy Coding: Five generic binarization schemes are defined for symbol encoding, and it is specified which of these is applied to each type of syntax element. *Context-adaptive binary arithmetic coding* (CABAC) is then used for entropy coding. The basic method is similar to the CABAC scheme in AVC, but has undergone a number of improvements, especially in regard to reducing the number of adaptive coding contexts, increasing the use of fast "bypass" coding, and improving the ability for parallel processing to increase the throughput.

In-Loop Filtering: One or two filtering stages can be optionally applied (within the inter-picture prediction loop) before writing the reconstructed picture into the decoded picture buffer. A *deblocking filter* (DBF) is used that is similar to the one in AVC; however the DBF design has been simplified with regard to its decision-making and filtering processes and also has been made more friendly to parallel processing. The second stage, called the *sample adaptive offset* (SAO) filter, is a non-linear amplitude mapping. The goal of SAO is to improve the reconstruction of the signal amplitude by adding an offset based on a look-up table mapping that is controlled by the encoder. Two types of SAO operation can be selected for each CTB—the band offset and edge offset modes, where depending on additional criteria (amplitude or local directional amplitude constellation) an offset value is added to the reconstructed sample amplitude.

Special "Transform Skip" Coding Modes: For certain types of content (especially screen content with graphics and text elements) more efficient compression is sometimes achieved when the transform is skipped (i.e. the residual is directly quantized and entropy coded). Furthermore, it is also possible to skip the quantization and loop filtering processes to enable lossless encoding of CUs.

III. Range Extensions

The drafted range extensions for HEVC include support for the 4:2:2 and 4:4:4 enhanced chroma sampling structures and sample bit depths beyond 10 bits per sample. Additional areas of work include coding of screen content (graphics and other non-camera-view or mixed content), very high bit-rate and lossless coding, coding of auxiliary pictures (e.g., alpha transparency planes), and direct coding of RGB source content. The range extensions are planned to be finalized in early 2014, and the draft can be found in [4].

As previously mentioned, the 4:2:0 chroma format supported in the version 1 profiles has chroma information that is half resolution both in the horizontal and vertical dimensions. This has been typical for consumer entertainment use, but the demands of higher-quality applications and screen content coding require use of the 4:4:4 format with full-resolution chroma representations, or of the 4:2:2 format in which half-resolution horizontal but full-resolution vertical chroma sampling is used.

In the 4:4:4 case, the draft range extensions support two modes of operation. The first, known as separate color plane coding, is to process each of the three color components separately, as if they were ordinary monochrome (luma) pictures. The second mode, known as joint color plane coding, is to process them jointly. Separate color plane coding is generally considered more difficult to support, so it is possible that this mode may not be supported in the final profile specifications.

When processed jointly, a single spatial segmentation is used to determine the CB, PB, and TB partitioning structure, and the MVs applied to the primary (nominally luma) component are

used to derive the MVs for inter-picture prediction of the other components. In this case, the decoding process is very similar to 4:2:0 processing, except for the different size dimensions of the chroma components. As a consequence, the quality of the motion compensation interpolation filtering is higher for luma (using 7 or 8 tap filters) than for chroma (using 4 tap filters). The same principle applies to other building blocks such as deblocking and SAO, which operate somewhat differently for luma and chroma components. If the video is coded directly in the RGB (red, green, blue) domain rather than being first pre-converted to luma (Y) and chroma (Cb and Cr) components, ordinarily G would be processed as Y, and B and R would be processed as Cb and Cr (although pre-conversion to YCbCr can ordinarily improve compression).

In the 4:2:2 case, only joint processing of the three components is foreseen. The basic decoding process can again remain unchanged, but with the addition of consideration of the different subsampling ratios for horizontal (2:1) and vertical (1:1), which can be mapped directly to a corresponding spatial segmentation. However, some cases require special considerations. Chroma regions that correspond to square luma regions are non-square rectangles (and vice versa). For the case of PBs, this is not really a problem; however, TBs are generally of square shape in luma, which would map to a rectangular TB of half width for chroma. To avoid the need for rectangular transform support in the design, such rectangular regions are split to form two square TBs of half height each. The DBF is not applied across the extra boundary introduced by this split, as the studies thus far indicate that this simplification is unlikely to cause visible artifacts for the envisioned 4:2:2 applications. Further, the prediction directions for angular chroma intra prediction (except for the horizontal, vertical, DC and planar modes) needed to be mapped to different angles relative to the prediction modes for luma, because of the non-equal horizontal/vertical sub-sampling [5]. For the case of motion compensation, the different chroma subsampling factors can be directly translated into MV position scaling factors for the chroma components, which no longer have equal scaling for horizontal and vertical displacements; otherwise, the decoding process is unchanged, e.g., 4-tap interpolation filters are still used for chroma.

Sample bit depths up to 10 bits per sample are already supported in the first edition of the standard. However, some applications require even higher precision—for example, some "ultra-high definition" formats are anticipated to use 12 bits per sample, and some medical, surveillance, military, and special-purpose applications may even need more. The planned range extensions are expected to include support for at least 14 bits per sample, and may include up to 16 bits. The version 1 syntax and semantics already provide support for higher bit depths, but the version 1 profiles include bit-depth restrictions, and some adjustments to the decoding process are necessary for best support of bit depths greater than 12 bits. As the bit depth and coding fidelity increase, some unusual phenomena can be exhibited in the compression behavior due to additional noise influence at the LSBs, and the dynamic range of the processing elements requires careful design for finite word-length arithmetic. The range extensions draft text includes an extended precision processing option that controls the processing word-length of the motion compensation and inverse transform stages to improve support for high-bit-depth coding.

TABLE I
BIT RATE REDUCTION OF HM 12.0 + REXT 4.1 VS. JM 18.5, FOR 4:4:4 INPUT

Configuration	Medium Rate Range			High Rate Range		
	Y	Cb	Cr	Y	Cb	Cr
All Intra	17.8%	14.7%	15.8%	13.3%	13.8%	14.2%
Random Access	35.1%	32.3%	27.4%	29.4%	32.1%	25.5%
Low Delay B	39.8%	45.6%	48.4%	32.8%	39.8%	41.2%

TABLE II
BIT RATE REDUCTION OF HM 12.0 + REXT 4.1 VS. JM 18.5, FOR 4:2:2 INPUT

Configuration	Medium Rate Range			High Rate Range		
	Y	Cb	Cr	Y	Cb	Cr
All Intra	15.9%	10.8%	12.6%	11.8%	8.7%	10.2%
Random Access	30.4%	12.0%	8.2%	28.0%	19.1%	14.1%
Low Delay B	35.2%	15.7%	12.8%	31.3%	21.9%	18.9%

Additionally, several relatively small changes to the decoding process have been developed for the range extensions that improve compression especially for screen content (graphics and text or mixtures of graphics and text with camera-view video), 4:4:4 chroma sampling, and near-lossless or lossless encoding. These modifications, which can provide substantial gains (sometimes as much as 30% or more bit rate reduction for 4:4:4 screen content coding with moderate-to-high fidelity), include the following:

- **Intra-picture block copying prediction**: With this feature, intra-picture prediction can operate by copying blocks of previously decoded regions within the same picture, in a similar manner to how motion compensation operates when referencing other decoded pictures.
- **Smoothing disabling for intra-picture prediction**: This feature allows the encoder to disable a smoothing pre-filtering that is otherwise applied to intra-picture spatial prediction signals.
- **Transform skip mode modifications**: These modifications, which apply both to lossy and lossless mode cases in which the inverse transform stage is skipped, include enabling horizontal and vertical DPCM coding modes for residual signals (with either intra-picture or inter-picture prediction), support of transform skipping for any block size (versus HEVC version 1 which supports this only for the 4×4 block size), rotation of 4×4 residual signals for more efficient entropy coding, and other small modifications of the entropy coding process for transform skip blocks.

Initial investigations show that HEVC retains its compression advantage relative to AVC also for the extended range applications. Tables I and II show the results of experiments measuring the average bit rate reduction for equal PSNR for example test sets of 10 bit 4:4:4 and 4:2:2 camera-view content video sequences, respectively, with various coding configurations. Since the project includes a significant focus on screen content coding (SCC), additional results are provided in Table III for some 4:4:4 sequences of this type of content. Each measurement was generated by coding seven video sequences with four QP values (22,

TABLE III
BIT RATE REDUCTION OF HM 12.0 + REXT 4.1 VS. JM 18.5, FOR SCC INPUT

Configuration	Medium Rate Range			High Rate Range		
	Y	Cb	Cr	Y	Cb	Cr
All Intra	53.5%	47.1%	48.5%	55.7%	47.6%	48.9%
Random Access	48.2%	44.0%	46.1%	49.3%	45.0%	46.9%
Low Delay B	48.0%	44.1%	46.0%	48.1%	44.0%	45.8%

27, 32 and 37 for the medium bit rate range and 17, 22, 27 and 32 for the high bit rate range). These results were obtained by running the current HM12.0+RExt4.1 range extensions draft reference software in comparison to the JM 18.5 software of AVC using similar configurations. Results are shown here for both luma and chroma measurements, since multiple color component consideration is an important part of the range extensions work (although the individual color component measurements are not strictly valid since they are based on the combined bit rate rather than isolating the bit rate used within the data for each color component separately). The results demonstrate the substantial compression improvement achieved by the HEVC range extensions for the tested content types. Note, moreover, that bit rate reductions for HEVC in perceptual terms generally exceed those measured by the PSNR metric used here, and we expect this to also be the case for the range extensions.

Furthermore, additional investigations are currently under consideration that may lead to additional future improvements for applications involving the coding of non-camera content, near-lossless coding, and coding in color domains other than YCbCr. This work may somewhat affect the near-term range extensions and is likely to result in an additional future phase of standardization activity. These include:

- **Cross-component decorrelation methods**. Correlation between different color components are typically larger in RGB color representation, compared to YCbCr (where the chroma components are already substantially decorrelated differences relative to the luma). Also, the penalty (in terms of bit rate increase) of not exploiting such correlations is naturally larger in color formats without subsampling of components. Therefore, methods for inter-component prediction are being investigated as possible additional elements to be applied within the encoding/decoding processes.
- **Improved compression in lossless and near lossless coding**. The HEVC first edition already enables lossless compression by skipping the transform, quantization and loop filtering, whereas prediction (motion-compensated or intra-picture) and entropy coding are used mostly "as is." Substantially different techniques have been proposed that may lead to additional improvements when coding at very high fidelities.
- **Special tools for screen content**. Whereas the HEVC first edition and its drafted range extensions enable improved compression of screen content by the simple options of the transform bypass mode and block copying, other, more sophisticated methods particularly suitable for coding synthetic image structures (which have characteristics such as sharp edges and repetitive patterns) are under investigation.

IV. HIGH-LEVEL SYNTAX FOR THE MULTI-LAYER EXTENSIONS

The scalability and 3D extensions to HEVC address many of the same applications as the scalable and multiview extensions of AVC, namely Scalable Video Coding (SVC) [6] and Multiview Video Coding (MVC) [7] as specified in Annexes G and H of the AVC specification [3], respectively. Both the SVC and MVC extensions of AVC are designed to be backward compatible to AVC for the base layer (or base view) and both incorporate temporal scalability to enable extraction and adaptation to different frame rates for the scalable or multiview bitstreams. SVC additionally provides spatial scalability, wherein multiple layers with different spatial resolutions are present, and so-called signal-to-noise ratio (SNR) scalability, wherein multiple layers may have the same spatial resolution but different fidelity. MVC provides the decoding of multiple views of the same scene, such as stereoscopic views or views from camera arrays. The high-level syntax designs of the SVC and MVC extensions of AVC are not fully aligned. Whereas the SVC extension of AVC uses a single-loop decoding process and involves joint decoding of the base and enhancement layers at the block level, MVC uses multi-loop decoding and does not change the core decoding process of an AVC High Profile decoder. Further, different NAL unit headers are used in the SVC and MVC extensions of AVC, such that no straightforward way exists to combine MVC view scalability with SVC spatial or SNR scalability.

In contrast, a common extension high-level syntax has been designed for all HEVC multi-layer extensions, including scalable, multiview, and depth map layers. While the initial profiles in development do not combine scalable and multiview layers, this high level syntax provides extensibility to enable future profiles that support combinations of different types of layers.

A. Layers, Sub-Layers, Pictures and Access Units

Some of the terminology in the multi-layer extensions of HEVC differs from the related concepts in the SVC and MVC extensions of AVC. In HEVC, a layer is generically defined as a set of NAL units with the same layer ID value in the NAL unit header. A *layer* may be a representation of the video which differs from other representations in terms of spatial resolution, quality (SNR), view angle, or for the same view, the property of being texture or depth. In the future, a layer may represent some other enhanced characteristics of the video scene which require sets of coded slices indexed by the time axis. In the AVC extensions, enhancing temporal frame rates is considered to be achieved by adding layers. However, in HEVC, temporal sub-layers corresponding to different temporal frame rates are defined within a layer and use the same value of layer ID [9].

In the HEVC extensions, a coded picture represents the coded samples of a single layer within an access unit, which contains the pictures from all layers with the same output time.

B. NAL Unit Header

The HEVC NAL unit design follows the same general principles as the AVC design, as described in [10], but has a different header length and contains some different syntax elements. The HEVC first edition and its extensions use the same two-byte NAL unit header. In the NAL unit header, six bits are allocated to a syntax element which represents a layer ID value. In the HEVC first edition, the layer ID value must be equal to zero, representing the base layer. A more detailed comparison of the

NAL unit header designs in AVC and HEVC as well as the motivation of the NAL unit header design in HEVC and its extensions can be found in [11].

C. Video Parameter Set

In both AVC and HEVC, all coded slices in a particular layer of a coded video sequence must refer to the same sequence parameter set (SPS), the ID for which is signaled through the picture parameter set (PPS), which is, in turn, identified in each slice header.

In addition to the PPS and SPS, which are defined similarly in AVC, a new type of parameter set is defined for HEVC. The video parameter set (VPS) provides information that is applicable to all layers in the entire coded video sequence. The VPS is intended for use in systems interfaces, capabilities exchange, and sub-bitstream extraction. A VPS identifier syntax element is added to the SPS, creating an additional hierarchy of parameter set levels. Each layer of a given video sequence, regardless of whether it has the same or different SPS as other layers, refers to the same VPS.

The VPS conveys information including 1) common syntax elements shared by multiple layers or operation points, in order to avoid unnecessary duplications; 2) essential information of operation points needed for session negotiation, including e.g., profile and level; and 3) other operation-point-specific information that does not belong to one specific SPS, e.g., hypothetical reference decoder (HRD) parameters for layers or sub-layers [11].

A VPS contains two parts, the base VPS and the VPS extension. The base VPS, as defined in the first edition, contains information related to the HEVC version 1 compatible layer, as well as operation points corresponding to layer sets [12], [13]. The base VPS also contains temporal scalability information, including the maximum number of temporal layers [9]. The VPS extension contains information related to the additional layers beyond the base layer.

In the VPS extension, the syntax can flexibly associate each layer ID with scalability parameters and inter-layer dependencies [14]. Layer dependencies are signaled, to indicate which layer(s) are used as reference layer(s) for inter-layer prediction when the current layer is coded. The pictures are coded in ascending order of layer ID, such that a layer can only depend upon another layer with a lower value of layer ID. In the AVC extensions, similar information may be present for SVC and MVC separately in different syntax structures, including e.g., different subset sequence parameter sets, and SEI messages, such as the scalability information SEI message for SVC [15] and the view scalability information SEI message for MVC [7].

In the VPS extension, the maximum number and type of scalability dimensions present in the coded video sequence are also signaled. For each possible layer ID value, values may be specified for a view ID (corresponding to the geometric location of each view), dependency ID (indicating different spatial or SNR scalability layers, typically with different resolution), and a depth map indication flag (indicating whether the current layer belongs to the texture or depth map of the 3D video content), and the syntax can enable signaling of additional scalability types in future extensions through reserved values [16]. Therefore, advanced adaptation based on a variety of video characteristics can be done by a media-aware network element (MANE) by first mapping the layer ID value to the characteristics specified in the VPS [17].

D. Sub-Bitstream Extraction

Sub-bitstream extraction, as specified in HEVC and its extensions, behaves similarly to the sub-bitstream extraction functions defined in the SVC and MVC extensions of AVC. Generally, target values of scalability parameters are provided as inputs, and a conforming sub-bitstream is output that contains only the target layers and sub-layers, based upon the scalability parameters.

In HEVC, the inputs to the sub-bitstream extraction process are the target temporal ID and a target layer identifier list. NAL units are removed which are in temporal sub-layers above the target temporal ID value and/or with layer ID values not included in the target layer identifier list.

A MANE can use the scalability dimensions per layer and the inter-layer dependencies signaled in the VPS extension to construct a target layer set appropriate for its desired function. For example, a MANE may wish to remove all views but the base view from the bitstream, or may wish to remove the highest spatial/SNR enhancement layer simply based on layer ID values. However, a simple MANE may perform simple sub-bitstream extraction without considering the inter-layer dependencies from the VPS extension, using only the temporal ID and layer ID values present in the NAL unit header by relying on the requirement that a given layer may only be dependent upon another layer with a lower value of layer ID. Such a simple MANE may safely remove layers with higher values of layer ID and be guaranteed that the extracted sub-bitstream will be conforming and decodable. With the scalability dimensions signaled in VPS extension, although certain video characteristics are not signaled as part of the NAL unit header, bitstream extraction based on various scalability dimension information can also be achieved. However, in AVC's SVC or MVC, such functionality requires signaling the scalability dimensions as part of NAL unit header, therefore four bytes are required for each NAL unit [7], [15].

V. Scalability Extensions

The scalability extension to HEVC enables spatial and coarse grain SNR scalability, and is referred to as "SHVC." The plan is to finalize this extension of HEVC by mid-2014, and the draft text can be found in [17]. Temporal scalability support was already provided in HEVC version 1, and may be combined with spatial and SNR scalability in SHVC [19]–[21]. The SHVC design uses a multi-loop coding framework, such that in order to decode an enhancement layer, its reference layers have to first be fully decoded to make them available as prediction references. This differs from AVC's SVC extension design, which used single-loop decoding for inter-coded macroblocks so that the motion compensation process would only need to be performed once when decoding. When two spatial or SNR layers are used, the base layer is the only reference layer, but for three or more

spatial or SNR layers, intermediate layers may also be used as reference layers. To some extent, an efficient single-loop decoding was only possible by defining reference and enhancement layer decoding processes closely dependent at the block-level, e.g. adding new prediction modes, using reference layer contexts for the enhancement layer's entropy coding etc. The high level design of the HEVC scalability extension, e.g., multi-loop coder/decoder and restrictions against block-level changes, were motivated by ease of implementation, especially the possibility to re-use existing HEVC implementations, even though the overall number of computations and memory accesses of the decoder would be higher than in a single-loop design. Beyond that, multi-loop coding also provides coding efficiency advantages over single-loop coding designs.

The coding tools in the HEVC scalability extension are limited to changes at the slice level and above. The reference layer picture, resampled if necessary, is used as additional reference picture for enhancement layer prediction, which enables inter-layer texture and motion parameter prediction. The multi-loop design is somewhat similar to AVC's and HEVC's multiview extensions, which require full decoding of the base view in the case of decoding dependent views. However, in the multiview case (as currently specified), all views have the same resolution so that no resampling is needed. The same applies for the case of SNR scalability, where scalable layers represent pictures of identical spatial resolution. The base layer bitstream can be interpreted by legacy decoders, and may be either an HEVC bitstream or an AVC bitstream. When the base layer is an AVC bitstream, only inter-layer texture prediction is performed, with inter-layer motion prediction not supported. Investigations have shown that the compression benefit would be small, and the AVC base layer motion vectors may not easily be accessible in existing decoder implementations.

In terms of performance and complexity, dependent coding of layers is often compared against simulcast (independent coding of equivalent signals). Typical applications where scalable coding or simulcast would be applied, such as flexible rate or resolution switching, would usually only output one of the layers. However, in the case of multi-loop decoding, it is still necessary to decode all reference layers, such that the overall decoding complexity increases compared to simulcast. This effect is more critical in SNR scalability, where the reference layers are not subsampled. On the other hand, dependent coding of layers has advantages over simulcast in terms of compression performance, as reported in the end of this section.

When spatial scalability is used, the decoded reference layer picture is resampled using a normatively defined upsampling filter for the spatial scalability case. Spatial scalability ratios in the current design are limited to 1.5× and 2× spatial resampling factors in each dimension (since these factors are sufficient to cover the primary anticipated use cases and the restriction simplifies implementation), and are described in the following sub-section.

A. Upsampling Filter

The upsampling filter in the HEVC scalability extension is used to map reconstructed sample values from the reference layer to the higher-resolution sampling grid of the enhancement layer [22]. This allows the use of the reconstructed reference layer sample values for enhancement layer prediction. In the scalability extension, the upsampling process is defined as a normative part of the standard and is further described in this subsection. The downsampling process used to create the source pictures of lower resolution as input to the encoding process of the reference layer is left outside the scope of the standard (as are most other aspects of the encoding process).

TABLE IV
FILTER COEFFICIENTS FOR THE LUMA UPSAMPLING FILTER

Phase	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
0	0	0	0	64	0	0	0	0
5/16	−1	4	−11	52	26	−8	3	−1
8/16	−1	4	−11	40	40	−11	4	−1
11/16	−1	3	−8	26	52	−11	4	−1

In the scalability extension, the upsampling filter is defined as an 8-tap polyphase finite-impulse-response (FIR) filter for luma resampling, and a 4-tap polyphase FIR filter for chroma resampling. One motivation for the number of taps is consistency with the HEVC motion compensation design for fractional-position interpolation, which also uses 8-tap and 4-tap FIR filters for luma and chroma interpolation, respectively. However, the corresponding reference layer position is defined with 1/16 sample precision, so filters for additional phase shifts are needed. (Motion compensation operates with only 1/4 sample precision for luma and 1/8 sample precision for 4:2:0 chroma.) The basic design enables the use of arbitrary upsampling ratios, in which filters for all 16 phase positions would be necessary, but the current specification is restricted to ratios of 1.5× and 2×, for which fewer positions are needed.

Scaled reference layer offsets may be signaled to enable the reference layer and enhancement layer the freedom to not fully correspond to the same region of a picture. Scale factors for the horizontal and vertical directions are computed as the ratio between the relevant enhancement and reference layer regions widths and heights, respectively. For each enhancement layer sample, the corresponding reference layer sample location and 1/16 sample phase is determined considering the scale factors and the scaled reference layer offsets. The 8-tap (or 4-tap) filter coefficients which correspond to the calculated phase are applied to the input reference layer samples, which are the sample at the reference sample location and its neighboring samples in the reference layer. Filter coefficients for the luma upsampling filter are shown in Table IV. The selection of the tap values is again analogous to the HEVC motion compensation interpolation process. The 0 and 8/16 phases are identical to the 0 and 1/2 phases of the HEVC process, and are needed for upsampling by the ratio 2×. The 0, 5/16 and 11/16 phases are needed for the ratio 1.5×, where the latter two are designed using the same approach as the 1/4 and 3/4 phases in the motion compensation interpolator and satisfy the same constraints on frequency response and the precision of the calculation.

TABLE V
FILTER COEFFICIENTS FOR THE CHROMA UPSAMPLING FILTER

Phase	T_0	T_1	T_2	T_3
0	0	64	0	0
4/16	−4	54	16	−2
5/16	−6	52	20	−2
6/16	−6	46	28	−4
8/16	−4	36	36	−4
9/16	−4	30	42	−4
11/16	−2	20	52	−6
14/16	−2	10	58	−2
15/16	0	4	62	−2

Similarly, coefficients for the chroma upsampling filter are shown in Table V. Here, chroma upsampling requires the definition of nine phases of the polyphase filter to support the upsampling ratios of $1.5\times$ and $2\times$. The reason for the larger number of phases necessary for chroma is the inherent phase shift between luma and chroma samples in 4:2:0 chroma subsampling, which is considered when mapping base and enhancement layer chroma positions. As in the luma filter, the phases corresponding to those used in motion compensation have the same tap values, while phases not used in the motion compensation satisfy the same constraints on frequency response and calculation precision.

B. Inter-Layer Texture Prediction

Use of the upsampling process described above enables the projection of reference layer reconstructed sample values to the enhancement layer resolution. To enable the selection of this upsampled information for prediction in the enhancement layer, the scalability extension employs a so-called "reference index" approach [23]. Conceptually, this approach requires an enhancement layer decoder to insert the upsampled reference layer picture into the enhancement layer RPL. The upsampled picture can then be signaled for reference in the same manner as usually in inter-frame prediction. That is, the enhancement layer bitstream signals an inter-mode CU, with the reference index corresponding to the upsampled picture inserted into the enhancement layer RPL (with a zero motion vector used for this specific reference picture).

The process for constructing the RPL at the decoder is relatively straightforward. First, an initial RPL is constructed in the same way as in HEVC version 1. That is, the short-term reference pictures and long-term reference pictures identified in the bitstream are added to the list. Following these pictures, the upsampled base layer picture is appended to the initial RPL and is marked as a long-term reference picture (so that motion vector predictors referring to these reference pictures are not scaled as a function of temporal distance). Again, this is consistent with the first edition of HEVC, except that the initial lists now contain the upsampled base layer picture and any additional reference layer pictures, when present.

The actual RPLs used by the enhancement layer decoder may be modified from their initial values, when RPL modification information is present in the bitstream (as is also the case in HEVC version 1). When this information is not present, the initial RPL is used directly. When RPL modification information is present, an encoder can signal to re-order the initial list before use, using the same process defined in HEVC version 1. This re-ordering allows the pictures corresponding to the reference layers, to be moved to a different location in the list. One benefit of this is improved coding efficiency, as pictures toward the end of the list require more bits to be indicated. When the upsampled reference layer samples are highly correlated to the enhancement layer, it is advantageous for an encoder to move the upsampled reference layer samples to an earlier location within the list.

The approach based on reference index signaling enables additional coding flexibility. For example, through the use of bi-prediction, an encoder can signal a prediction that averages information from the reference layer and reconstructed enhancement layer pictures from different time positions. This could also employ weighted prediction. To limit memory bandwidth and complexity, the reference index approach also specifies a bitstream restriction that the motion vector must be zero when referencing the upsampled reference layer samples. This simplifies the decoder design, especially for implementations that might perform the upsampling "on the fly" as part of the prediction process, rather than upsampling whole reference pictures as a pre-processing step.

In addition to the use of the upsampled reference layer samples, the scalable extension also supports the prediction of motion information from the decoded reference layer. This is accomplished by associating motion data from the reference layer with the upsampled reference layer picture that is inserted into the enhancement layer RPL. This motion field mapping process is described in the next sub-section.

C. Inter-Layer Motion Prediction

In the scalable extension, motion field mapping is the process of using the reference layer motion information when coding the enhancement layer motion vectors by making use of the existing TMVP process of HEVC version 1 [24].

In HEVC, TMVP is used to predict motion information for a current PU from a co-located PU in the reference picture. The process is defined to require the prediction modes, reference indices, luma motion vectors and reference picture order counts (POCs) of the co-located PU. This information is stored on a 16×16 luma block basis, which may be a lower resolution than what is transmitted in the bitstream in cases of small PU sizes. This reduces the worst-case memory size and bandwidth requirements for storing the reference layer motion information [25]. The goal of the motion field mapping process is then to project this motion information from the reference layer to the enhancement layer's resolution, while also accounting for the 16×16 TMVP storage units in the reference layer.

The first step in the mapping of the motion information is to determine for the current enhancement layer PU the co-located position in the stored reference layer motion information, taking into account the reduced motion information storage resolution as well as the upsampling ratio between the two layers and any reference layer offsets. Once the co-located position is determined and the motion information from the co-located reference layer PU is available, a scaling operation is applied to those motion vectors to account for the upsampling ratio (since

TABLE VI
BIT RATE REDUCTION OF SHVC VS. SIMULCAST: 2× SPATIAL SCALABILITY

Sequence	ΔQP=0		ΔQP=2	
	EL+BL vs. simulcast	EL only vs. high res. single layer	EL+BL vs. simulcast	EL only vs. high res. single layer
Kimono	19.8%	29.2%	27.3%	47.5%
ParkScene	12.6%	17.6%	17.6%	27.8%
Cactus	11.6%	16.6%	16.7%	27.7%
BasketballDrive	14.5%	19.9%	20.8%	33.0%
BQTerrace	6.0%	7.3%	8.5%	12.1%
Average	12.9%	18.1%	18.2%	29.6%

TABLE VII
BIT RATE REDUCTION OF SHVC VS. SIMULCAST: 1.5× SPATIAL SCALABILITY

Sequence	ΔQP=0		ΔQP=2	
	EL+BL vs. simulcast	EL vs. high res	EL+BL vs. simulcast	EL vs. high res
Kimono	29.0%	49.5%	40.7%	78.1%
ParkScene	22.3%	36.0%	31.7%	58.8%
Cactus	21.1%	34.7%	30.8%	58.5%
BasketballDrive	24.6%	38.6%	34.6%	61.9%
BQTerrace	13.5%	18.0%	20.3%	33.3%
Average	22.1%	35.4%	31.6%	58.1%

motion vectors also grow with the picture resolution). However, no further scaling depending on temporal distance is applied due to the fact that the reference layer picture is indicated as "long term" picture.

The motion mapping process can be enabled or disabled within the bitstream, and it is disabled when an AVC base layer is used. Combined with the upsampling of reference layer sample values and the reference index signaling mechanism, motion mapping provides a means to leverage a significant amount of reference layer information without changing the block level design of an HEVC decoder.

D. Performance Evaluation

To evaluate the compression efficiency of the SHVC design during the standardization process, a set of common test conditions (CTC) [26] have been defined, which include a wide range of test conditions. Although SHVC can accommodate more than two scalable layers, the CTC only uses two layers. Experimental results are provided in this section for a subset of the CTC, using the scalable HEVC model (SHM) reference software SHM 2.0. Only those sequences using a 1920 × 1080 resolution at the enhancement layer are included, with corresponding base layers of 960 × 540 or 1280 × 720, for 2× and 1.5× spatial scalability, respectively. Only the Random Access test configuration is used, in which intra-coded pictures are provided once per second in the video sequence. The coding efficiency of scalability is highly dependent upon the relative bit rate allocation between the base and enhancement layers. The CTC include two different QP offsets between the base and enhancement layers, where the enhancement layer QP = ΔQP + base layer QP, with ΔQP equal to 0 or 2.

Tables VI and VII show the average bit rate savings for equal luma PSNR across four base layer QP values (22, 26, 30, and 34). Two types of comparisons are made, as shown in separate columns. A simulcast comparison is made, in which the enhancement layer (EL) plus base layer (BL) are compared to simulcast of a high-resolution single-layer bitstream at the same resolution as the enhancement layer plus the identical base layer. Additionally, a comparison is made where only the enhancement layer is compared to the high-resolution single-layer bitstream. The latter number is deemed relevant for the cost savings when introducing an additional service based on scalable technology (e.g. Ultra-HD broadcast when an HD broadcast of the same program already exists).

VI. 3D VIDEO EXTENSIONS

3D and multiview video formats can enable depth perception for a visual scene when used with an appropriate 3D display system. The available types of 3D displays include stereoscopic displays that are viewed with special glasses to enable the display of different views to each eye of the viewer, and auto-stereoscopic displays that emit view-dependent pixels and do not require glasses for viewing. The latter kind of displays often employ depth-based image rendering techniques, where it is desirable to use high-quality depth maps as part of the coded representation. Therefore, video plus depth is another important and emerging class of 3D formats. These can also allow for advanced stereoscopic processing, such as adjusting the level of depth perception in conventional stereo displays according to display size, viewing distance, user preference, etc. The depth information itself may be extracted from a stereo pair by solving for stereo correspondences or may be obtained directly through special range cameras; it may also be an inherent part of the content, e.g. in 3D computer graphics generated imagery.

To support these applications, HEVC extensions for the efficient compression of stereo and multiview video are being developed by JCT-3V, and the inclusion of depth maps to support advanced 3D functionalities is also under study. An analysis of the different schemes in terms of compression performance is also provided at the end of this section.

A. Multiview HEVC

The most straightforward architecture is a multiview extension of HEVC that is referred to as MV-HEVC. It uses the same design principles of the prior MVC extension in the AVC framework [7], [8]. The plan is to finalize this extension of HEVC by early 2014, and the draft text can be found in [27]. This scheme enables inter-view prediction by modifications to the RPL construction, such that pictures from other views at the same time instances can be used for prediction, where the disparity shift between the views is compensated for in the prediction process instead of the motion shift due to time differences. The whole approach is simply defined by a) extending the high-level syntax appropriately, and b) defining a process by which decoded pictures of other views are stored as reference pictures as needed.

The extensions to high-level syntax include signaling the prediction dependencies between different views, identification of which pictures belong to each view, and syntax elements to facilitate extraction of the base view. A key benefit of this architecture is that it can be implemented without changing the syntax or decoding process of single-layer HEVC below the

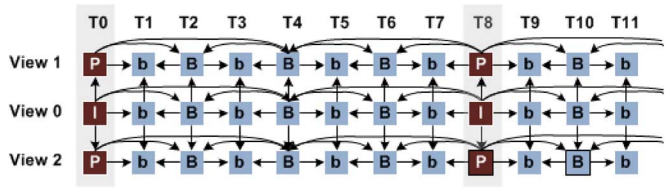


Fig. 3. Example multiview prediction structure for a 3-view case. View 0 denotes the base view and a picture in a non-base view (view 1 or view 2) can be predicted from pictures in a dependent (base) view of the same time instance. Pictures denoted by "I" use only intra-picture prediction, pictures denoted by "P" additionally use uni-predictive inter-picture prediction, and pictures denoted by "B" or "b" additionally use bi-predictive interpicture prediction. Pictures with a darker color belong to temporal random access points, and pictures associated with "b" are not used for temporal reference.

slice header level, which allows re-use of existing HEVC encoder and decoder implementations without major changes for stereo and multiview applications.

An example prediction structure is shown in Fig. 3. Inter-view sample prediction is enabled through the flexible reference picture management capabilities of HEVC. Essentially, the decoded pictures from other views are inserted into the RPLs of the current view for use in prediction processing. As a result, the RPLs include the temporal reference pictures of the current view that may be used to predict the current picture along with the inter-view reference pictures from neighboring views of the same time instance. With this design, block-level decoding modules remain unchanged, and only small changes to the high-level syntax are required as noted above, e.g., indication of the prediction dependency across views. The prediction is adaptive, so the best predictor among temporal and inter-view references (or an average employing bi-prediction or weighted prediction) can be selected on a block basis (e.g., in terms of rate-distortion cost).

In this way, more efficient compression of stereo content is achieved than by using so-called frame-compatible formats, which place the pictures from different views into a monoscopic frame (e.g., left/right, top/bottom), but cannot derive benefits from inter-view redundancy. Through the high-level syntax concepts described in Section IV-D, the multiview extension is backward compatible with monoscopic decoders which can simply extract the sub-bitstream of the base view. This part of the design could also be used for the hybrid architectures discussed in Section VII.

B. Multiview HEVC With Modified Block-Level Tools

To achieve higher compression efficiency, yet still maintain backwards compatibility with monoscopic video coded by HEVC, an alternative coding architecture could be designed to leverage the benefits of modified block-level coding tools. In such an architecture, and similar to the architecture described in previous sub-section, the base view could still be fully compatible with HEVC version 1 in order to extract monoscopic video, such that only the dependent views would use additional coding features. By block-level changes, it is possible to exploit the correlation of motion and residual data between views. Since scene objects projected to different viewpoints have similar motion and texture characteristics, identifying and exploiting such correlations could lead to substantial bit rate savings. For instance, in the context of the coding of multiple views, it is sometimes possible to infer some of the information used in the decoding process, e.g., motion vectors for a particular block, based on other available data, e.g., motion vectors from other blocks (see Fig. 4).

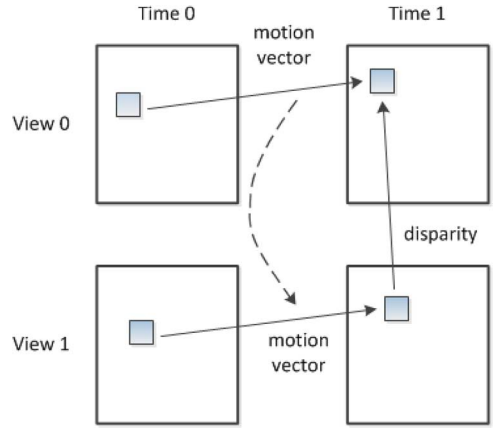


Fig. 4. Illustration of motion prediction between views, where the motion vector of view 1 is inferred from the motion vector of view 0 from corresponding blocks at time 1 based on the NBDV disparity between those blocks.

The JCT-3V has defined a reference test model and associated working draft text specification for a candidate extension design known as 3D-HEVC [28], [29] in order to perform study on advanced tools for coding multiple views. The basic design for 3D-HEVC originated from the proposal in [30], with further improvements and simplifications being implemented since then. No decision has yet been made for including these 3D-HEVC tools in an upcoming extension; however, the 3D-HEVC reference model captures the collective state of key proposals in the area for coordinated study and further consideration. The following paragraphs describe the most notable 3D-HEVC tools in more detail.

Neighboring Block-Based Disparity Vector Derivation: To identify the corresponding blocks of different views, neighboring block based disparity vector (NBDV) derivation is used in 3D-HEVC, which is designed in a way similar to AMVP and merge modes in HEVC (see Section II). However, disparity vectors are uniquely derived from neighboring blocks (depending on availability), so no additional bits are spent for signaling or refinement.

The basic idea of NBDV is to make use of available disparity vectors used for inter-view sample prediction of spatial and temporal neighboring blocks [31].

The spatial neighboring blocks are the same as those used in HEVC AMVP/merge modes, with the same order of block access as in merge: A_1, B_1, B_0, A_0, and B_2, as shown in Fig. 5. However, as it is highly possible that none of them uses inter-view references, temporal neighboring blocks are also checked [32], [33]. Once a disparity vector is identified, the disparity vector of the current block is derived to be the same as the disparity (motion) vector of the neighboring block and the whole NBDV process terminates. The disparity vector is used for identifying the reference block in the inter-view reference picture, as required in e.g., inter-view motion prediction and inter-view residual prediction. If no disparity vector is found from neighboring blocks, the NBDV process returns a zero disparity vector.

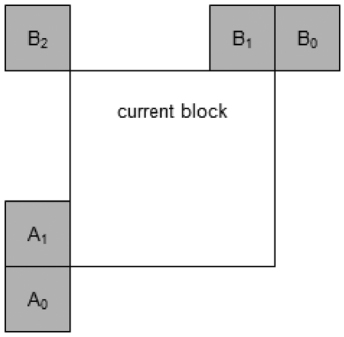


Fig. 5. Spatial neighboring blocks accessed for NBDV.

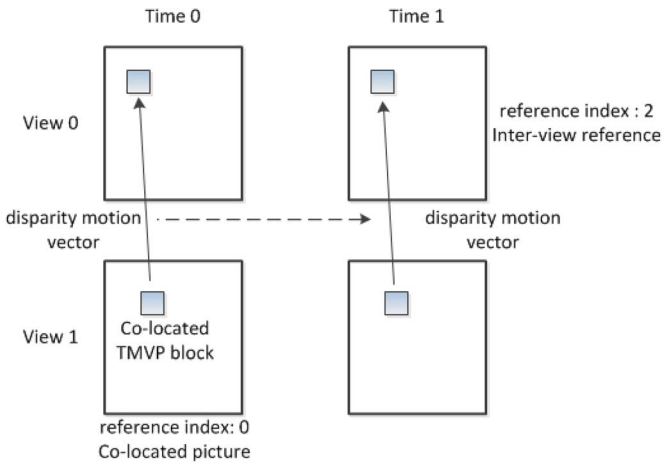


Fig. 6. Temporal motion vector prediction in 3D-HEVC. The target reference index of the TMVP candidate is changed from 0 to 2, so that TMVP candidate is made available by reusing the disparity motion vector.

Inter-View Motion Prediction: Inter-view motion prediction in 3D-HEVC is realized by introducing additional candidates into the list of the merge mode, whereas the AMVP mode has been kept unchanged. The merge list then contains six candidates (compared to five in the HEVC first edition). While the list still contains the candidates constructed as usual in HEVC, two additional candidates can be interspersed as described below.

The first candidate is the motion vector and corresponding reference picture index of the block found by NBDV in the inter-view reference picture, as shown in Fig. 4. This first candidate is called the inter-view candidate [34]. The second candidate is the disparity vector derived by NBDV with the inter-view reference picture index. The second candidate is inserted regardless of the availability of inter-view candidate [35]. Similar to the merge process in HEVC version 1, pruning is applied to additional candidates, by comparing with only the candidates from spatial neighbors denoted by A_1 and B_1, as shown in Fig. 5 [35].

The TMVP candidate is also modified to accommodate the case when the target reference index (which is always 0 in the HEVC first edition specification) and the reference index of the co-located block correspond to different types of references—i.e., when one is a temporal reference picture and the other is an inter-view reference picture. In this case, to improve the coding efficiency, the target reference index of the TMVP candidate is changed to align with that of the co-located block [36]. As shown in Fig. 6, for the current block of view 1 at time 1, its co-located TMVP block contains a disparity motion vector and the reference index 0 corresponds to a temporal reference of the current picture, therefore the TMVP candidate would usually be considered as unavailable. In 3D-HEVC, the candidate is considered as available by reusing the motion vector but changing the target reference index to 2, which corresponds to the inter-view reference picture.

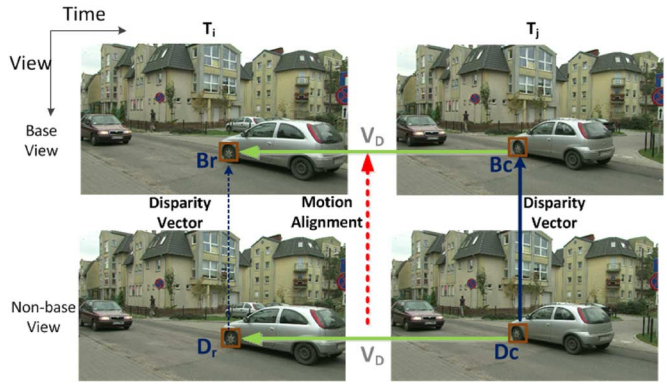


Fig. 7. Prediction structure of advanced residual prediction.

Inter-View Residual Prediction: Advanced residual prediction (ARP) was designed to take advantage of the correlation between the motion-compensated residual signal of two views [37].

As shown in Fig. 7, motion compensation is performed for the block D_c in the current non-base view, using the motion vector V_D. First, an inter-view reference block B_c is identified by the NBDV vector. Motion compensation (using V_D) is invoked between the reconstructed B_c and the corresponding reconstructed B_r of the base view. The predicted residual is added to the prediction signal (motion compensation from the block D_r). As the same vector V_D is used, the residual signal of the current block can be more precisely predicted. When ARP is enabled, the prediction of the residue can be weighted by 0.5 or 1.

Since additional motion compensation at the base (reference) view may require a significant increase of memory accesses and calculations, several ways to make the design more practical with a minor sacrifice of coding efficiency have been identified [37], e.g., bi-linear filters are used for the motion compensation of both the reference block and the current block.

Illumination Compensation: Prediction may fail when cameras capturing the same scene are not calibrated in color transfer or by lighting effects. To deal with this issue, a technique known as illumination compensation has been developed to improve the coding efficiency for blocks predicted from inter-view reference pictures [38]. This mode only applies to blocks which are predicted by an inter-view reference picture.

For the current PU, its neighboring samples in the top neighboring row and left neighboring column and the corresponding neighboring samples of the inter-view reference block are the input parameters used to form a linear model characterized by a scaling factor a and an offset b. The values of a and b are determined by a least-squares solution, considering the constraint that a should be close to 1. The corresponding neighboring samples in the reference view are identified by the disparity motion vector of the current PU, as shown in Fig. 8.

After disparity motion compensation from an inter-view reference, the gain/offset model is applied to each value, scaling it by a, and adding the offset b.

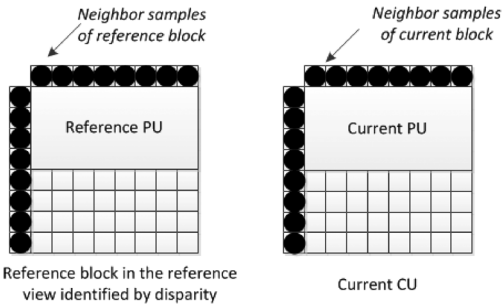


Fig. 8. Neighboring samples for the derivation of illumination compensation parameters.

C. Multiview HEVC With Depth

To investigate video-plus-depth compression formats, the 3D-HEVC model also includes compression of depth map information. For the efficient compression of 3D video data with multiple video and depth components, a number of coding tools are investigated to exploit dependencies among the components. It is assumed that the first video component is independently coded by a conventional 2D HEVC, to provide compatibility with existing 2D video services. For each additional 3D video component, i.e., the video component of a dependent view as well as the depth maps, additional coding tools can be employed. Thus, a 3D video encoder can select the best coding method for each block from a set of conventional 2D coding tools and additional coding tools, some of which are described in the following subsections. It is noted that the additional texture coding tools described in this section use depth information, while the ones described in Section IV-B do not.

Beyond the advanced multiview coding tools described in the previous section, video-plus-depth compression can make use of new coding tools specifically designed to exploit the unique characteristics of depth; and view synthesis prediction, which uses depth information for texture coding.

Depth map images are characterized by large homogeneous areas and sharp edges. The preservation of edges in depth maps is important since inaccurate edge reconstruction may lead to significant objective distortion and perceptual artifacts for synthesized views. Another interesting characteristic of depth images is that the edge information in the depth image, which corresponds to depth discontinuities in the scene, is typically a subset of the edge information that could be extracted from the corresponding texture component.

Two major coding modules have been proposed: partition-based depth intra coding and motion parameter inheritance. In addition, as depth is generally characterized by sharp edges, the interpolation filters used for motion compensation in HEVC version 1 have not been found to be beneficial in preserving the edges in depth map. Furthermore, motion compensation is applied with integer-sample accuracy for depth map coding, and encoder optimizations are applied by turning off in-loop filtering processes, including the DBF and SAO loop filter, for depth coding. In addition, view synthesis prediction has been proposed for the coding of texture using depth. These tools are further described below.

Partition-Based Depth Intra Coding: To better represent the depth information, several depth-specific coding tools have been introduced in the current 3D-HEVC design, all allowing separating of depth blocks into non-rectangular partitions. Such partition-based depth intra coding modes include depth modeling modes (DMM) [39], region boundary chain (RBC) coding [40] and simplified depth coding (SDC) [41]. In all of these modes, each depth PU can be divided as one or two parts, where each part is represented by a constant value, i.e., DC value, as illustrated in Fig. 9. The DC value for each partition is predicted using neighboring reference samples and a residual value may be further coded to compensate the prediction error.

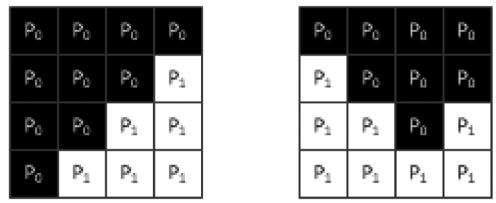


Fig. 9. Examples of depth PU partitioning in depth coding.

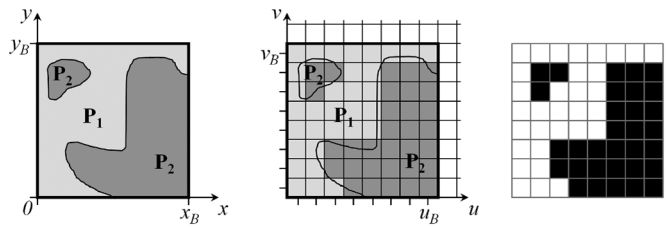


Fig. 10. Contour partition of a block: continuous (left) and discrete signal space (middle) with corresponding partition pattern (right) [29].

Although both DMM and RBC partition a depth PU into two parts, they differ on the representation of the partitioning pattern. In DMM, two types of partitioning patterns are applied, including the wedge-shaped pattern and a contour pattern. The wedge-shaped pattern segments a depth PU with a straight line as shown in Fig. 9(a). Different from the wedge-shaped patterns, RBC represents the partitioning pattern explicitly using a series of connected chains, where each chain is a connection of one sample and one of its eight-connectivity samples, indexed from 0 to 7, so the partition boundary can be different from a straight line, as shown in Fig. 9(b). A contour pattern can support two irregular partitions, each of which may contain separate sub-regions, as shown in Fig. 10. The contour (partition boundary) of a depth block is determined by analyzing the co-located texture block. Moreover, different methods for signaling the partitioning pattern are used in wedge modes, including 1) explicit signaling of a wedge-shaped pattern index selected from a pre-defined set of wedge-shaped patterns; and 2) deriving the partitioning pattern based on the reconstructed co-located texture block.

SDC is built on top of the DMM and RBC and featured by adding: 1) one partition per PU which is used to model smooth regions; 2) skipping the transform and quantization process and coding the residual samples directly; 3) a depth look-up table (DLT) for conversion of depth values to reduce the dynamic range of depth representation, especially in case the depth map doesn't use the full range of available depth values, (typically the range from 0 to 255) [41].

Motion Parameter Inheritance: In 3D-HEVC, inheritance of the texture's motion parameters for depth data is achieved by adding one more merge candidate to the merge list of the current depth block, in addition to the usual spatial and temporal

candidates from the HEVC version 1 merge mode. The extra candidate is generated from the motion information of the co-located texture block [42].

View Synthesis Prediction (VSP): VSP is an effective approach to reduce the inter-view redundancy, whereby the depth information is used to warp texture data from a reference view to the current view such that a predictor for the current view can be generated [43].

In depth-based rendering, view synthesis is typically implemented as forward warping, where the depth image of a given view is used to warp it into a synthetic view. In the context of VSP coding, this is not practical, as it would require first generating an entire synthetic picture and storing it in the reference picture buffer before encoding or decoding the current picture, which would lead to a significant complexity increase at the decoder. Instead, a block-based backward VSP (BVSP) scheme has been introduced in the 3D-HEVC design, where the depth information of the current block is inferred to determine the corresponding pixels in the inter-view reference picture [44], [45]. Since texture is typically coded prior to depth, the depth of the current block can be estimated using the same NBDV process described earlier. This way, a depth block can be inferred by assuming that the current block has the same depth (and inter-view displacement vector) as the neighboring block. The depth block to which the displacement vector points in the reference view can be used for backward warping in the current view. As an extension of this, the maximum depth from this depth block is converted to a disparity vector, and then this refined disparity vector would be used to do motion inheritance and perform the BVSP operation [46]. The BVSP process described above can be designed to use the motion compensation engine of HEVC version 1, but with smaller blocks, e.g., 4×4 blocks for each PU, each with a different disparity (motion) vector.

In the current 3D-HEVC design, the usage of the VSP mode is signaled through a view synthesis prediction (VSP) merge candidate in the merge candidate list. A VSP merge candidate is derived to have a tag indicating the usage of BVSP, therefore other normal candidates are tagged to not to use BVSP during the merge candidate generation process. Such a VSP merge candidate contains a motion vector which is a disparity (motion) vector and a reference index indicating the inter-view reference picture from which the current block is predicted [43]. Note that the disparity vector (derived from NBDV) of this candidate is used further to determine refined disparity vectors for each smaller block (e.g., 4×8 or 8×4) within the PU as described above.

D. Performance Evaluation

To evaluate the compression efficiency of the different architectures and coding techniques, simulations were conducted using the reference software and experimental evaluation methodology that has been developed and is being used by the standardization community [47], [48]. In the experimental framework, multiview video and corresponding depth are provided as input, while the decoded views and additional views synthesized at selected positions are generated as output. As defined in the common test conditions (CTC), the base view (view 0) is coded as the center view of each input test sequence and two non-base (dependent) views positioned to the left and right of the center view are also coded; these are denoted as view 1 and view 2. The total results for the two-view stereo case are generated based on the average luma PSNR values of the base view and view 1 and corresponding bit rates for these two views, while the total results for the three-view case are generated by average luma PSNR values and bit rates for all three views.

TABLE VIII
BIT RATE REDUCTION OF MV-HEVC VS. SIMULCAST

Sequence	View 1 only	View 2 only	Total 2-view	Total 3-view
Balloons	53.9%	49.7%	23.5%	31.5%
Kendo	52.5%	47.2%	23.3%	30.4%
Newspaper	56.4%	54.4%	23.3%	33.2%
GT_Fly	82.0%	81.3%	38.7%	52.4%
Poznan_Hall2	53.5%	53.9%	23.3%	32.8%
Poznan_Street	69.7%	69.4%	29.7%	41.4%
Undo_Dancer	74.5%	76.0%	34.0%	47.3%
1024×768	54.2%	50.4%	23.4%	31.7%
1920×1088	69.9%	70.2%	31.4%	43.5%
Average	63.2%	61.7%	28.0%	38.4%

The first set of simulations provides a comparison between MV-HEVC and HEVC simulcast coding of two or three views, and coding of depth maps is not considered in this case—i.e., PSNR and bit rate values are calculated based on texture information only. The results are shown in Table VIII. The results indicate that MV-HEVC provides an average bit rate savings of 28% for the two-view (stereo) case and 38% for the three-view case, relative to simulcast, which demonstrates the effectiveness of the inter-view sample prediction of texture. The bit rate savings for predictively coding the dependent views (views 1 and 2) from the base view (view 0) relative to independently coding these views are also included. It is shown that each dependent view can be coded with more than a 60% reduction in bit rate. The complexity is not increased compared to simulcast, since in multiview applications all views would need to be decoded anyway.

The second set of simulations reports the performance of the additional block-level coding tools that are supported by the current 3D-HEVC design, considering both texture and depth map coding. Specifically, bit rate savings are reported for the three-view case relative to HEVC simulcast as well as relative to MV-HEVC, where in the latter case an independent HEVC encoding is operated for the depth maps. It is noted that the current software for 3D-HEVC uses a view synthesis optimization (VSO) tool [39] which codes the depth information such that the trade-off between bit rate and synthesis quality is optimized. Furthermore, in contrast to the first set of simulations, where only the compression efficiency of multiview texture was being evaluated, this second set of simulations must account for the quality of the depth map coding. To do this, the PSNR of synthesized views are reported [47], [48] since any improvements in depth map coding (either from the depth coding tools or encoder optimization) would be reflected by this measure. The bit rate is calculated as the total coded bits for both texture and depth components. The results for the second set of simulations are reported in Table IX and indicate that 3D-HEVC with VSO turned off provides an average bit rate savings of 41% relative to HEVC simulcast coding, i.e., where all texture and depth views are coded independently. Furthermore, when compared to MV-HEVC, which uses inter-view sample

TABLE IX
BIT RATE REDUCTION OF 3D-HEVC (3-VIEW CASE)

Sequence	3D-HEVC (VSO OFF) vs. Simulcast	3D-HEVC (VSO OFF) vs. MV-HEVC	3D-HEVC (VSO ON) vs. MV-HEVC
Balloons	34.2%	12.6%	25.1%
Kendo	31.3%	12.5%	30.9%
Newspaper	34.7%	9.8%	29.8%
GT_Fly	54.1%	21.0%	32.9%
Poznan_Hall2	36.6%	14.3%	30.4%
Poznan_Street	39.6%	9.3%	19.5%
Undo_Dancer	56.8%	29.0%	45.5%
1024×768	33.4%	11.6%	28.6%
1920×1088	46.8%	18.4%	32.1%
Average	41.0%	15.5%	30.6%
Decoding time	111%	118%	118%

prediction for both texture and codes each depth view independently with HEVC, 3D-HEVC can achieve an average bit rate savings of 15.5% with VSO turned off for both configurations. Comparable bit rate savings are observed when VSO is turned on for both MV-HEVC and 3D-HEVC. Additionally, when enabling VSO for 3D-HEVC only, an average bit rate savings of 30.6% can be achieved. It should, however, be observed that MV-HEVC could also potentially save bit rate for the depth information, by not encoding details of the depth map whenever they are not relevant for texture synthesis.

The complexity of 3D-HEVC in terms of decoder run time is also evaluated. As reported in Table IX, an average increase in run time of 11% and 18% is incurred relative to the simulcast and MV-HEVC references, respectively.

VII. HYBRID ARCHITECTURES

From a pure compression efficiency point of view, it is always best to use the most advanced compression technologies. However, when introducing new services (such as higher resolution video or 3D video), providers must also consider the capabilities of existing receivers and establish an appropriate transition plan. Considering that most terrestrial broadcast systems are based on H.262/MPEG-2 or AVC, it may not be easy to simply switch technologies for all transmission environments in the near-term.

One solution to this problem is to continue transmitting the existing service in the legacy format, and deliver an HEVC enhancement layer as a supplemental stream for an upgraded service. The HEVC enhancement layer could be an additional spatial scalability layer that enables a higher resolution video output or an additional view to support stereo services.

The obvious advantage is that backward compatibility with the existing system is provided with significant bandwidth savings relative to simulcast in the legacy format. One drawback of this approach is that it requires legacy technologies to operate synchronously with the newer one, where the decoding and output time for each picture must be synchronized; this may pose implementation challenges for certain receiver designs. Also, in the case of 3D video, the 3D program becomes tightly coupled with the 2D program; in this way, it is not possible to have independent 2D and 3D content programs, which is sometimes desired from the content-production and user-experience perspectives. Nevertheless, stereoscopic broadcasting trials of hybrid H.262/MPEG-2 and AVC based systems are currently being conducted in Korea, and a hybrid transmission format with one view coded as H.262/MPEG-2 and another view coded with AVC has recently been standardized by the ATSC [48]. Similar hybrid formats involving HEVC will also be possible. Moreover, the high-level syntax defined in the HEVC extensions supports the capability to signal that the base layer/view is encoded with AVC rather than HEVC.

In the context of depth-based 3D formats, there are clearly many variations that could be considered. For instance, in an AVC-compatible framework, the base view would be coded with AVC, while additional texture views and supplemental depth videos could be encoded with HEVC. In general, the hybrid codec variations that are supported or deployed would be determined by specific application delivery requirements.

VIII. CONCLUSIONS AND OUTLOOK

While the first version of HEVC is sufficient to cover a wide range of applications, needs have been identified to enhance the standard in several ways. As can be seen from the information presented in this paper, the development of these extensions in the relevant standardization groups has been an active area of recent research and development. These extensions will further enhance the utility of the HEVC standard and broaden its range of applications. While the standardization of the extensions discussed in this paper is not yet fully completed, the basic design is in place for several of these extensions, and the state of work in the committees represents the current state of the art for developments in video coding and its applications. Much of the technology described in this paper will be finalized as standard extensions within 2014, and further extension work beyond this timeframe is planned.

ACKNOWLEDGMENT

The authors thank Jizheng Xu and Bin Li of Microsoft Research for the simulation results reported in Tables I–III. The authors thank all the contributors to the work of the ITU-T Video Coding Experts Group, the ISO/IEC Moving Picture Experts Group, the JCT-VC and the JCT-3V, as their important contributions to the HEVC standard are the technical substance described in this paper. The manuscript reviewers are also thanked for their helpful comments.

REFERENCES

[1] *High Efficiency Video Coding*, Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013.
[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[3] *Advanced Video Coding for Generic Audiovisual Services*, Rec. ITU-T H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), 2012.
[4] D. Flynn, J. Sole, and T. Suzuki, "Range extensions draft 4," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-N1005, 14th Meeting: Vienna, AT*, Jul. 25–Aug. 2 2013.
[5] H. Nakamura, M. Ueda, S. Fukushima, and T. Kumakura, "Unified intra prediction angles for 4:2:2 chroma format," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-M0127, 13th Meeting: Incheon, KR*, Apr. 18–26, 2013.
[6] 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.
[7] Y. Chen, Y.-K. Wang, K. Ugur, M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, Jan. 2009, Article 8.

[8] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[9] J. Boyce, S. Wenger, W. Jang, D. Hong, Y.-K. Wang, and Y. Chen, "High level syntax hooks for future extensions," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-H0388, 8th Meeting: San José, CA, USA*, Feb. 1–10, 2012.

[10] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Systems Video Technol.*, vol. 13, no. 7, pp. 645–656, Jul. 2003.

[11] R. Sjöberg, Y. Chen, A. Fujibayashi, M. M. Hannuksela, J. Samuelsson, T. K. Tan, Y.-K. Wang, and S. Wenger, "Overview of HEVC high-level syntax and reference picture management," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1857–1869, Dec. 2012.

[12] M. M. Hannuksela and Y.-K. Wang, "AHG9: Operation points in VPS and nesting SEI," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-K0180, 11th Meeting: Shanghai, China*, Oct. 10–19, 2012.

[13] J. Boyce, "AHG9: Operation points in VPS and nesting SEI," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-L0180, 12th Meeting: Geneva, Switzerland*, Jan. 14–23, 2013.

[14] J. Boyce, S. Wenger, W. Jang, D. Hong, Y.-K. Wang, and Y. Chen, "Information for scalable extension high layer syntax," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-H0386, 8th Meeting: San José, CA, USA*, Feb. 1–10, 2012.

[15] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and transport interface of SVC," *Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1049–1063, Sep. 2007.

[16] M. M. Hannuksela, "AHG10 hooks for scalable coding: Video parameter set design," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-J0075, 10th Meeting: Stockholm, Sweden*, July 11–20, 2012.

[17] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, "Scalable high efficiency video coding draft 3," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-N1008, 14th Meeting: Vienna, Austria*, July 25–Aug. 2 2013.

[18] J. Boyce, D. Hong, and S. Wenger, "Extensible high layer syntax for scalability," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-E279, 5th Meeting: Geneva, China*, Mar. 16–23, 2011.

[19] A. Luthra, J.-R. Ohm, and J. Ostermann, "Requirements of the scalable enhancement of HEVC," ISO/IEC JTC 1/SC 29/WG 11 (MPEG) document N12956, Jul. 2012.

[20] "Requirements for high efficiency video coding (HEVC) scalability extension," ITU-T SG16, Geneva, Switzerland, Mar. 2011, Annex Q6.A to doc. TD 190 (WP 3/16).

[21] G. J. Sullivan and J.-R. Ohm, "Joint call for proposals on scalable video coding extensions of high efficiency video coding (HEVC)," ITU-T Study Group 16 Video Coding Experts Group (VCEG) document VCEG-AS90 and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) document N12957, Jul. 2012.

[22] E. Alshina, H. Lakshman, J. Dong, J. Chen, and A. Luthra, "Suggested up-sampling filter design for tool experiments on HEVC scalable extension," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-K0378, 11th Meeting: Shanghai, China*, Oct. 10–19, 2012.

[23] J. Dong, Y. He, Y. He, G. McClellan, E.-S. Ryu, X. Xiu, and Y. Ye, "Description of scalable video coding technology proposal by InterDigital," in *Communications Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-K0034, 11th Meeting: Shanghai, China*, Oct. 10–19, 2012.

[24] X. Xiu, Y. He, Y. He, and Y. Ye, "TE C5: Motion field mapping," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-L0052, 12th Meeting: Geneva, Switzerland*, Jan. 14–23, 2013.

[25] J. Chen, V. Seregin, L. Guo, and M. Karczewicz, "Non-TE5: On motion mapping in SHVC," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-L0336, 12th Meeting: Geneva, Switzerland*, Jan. 14–23, 2013.

[26] X. Li, J. Boyce, P. Onno, and Y. Ye, "Common SHM test conditions and software reference configurations," in *Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-M1009, Incheon, Korea*, Apr. 20–26, 2013.

[27] G. Tech, K. Wegner, Y. Chen, M. M. Hannuksela, and J. Boyce, "MV-HEVC draft text 5," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-E1004, 5th Meeting: Vienna, Austria*, Jul. 27–Aug. 2 2013.

[28] G. Tech, K. Wegner, Y. Chen, and S. Yea, "3D-HEVC draft text 1," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-E1001, 5th Meeting: Vienna, Austria*, July 27–Aug. 2 2013.

[29] L. Zhang, G. Tech, K. Wegner, and S. Yea, "3D-HEVC test model 5," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-E1005, 5th Meeting: Vienna, Austria*, Jul. 27–Aug. 2 2013.

[30] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Müller, H. Rhee, G. Tech, M. Winken, and T. Wiegand, "Description of 3D video technology proposal by Fraunhofer HHI (HEVC compatible; configuration A)," ISO/IEC JTC 1/SC 29/WG 11 (MPEG) document m22570, Nov. 2011.

[31] L. Zhang, Y. Chen, and M. Karczewicz, "Disparity vector based advanced inter-view prediction in 3D-HEVC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 1632–1635.

[32] J. Kang, Y. Chen, L. Zhang, and M. Karczewicz, "3D-CE5.h related: Improvements for disparity vector derivation," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-B0047, 2nd Meeting: Shanghai, China*, Oct. 13–19, 2012.

[33] J. Sung, M. Koo, and S. Yea, "3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-A0126, 1st Meeting: Stockholm, Sweden*, Jul. 16–20, 2012.

[34] J. An, Y. W. Chen, J. L. Lin, Y. W. Huang, and S. Lei, "3D-CE5.h related: Inter-view motion prediction for HEVC-based 3D video coding," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-A0049, 1st Meeting: Stockholm, Sweden*, Jul. 16–20, 2012.

[35] L. Zhang, Y. Chen, and L. Liu, "3D-CE5.h: Merge candidates derivation from disparity vector," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-B0048, 2nd Meeting: Shanghai, China*, Oct. 13–19, 2012.

[36] L. Zhang, Y. Chen, and M. Karczewicz, "3D-CE5.h: Improved temporal motion vector prediction for merge," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-C0047, 3rd Meeting: Geneva, Switzerland*, Jan. 17–23, 2013.

[37] L. Zhang, Y. Chen, X. Li, and M. Karczewicz, "CE4: Advanced residual prediction for multiview coding," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-D0117, 4th Meeting: Incheon, Korea*, Apr. 20–26, 2013.

[38] H. Liu, J. Jung, J. Sung, J. Jia, and S. Yea, "3D-CE2.h: Results of illumination compensation for inter-view prediction," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-B0045, 2nd Meeting: Shanghai, China*, Oct. 13–19, 2012.

[39] K. Müller, P. Merkle, G. Tech, and T. Wiegand, "3D video coding with depth modeling modes and view synthesis optimization," in *Proc. Asia-Pacific Signal and Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Oct. 2012.

[40] J. Heo, E. Son, and S. Yea, "3D-CE6.h: Region boundary chain coding for depth-map," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-A0070, 1st Meeting: Stockholm, Sweden*, Jul. 16–20, 2012.

[41] F. Jäger, "3D-CE6.h results on simplified depth coding with an optional depth lookup table," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-B0036, 2nd Meeting: Shanghai, China*, Oct. 13–19, 2012.

[42] Y.-W. Chen, J.-L. Lin, Y.-W. Huang, and S. Lei, "3D-CE3.h results on removal of parsing dependency and picture buffers for motion parameter inheritance," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-C0129, 3rd Meeting: Geneva, Switzerland*, Jan. 17–23, 2013.

[43] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," *Signal Processing: Image Commun.*, vol. 24, no. 1–2, pp. 89–100, Jan. 2009.

[44] D. Tian, F. Zou, and A. Vetro, "CE1.h: Backward view synthesis prediction using neighbouring blocks," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-C0152, 3rd Meeting: Geneva, Switzerland*, Jan. 17–23, 2013.

[45] D. Tian, F. Zou, and A. Vetro, "Backward view synthesis prediction for 3D-HEVC," in *Proc. IEEE Int. Conf. Image Process.*, Melbourne, Australia, Sep. 2013.

[46] Y.-L. Chang, C.-L. Wu, Y.-P. Tsai, and S. Lei, "3D-CE5.h related: Depth-oriented Neighboring Block Disparity Vector (DoNBDV) with virtual depth," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-B0090, 2nd Meeting: Shanghai, China*, Oct. 13–19, 2012.

[47] D. Rusanovskyy, K. Mueller, and A. Vetro, "Common test conditions of 3DV core experiments," in *Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-E1100, 5th Meeting: Vienna, Austria*, Jul. 27–Aug. 2 2013.

[48] Advanced Television Systems Committee (ATSC), Washington, D.C., "3D-TV Terrestrial Broadcasting, Part 2-SCHC Using Real-Time Delivery," Doc. A/104:2012, Dec. 26, 2012.

Gary J. Sullivan (S'83–M'91–SM'01–F'06) received B.S. and M.Eng. degrees in Electrical Engineering from the University of Louisville in 1982 and 1983, respectively, and Ph.D. and Engineer's degrees in electrical engineering from the University of California at Los Angeles in 1991. He has been a longstanding chairman or co-chairman of various video and image coding standardization activities in ITU-T VCEG and ISO/IEC MPEG and JPEG. He is best known for leading the development of the "Advanced Video Coding" (AVC) standard ITU-T H.264 | ISO/IEC 14496-10 and its Scalable Video Coding (SVC) and 3D/Stereo/Multiview Video Coding (MVC) extensions. More recently, he led the development of the new "High Efficiency Video Coding" (HEVC) standard ITU-T H.265 | ISO/IEC 23008-2.

He is a Video and Image Technology Architect in the Windows division of Microsoft Corporation. At Microsoft he has been the originator and lead designer of the DirectX Video Acceleration (DXVA) 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 loss resilient video coding.

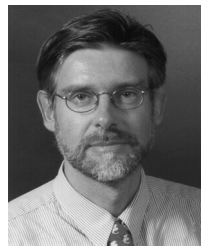
Dr. Sullivan is a Fellow of the IEEE and SPIE. He has received the IEEE Masaru Ibuka Consumer Electronics Award, the IEEE Consumer Electronics Engineering Excellence Award, the IEEE Circuits and Systems CSVT Transactions Best Paper Award, the INCITS Technical Excellence Award, the IMTC Leadership Award, and the University of Louisville J. B. Speed Professional Award in Engineering. The team efforts that he has led have been recognized by an ATAS Primetime Emmy Engineering Award and a pair of NATAS Technology & Engineering Emmy Awards.

Jill M. Boyce received a B.S. in Electrical Engineering from the University of Kansas in 1988 and an M.S.E. in Electrical Engineering from Princeton University in 1990. She is Director of Algorithms at Vidyo, Inc. where she leads video and audio coding and processing algorithm development. She represents Vidyo at the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC 1/SC 29/WG 11 (MPEG), where she is an editor of the Working Draft and Test Model of the Scalability HEVC Extension. She was formerly VP of Research and Innovation Princeton for Technicolor, formerly Thomson. She was formerly with Lucent Technologies Bell Labs, AT&T Labs, and Hitachi America. She was Associate Editor from 2006 to 2010 of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. She is the inventor of over 100 granted U.S. patents, and has published more than 40 papers in peer-reviewed conferences and journals. She is an IEEE Senior Member.

Ying Chen (M'05–SM'11) received a B.S. in Applied Mathematics and an M.S. in Electrical Engineering & Computer Science, both from Peking University, in 2001 and 2004 respectively. He received his PhD in Computing and Electrical Engineering from Tampere University of Technology (TUT), Finland, in 2010.

He is currently a Senior Staff Engineer/Manager at Qualcomm Incorporated, San Diego, CA, USA. Dr. Chen joined Qualcomm in Mar. 2009. His earlier working experiences include Researcher in TUT and Nokia Research Center, Finland from 2006 to Feb. 2009 and Research Engineer in Thomson Corporate Research, Beijing, from 2004 to 2006.

Dr. Chen has been actively contributing to MPEG, JVT, JCT-VC, and JCT-3V, on Scalable Video Coding (SVC), Multiview Video Coding (MVC), and 3D Video (3DV) Coding extensions of H.264/AVC, as well as high-level syntax (HLS), scalable extension, and 3DV extension of HEVC. Dr. Chen has also been involved in standardization activities of MPEG systems, including the ISO Base Media File Format, H.222.0/MPEG-2 Systems and DASH (Dynamic Adaptive Streaming over HTTP). Dr. Chen has served as the editor of MVC reference software, co-editors of H.264/AVC based 3DV standards and co-editors of the multiview HEVC (MV-HEVC) standard and 3D-HEVC. Dr. Chen has co-authored more than two hundred standardization contribution documents to JVT, JCT-VC, JCT-3V, and MPEG and around 40 academic papers in the fields of image processing, video coding, and video transmission.



Jens-Rainer Ohm (M'92) received the Dipl.-Ing. degree in 1985, the Dr.-Ing. degree in 1990, and the habil. degree in 1997, all from Technical University of Berlin (TUB), Germany. From 1985 to 1995, he was a research associate with the Institute of Telecommunication at TUB. From 1996 to 2000, he was project coordinator at Heinrich Hertz Institute (HHI) in Berlin. In 2000, he was appointed full professor and since then has held the chair position of the Institute of Communication Engineering at RWTH Aachen University, Germany. His research and teaching activities cover the areas of motion-compensated, stereoscopic and 3-D image processing, multimedia signal coding, transmission and content description, audio signal analysis, as well as various topics of signal processing and digital communication systems.

Since 1998, he has participated in the work of the Moving Picture Experts Group (MPEG). He has been chairing/co-chairing various standardization activities in video coding, namely the MPEG Video Subgroup since 2002, the Joint Video Team (JVT) of MPEG and ITU-T SG 16 VCEG from 2005 to 2009, and currently, the Joint Collaborative Teams on Video Coding (JCT-VC) and on 3D Video Coding Extensions (JCT-3V).

Prof. Ohm has authored textbooks on multimedia signal processing, analysis and coding, on communication engineering and signal transmission, as well as numerous papers in the fields mentioned above. He is member of various professional organizations including IEEE, VDE/ITG, EURASIP and AES.

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 Manager at Sharp Laboratories of America, Camas, WA, where he leads groups performing research on 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.

Anthony Vetro (S'92–M'96–SM'04–F'11) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY. He joined Mitsubishi Electric Research Labs, Cambridge, MA, in 1996, where he is currently a Group Manager responsible for research and standardization on video coding, as well as work on display processing, information security, sensing technologies, and speech/audio processing. He has published more than 150 papers in these areas. He has also been an active member of the ISO/IEC and ITU-T standardization committees on video coding for many years, where he has served as an ad-hoc group chair and editor for several projects and specifications. He was a key contributor to the Multiview Video Coding extension of the H.264/MPEG-4 AVC standard, and current serves as Head of the U.S. delegation to MPEG.

Dr. Vetro is also active in various IEEE conferences, technical committees, and editorial boards. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, and as a member of the Editorial Boards of IEEE MultiMedia and IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He served as Chair of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and on the steering committees for ICME and the IEEE TRANSACTIONS ON MULTIMEDIA. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2010–2013) and IEEE Signal Processing Magazine (2006–2007) and, and later served as a member of the Editorial Board (2009–2011). He also served as a member of the Publications Committee of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS (2002–2008). He has also received several awards for his work on transcoding, including the 2003 IEEE Circuits and Systems CSVT Transactions Best Paper Award. He is a Fellow of IEEE.