Overview of the Versatile Video Coding (VVC) Standard and its Applications

Benjamin Bross, *Member, IEEE*, Ye-Kui Wang, Yan Ye, *Senior Member, IEEE*,
Shan Liu, *Senior Member, IEEE*, Jianle Chen, *Senior Member, IEEE*,
Gary J. Sullivan, *Fellow, IEEE*, and Jens-Rainer Ohm, *Member, IEEE*

***Abstract*—Versatile Video Coding (VVC) was finalized in July 2020 as the most recent international video coding standard. It was developed by the Joint Video Experts Team (JVET) of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) to serve an ever-growing need for improved video compression as well as to support a wider variety of today's media content and emerging applications. This paper provides an overview of the novel technical features for new applications and the core compression technologies for achieving significant bit rate reductions in the neighborhood of 50% over its predecessor for equal video quality, the High Efficiency Video Coding (HEVC) standard, and 75% over the currently most-used format, the Advanced Video Coding (AVC) standard. It is explained how these new features in VVC provide greater versatility for applications. Highlighted applications include video with resolutions beyond standard- and high-definition, video with high dynamic range and wide color gamut, adaptive streaming with resolution changes, computer-generated and screen-captured video, ultralow-delay streaming, 360° immersive video, and multilayer coding e.g., for scalability. Furthermore, early implementations are presented to show that the new VVC standard is implementable and ready for real-world deployment.**

***Index Terms*—Video coding, Video compression, Standards, H.266, VVC, H.265, HEVC, MPEG, VCEG, JVET**

I. Introduction

VVC, or Versatile Video Coding [1], standardized in ITU-T as Recommendation H.266 and in ISO and IEC as International Standard 23090-3 (MPEG-I Part 3), is the new generation of international video coding standard jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). Its predecessor, the High-Efficiency Video Coding (HEVC) standard [2], [3] (a.k.a. H.265 and MPEG-H Part 2) was finalized in 2013, offering about 50% bit rate reduction over the previous Advanced Video Coding (AVC) standard [4] (a.k.a. H.264 and MPEG-4 Part 10). The increased coding efficiency of HEVC enabled broadcast and streaming of 4K video with increased fidelity and with high dynamic range (HDR) and wide color gamut (WCG). Currently, along with continuing major increases in the reach and speed of broadband internet services, the share of internet-based video in global data traffic is currently about 80% and still continuing to grow [5]. Additionally, the proportion of household TV sets with 4K resolution is projected to reach two thirds by 2023 [6], and these higher resolution TVs require higher-quality video content in order to reach their full potential. This illustrates the need for even more efficient compression than the HEVC standard, and it motivated the ITU-T's VCEG and ISO/IEC's MPEG to join forces again in 2015 to explore new video coding technologies with higher coding efficiency by forming a joint group called the Joint Video Exploration Team (JVET).

Two years later, in 2017, the exploration activities resulted in the Joint Exploration Test Model (JEM) [7] that had already demonstrated more than 30% bit rate reduction compared to HEVC. This was considered as sufficient evidence to start a new standard development effort, so the Joint Video Exploration Team was converted to the Joint Video Experts Team (with the same JVET acronym) and a Joint Call for Proposals for new video coding technology was issued in October 2017. The Call attracted the submission of proposals from 33 organizations for the coding of three categories of video content: standard dynamic range (SDR), HDR, and 360º video [8], [9], [10]. A subjective evaluation test in April 2018 showed that all submissions were superior to HEVC in terms of subjective quality in most test cases, and several submissions were superior to the JEM in a relevant number of cases [11]. Based on the best-performing compression technology elements identified from the submissions, the formal project for development of the VVC standard started in April 2018, and the first drafts of the specification document and the software for the VVC test model (VTM) were generated in the same month.

From the beginning, VVC was designed not only to provide a substantial bit rate reduction compared to its predecessor (HEVC) but also to be highly versatile, i.e., to cover all current and emerging media needs. These include video beyond

B. Bross is with Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, Germany (e-mail: first.lastname@hhi.fraunhofer.de).
Y.-K. Wang is with Bytedance, San Diego, CA, USA (e-mail: yekui.wang@bytedance.com).
Y. Ye is with Alibaba Group U.S., Sunnyvale, CA, USA (e-mail: yan.ye@alibaba-inc.com).
S. Liu is with Tencent, Palo Alto, CA, USA (email: shanl@tencent.com).
J. Chen is with Qualcomm, San Diego, CA, USA (email: cjianle@qti.qualcomm.com)
G. J. Sullivan is with Microsoft Corporation, Redmond, WA, USA (email: garysull@microsoft.com).
J.-R. Ohm is with the Institute of Communications Engineering, RWTH Aachen University, Aachen, Germany (email: ohm@ient.rwth-aachen.de).

standard- and high-definition with SDR, including even higher resolution (up to 8K or larger), HDR and WCG; computer-generated or screen content, as occurs especially in computer screen sharing and gaming; 360^o video for immersive and augmented reality; and applications requiring ultralow delay such as wireless display and online gaming.

Another difference compared to previous video coding standards is the handling of video usability information (VUI) parameters and supplemental enhancement information (SEI) messages. Both can assist in processes related to decoding, display or other purposes. Although not required by the decoding process to obtain correct values of decoded pictures, some of the VUI parameters and SEI messages are used for specifying bitstream conformance and are used in some systems for signaling the proper interpretation of HDR or 360^o video bitstreams. Unlike AVC and HEVC, for which the VUI parameters and all SEI messages have been specified directly within the same video coding standard that specifies the coding tools, for VVC, only those SEI messages that directly affect the bitstream conformance definition have been included in the VVC standard itself, while VUI parameters and other SEI messages have been generalized and put into a separate specification, the versatile SEI (VSEI) standard, standardized in ITU-T as Recommendation H.274 and in ISO/IEC as 23002-7 [12]. Using a separate and generalized specification for these metadata will allow elements of the VSEI standard to be referenced for use in other contexts besides VVC and will ease the effort of document maintenance and enhancement by reducing the need to revise several parts of one large document when working on enhancements of both the coding tools and the supplemental information.

Despite its versatility, the VVC core design continues to basically follow the conventional block-based hybrid video coding scheme, and the system and transport interface of VVC continues to follow the basic bitstream structure based on network abstraction layer (NAL) units and parameter sets, similar as specified in AVC and HEVC. Consequently, the mechanisms for encapsulating VVC bitstreams in the ISO base media file format (ISOBMFF, a.k.a. the mp4 file format) tracks [13], in MPEG-2 transport streams [14], and in real-time transport protocol (RTP) packets and RTP streams [15] can be kept similar as for AVC and HEVC, with limited VVC-specific extensions. These encapsulation mechanisms enable the use of VVC in a broad variety of applications, including streaming, broadcast, and video conferencing.

This paper is structured as follows. Section II provides an overview of the technologies introduced in VVC. This includes the functionalities enabled by high-level syntax as well as the core compression technologies. Section III discusses how these functionalities can be applied to efficiently code video for a wide range of applications, including results comparing the coding efficiency for an application with HEVC. Application-related profiles, tiers and levels are outlined in this section as well. Early implementations demonstrating VVC's readiness for deployment are reviewed in Section IV. This includes decoders, conformance testing bitstreams, and an open-source encoder. Section V outlines possible near-term extensions to VVC as well as a more long-term outlook.

II. Technology Overview

VVC was designed to be truly versatile, a mission which comes along with a multitude of new high-level functionalities. From a technology point of view, this includes methods to packetize, process and access the video data in the compressed domain as well as new or refined coding tools to facilitate these scenarios. The technology aspects related to high-level functionalities are reviewed first in this section, followed by an overview of core compression technologies. This includes major improvements and new coding tools for the block-based hybrid video coding design. All of these advances contribute to the bit rate savings over the prior standards for the same video quality. Furthermore, content- or application-specific bit rate reduction is achieved by special coding tools. For a more detailed overview of the VVC coding tools in the context of video coding standard evolution, the reader is referred to [16]. Rather than repeating here much of the content of [16] and that of the other papers that appear in this issue of the *IEEE Trans. CSVT*, the reader is referred to those other papers for additional information.

A. *High-Level Functionalities*

VVC inherited much of the high-level syntax (HLS) designs from AVC and HEVC, including the structuring of the bitstream into NAL units and the use of cached parameter sets with indexed referencing. This contrasts with the international video coding standards developed before AVC, e.g., H.262/MPEG-2 and H.263, in which a start-code based bitstream structure with simple headers has been used. A main objective of the HLS design, which plays a major role in the systems and transport interface for a video coding design, is to enable and expose the design's high-level functionalities, including, for example, random access capability, parallel processing capability, and layered coding scalability. A more comprehensive overview of the HLS design in VVC can be found in [17]. This section discusses some of the high-level functionalities provided by the VVC HLS features and their uses in various applications.

1. Random Access

Random access capability refers to the ability to start consuming video content from positions other than the very beginning of the bitstream. Such an ability is necessary in order to enable many fundamental video application functionalities, e.g., seeking to or starting from an arbitrary media time, joining an ongoing broadcast/multicast, switching to a different program channel, and switching to a different bitstream in order to adapt to varying network conditions.

A complicating factor in enabling random access is that in modern video coding designs, most of the pictures in a video bitstream are coded using inter-picture prediction from other previously decoded pictures. To avoid excess buffering and delays in the decoding process, the bitstream is constructed so that data that cannot be decoded until after some other data is decoded is sent after that other data. Thus, the data dependencies determine the *decoding order* – i.e., the order in which the pictures are placed into the bitstream by the encoder and are decoded from it by a decoder. To maximize coding efficiency, the decoding order may be different from the *output order*, i.e., the order in which the decoded pictures are to be displayed after decoding. The output order corresponds to the order in which the pictures are captured by a camera before encoding.

The most basic feature for random access to a VVC bitstream is the instantaneous decoding refresh (IDR) picture. When the decoding starts from an IDR picture, the IDR picture itself and all the pictures that follow it in decoding order can be correctly decoded. However, IDR pictures interrupt the use of inter-picture referencing, which affects the coding efficiency. A more efficient form of random access can be achieved using clean random access (CRA) pictures. When the decoding starts from a CRA picture, the CRA picture itself and all the pictures that follow it in output order can be correctly decoded, but there may be some pictures that follow the CRA picture in decoding order and precede it in output order that cannot be properly decoded because they use inter-picture prediction from pictures that precede the CRA picture in decoding order, and these thus need to be discarded when performing a random access. Both IDR and CRA pictures are known as intra random access point (IRAP) pictures, since both of them are pictures that are coded only using “intra” coding – i.e., using prediction only from within the current picture.

Another issue is that the number of bits needed to encode an IRAP picture with adequate fidelity is relatively high, since these pictures are coded without the use of inter-picture prediction. In applications in which it is critical to minimize end-to-end delay, the spikes in bit usage at the locations of the IRAP pictures can cause a data buffering delay that can be avoided by using a different random-access feature called gradual decoding refresh (GDR) pictures. When the decoding starts from a GDR picture, the GDR picture itself can use inter-picture prediction from previous pictures in decoding order and may not be fully correct in content when performing a random access. However, an identified recovery point picture, which is usually several pictures after the GDR picture in decoding order, and all the pictures that follow the recovery point picture in output order can be correctly decoded. Typically, a refreshed region starts at a GDR picture and grows gradually to include the entire picture at the recovery point. The refreshed region for any picture may only refer to the refreshed regions in the reference pictures associated with the same GDR picture; this motion constraint causes significantly lower coding efficiency.

IRAP pictures can have *leading pictures* – pictures that follow the IDR/CRA picture in decoding order but precede it in output order. To support the use of leading pictures with IDR pictures, VVC defines two types of IDR pictures – one type that has leading pictures and another that does not. The leading pictures associated with an IDR picture can be correctly decoded when randomly accessing from the IDR picture, and hence they are also referred to as random access decodable leading (RADL) pictures. Leading pictures associated with a CRA picture may or may not be correctly decodable when randomly accessing from the CRA picture, and thus they are classified either as RADL pictures or as random access skipped leading (RASL) pictures, with the latter type being those pictures that cannot be correctly decoded when random accessing from the associated CRA picture.

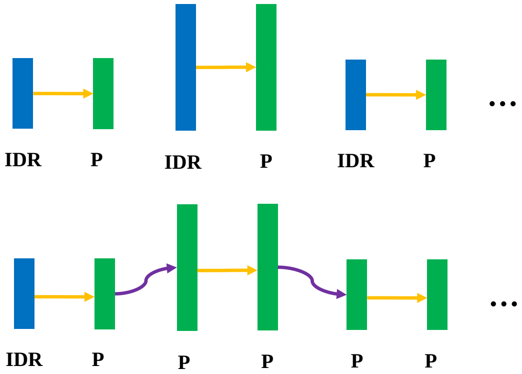

Fig. 1. Examples of spatial resolution changes at IDR pictures (the upper half) and at inter-coded pictures (the lower half) with the use of reference picture resampling, where the height of a box representing a picture indicates the spatial resolution.

The use of leading pictures in a bitstream can help improve coding efficiency, but it increases the end-to-end delay (i.e., the delay between capturing a picture at the sender side and displaying it at the receiver side) because picture reordering before encoding and after decoding is needed whenever there are leading pictures. Among the different types of pictures that can be used to provide random access capability, CRA pictures with RASL pictures can provide the highest coding efficiency, but at the cost of longer end-to-end delay. GDR pictures without picture reordering can provide the shortest end-to-end delay since this can avoid both the picture reordering delay and the data buffering delay that is often needed to smooth out the bit rate spikes caused by IRAP picture usage, but GDR pictures have a cost of lower coding efficiency and cause a time lag between the random access point and the recovery point. IDR pictures without leading pictures are in between for both coding efficiency and end-to-end delay. Some analysis of end-to-end delay is provided in Section III.F.

In typical streaming and broadcast applications, where a moderate-to-long end-to-end delay does not affect the user experience, it can be desirable to use CRA pictures to provide random access capability or bit rate adaptation, although stream adaptation through bitstream switching with CRA pictures can be more complicated than using IDR pictures (see the analysis in Section III.D). In real-time conversational applications such as video conferencing, out-of-order picture coding is often avoided, and IDR pictures may be preferred over CRA pictures to provide random access capability (e.g., in multiparty scenarios), since a significant end-to-end delay, e.g., above 150 ms, can impact the conversational communications and thus significantly degrade the user experience. In applications such as wireless display and online gaming, ultralow end-to-end delay is desirable. In such cases, using GDR to provide random access capability can help minimize the end-to-end delay.

2. Reference Picture Resampling

Conventionally, e.g., in HEVC, the spatial resolution of a video bitstream can only change at an IDR picture or equivalent, as illustrated by the upper half of Fig. 1. VVC also allows the spatial resolution to change at inter-coded pictures, as illustrated by the lower half of Fig. 1, through the support of the feature referred to as reference picture resampling (RPR). As the name RPR implies, when such a resolution change occurs, the decoding process of a picture may refer to one or more previous reference pictures that have a different spatial resolution for inter-picture prediction, and consequently a resampling of the reference pictures for operation of the inter-picture prediction process may be applied. Compared to forcing the insertion of an IDR picture in order to switch resolution, allowing inter-picture prediction from reference pictures of different resolutions improves coding efficiency and mitigates the problem of bit rate spikes associated with IRAPs.

In VVC, RPR allows either downsampling or upsampling of a reference picture to predict a current picture having a different resolution. In order to avoid additional processing steps, the RPR process in VVC is designed to be embedded in the motion compensation process and performed at the block level. In the motion compensation stage, the scaling ratio is used together with motion information to locate the reference samples in the reference picture to be used in the interpolation process. Depending on the RPR scaling ratio, three sets of interpolation filters with different frequency responses are defined in VVC. In the case of downsampling from the reference picture, two sets of 16-phase 8-tap interpolation filters are used for the luma component, and two sets of 32-phase 4-tap interpolation filters are used for the chroma components. When the chroma planes have the same resolution as the luma plane, only 16 of the 32 chroma filter phases are used. The first set of luma and chroma downsampling filters is designed for a scaling ratio of 1.5, and is applied when the actual scaling ratio is between 1.25 and 1.75. The second set of luma and chroma downsampling filters is designed for a scaling ratio of 2, and is applied when the actual downscaling ratio is greater than 1.75. These downsampling filters are all based on the windowed sinc family of low-pass filters, a well-known concept in digital signal processing. For upsampling from the reference picture (i.e., when the current picture is larger than the reference picture) and for downsampling with a downscaling ratio less than 1.25, the same interpolation filters as those for normal motion compensation are used. In fact, the conventional motion compensation interpolation process can be deemed as a special case of the resampling process with the scaling ratio in such a range. In addition to conventional translational block motion, VVC also supports an affine motion mode, and the affine mode has three sets of 6-tap interpolation filters that are used for the luma component to cover the different scaling ratios in RPR. Whether downsampling or upsampling is performed is determined separately for the horizontal and vertical dimensions using the effective widths or heights of the reference picture and the current picture.

Any application scenario that involves changing the video spatial resolution can potentially benefit from using the RPR feature. These especially include conversational applications, e.g., video telephony, where changing the video spatial resolution can help to minimize delay and adapt to changing network conditions. It can be useful for speaker changes in multiparty video conferencing, in which it is common that the active speaker is displayed with a larger picture size than what is used for the rest of the conference participants. It can also enable fast start-up in streaming or conversational video, by starting the bitstream with a lower resolution and then changing to a higher resolution after the start-up. If the amount of action in the video content temporarily becomes very high, e.g., due to a scene cut, occluding foreground action or global camera motion, temporarily reducing the picture resolution can help with maintaining a high enough frame rate to maintain perceptual motion continuity. Changing the picture resolution can also be helpful with adaptive bit rate streaming based on a library of pre-encoded alternative video bitstreams, by using CRA pictures for switching between bitstreams with different spatial resolutions. Section III.D provides more detail on adaptive streaming with resolution changes.

Another benefit brought by RPR is that it simplifies the support of spatial scalability, because with RPR an additional module for resampling of inter-layer reference pictures for spatial scalability becomes unnecessary, as the same resampling filters can be used for spatial scalability.

3. CTUs, Slices, Tiles, and Wavefronts

The basic processing unit within a picture in VVC, as in HEVC, is the coding tree unit (CTU), which contains the luma and chroma samples of a square region of the picture (except for truncation of the CTUs at the right or bottom edges when the width or height is not divisible by the CTU size). In VVC the CTUs can be larger than in HEVC, but the concept is the same, and is similar to the concept of a macroblock in AVC.

Similarly as in AVC and HEVC, in VVC a picture can be segmented into regions called *slices*, each of which is sent in its own separate NAL unit. VVC also inherited two other high-level picture partitioning and parallelism features from HEVC: *tiles* and *wavefronts*, with some minor differences as further described in [17]. Each tile is a rectangular subset of a picture and is typically not forming a separate NAL unit. When the wavefronts tool is in use, the decoding of the next row of CTUs can begin before the current row has been completely processed, with the lag of one CTU for each subsequent row. This requires the probability estimates of the entropy decoder (see Section II.B.5) to be stored after decoding the first CTU of each row within a tile, to be used to initialize the entropy coding state for decoding the next row. In HEVC, the lag was two CTUs, but the CTUs were typically smaller.

In earlier standards, the partitioning of a picture into slices was mainly for the purpose of maximum transmission unit (MTU) size matching, to avoid having a coded picture fragmented into multiple packets by the transport protocols. When a picture is encoded into multiple slices and each slice is encapsulated into one transport packet, the loss of some of these slice packets does not affect the decoding of the received slices. However, if a picture is not encoded into multiple slices but the

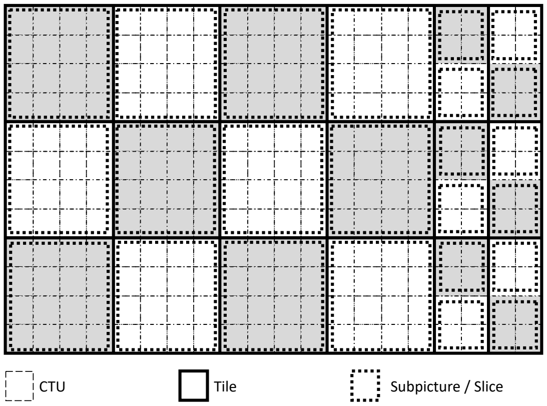

Fig. 2. A picture partitioned into 18 tiles, 24 slices and 24 subpictures (each subpicture contains one slice in this example).

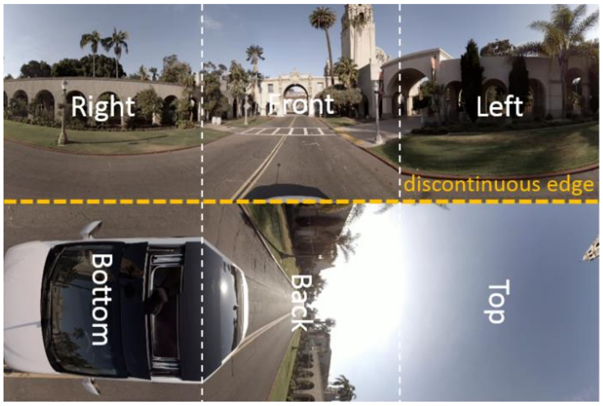

Fig. 3. An example of a cubemap projected picture.

coded picture is fragmented into multiple packets by the transport protocols, then the loss of any of those packets may cause the received slices of the picture to become non-decodable, and the loss of the packet that contains the header information will definitely cause the entire received slice to become non-decodable. Therefore, in the designs of AVC and HEVC, generally a slice consists of a number of macroblocks or CTUs in raster scan order. During the development of VVC, MTU-size-matching was no longer considered the main purpose of the use of slices; instead, enabling regional access of a picture and ultralow end-to-end delay were intended. In the design of VVC, the division of the picture into slices can use one of two modes: the rectangular slice mode and the raster scan slice mode. In the rectangular slice mode, a slice covers a rectangular region of a picture and consists of either a set of tiles or a sequence of CTU rows in a tile. The latter enables a more flexible partitioning of pictures into rectangular slices beyond what is allowed by tiles being constrained to be always in complete rows of tiles with the same height and complete columns of tiles with the same width. In the raster scan slice mode, each slice contains a series of complete tiles in raster scan order and could be covering a non-rectangular region. The use of the raster scan slice mode can be beneficial for low-delay applications where the number of tiles to be included in a slice might not be known before the start of the encoding of the slice.

4. Subpictures

A functionality that is useful for some applications and is especially needed for high-resolution immersive video is the support of so-called bitstream extraction and merging (BEAM) operations. BEAM support was an important design goal in the development of VVC HLS. For example, in 360° video, at any moment a viewer usually sees only a small spatial portion of the entire coded video. Therefore, for transmission and/or decoding efficiency, a large spatial portion of the encoded video may not need to be transmitted and/or decoded. To be able to do that in an efficient and convenient manner, the bitstream needs to be coded in the way such that a region of the picture can be extracted and decoded independently without accessing the other regions. Later, when the viewer turns his or her head and changes the viewing orientation, usually the new field of view (FOV) is spatially overlapping with the old FOV, thus merging of the bitstream representing the overlapping region and the bitstream representing the region that newly came into the FOV is needed. Note that completely switching to a new bitstream representing both regions would also be possible to avoid the needs of bitstream merging, however, it is less efficient from coding efficiency point of view, as the region newly coming into the FOV in this approach has to be intra refreshed.

In VVC, BEAM operations can be realized based on the subpicture feature. A picture consists of one or more subpictures. The layout of the positions and sizes of subpictures is the same for all pictures in a coded video sequence, which is a self-contained sequence of coded pictures. (More precisely, to account for the possibility of using layered video coding as discussed in Section II.A.7, this is within the sequence of coded pictures within a particular layer.) Each subpicture sequence may be coded such that it can be extracted and decoded without the presence of any of the other subpicture sequences. A subpicture consists of one or more rectangular slices that together form a rectangular region, e.g., as shown in Fig. 2.

The HLS related to subpictures, including syntax for picture header and slice header, has been designed such that the extraction of a subpicture sequence won't need to change any part of the picture headers or slice headers. When a subpicture sequence is motion-constrained to be extractable, motion vectors (MVs) of blocks in the subpictures are still allowed to point out of the subpicture boundary, and when that occurs, reference sample padding is used in motion compensation at subpicture boundaries, similar to the scenario when MVs would point outside of the picture boundaries. This way, higher coding efficiency for extractable subpicture sequences can be achieved than by disallowing MVs that point outside of the boundaries of the extracted region. For example, average coding gains of about 4.1% and 6.5% were reported in [18] for the cases of 6×4 and 12×8 subpictures per picture, respectively.

5. Virtual Boundaries

Virtual boundaries are boundaries within pictures where the in-loop filter operations that would apply across the boundaries are disabled. The granularity of the possible locations of virtual boundaries is eight luma samples. This feature serves two purposes. The first is for avoiding seam artifacts introduced by applying the in-loop filters across an artificial boundary created by a preprocessing step before encoding, e.g., a boundary that can occur when a multi-face projection format such as the cubemap projection (CMP) [19] is used to represent a 360º video during coding. Each CMP picture contains a layout of six cubemap faces, as shown in Fig. 3. However, while the upper and lower rows of three faces are continuous in the spherical domain, a discontinuity exists between the upper and lower rows, shown by the horizontal "discontinuous edge" in Fig. 3. Such a boundary can be signaled as a virtual boundary to disable the in-loop filter operations across the boundary, to avoid mixing of spherically non-neighboring samples together during in-loop filtering, which could harm the subjective quality. Although it is possible to partition the picture such that the boundary between the upper and lower face rows coincide with tile boundaries or slice boundaries, across which the in-loop filters could be disabled to achieve similar subjective effect, defining virtual boundaries and disabling loop filtering across them provides more flexibility for 360º video coding, because the latter does not require the cubemap face width or height to be a multiple of the CTU size.

Since the locations of such virtual boundaries usually stay unchanged over the entire coded video sequence, a virtual boundary signaling option is provided to signal virtual boundaries that remain constant for the entire sequence. The second purpose of virtual boundaries is for use with GDR, when the boundary between the refreshed region and the unrefreshed region is changing from picture to picture and may not be aligned with the boundaries of slices or tiles, so a second virtual boundary signaling option is provided to ensure mismatch-free decoding of the refreshed regions in a GDR picture and in its associated recovering pictures when random accessing from the GDR picture.

6. Parameter Sets and Other "Header" NAL Units

Parameter sets are syntax structures that are stored in a cache and have an associated index for identifying which parameter set of a given type is being referenced. They were first introduced in the AVC standard. In AVC, only two types of parameter sets were specified, the sequence parameter set (SPS) and the picture parameter set (PPS). The SPS and PPS were introduced in order to provide encoders and system designers the freedom to decide for themselves how often to send this sequence-level and picture-level information that is important but often rather repetitive. The parameter set design also enables the transmission of such header information using an out-of-band, reliable mechanism in systems that support such a method. For example, in video telephony and conferencing applications, parameter sets are typically transmitted as part of a session description protocol (SDP) file during the session negotiation process, using the transmission control protocol (TCP). This can ensure successful transmission of the very important header information carried in the parameter sets, although it comes at the cost of possible longer initial delay due to the retransmissions that could be needed when packet losses occur. In contrast, the coded video data in such applications is transmitted using RTP on top of the user datagram protocol (UDP), which does not guarantee the arrival of all the data at the receiver or the arrived data to be error free but can guarantee bounded end-to-end delay.

In addition to the SPS and PPS, another type of parameter set, the video parameter set (VPS), was introduced to HEVC and is also used in VVC. The VPS contains cross-layer sequence header information that provides a "big picture" characterization of the properties and dependencies of an entire multilayer bitstream and the header parameters common to all the layers. The two purposes described for the SPS and PPS also apply for the VPS.

Besides the VPS, SPS, and PPS, a new type of parameter set, the adaptation parameter set (APS), was introduced in VVC. APSs are used to carry control parameters that affect particular low-level coding functions, and three types of APSs are defined for carrying the parameters applied in the adaptive loop filter (ALF), another in-loop filter known as luma mapping with chroma scaling (LMCS), and the inverse quantization scaling lists, respectively. The purpose of saving signaling overhead still applies for APSs, as an APS can be used to avoid repeated transmission of those parameters that are common for multiple slices of a picture or for slices of different pictures. However, the original purpose of out-of-band transmission, as for other types of parameter sets, does not apply for APS, as updated APS content would typically be sent quite frequently and thus these parameters cannot just be transmitted out-of-band during the session negotiation stage. On the other hand, when pictures contain multiple extractable subpictures, APSs can be carried in separate file format tracks or dynamic adaptive streaming over HTTP (DASH) representations, separate from the tracks/representations each carrying an extractable subpicture sequence. This capability of systems encapsulation of VVC bitstreams is important to enable convenient and efficient BEAM operations.

Another type of NAL unit used in VVC is the picture header (PH). Similar to the PH in prior video coding standards that had a start code-based bitstream structure like H.263 and MPEG-2, the VVC PH carries picture-level parameters that are common for all slices of a picture. However, in the context of the NAL-unit-based syntax structure, having PHs embedded in their own NAL units conveniently enables their carriage in a file format track or DASH representation separate from other tracks/representations, which, as mentioned above, enables convenient and efficient BEAM operations. Alternatively, the VVC PH does not have to be in its own NAL unit; the PH syntax structure could be directly included in the slice header instead, similarly as in AVC and HEVC which do not have the PH concept, but this only applies when all pictures are coded using only one slice for an entire coded video sequence.

7. Scalability and Layered Coding

Scalable video coding refers to the structuring of a coded video bitstream by an encoder in a way that enables the extraction and decoding of subsets of the coded data to produce decoded content with lower quality or to produce alternative or supplemental decoded content. The simplest form of scalability is temporal scalability, which makes it possible to extract and

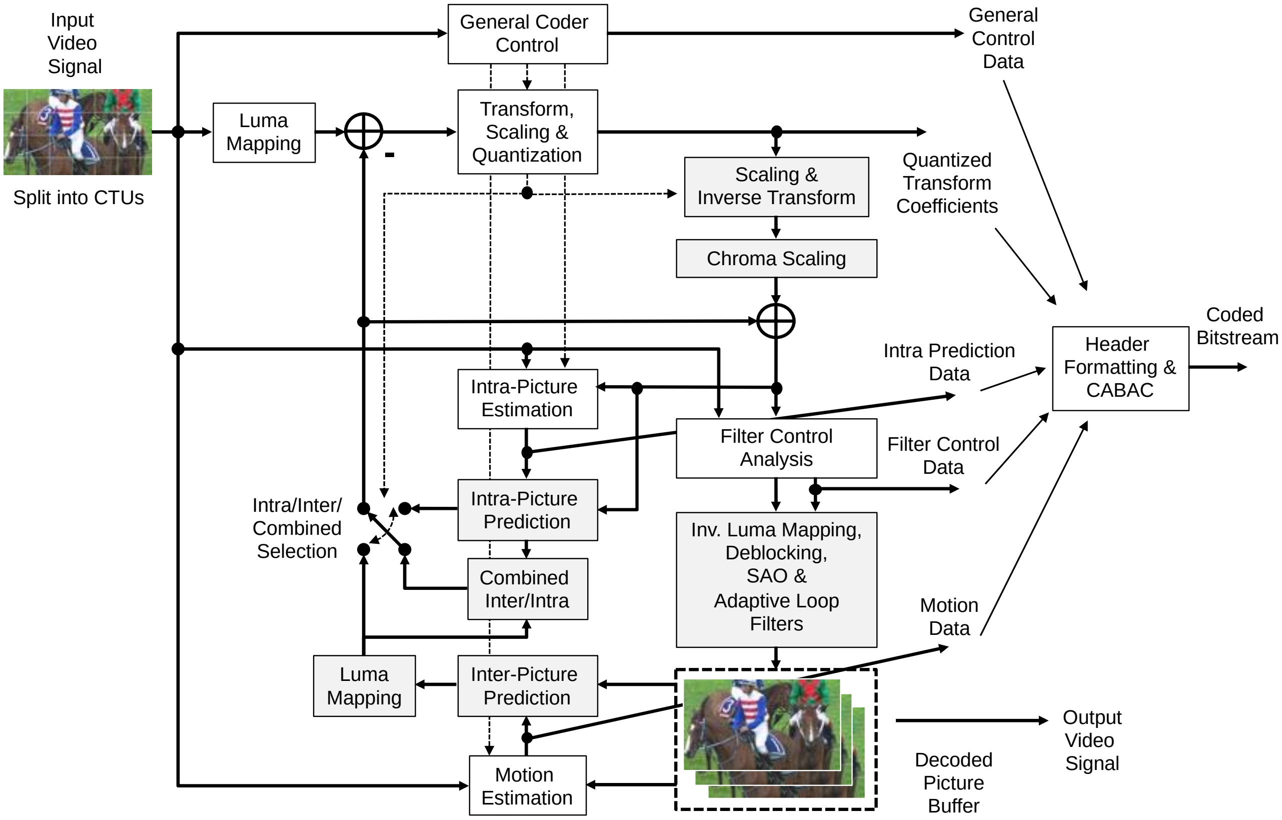

Fig. 4. Typical VVC encoder.

decode a subset that produces decoded video with a lower picture rate, a.k.a. frame rate. Subsets for a particular decoded output are called layers, and temporal scalability subsets are called temporal sublayers. Other forms of scalability, which involve multilayer coding, include (but are not limited to) quality scalability, a.k.a. signal-to-noise ratio (SNR) scalability, spatial scalability, and multiview scalability, for which extracting and decoding more data can result in higher quality, higher spatial resolution, and more views, respectively.

Scalability can be used for bit rate saving compared to simulcast coding of multiple, independent bitstreams, among other benefits [20], [21], in applications wherein a particular content may be consumed by different classes of clients that are differentiated by connecting bandwidths, decoding capabilities, display sizes, etc., providing the same or an even larger set of operation points in terms of picture rates, qualities, and resolutions as simulcast. Layered coding can also be used to encode alternative or supplemental content, such as to encode depth maps or transparency maps that are associated with the video pictures, and can even be used to carry synchronized independently decodable "simulcast" content. Applications that can benefit from scalability and layered coding include (but are not limited to) conferencing, broadcasting, and 3D video applications. These applications are further introduced in Sections III.H and III.I, whereas this section focuses on the design of these features in VVC.

The basic design for temporal scalability support in VVC is similar to that in HEVC. The NAL unit header includes fields that specifies a temporal sublayer ID, and a picture is not allowed to refer to another picture with a higher value of temporal ID for inter-picture prediction. The basic multilayer coding design for quality, spatial, and multiview scalabilities in VVC is also similar to the multilayer extensions of HEVC [22], [23]. The NAL unit header also includes a field that specifies a layer ID, and for inter-layer picture referencing, a picture can only refer to a decoded picture of a lower layer in the same access unit, which contains all coded pictures of all layers pertaining to the same output time. The inter-layer prediction process is basically the same as the temporal inter-picture prediction process, i.e., the multilayer coding is based on multi-loop decoding, where inter-layer prediction is based on fully decoded inter-layer reference pictures, as contrasted with the single-loop decoding scheme on which the scalable extension of AVC is based, where inter-layer prediction can be based on an inter-layer reference picture that is only partially decoded or even only parsed.

However, there are some key differences of the scalability support in VVC compared to earlier video coding standards. First, the capability of multilayer scalabilities is provided already in the first version of VVC, as opposed to being only in later versions in earlier video coding standards such as AVC and HEVC. Second, for spatial scalability, thanks to the existence of the RPR feature in VVC which provides the resampling functionality between the current picture and its temporal reference pictures, no additional signal-processing features are needed to support inter layer prediction of spatial scalability. Rather, spatial scalability is achieved with just some HLS changes to the single-layer coding design. Third, the scalability-specific HLS in VVC has been designed to be significantly simpler than in the multilayer extensions of HEVC. Additionally, the decoded picture buffer management design and the definitions of the levels for the profiles (see Section III.A) have been made in a manner such that the same design applies to both single-layer and multilayer bitstreams, which enables single-layer decoders to be easily adapted to support the decoding of multilayer bitstreams.

VVC introduces two additional types of NAL units, namely the decoder capability information (DCI) NAL unit and the operating point information (OPI) NAL unit, for scalability support. The DCI NAL unit indicates the minimum decoding capability that is needed for decoding the entire bitstream. The VPS and SPS also indicate minimum decoding capability in the context of both multi-layer and single-layer bitstreams, but only for the relevant coded video sequences for the current layer rather than for the entire bitstream. The OPI NAL unit identifies the target operation point the decoder is supposed to be operating at from both the decoding point of view, i.e., which coded pictures of which layers or sublayers are to be decoded, and the outputting point of view, i.e., which decoded pictures of which layers are to be output. Both DCI and OPI are optional. In the absence of DCI, no in-band bitstream capability information is provided, unless it is known (e.g., by some system-provided means) that what is provided in a VPS or SPS applies to the entire bitstream. In the absence of OPI, a default operating point is inferred.

B. Core Compression Technologies

All of the aforementioned new high-level functionalities have been put into version 1 of VVC so that the new standard will already be in a position to efficiently code video for a very wide range of different applications. However, given the exponentially increasing use of video and the associated high amount of data needed to send video, significantly increased compression capability remains crucial. While the basic well-known block-based hybrid video coding scheme used in all previous major standards has been retained in VVC, it has been improved in many ways. The improvements affect all the functional elements of hybrid video coding. As with past designs, the represented video content consists of either one or three color plane arrays of sample values that have a represented bit depth. There is a primary color plane called the luma color plane, and the others (if present) are referred to as the chroma planes. The luma plane ordinarily represents local brightness information and the chroma planes ordinarily represent color hue and saturation. The format known as 4:4:4 has equal resolution for all three color planes, and the format known as 4:2:0 has chroma planes with half the width and half the height of the luma plane. Content for consumer applications typically uses the 4:2:0 color format, while remote computer desktop sharing and wireless display application may use 4:4:4. A lesser-used format is 4:2:2, where the chroma has half the width but the same height as the luma. In older consumer applications, the video bit depth was typically 8 bits, but 10 bits are needed for HDR, and all profiles in VVC version 1 support bit depths up to 10 bits.

Fig. 4 shows the functional diagram of a typical hybrid VVC encoder, including a block partitioning that splits a video picture into CTUs, block-based intra- and inter-picture prediction, spatial transformation and quantization of the prediction residual, in-loop filtering of the reconstructed signal after scaling (a.k.a. "inverse quantization") and inverse transformation, followed by header formatting and context adaptive binary arithmetic coding (CABAC) entropy coding for bitstream generation. Compared to a typical HEVC encoder, e.g., as shown in [3], it can be seen that VVC introduces additional elements such as combined inter-/intra-picture prediction (see Section II.B.3), luma mapping with chroma scaling, and additional loop filters (see Section II.B.6). While most of the design uses rectangular blocks, some non-rectangular partitioning of prediction regions is also supported (see the geometric partitioning mode discussed in Section II.B.3). Although screen content coding (SCC) tools are discussed separately in Section II.B.7, they can be located in the respective modules shown in Fig. 4, e.g. the adaptive color transform could be considered as part of transform and inverse transform.

Like HEVC, the VVC design does not include special coding tools for handling interlaced video; instead it has a simple ability to indicate whether the coded video pictures are to be interpreted as complete frames or as top and bottom fields, as this approach has been proved as sufficient and effective for coding of interlaced video. The main advances in each module as well as the new elements are summarized in the following.

1. Block Partitioning

The HEVC quadtree partitioning of a CTU has been extended in VVC by enabling a more flexible partitioning and supporting larger block sizes. This includes recursive non-square splits and separate partitioning for luma and chroma. Additional constraints facilitate pipelined processing in hardware decoder implementations. The main VVC partitioning techniques are listed in the following and a more detailed description can be found in [24].

- **Quadtree plus multi-type tree (QT+MTT)** extends the quadtree-based partitioning from HEVC with a nested multi-type tree using binary and ternary splits. Together with increasing the maximum block size from 64×64 in HEVC to 128×128 in VVC, this allows for a better adaptation to image content for high spatial resolutions while also supporting small and narrow rectangular blocks for the prediction and transform processing stages. As done in HEVC, VVC uses the CTU as the basic processing unit and root of the recursive partitioning tree, and coding units (CU) containing coding blocks as leaves. However, the prediction units introduced in HEVC, which can take non-square shapes, are no longer used because the generalized new CU shapes can be non-square in VVC. Unless the CU is too large for the maximum transform length (see Section II.B.4), it is used directly as the processing unit of the prediction and transform stages, without any further partitioning. Furthermore, the HEVC quadtree partitioning of the CU residual into square transform units is replaced by using rectangular transform units (TU), where certain new VVC modes, i.e. intra sub-partition (ISP) mode and subblock transform (SBT) mode, allow horizontal or vertical subdivision of a CU into TUs.

- **Chroma separate tree (CST)**: VVC introduces the CST which enables the use of a separate partitioning for luma and chroma in an intra-coded slice. This is motivated by the fact that luma usually has finer texture or sharper edges than chroma, which allows for larger chroma CUs. In encoder implementations that consider both luma and chroma rate-distortion costs in the decision to split a block, the CST provides a speed up in cases where chroma is not further split into smaller CUs.
- **Virtual pipeline data units (VPDUs)**: In hardware video decoders for VVC, block regions of a CTU that are known as VPDUs can be processed separately in parallel to increase the throughput. Although the VVC standard does not discuss the VPDU concept explicitly, the syntax is designed to disallow certain binary and ternary splits for large CUs to enable a VPDU size that is smaller than the maximum possible CU size. It is important to keep the VPDU size as small as possible because the memory buffer size in the pipeline stages is proportional to it. The maximum CU size in VVC is 128×128 luma samples, but the VPDU size is limited to 64×64 luma samples.

2. Intra-Picture Prediction

Advanced intra-picture prediction techniques in VVC include the DC and planar modes similar to HEVC plus finer-granularity angular prediction with more angles compared to HEVC (93 vs 33), additional matrix-based prediction modes for luma, and cross-component prediction modes for chroma (not counting the intra-picture block copy and palette modes discussed in Section II.B.7, which could also be considered intra modes). The new intra coding tools are summarized in the following, and a more detailed description of intra-picture prediction techniques can be found in [25].

- **Finer-granularity angular prediction:** For each block size there are 65 angular prediction directions, while the set of angles depends on the block size. For square blocks, the 65 angular prediction directions are defined from 45 degrees to −135 degrees in a clockwise direction for a square shape coding block.
- **Wide-angle intra prediction (WAIP)**: For blocks that are not square, 14 angles using prediction from the shorter side of a block are replaced by more extreme angles using prediction from the longer side, bringing the total number of angles supported in the design to 93 while the number of angular modes that can be signaled for any particular block size remains 65.
- **4-tap fractional sample interpolation filters**: Two sets of 4-tap filters are used to interpolate the luma sample values when the intra prediction angle points to a fractional position with 1/32 sample accuracy, as contrasted with 2-tap bi-linear interpolation in HEVC. One set of filters corresponds to the DCT-based filters applied in chroma motion compensation. The other set consists of smoothing filters which are derived by convolving the bi-linear filter of HEVC with the {1, 2, 1}/4 reference sample smoothing filter from HEVC. As a consequence, the reference sample smoothing from HEVC is restricted to the angles pointing to full sample positions, which do not require interpolation. The selection of the interpolation filter is determined based on the block size and the prediction mode, with the sharpening DCT-based filter being applied for smaller blocks and modes around the horizontal and vertical direction. For chroma components, 2-tap linear interpolation is used for simplicity.
- **Position-dependent prediction combination (PDPC)** further modifies the prediction of the planar, DC and selected angular modes (the horizontal and angles above 0 plus vertical and angles below −90 degrees) by combining the initial prediction with the boundary reference samples. The combination weights depend on the prediction mode and sample locations. PDPC can be seen as a more generalized version of HEVC’s boundary value smoothing applied to remove discontinuities for boundary prediction samples in the prediction direction for the horizontal (using the first column), vertical (first row) and DC (both) predictors.
- **Multiple reference lines (MRL)**: In addition to the directly adjacent line of neighboring samples, one of the two non-adjacent reference lines can be used as the reference line for intra-picture prediction of luma samples. The non-adjacent reference line can be two or three lines away from the current block. Since using the extended reference lines is only beneficial for sharp content, combining them with the planar mode, reference sample smoothing and PDPC is disallowed.
- **Matrix-based intra-picture prediction (MIP)** can be regarded as a low-complexity variant of a neural-network-based intra-picture prediction. The neural network used to predict the current block from the reference samples has been simplified to a matrix-vector multiplication, with the matrix being selected from a set of pre-trained matrices and the vector being constructed from the reference samples. In order to derive the matrix coefficients, key aspects of the data-driven neural network training algorithm have been reused. Other elements of MIP, such as downsampling of the reference samples and an upsampling step with linear interpolation for the final prediction, have been introduced to reduce the complexity and memory requirements. The selection index of the matrix to be employed for the MIP mode is signaled using a truncated binary code. In the most common use cases, MIP is applied only for luma samples, but it can also be applied to chroma samples in the case of 4:4:4 chroma sampling when the CST is disabled.
- **Intra sub-partition (ISP) mode** splits the luma block of a CU vertically or horizontally into 2 or 4 sub-partitions for separate processes of prediction and transform with all sub-partitions sharing the coding mode information. For small block sizes, the prediction can span over multiple transform blocks to prevent prediction blocks that are only one or two

samples wide, which would be a burden in some hardware implementations.

- **Cross-component linear model (CCLM)** predicts chroma component samples from corresponding reconstructed luma samples. The parameters of the linear model can be derived by minimizing the regression error between neighboring luma and chroma samples. The three different CCLM modes at the CU level in VVC specify different neighboring luma and chroma sample locations: these can be located to the left, above or above-left of the current block. For non-4:4:4 sampling formats, the reconstructed luma samples need to be downsampled in a way that takes into account whether the chroma samples are vertically and co-sited with the luma samples. This is specified by an SPS flag.
- **Extended most probable mode (MPM) signaling:** In HEVC, the DC, planar and 33 angular luma intra-picture prediction modes are signaled using either an index into a list of three MPMs derived from neighboring modes, or a 5-bit fixed length code for the remaining 32 modes. VVC signals the DC, planar and 65 angular modes by extending the list of MPMs to six, with the first mode always being the planar mode. The remaining 61 luma intra-picture prediction modes are signaled using truncated binary coding. When MRL prediction is used, the signaled MRL mode is restricted to be one of the 5 non-planar MPMs.

3. Inter-Picture Prediction

As in AVC and HEVC, inter-picture prediction in VVC uses either single-MV uni-prediction referencing a picture in a list of previously decoded reference pictures or bi-prediction using two MVs and indices into two lists of pictures called list 0 and list 1 to select the reference pictures to be used with the MVs to form two prediction signals that are then averaged together. Beyond that, VVC introduces a variety of new coding tools for more efficient representation, prediction and coding of motion compensation control information, as well as for enhancing the motion compensation processing itself. These techniques can be categorized into: a) advances in coding motion information; b) advances in CU-level motion compensation; c) refined motion compensation processes using subblock based motion derivation and prediction refinement at the decoder; and d) horizontal wrap-around motion compensation as a special feature for immersive video. These four categories are described in this section.

a) Advances in coding motion information

Using motion information from temporally and spatially neighboring blocks, HEVC introduced the merge mode and the advanced MV prediction (AMVP) mode for the prediction and coding of motion parameters in inter-picture prediction. In VVC, both of these modes are extended using improved predictors, enabling MV differences (MVDs) for the merge mode, and providing a more flexible MVD signaling for the AMVP mode to improve the tradeoff between motion accuracy and motion overhead bits. These enhancements are described in the following, with a more detailed description provided in [26].

- **History-based MV prediction (HMVP)**: In addition to spatial and temporal neighbor MV predictions, VVC adds this new type of MV prediction in the merge mode and AMVP candidate list. The motivation of having HMVP candidates is to re-use the MVs of previously coded CUs, especially those of non-adjacent CUs. The HMVP candidates are established using a five-entry table that is maintained and updated using a first-in-first-out (FIFO) rule.
- **Symmetric MVD (SMVD**) is used to code the motion information for bi-prediction using a shortcut when feasible. In SMVD mode, only the AMVP indices for list 0 and list 1 and the MVD of list 0 are signaled. The other motion information, i.e. the reference picture indices and MVD of list 1, are implicitly derived by the decoder based on an inference of a constant motion trajectory.
- **Adaptive MV resolution (AMVR)**: VVC supports a selection of the MV resolution at the CU level to provide a more customized tradeoff between MV overhead bits and prediction quality. For inter-predicted CUs with translational motion, the selected MV resolution can be one quarter, one half, whole integer, or four, in units of luma samples. If half luma sample resolution is selected, an alternative luma interpolation filter is used for the half-sample position in this block. This aspect of AMVR is also known as switchable interpolation filter (SIF). The frequency response of the alternative filter is much smoother for the purpose of attenuating high-frequency noise components. For CUs coded in the affine AMVP mode, MV resolution can be switched among one quarter, whole integer, or one sixteenth luma samples (see Section II.B.3.c) for affine motion compensation).
- **Pairwise average MV merge candidate** is generated using the first two existing candidates in the merge candidate list. The MVs of this new merge candidate are calculated separately for each reference picture list. If both existing merge candidates contain an MV for the same list, these two MVs are averaged to obtain the MV for that list, even when they point to different reference pictures. In that case, the reference picture index from the first existing candidate is used. If only one existing candidate contains an MV for that particular list, it is used directly without averaging.
- **Merge with MVD (MMVD)** refines the MV of the merge mode by a signaled MVD, and can be deemed as a new tradeoff between the AMVP mode and the merge mode. The MVD in this mode can only be purely horizontal or purely vertical. The refining MVD is represented by indicating the selected direction and a distance in that direction.

b) Advances in CU-level motion compensation

VVC enhances CU-level motion compensation by introducing more flexible weighting of the prediction signals. This includes the ability to predict non-rectangular partitions inside a CU by applying weighting matrices to each prediction signal for bi-prediction combinations known as the geometric partitioning mode. Furthermore, VVC allows the combination of merge mode with intra-picture prediction and signaling of bi-

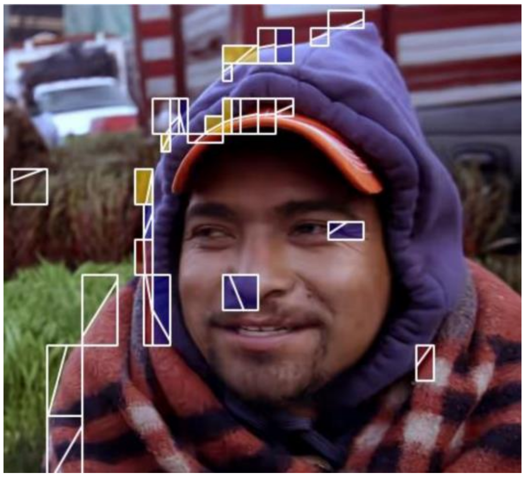

Fig. 5. Example of blocks using the geometric partitioning mode.

prediction weights at the CU-level. A short summary of the three tools is provided in the following, and the reader is referred to [26] for further details.

- **Geometric Partitioning Mode (GPM)**, sometimes called wedge partitioning, aims to expand the partitioning flexibility to better fit the non-rectangular boundaries of moving objects within a CU. When GPM is applied, the current CU is split into two parts by a geometrically located straight line, which is parameterized by an angle and an offset. An example of GPM-coded blocks is shown in Fig. 5. In total, 64 selectable partitioning lines are supported for CUs with sizes between 8×8 and 64×64, except sizes 8×64 and 64×8. GPM mode is an extension of the merge mode in which each partition inherits one MV from the merge candidate list. For each partition, a block-based motion compensation prediction is performed, resulting in two intermediate prediction blocks. The final prediction block is generated by performing a blending process using a weighting matrix that is derived based on the position of each sample location relative to the partitioning boundary.
- **Combined intra/inter-picture prediction (CIIP)** is introduced to take advantage of both an inter-prediction merge mode and intra prediction. In CIIP mode, the final prediction is a weighted combination of a merge mode inter-picture prediction and a planar mode intra-picture prediction. The combining weight is implicitly derived based on whether the above and left neighboring CUs are coded using an intra-picture prediction mode or not.
- **Bi-prediction with CU-level weights (BCW)** provides the choice of a non-equal weighting combination for bi-prediction at the CU level, in addition to the traditional weighted prediction (WP) for which weights are specified in syntax at the slice level for each reference picture. A set of 5 fixed weights equal to $\{-2, 3, 4, 5, 10\}/8$ is pre-defined for BCW, and an index into this set is signaled at the CU level to specify the selected weight *w* of the prediction block from list 1. The weight for list 0 is then set as 1 – *w*. When all the reference pictures are temporally preceding the current picture, all five weights may be used. Otherwise, only the weight value subset $\{3, 4, 5\}/8$ can be used.

c) Refined subblock-based motion compensation

VVC also introduces technologies that represent motion in higher granularity, e.g. with a subblock level, or further refine motion estimates at the decoder instead of using explicit signaling. VVC further increases the MV precision and fractional sample motion compensation to 1/16 luma sample in some modes. These new features are summarized in the following, and a more detailed description is provided in [27].

- **Subblock-based temporal MV prediction (SBTMVP)**: In addition to the CU-wise merge mode used in HEVC, VVC includes a subblock merge mode, which can be applied to CUs with both width and height larger than or equal to 8 luma samples. The subblock merge candidates consist of the SBTMVP candidate at the first place in the list, followed by affine motion merge candidates. SBTMVP inherits the motion information from a particular identified reference picture called the collocated picture, in units of 8×8 subblocks. SBTMVP derivation uses two major steps: a) establish a displacement vector (DV) for the current CU, and b) derive motion information for each subblock based on the motion identified by the DV. If the MV of the left neighboring block refers to the collocated picture, that MV is used as the DV to find the corresponding area for the motion information of the current CU in the collocated picture. Otherwise, the DV is set to zero.
- **Affine motion**: A high-order deformation model can capture non-translational motion, such as zooming and rotation, in addition to representing translational motion between the current picture and its reference picture. As a representative high-order deformation model, an affine motion model with CU-level signaling is introduced for luma in VVC. The CU-level affine motion can use a 4-parameter model or a 6-parameter model. The 4-parameter model is described by MVs of two control points located at the top-left and top-right corners of the CU and the 6-parameter model is described by MVs of three control points located at the top-left, bottom-left and top-right corners. When a CU is coded in affine motion mode, the luma block of the CU is spilt into 4×4 subblocks and the MV at the central sample position of each subblock is calculated according to the affine motion model and set as the subblock MV. The subblock MV is rounded to 1/16 luma sample precision during the calculation and a set of 6-tap interpolation filters is applied to generate the prediction of each subblock. Similarly as for translational motion, an affine merge mode and affine AMVP mode are also supported in VVC for efficient prediction and coding of affine motion parameters.
- **Prediction refinement with optical flow (PROF)** Motion compensation of the affine mode is conducted on the basis of 4×4 subblocks to achieve a balance of prediction quality and computational as well as memory access complexity. PROF is applied to refine each luma prediction subblock to mimic sample-wise motion compensation. PROF adjusts each prediction sample by an offset that is derived based on the gradient around the prediction sample and an MV offset

relative to the centered subblock MV. PROF is only applied to luma prediction blocks.

- **Decoder MV refinement (DMVR)** refines the bi-prediction motion of the regular merge mode by using a bilateral search step without transmitting additional side information. When the merge MVs point to two reference pictures that have equal and opposite temporal distance to the current picture, DMVR searches candidate MVs around the initial MVs in list 0 and list 1 with a mirrored MV offset. The searching process consists of an integer sample MV offset search and a fractional sample MV offset refinement. In the integer sample search, the sums of absolute differences (SADs) of each pair of candidate reference blocks are compared, within the search range of ±2 integer luma samples from the initial MVs in list 0 and list 1. When the initial MV has a fractional value, a bi-linear two-tap interpolation filter is used to generate reference blocks for searching. The fractional sample refinement is derived by minimizing a parametric error surface function instead of using additional SAD searching. When the width or height of a CU is greater than 16 luma samples, the CU is divided into subblocks with a maximum of 16 luma samples in width and height, and DMVR is applied to each subblock independently.
- **Bi-directional optical flow (BDOF)** is the second feature in VVC to improve bi-prediction efficiency by using a motion refinement technique performed by the decoder. Different from the motion searching applied in DMVR, BDOF is built upon the optical flow concept and is applied to CUs coded either in merge mode or AMVP mode. As done for DMVR, BDOF is applied only when the initial MVs point to two reference pictures that have equal and opposite temporal distance from the current picture. Based on an assumption of constant motion trajectory, a motion difference relative to the CU MVs is calculated for each 4×4 subblock by solving an optical flow differential equation which minimizes the difference between the prediction subblocks of list 0 and list 1. The derived motion differences are used together with the prediction sample gradients to adjust each bi-predicted sample value in the 4×4 subblock. When both DMVR and BDOF are applied to a CU, DMVR is preformed first and then followed by BDOF.

d) Horizontal wrap-around motion compensation

For some specific immersive video projection formats further detailed in Section III.G, a special case of motion compensation can be applied to alleviate the appearance of seam artifacts in 360^o video coded in the equirectangular projection (ERP) format or other 360^o video projection formats that share similar properties [28]. Unlike conventional motion compensation that applies repetitive padding in the decoding process when an MV refers to samples beyond the picture boundaries of the reference picture, horizontal wrap-around motion compensation fetches samples from the opposite side of the reference picture when part or all of the reference block is outside of the picture’s left or right boundary – hence the name “horizontal wrap-around.” This mode of motion compensation can be combined with an encoder padding method that is often used in 360^o video coding, such as for the padded ERP format, and such a combination is especially effective in improving the visual quality of extracted viewports of coded 360^o immersive video.

4. Transforms and Quantization

The basic concept of applying an integer transform to the prediction residual followed by quantization of the transform coefficients is well known from previous standards and is retained in VVC. Beyond this, VVC achieves better energy compaction of the prediction residual by extended transforms complemented by refined quantization and residual coding. The new transform and quantization aspects are described in the following. Details of the VVC transform design can be found in [29]. VVC quantization and residual coding is further detailed in [30].

- **Larger and non-square transforms**: HEVC employs separable square transforms with kernel sizes from 4×4 to 32×32. VVC additionally supports non-square transforms by combining different kernel sizes that are dyadically increasing from length-2 to length-64 horizontally and vertically. Additionally, 1D transforms are used with the ISP mode (see Section II.B.2). The main transform kernel type used in VVC is a straightforward extension of the DCT type-II based HEVC core integer transform to length 64. The transform of length 64 constitutes a special case because all coefficients outside a 32×32 area of a transform block are required to be equal to zero. For this purpose, an encoder would typically just set these coefficients to zero after applying the forward transform (a.k.a. "zeroing them out"), and if this zeroing introduces excessive distortion, the encoder would simply choose a different block size for that region. The zeroing out aspect was introduced to enable use of a longer transform while keeping the decoder implementation complexity similarly to that of a shorter one.
- **Multiple transform selection (MTS)**: In some cases, alternative transforms can better decorrelate the prediction residual, e.g., for the intra prediction residual where the prediction error tends to increase with increasing distance from the boundary samples. HEVC exploits this by including an additional 4×4 integer approximation of the DST type-VII for intra prediction luma residuals. VVC further extends this capability by defining four additional horizontal/vertical combinations of separate DST type-VII and DCT type-VIII integer kernels for all square and non-square luma block sizes from 4×4 to 32×32. The selection is either done explicitly by signaling an index per CU or by implicitly deriving it based on the width and height of the transform block. Similar to the DCT type-II based transform of length 64, zeroing out is applied to the non-DCT type-II coefficients outside a 16×16 area in order to reduce the implementation complexity of the additional transforms.
- **Low frequency non-separable transform (LFNST)**: In intra-coded blocks, a set of non-separable mode-dependent transforms can be applied by the encoder to the low

frequency coefficients of the DCT type-II based primary transform. The additional inverse transform kernels were derived from training data in order to exploit the remaining directionality characteristics of the intra-picture prediction residual signals. For that reason, LFNST can be seen as the second coding tool in VVC in addition to MIP that was developed using data-driven methods of machine learning. Inverse transform output sizes for this additional transform stage are either within a 4×4 or 8×8 region of frequencies, and for each size, four sets with two LFNST transforms each are defined. The set is selected based on the intra mode, and an index is explicitly signaled per CU to determine whether to apply LFNST to the block, and if so, which of the two inverse transforms in the set is applied. The number of outputs of the LFNST inverse transform process is 16 for blocks with a width or height of 4, and 48 for larger ones, and these become the low-frequency input to the ordinary DCT type-II based inverse transform. Similar to the zeroing out used with the core transforms, coefficients outside of the LFNST output region are set to zero. Further reduction of the storage size for LFNST matrices is achieved by limiting the number of input coefficients to the LFNST inverse transform to 8 for small blocks and 16 for the others.

- **Subblock transform (SBT) mode** chooses a sub-partition of the residual block of an inter-predicted CU to be coded and skips the remaining portion. The coded residual sub-partition can be half or one-quarter size of the CU and an MTS transform type for the coded residual is implicitly inferred. The coded half or quarter sub-partition can be either the left, right, top or bottom part, leading to an overall number of 8 modes to be signaled per CU.
- **Extended quantization control**: VVC maintains the uniform reconstruction quantizer design from HEVC with a quantization parameter (QP) controlling the quantizer step size. To be able to achieve lower bit rates, the largest QP value supported in VVC is larger than in HEVC. Aside from the constant offset of $6\cdot(b-8)$ that depends on the bit depth b of the decoded video samples, the maximum QP value in HEVC is 51 while VVC supports QP values up to 63. This increases the maximum inverse quantization scaling step size by a factor of 4. As in HEVC, the QP can be adjusted locally for the purpose of rate control and perceptual optimization. For that purpose, the concept of quantization groups as an area to signal a luma QP offset and scaling lists for frequency-dependent inverse quantization scaling from HEVC is kept in VVC and adapted to support the non-square block structures.
- **Adaptive chroma QP offset**: For the chroma components, QP values are derived from the QP of the corresponding luma block via look-up tables. In VVC, these are defined by piece-wise linear mapping functions that are coded in the SPS. This enables a VVC encoder to adapt the mapping to content with HDR and WCG characteristics. The QP values for the two chroma components can also be adjusted individually using syntax on a spatially localized basis, which was not supported in HEVC version 1.
- **Dependent quantization (DQ)** enables a switching between two scalar inverse quantizers for decoding each transform coefficient, depending on the previous quantized coefficient's value. Although using a scalar inverse quantizer at the decoder, it can be interpreted as a form of vector quantization, as it jointly codes the transform coefficients in an interdependent way by using a trellis-based search at the encoder. Another trick of dependent coding, introduced in HEVC as sign data hiding (SDH), is also retained as a lower-complexity alternative to DQ.
- **Joint coding of chroma residuals (JCCR)**: Remaining correlations in the quantized chroma residual signal can be efficiently exploited in VVC using a JCCR mode in which only one residual block is signaled and is used to derive residual blocks for both chroma components.

5. Entropy Coding

The entropy coding in VVC is performed using CABAC as the sole entropy coding method, as in HEVC. In VVC, the efficiency of CABAC is further improved by the following changes in the coefficient coding and probability estimation. More details on the VVC entropy coding can be found in [30].

- **Improved coefficient coding** changes the HEVC design by adding additional coefficient group sizes for new transform block sizes, using a reverse diagonal coefficient scan as the only scanning method and improving the probability model selections for the absolute transform coefficient levels.
- **High-accuracy multi-hypothesis probability estimation** employs two probability estimates associated with each context model. These estimates are updated independently with different pre-trained adaptation rates for each coded binary decision. The average of the two estimates constitutes the probability estimate used for interval subdivision in the binary arithmetic coder.

6. In-loop Filters

In VVC, improved and new signal-enhancing in-loop filters are applied to the reconstructed video signal before the pictures are used for output and as references for subsequent motion compensated prediction. This includes a new luma mapping with chroma scaling tool, where the inverse luma mapping part is applied before all other in-loop filters. The deblocking filter is well known from previous standards and aims at reducing blocking artifacts while preserving edges in the original video signal. Blocking artifacts tend to be introduced by differences in mean sample values between adjacent prediction blocks and adjacent transform blocks. In VVC the deblocking is modified to take into account the new block structures and coding tools with long deblocking support as well as different types of video signals by using luma-adaptive deblocking. The second new filter is an adaptive loop filter, which targets to enhance the reconstructed video signal, e.g. by using Wiener filter approaches at the encoder. The sample adaptive offset (SAO) filter from HEVC is kept without modification [31]. SAO operates after deblocking and feeds its output into ALF. SAO provides its own benefit on coding efficiency, and is of rather low complexity both for encoder and decoder implementations.

The novel in-loop filtering tools in VVC are described in the following; for more detail, the reader is referred to [32].

- **Luma mapping with chroma scaling (LMCS)**: Before deblocking and subsequent filters, VVC introduces a way to modify the dynamic range of the input signal using a luma inverse mapping function that is applied to the reconstructed video before loop filtering to match a forward mapping function applied by the encoder. From the block diagram shown in Fig 4, it can be seen that the forward mapping has to be applied to the inter-picture prediction signal as well because the inverse mapping has been applied to the reference pictures. Additionally, chroma residual scaling can be applied to adjust the chroma signal depending on the luma mapping, which can be used to balance the bits used to code luma and chroma samples. Originally designed for HDR signals, the redistribution of amplitude values by LMCS was shown to be beneficial for SDR signals as well.
- **Long deblocking filters**: The main deblocking filter design of HEVC is kept in VVC. The HEVC design includes the determination of a boundary strength (BS) for a set of four samples on an 8×8 grid, based on whether the prediction mode is intra or inter, the difference in motion across the boundary, non-zero transform coefficients, and QP-dependent spatial activity. For BS equal to zero, no filtering is applied in order to preserve strong edges in picture content. A BS greater than 0 indicates either weak or strong filtering for luma and normal filtering for chroma. Due to finer block partitioning in VVC, e.g. as introduced by SBT, ISP (see Section II.B.2) and subblock motion compensation (see Section II.B.3), the decision part from HEVC is extended to consider 4×4 luma sample transform blocks and 8×8 luma sample prediction subblock boundaries. Furthermore, the set of filters is extended by longer strong filters for luma and strong deblocking for chroma to reduce artifacts in relatively smooth areas of larger blocks. Compared to HEVC, the long deblocking filters in VVC increase the area of luma blocks to be processed in parallel from 8×8 to 16×16.
- **Luma-adaptive deblocking** controls the deblocking filter strength based on the average luma level of the reconstructed sample values. Up to four luma level threshold values and associated offset values for stronger deblocking can be signaled in the SPS. This is especially helpful for HDR content which has different non-linear transfer characteristics compared to SDR video. At the display, a corresponding non-linear process is invoked to transform the decoded video signal into linear light. Independent of the HDR scheme, this process tends to make distortions introduced by quantization more visible in areas of high or low brightness. Luma-adaptive deblocking allows for a stronger deblocking in these cases to alleviate visible distortions in such areas.
- **Adaptive loop filter (ALF):** applies a spatial filtering process to enhance the reconstructed video signal. For the luma component, the filter has a 7×7 diamond-shaped region of support, and for the chroma components the similar region of support is 5×5. Within this otherwise-linear filtering operation, a non-linear clipping can be applied to the difference between the current sample and its neighboring samples. This allows an encoder to also take into account the value similarity between current and neighboring samples by selecting and signaling appropriate clipping parameters. At the decoder, the selection of the luma filter is based on local classification of a 4×4 block into 25 classes using directionality and 2D Laplacian activity. Signaling an index into a set that contains 25 luma filters and an index for one of 8 chroma filters at the CTU-level enables a high degree of local adaptivity. The filter coefficients and clipping parameters are determined by the encoder, and multiple sets per coded video sequence can be signaled using an APS (see Section II.A.6).
- **Cross-component ALF (CC-ALF)** applies a 3×4 diamond-shaped high-pass filter per chroma component to the luma input samples of ALF. For each chroma sample after ALF, the filtered corresponding luma sample is used as a corrective offset. An encoder can determine four sets of CC-ALF filter coefficients per chroma component. Local adaptivity is provided by signaling one of the four filter sets per component at the CTU level, and the CC-ALF filter sets are conveyed together with the ALF parameters in an APS.

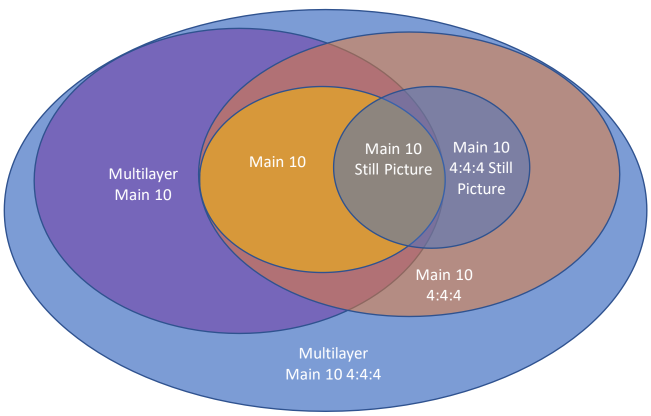

Fig. 6. The six profiles defined in VVC version 1.

7. Screen Content Coding Tools

Special coding tools are included in VVC to increase the coding efficiency for video that has different characteristics from camera-captured content. This particularly addresses the screen-captured content used in screen sharing applications and the computer-generated content dominant in gaming applications. These applications are further reviewed in Section III.E. The SCC tools in VVC are summarized below, and a more detailed description can be found in [34].

- **Palette mode** makes use of the fact that screen-captured content tends to use a limited number of color values for samples inside a local area. Therefore, the sample values in a palette-coded block are mapped to a reduced set of colors, i.e. the block's palette table, and each sample is represented by an index into the palette table or an index that indicates an "escape" color, in which case the quantized sample values are directly coded. Although applicable to video with 4:2:0 or 4:2:2 chroma sampling, the palette mode is especially effective for video with 4:4:4 chroma sampling. For this reason, the use of the palette mode in VVC is restricted to the 4:4:4 profiles (see the profile descriptions in Section III.A). The concept of the VVC palette mode is a straightforward adaptation from the palette mode in HEVC's SCC extension, with adjustment to support CST operation and some simplification in palette construction by re-ordering syntax elements for improved CABAC throughput.

- **Adaptive color transform (ACT)** enables a switchable decorrelating color transform for video with 4:4:4 chroma sampling in *RGB* color spaces, using a reversible YCgCo-R luma-chroma color transform which can be applied adaptively on a CU basis [33]. The ACT was basically carried over from the HEVC SCC extension with the difference that HEVC SCC additionally included a non-reversible transform, which has not been carried over. To avoid memory buffering issues, ACT cannot be combined with CST and cannot be used with the 64-length transform.
- **Intra-picture block copy (IBC)** exploits repeating patterns inside a picture by copying a spatially neighboring block of samples as the prediction of a block. Such patterns occur frequently in screen-captured or computer-generated content. Although conceptually similar to the IBC in HEVC SCC, its complexity has been significantly reduced by restricting the neighboring area from which a block can be copied and by using simpler methods to derive and signal the block displacement vectors.
- **Block-based differential pulse-code modulation (BDPCM)** is a concept known from HEVC as well. BDPCM exploits the same effect as the residual DPCM used in the HEVC range extensions (RExt). BDPCM applies horizontal or vertical DPCM on the residual samples resulting from either horizontal or vertical intra prediction instead of applying a conventional spatial transform to the residual samples. The AVC and HEVC fallback mode known as pulse-code modulation (PCM), which directly codes the video samples of a block using fixed length codes, is not carried over to VVC.
- **Transform skip residual coding (TSRC)** applies an alternative residual coding process that accounts for the different signal characteristics of the spatial residual resulting from skipping the transform. Since skipping the transform turned out to be beneficial for screen content residuals as well, VVC also incorporates the transform skip modes from HEVC RExt as well.

III. Applications, Test Results and Profiles

In the past, applications beyond typical standard- and high-definition camera-captured content coding required additional profiles, which have been added in later versions of a specification. In contrast, the first version of VVC already contains various new functionalities which we reviewed in the previous section. Most new functionalities are supported by the Main 10 profile of VVC for mainstream 4:2:0 YC_BC_R (a.k.a. *YUV*) video applications. Beyond this, version 1 of VVC also includes profiles for other video formats, such as the 4:4:4 chroma format typically used with *RGB* video, as well as still picture profiles to apply the VVC intra technology to image coding as well.

Following a description of the profiles of VVC version 1 and their levels and tiers, the rest of this section describes how these functionalities enable VVC to efficiently compress and handle the compressed video data for various types of applications. Additional applications and use cases beyond what has been envisioned during the development of VVC may also emerge in the future. One example of such a prior instance of unforeseen use is the tiles feature in HEVC. Originally proposed for parallel processing, tiles are now also used in combination with encoder motion search constraints and later-developed SEI messages to support the streaming of 360° video for localized viewport access.

A. Profiles, Tiers, and Levels

VVC version 1 specifies six defined sets of feature capabilities, which are called profiles, as shown in Fig. 6 and described below:

- **Main 10 and Main 10 4:4:4**: these single-layer coding profiles basically support all VVC coding tools (with the exception that the palette mode is supported in the Main 10 4:4:4 profile but not the Main 10 profile) and restrict the bitstream to contain only one layer, although there is no restriction on the use of temporal sub-layer scalability;
- **Multilayer Main 10 and Multilayer Main 10 4:4:4**: multi-layer profiles with the only difference compared to the two corresponding single-layer video profiles being that the bitstream can contain multiple layers; and
- **Main 10 Still Picture and Main 10 4:4:4 Still Picture**: single-picture profiles with the only difference compared to the two corresponding single-layer video profiles being that the bitstream can contain only one picture, which needs to be intra-picture coded.

VVC also provides general constraint syntax for constraining the individual features used in a VVC bitstream and supports the indication of externally specified subprofiles within the specified profiles based on these constraints.

Similar to HEVC, VVC specifies two *tiers* for the profiles, the Main tier and the High tier, with the High tier having 2 to 4.5 times higher capability in terms of bit rate and coded picture buffer size. One difference compared to HEVC is that in VVC the highest frame rate supported for the High tier has been raised from 300 frames per second (fps) to 960 fps, for support of video content captured with ultrahigh picture rate.

The definitions of the levels of capability for the profiles in VVC are basically similar as in HEVC for single-layer bitstreams, supporting a large range of spatial resolutions and frame rates, from as low as 176×144@15 fps in level 1.0 to as high as 8192×4320@120 fps in level 6.2. For multilayer VVC bitstreams, basically the same level definitions as for single-layer bitstreams apply, aiming at easy and simple upgrading of single-layer decoder designs to be capable of decoding both single-layer and multilayer bitstreams.

Table I
PSNR BD BIT RATE SAVINGS OF VVC (VTM-11.0) OVER HEVC (HM-16.22) FOR THREE EXAMPLE 10-BIT CONFIGURATIONS

Sequence	Random Access (entertainment)						Low Delay B (interactive)						All Intra					
class	Y	U	V	YUV	EncT	DecT	Y	U	V	YUV	EncT	DecT	Y	U	V	YUV	EncT	DecT
A1 (2160p)	39.6%	39.4%	46.2%	40.4%	676%	157%	–	–	–	–	–	–	29.0%	32.2%	34.1%	29.0%	1545%	169%
A2 (2160p)	43.4%	41.0%	40.2%	43.5%	762%	169%	–	–	–	–	–	–	29.3%	23.9%	21.1%	29.3%	2505%	177%
B (1080p)	36.5%	49.3%	47.7%	39.4%	751%	150%	30.8%	37.4%	35.5%	30.8%	744%	152%	21.7%	27.0%	30.8%	21.7%	2780%	177%
C (832×480)	32.8%	35.2%	37.2%	33.6%	1031%	175%	29.1%	22.6%	22.4%	29.1%	897%	157%	22.5%	19.0%	22.7%	22.5%	3886%	192%
E (720p)	–	–	–	–	–	–	33.4%	40.1%	34.2%	33.4%	357%	125%	25.8%	25.9%	24.5%	25.8%	2249%	170%
Average	**37.5%**	**41.9%**	**43.1%**	**38.9%**	**803%**	**162%**	**30.9%**	**33.2%**	**30.8%**	**30.9%**	**659%**	**147%**	**25.1%**	**25.4%**	**26.9%**	**25.1%**	**2576%**	**178%**
D (416×240)	30.7%	31.8%	31.4%	31.0%	1130%	170%	26.0%	16.6%	15.9%	26.0%	932%	165%	18.5%	13.3%	13.4%	18.5%	4414%	182%

B. Conventional Applications

VVC was designed with consideration of emerging applications and content characteristics from the beginning, although conventional applications such as SDR video and spatial resolutions below UHD still make up the majority of coded video today. In order to track the performance on conventional applications, JVET defined common test conditions (CTCs) for three example configurations, namely: random access (RA) for entertainment applications with 1 s random access periods; low delay bi-predictive (LDB) with temporally forward bi-predictive pictures for interactive application such as video conferencing; and all-intra (AI), restricting all pictures to use intra-picture prediction only. The CTCs also define classes of representative video test sequences with varying resolutions from 416×240 (class D) to 2160p UHD (class A) [36].

Table I reports the Y, U, V and combined YUV (using 6:1:1 weighting as defined in [37]) Bjøntegaard Delta (BD) bit rate savings measured (as positive numbers, otherwise computed as in [37]) for the VTM over the HEVC reference software model (HM) for the CTCs using the peak signal-to-noise ratio (PSNR) objective quality measure. Also shown in the table are the encoding times (EncT) and decoding times (DecT) of the VTM relative to the HM on similar computing platforms, to provide rough illustrations of relative complexity. To reflect the relevance of the test material to application use cases, some test sequence classes are not tested in combination with some configurations, as identified by "–" marks in Table I, and the low-resolution class D material is not included in the overall average measurements since it is considered not to represent the high priority applications for the project. It should be noted that for RA, the VTM-11.0 software employs by default an encoding technique known as motion compensated temporal filtering (MCTF) as a denoising prefilter to improve the motion prediction performance. Although not enabled by default in HEVC CTCs, MCTF is used for the RA configuration of HM-16.22 as well to allow for a fair comparison in these measurements. As a general tendency, it can be observed that VVC's PSNR coding efficiency improvement increases with increasing spatial resolution. The highest savings are reported for RA, with 42% YUV PSNR bit rate reduction on average for UHD (2160p). For AI, the measured 25.1% savings on average can be considered as quite substantial when considering that bit rate reduction in intra coding is particularly hard to achieve.

The coding efficiency for the UHD and HD SDR RA entertainment application scenario has been assessed in recent VVC verification test activities [38], [39] using VTM software and test sequences outside the CTCs. The YUV PSNR-based bit rate savings for each verification test sequence as well as the estimated mean opinion score (MOS) subjective quality savings based on formal subjective visual assessment are shown in Table II. In general, the tests confirm that the MOS-based bit rate savings seem to be somewhat higher than the PSNR-based savings. At almost 50%, the average HD SDR savings for entertainment application are higher than the ones reported for UHD, which seems surprising at first. Having a closer look, it can be seen that the bit rate savings vary across sequences and the highest bit rate saving of 62% over HEVC HM is reported for the UHD sequences DrivingPOV3. Thus, a direct comparison between UHD and HD results cannot be made but the tests confirm that VVC achieves significant bit rate reductions over HEVC HM in both cases.

Formal assessment for HD SDR LDB low delay applications has been performed as well [39] and the results are also shown in Table II. The results for HD SDR LDB confirm again the higher MOS-based savings compared to the YUV PSNR-based ones. Given the fact that several new inter coding tools in VVC

Table II
VVC VERIFICATION TEST YUV PSNR AND MOS BD BIT RATE SAVINGS OF VVC (VTM) OVER HEVC (HM-16.22) FOR 10-BIT SDR CONTENT AND VARIOUS APPLICATION SCENARIOS

UHD SDR (VTM-10.0)			HD SDR (VTM-11.0)								
Entertainment (RA)			Entertainment (RA)			Conversational (LDB)			Gaming (LDB)		
Sequence	PSNR	MOS	Sequence	PSNR	MOS	Sequence	PSNR	MOS	Sequence	PSNR	MOS
DrivingPOV3	42.8%	61%	BarScene	40.6%	48%	Beatriz	24.7%	41%	DOTA2	24.3%	34%
Marathon2	32.9%	37%	DrivingPOV	40.2%	58%	OfficeWalkAtWall	35.2%	28%	EuroTruckSimulator2	30.1%	42%
MountainBay2	38.6%	37%	Meridian2	40.0%	50%	OfficeWalkAtCeiling	31.8%	37%	Starcraft	31.6%	38%
NeptuneFountain3	26.8%	38%	Metro	30.7%	38%						
TallBuildings2	37.3%	41%									
Average	**35.7%**	**43%**	**Average**	**37.9%**	**49%**	**Average**	**30.6%**	**35%**	**Average**	**28.7%**	**38%**

require bi-prediction from opposite temporal directions, which is not allowed in LDB operation, the 35% bit rate savings for conversational and 38% for gaming low delay applications can be considered as significant as well. The higher savings for gaming content confirm the effectiveness of VVC in coding computer generated content as further discussed in Section III.E.

C. Video Beyond Standard- and High-Definition

Steady advances in display and camera technology in recent years have led to an increased use of higher definition video. This not only includes an increase in spatial resolution up to 8K and larger but also HDR and WCG for more vivid pictures. Although many televisions, personal computers, smartphones and tablets are equipped with HEVC encoders and decoders for capture and playback, the associated data rates are still rather high, especially for 4K and higher resolutions. For streaming, this stretches the limits of broadband and especially mobile network capacity. In broadcast, coding efficiency also limits the number of 4K channels, e.g., as currently used for sport events and movies, as well as preventing cost efficient broadcast of 8K video. Considering that more and more cameras and smartphones are now capturing video in 4K, increased coding efficiency saves local and cloud storage, which typically translates directly into cost savings.

All of the coding tools of VVC's core compression technology (Section II.B) contribute to its increased compression capability when coding video beyond standard- and high-definition video. In particular, the larger block sizes in VVC (see Section II.B.1) are a simple yet efficient way to represent flat areas which cover a larger area of samples in higher resolution video compared to the same content in lower spatial resolutions. In addition, adaptive chroma QP offsets (see Section II.B.4) allow encoders to adjust the chroma quantization to address the rich color experience offered by WCG. Among the in-loop filters, an encoder can select appropriate LMCS mapping functions appropriate for HDR transfer functions such as the hybrid log gamma (HLG) and perceptual quantizer (PQ) representations defined in Rec. ITU-R BT.2100 [40] and can also employ the luma-adaptive deblocking filter for HDR use (see Section II.B.6).

In addition to the new coding features in VVC, many HDR schemes will also use metadata that is conveyed to display devices by means of VUI and SEI messages specified in the VSEI specification [12], e.g. to indicate the electro-optical light transfer function, mastering display color volume, frame-average light level, content light level, alternative transfer characteristics for interpretation, reference ambient viewing environment, or content color volume. This enables the devices to interpret the video content properly and apply tone mapping that is optimized to the display capabilities.

In order to measure coding efficiency for HDR/WCG applications, JVET defined HDR CTCs with a class H1 of 1080p PQ sequences and a class H2, containing 2160p HLG sequences [41]. The HDR CTCs also include specific VTM encoder configurations with LMCS and chroma QP mapping parameter optimized to HLG and PQ transfer curves and WCG.

Table III
BD-RATE SAVINGS OF VVC (VTM-11.0) OVER HEVC (HM-16.18) FOR 10-BIT HDR CONTENT USING VARIOUS METRICS

Sequence class	**DE 100**	**PSNR L100**	**wPSNR Y**	**U**	**V**	**PSNR Y**	**U**	**V**
H1 (1080p)	44.3%	41.8%	37.6%	53.4%	46.5%	34.9%	48.9%	39.9%
H2 (2160p)	–	–	–	–	–	31.8%	66.0%	60.2%
Average	**44.3%**	**41.8%**	**37.6%**	**53.4%**	**46.5%**	**33.8%**	**55.1%**	**47.2%**

Besides PSNR, the HDR CTCs define additional objective distortion measures such as weighted PSNR (wPSNR) and two metrics for PQ evaluating the distortion in the CIE 1976 Lab color space, namely deltaE-100 (DE100) and PSNR-L100. Table III shows the results of the VTM compared to the HM according to the HDR CTCs. It can be observed that the increase in chroma efficiency is higher (mostly caused by an increased fidelity) and that the DE100 and PSNR-L100 metrics report a higher efficiency than wPSNR and PSNR. For the HDR application scenario, verification testing efforts using formal subjective assessments are still ongoing. At the time of this writing, no verification tests are yet planned by the JVET for 8K due to viewing equipment-related challenges and the circumstances of the COVID-19 pandemic. However, similar or better results would be expected for 8K, and the industry has already started evaluating VVC for 8K with particular use in live broadcast. It was reported that a commercial VVC encoder in a live context already showed 24% coding gain compared to live HEVC encoding, in spite of the fact of that this was still at an early development stage [42].

D. Adaptive Streaming with Resolution Changes

In streaming applications, due to the fact that network

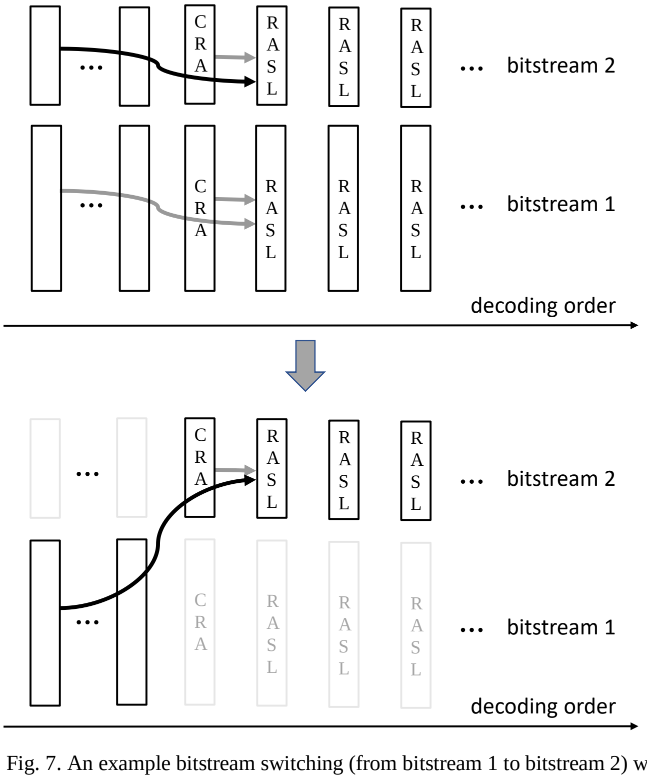

Fig. 7. An example bitstream switching (from bitstream 1 to bitstream 2) with different spatial resolutions at a CRA picture with associated RASL pictures.

bandwidth can change from time to time, adaptive streaming through bitstream switching is widely applied to optimize the user experience. If the network capacity drops below the media bit rate, the playback would soon be disrupted due to rebuffering. In this case, a bitstream with a lower bit rate should be served to minimize rebuffering and playback freezes. If the network bandwidth stays higher than the media bit rate in a stable manner, a bitstream with higher bit rate and hence higher quality should be served so that the user can enjoy higher video quality. Switching to a different bit rate often also involves switching to a different spatial resolution, as beyond a certain bit rate the quality of another spatial resolution can be better for the given video content.

As discussed earlier in Section II.A.1, CRA pictures with associated RASL pictures provide the highest coding efficiency while providing random access capability. For example, by using such CRA pictures and the RPR feature in some cases versus using IDR pictures in adaptive streaming, average coding gains ranging from 2.4% to 8.7% for different configurations were reported in [43] and [44]. However, the handling of bitstream switching involving resolution changes at such CRA pictures has long been a challenge, because of the RASL pictures associated with the CRA picture at the switching point. Fig. 7 shows an example of bitstream switching between different spatial resolutions at a CRA picture with associated RASL pictures without the RPR feature. After the switching, the decoder simply cannot decode these RASL pictures, because when a reference picture for the RASL picture is from the previous bitstream before the switching-point picture in decoding order, the reference picture will have a different spatial resolution, as shown by the arrow in the solid line in the lower part of Fig. 7. Simply discarding these RASL pictures would result in a gap in the displayed video, and consequently the bitstream switching is not a seamless switching. Another option is to fetch not only the CRA picture of the new bitstream but also the CRA picture at the same location of the previous bitstream and its associated RASL pictures, use the RASL pictures of the previous bitstream for playback, and discard the RASL pictures of the new bitstream. In this case, at the location of the switching point, two CRA pictures need to be transmitted and decoded. Actually, the sets of RASL pictures of both bitstreams would typically need to be transmitted, as normally a CRA and its associated RASL pictures are encapsulated in the same media segment in a streaming environment such as DASH [45].

The RPR feature of VVC can be employed to solve the above issue. With RPR in use, after switching to a bitstream with a different spatial resolution at a CRA picture, an associated RASL picture can use a reference picture that has been decoded from the previous bitstream, again as shown by the arrow in solid line in the lower part of Fig. 7. Indeed, the decoded RASL picture could have a mismatch in this case. However, the content provider can control both the encoding of the video bitstreams and the DASH signaling to allow such bitstream switching only at positions for which the decoding mismatch of the RASL pictures won't result in noticeable user experience degradation.

E. Computer Generated and Screen Captured Content

Previous video coding standards have been developed targeting the compression of camera-captured video content. Such video signals typically exhibit camera sensor noise as well as rather smooth transitions on edges. Nowadays the dominance of that kind of content is challenged by emerging applications such as online gaming and screen sharing in teleconferencing. In such cases, the video is not captured by a camera but rather by capturing the graphics buffer of the computing device instead. This computer-generated content can be characterized by sharper edges and a greater amount of flat, uniform-colored areas, which results in a different distribution of spatial frequencies within the video signal. With the increasing popularity of online gaming and the streaming of such content, the amount of computer-generated video data in global traffic is steadily increasing. Moreover, the recent massive increase of the use of personal and business video conferencing due to the COVID-19 pandemic means that screen sharing is also becoming more common, and this adds to the increased need for efficient coding of screen content. The encoding of mixtures of computer-rendered and camera-captured content has also become increasingly common, either using graphics overlays or windowed regions. When it comes to gaming content, it should be noted that video resulting from super realistic rendering has signal characteristics comparable to camera-captured content and only lacks camera sensor noise.

HEVC was the first video coding standard to include coding tools designed especially to exploit the different characteristics of screen content for more efficient compression. However, these profiles were only added in its version 4 with the SCC extension [35], excluding them from all devices already on the market which mainly support only the version 1 profiles of HEVC. It is also worth noting that some functionalities beneficial for coding screen content had already been introduced in the second version of HEVC, i.e. in the RExt extensions [46]. Given the increasing amount of SCC video data, VVC version 1 already includes the following tools and functionalities to efficiently code screen content:

- **4:4:4 profiles** are defined already in version 1 of VVC. This enables the coding of video from a graphics buffer in its native *RGB* color space with the 4:4:4 color sampling format. For HEVC, similar profiles had been introduced only in the second version, with HEVC RExt.
- **Lossless coding** is supported in VVC by using transform skip with a QP value that skips scaling at the decoder. Since mathematically lossless coding is an end-to-end system property, it requires an encoder to not modify the residual

Table IV
BD BIT RATE SAVINGS OF VVC SCC (VTM-9.0) OVER HEVC (HM-16.20) AND HEVC SCC (SCM-8.6)

Test Sequences	Over HM-16.20			Over SCM-8.6		
	AI	RA	LB	AI	RA	LB
Class F 4:2:0	39.84%	42.52%	43.23%	23.85%	30.58%	34.19%
TGM 4:2:0	62.61%	61.14%	61.85%	13.99%	27.58%	34.41%
SCC YUV 4:4:4	52.56%	51.22%	49.22%	15.80%	24.89%	31.12%
SCC RGB 4:4:4	55.83%	53.42%	51.32%	13.68%	21.90%	26.80%

samples and to disable tools that can alter the residual and reconstructed samples, such as JCCR and in-loop filters. If lossless coding is applied to the whole picture in VVC, its coding efficiency is not significantly different from that of HEVC RExt. Therefore, the main application benefit is considered to be region-wise lossless coding, e.g. to preserve unaltered quality for semantically important parts of the video, e.g. street signs, faces and license plates in natural content and text or logos in screen-captured content. Since the modified residual coding for transform skip was found to be particularly more efficient for screen-captured content, VVC also allows use of the regular residual coding method for lossless compression of camera-captured content by means of a slice-level switch.

The PSNR compression performance of coding screen content using VVC version 1, compared with using HEVC version 1 and version 4 (with the SCC extension), was studied in [47]. The summary result of combined YUV (using 6:1:1 weighting as defined in [37]) and RGB (with 1:1:1 equal weighting) PSNR-based BD bit rate savings (as positive numbers, otherwise computed as in [37]) is presented in Table IV. It is seen that compared with HEVC version 1, VVC version 1 provides substantial coding gain for such content – more than 50% on average. The VVC compression benefit also remains significant (more than 20% on average) when compared with the HEVC SCC extension. The verification test results on HD SDR gaming content for the LDB low delay configuration [39] confirm the effectiveness of VVC in coding computer generated content, with almost 40% bit rate savings over HEVC HM on average (see Table II).

In addition to the aforementioned features, most use cases such as screen sharing in teleconferences, wireless displays and online gaming requires low delay streaming which is discussed in the next section.

F. Ultralow-Delay Streaming

In addition to the delay caused by capturing, pre-processing, and post-processing, the end-to-end delay in a video communication system consists of five parts: 1) the encoding time, 2) the transmission time, 3) the initial buffering delay, i.e., the coded picture buffering delay needed to cope with bit rate variations that cause transmission time variations, 4) the decoding time, and 5) the initial output delay, i.e., the decoded picture buffering delay needed to cope with the maximum amount of picture reordering. With the assumption of the delays introduced by capturing, pre-processing, picture reordering, and post-processing all being zero, the first four elements of the end-to-end delay listed above are illustrated as A, B, C, and D, respectively, in Fig. 8.

Conversational applications, e.g., video telephony and video conferencing, require low end-to-end delays. In conversational applications, picture reordering before encoding is usually not allowed, and consequently, the decoding order of pictures is the same as the output order, and the initial output delay is assumed to be zero. Some other applications, e.g., wireless display and some online gaming applications, have even more stringent requirements on end-to-end delay. For such applications, picture reordering would not be used, and the initial output delay is also assumed to be zero.

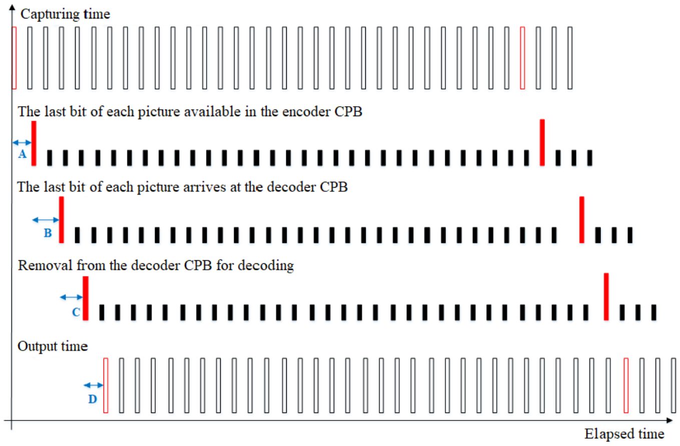
Capturing time
The last bit of each picture available in the encoder CPB
A
The last bit of each picture arrives at the decoder CPB
B
Removal from the decoder CPB for decoding
C
Output time
D
Elapsed time

Fig. 8. Illustration of four elements of end-to-end delay: A) encoding time, B) transmission time, C) initial buffering delay, and D) decoding time, where there is no picture reordering (i.e., decoding order is the same as output order) and the pictures shown in red are IDR pictures. The height of each solid rectangle, representing a coded picture, provides a rough indication of the number of bits used for the coded picture.

To further reduce the end-to-end delay, using more computing power could be considered to reduce encoding time and decoding time, and improving the network bandwidth could be considered to reduce the transmission time. However, in a given environment with given computing and networking resources, one can only rely on other approaches.

One such approach targets at reducing the initial buffering delay. Since this delay is caused by bit rate variations, it is straightforward to see that smoothing out the bit rate can reduce this delay. One way to smooth out the bit rate while avoiding skipping the encoding of some of the source pictures is to apply QP-based bit rate control such that the IRAP pictures are coded with similar amounts of bits as the inter-coded pictures. However, this results in low quality for the IRAP pictures, which could not only cause lower quality or higher bit rate for other pictures that reference them, but would also produce a significant quality variation between consecutive pictures that can be visually annoying. Another approach is to skip a few pictures following an IRAP picture during encoding. This would not reduce the end-to-end delay of the IRAP picture, but after displaying the IRAP picture, the next decoded pictures can be displayed earlier by not waiting until after the same time difference between their capturing time and the capturing time of the IRAP picture, thus the end-to-end delay of those pictures can be reduced. However, this results in jitter in the temporal domain due to the speeding up in displaying of pictures after an IRAP picture.

Fortunately, using GDR (see Section II.A.1) can smooth the bit rate while avoiding these problems and avoiding the skipping of source pictures by the encoder. This is because GDR effectively distributes the intra coded samples, which are the main cause of the bit rate spikes, from all being within one picture in the IDR case to being spread across multiple pictures within the same random-access period, e.g., as shown in Fig. 9. Once the bit rate spikes are removed, the sum of the worst-case transmission time for a coded picture and the initial buffering

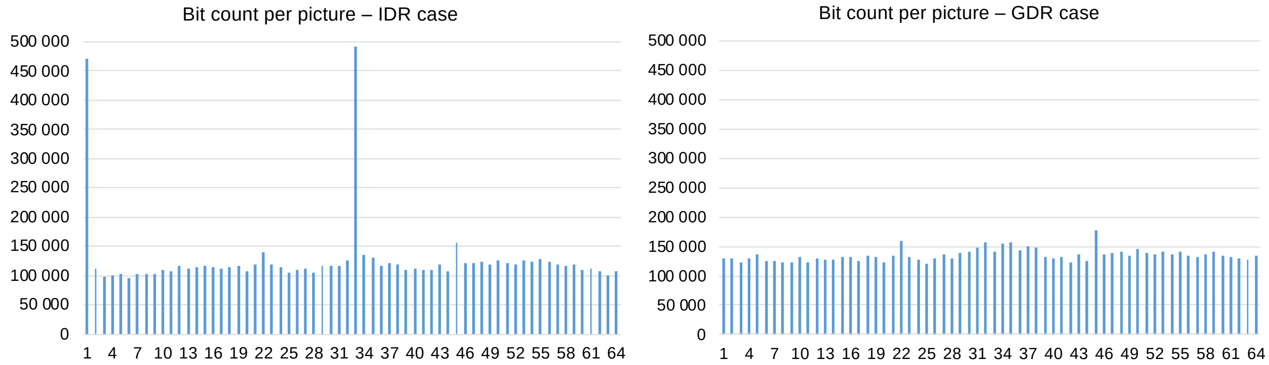

Fig. 9. Example per-picture bit counts for using IDR and GDR pictures, respectively, for random access, where the random access period is 32 pictures. As can be seen, for GDR the median bit count for each picture is somewhat higher, but the peak bit count is much lower.

delay (i.e., the delay elements B and C, respectively, shown in Fig. 8) can be significantly reduced, which means that the overall end-to-end delay is significantly reduced.

There are two other approaches to further reduce the end-to-end delay, and these approaches can be applied together with GDR. The first is early sending, i.e., to start sending bits of a coded picture before the entire picture is encoded. The second is early decoding, i.e., to start decoding part of a picture before the entire coded picture is received. Assuming that each picture is coded into a number of slices, each slice is in its own VCL NAL unit, and each slice is placed in the network sending buffer immediately after its encoding, this can significantly reduce the gap between encoding time and transmission time (the distance between elements A and B in Fig. 8) compared to sending the bits of a coded picture only after the entire picture is encoded. Likewise, when it is possible to start the decoding of the first part of a coded picture before the reception of all bits of that picture, the distance between transmission time and decoding time (elements B and C in Fig. 8) can be significantly reduced.

An analysis in [49] showed that, in an example setting for a video sequence with random access period of 32 pictures, the end-to-end delay can be reduced from 253 ms to 91 ms by applying GDR, from 91 ms to 61 ms by further applying early sending, and from 61 ms to 36 ms by further applying early decoding.

Note that the early sending approach can be applied by implementations without the need of a change to the video coding specification, which specifies the bitstream format and the decoding process and does not specify the encoding process. As detailed in [17], VVC specified GDR in a "more normative" manner, such that the GDR picture is indicated by a distinct NAL unit type, and a conforming bitstream can start with a GDR picture and is not required to contain any IRAP pictures. To support the earlier decoding approach, VVC also inherited the decoding-unit-based buffering model [50] as part of the HRD specification, to support it in a clearly specified manner.

G. Immersive Video

In a perfect immersive media application, when a user consumes the media, the user would feel like as if they were in the environment where and when the media such as video and audio were captured. One of the immersive media applications is 360^o video, wherein, when a user turns the head from one direction to another, the video perspective they see would transition from the old viewing direction to the new direction accordingly. To achieve such a user experience, the video is usually captured by a camera rig with multiple lenses that can observe the video scene from all directions around the camera rig. The captured video signals are then stitched into a spherical signal, often with two views to generate a stereoscopic effect.

To be able to apply 2D video coding schemes for compression of spherical video, the spherical video signal is projected onto a 2D rectangular raster signal using a certain projection format, e.g., ERP or CMP. The mapped pictures are then encoded and transmitted to the user and an inverse mapping is applied in the decoding system for interpreting the video content. More details on such a framework for 360^o video usage can be found in [51].

For 360^o video signaling, the ERP, sphere rotation, region-wise packing, and omnidirectional viewport SEI messages have been inherited from HEVC and AVC for use with VVC. These SEI messages are specified in the VSEI standard [12]. A generalized cubemap (GCMP) projection SEI message has also been specified to provide an extended functionally over the cubemap projection SEI message previously specified for HEVC and AVC.

Efficient support for immersive video applications has been one of the main goals during the development of the VVC standard. Most VVC tools and features discussed in Section II that help conventional video applications can also help better compress immersive video content. Besides those, the following tools or features have been designed more specifically for immersive applications: extractable subpictures for BEAM operations (see Section II.A.4), wrap-around motion compensation (discussed in Section II.B.3.d) and further below in this section), and disabling in-loop filters across virtual boundaries (see Section II.A.5).

The ERP projection format is one of the most commonly used 360^o video projection formats. For the ERP format, the left and right edge of the pictures are continuous in the spherical domain, but discontinuous in the projected rectangular domain [19]. In conventional motion compensation, when an MV refers to samples beyond the picture boundaries of the reference picture, repetitive padding is applied to derive the values of the out-of-bounds samples by copying from those

Table V
BIT RATE REDUCTION OF VVC (VTM-10.0) OVER HEVC (HM-16.16) FOR 360° VIDEO USING THE PADDED ERP FORMAT AND THE PADDED CMP/GCMP FORMATS, BIT RATE SAVINGS MEASURED USING E2E WS-PSNR YUV

	Padded ERP			**Padded CMP / GCMP**		
	Y	U	V	Y	U	V
Class S1	25.6%	33.6%	36.4%	29.4%	34.9%	37.2%
Class S2	35.1%	33.0%	35.6%	36.7%	34.5%	36.9%
Average	**29.4%**	**33.4%**	**36.1%**	**32.3%**	**34.8%**	**37.1%**

nearest neighbors on the corresponding picture boundary. Thus, conventional motion compensation causes the left and right edge of an ERP picture to be coded in a disjoint manner, which in turn often leads to visible seam artifacts when a viewport that encompasses the left and right edges of the ERP picture is generated after compression. Horizontal wrap-around motion compensation solves this problem by addressing the discontinuity issue between the reference picture's left and right boundaries. As depicted in Fig. 10, when a part of the reference block is outside of the reference picture's left (or right) boundary in the projected domain, instead of repetitive padding, the "out-of-boundary" part is taken from the corresponding spherical neighbors that are located within the reference picture toward the right (or left) boundary in the projected domain. Samples along the top and bottom picture boundaries also may have corresponding spherical neighbours within the same (not the opposite) picture boundaries, but it is less straightforward to identify those; further, viewing experiments indicate that discontinuities along the top and bottom boundaries tend to have less visual impact. Therefore, wrap-around padding only applies in the horizontal dimension, and the conventional repetitive padding is still used for the top and bottom picture boundaries. Besides the ERP format, horizontal wrap-around motion compensation can also be used for other projection formats with constant sampling density in the horizontal dimension. In VVC, an SPS flag is signaled to indicate whether horizontal wrap-around motion compensation is enabled, followed by a wrap-around offset used to calculate coordinates of the corresponding samples from the other side of the reference picture.

Table V shows the VTM-10.0 performance compared to HM-16.16 for the padded ERP and the padded CMP/GCMP projection formats, according to the JVET CTCs for 360° video [52]. Positive numbers indicate BD-rate reduction. The JVET 360Lib-11.0 software package [53] is used to perform projection format conversion and compute spherical quality metrics. The spherical quality metric shown is weighted spherical PSNR (WS-PSNR [54]) measured in the end-to-end manner according to the 360° video processing and compression workflow defined in [19]. For the cubemap projection formats, the HM uses the CMP projection format with padding as described in the CMP SEI message in HEVC, and the VTM uses the GCMP projection format as described in the corresponding new VSEI message, and the improved blending for GCMP from [55] is included in post-processing to compute the end-to-end WS-PSNR. Compared to HEVC, VVC achieves average luma bit rate reductions of 29.4% and 32.3% for the padded ERP and the padded CMP/GCMP formats, respectively, for 360° video content; and higher gains can be achieved for the chroma components. Subjective verification testing efforts for 360° video have been conducted [39]; the results for both padded ERP and padded CMP / GCMP formats are shown in Table VI. On average, VVC achieves MOS-based BD bit rate savings of 50% and 56% for the padded ERP and padded CMP / GCMP formats, respectively. Such gains in 360° video coding efficiency confirm the suitability of VVC for immersive video applications, which are expected to grow significantly in the coming years.

Table VI
VVC VERIFICATION TEST MOS BD BIT RATE SAVINGS OF VVC (VTM-11.0) OVER HEVC (HM-16.22) FOR 360° VIDEO PADDED ERP AND PADDED CMP/GCMP CONTENT

Sequence	**Padded ERP MOS**	**Padded CMP / GCMP MOS**
GTSheriff	40%	47%
HarborBiking2	49%	50%
KiteFliteWalking2	53%	62%
SkateBoardAtBridge	58%	67%
Average	**50%**	**56%**

H. Conferencing and Broadcasting Applications

Many video applications, such as video conferencing, broadcasting and streaming, can benefit from scalability. These applications need to fulfill the requirements of backward compatibility to a lower spatial resolution and/or lower frame rate to accommodate the co-existence of legacy devices along with new devices, and/or to adapt to network bandwidth

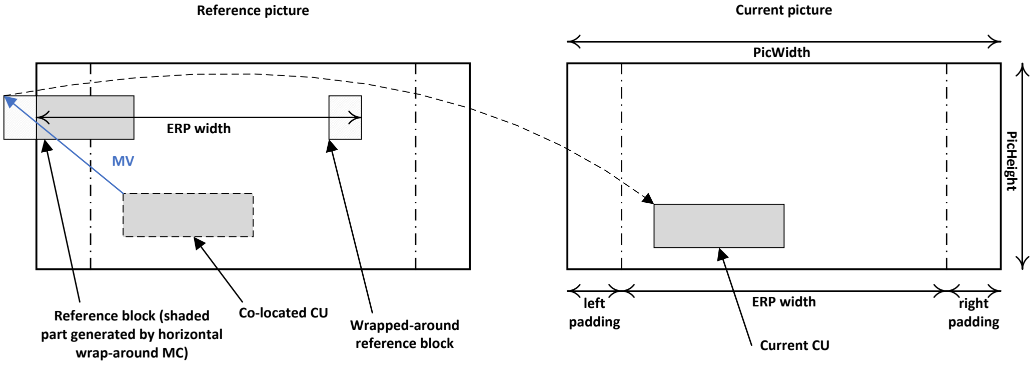

Fig. 10. Illustration of horizontal wrap-around motion compensation.

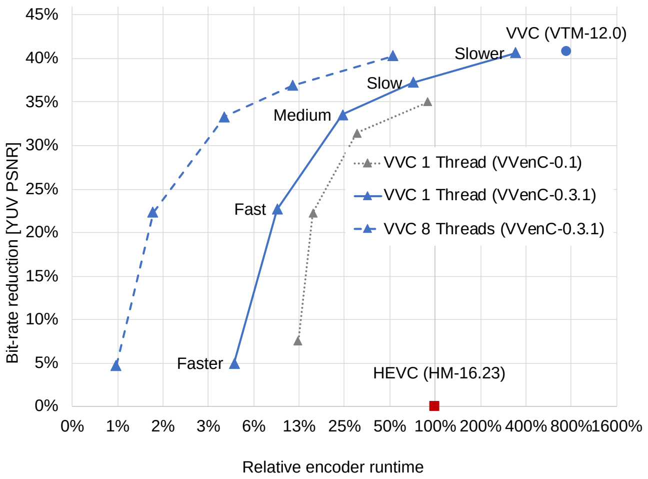

Fig. 11. YUV PSNR bit rate reduction versus encoder runtime of VTM and VVenC relative to the HEVC reference software (HM-16.23) for JVET CTC class B (HD) and class A (UHD) test sequences in random access configuration.

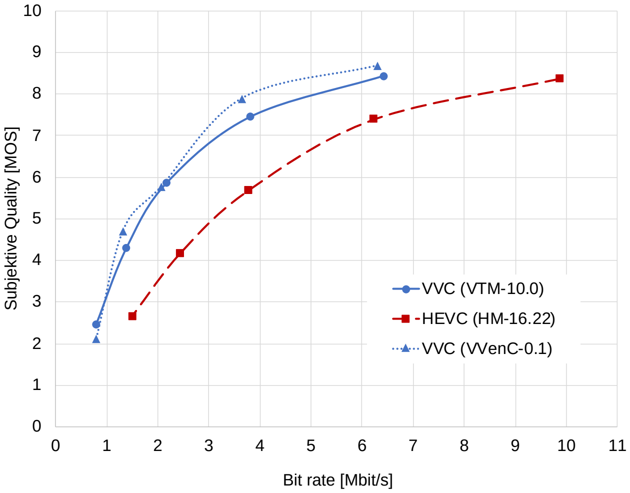

Fig. 12. Subjective quality versus bit rates, pooled from the five UHD SDR sequences used in formal subjective assessment of VTM and VVenC ("medium" preset) relative to the HEVC reference software (HM-16.22).

fluctuation quickly. For broadcasting applications, ATSC 3.0 [56] adopted the scalable extensions of HEVC (SHVC) [22] as an optional format to support spatial scalability. Combined with the MIMO technology in wireless communications that uses multiple antennas to enhance connectivity and offer better speeds, spatial scalability can be used to serve a diverse set of devices with high coding efficiency. If a device (a TV or set-top-box or mobile phone) only receives the base layer video on one (e.g. the vertical) antenna, the consumer can watch the baseline service, e.g. at 4K or 1080p quality. If the device receives both the base layer and enhancement layer on two (e.g. both vertical and horizontal) antennas, then the consumer can combine the two together and enjoy the higher quality service, e.g. at the 8K or 4K quality. For video conferencing, temporal scalability using the hierarchical-B prediction structure with only temporally forward prediction is an effective mechanism to cope with bandwidth fluctuation, as it allows adaptation of the frame rate to the currently available bandwidth by coding pictures in different temporal layers. VVC version 1 supports scalability features and layered coding (see Section II.A.7 for design details) so as to facilitate the implementation and deployment of decoders with multi-layer decoding capability for conferencing, broadcasting and other applications that can benefit from scalable video coding.

I. 3-Dimensional Video

The multilayer coding functionality in VVC also enables coding of 3-dimensional (3D) video signals. This is conceptually similar to the HEVC multiview extension (MV-HEVC) [23] introduced together with the scalable extensions in version 2 of HEVC. For 3D video, each layer can represent a different camera perspective view or one view and a corresponding depth map that the rendering of arbitrary views. It should be noted that inter-layer prediction, e.g. between different views is possible in VVC, similar to MV-HEVC. However, specific tools for coding depth maps as introduced with the HEVC 3D video coding extension (3D-HEVC) [23] in the third version of HEVC are not supported in VVC. It can however be expected that also depth maps can be compressed much better already with the basic tools of VVC, and furthermore, depth maps generally require much less bit rate than video pictures.

IV. Early Implementations

During the development of a video coding standard, the effectiveness of compression algorithms under consideration is usually evaluated by implementing them in a software testbed also referred to as reference software or as a test model. Such a reference model is designed primarily for completeness and as a flexible platform for experimentation to test and demonstrate the full functional capabilities of the standard and to test possible extensions and modifications, which makes the reference software typically very slow and gives it a large memory footprint. This is one of the reasons why reference software codebases are hardly ever used directly in practical applications, although such software is often used as a starting basis for developing such practical solutions. However, when compared to the similarly structured reference software of a previous standard, some tentative conclusions with regard to coding efficiency and complexity increase can be drawn. A more precise analysis of VVC implementation complexity, which includes the VTM reference software as well as some of the optimized implementations presented here, is provided in [57].

This section aims at providing an overview of early implementations of VVC and supporting tools that have already become available at the time of writing in addition to the VTM reference software, with a particular focus on packages developed with the involvement of coauthors of this paper. This includes an open, publicly available encoder implementation and several fast decoder implementations, as well as tools mainly used in research and development of VVC conforming implementations. Here, bitstream analyzers are of particular interest to visualize bitstream properties such as block partitioning and mode distribution as well as special bitstreams, designed to test decoder conformance to the VVC standard. A survey of early VVC implementations and supporting technology is available in [58].

A. Open, Optimized Versatile Video Encoder (VVenC)

In September 2020, two months after the design work on VVC was officially completed, Fraunhofer HHI published the first version of its source code for an optimized VVC encoder implementation called VVenC [59]. This section focuses on that encoder, although other encoders supporting VVC have been announced as well (e.g., [60], [61]). The main goal of VVenC is to make a VVC software encoder available that can achieve the coding efficiency of the VTM reference software at a fraction of its runtime. Beyond that, it includes additional functionalities particularly useful for real-world applications that are not of critical importance for a reference software. As one example for a real-world application, VVenC has been integrated in a cloud encoding platform [62]. In summary, the current version 0.3.1 of VVenC has the following features:

- **Five predefined quality/speed presets** (called *slower*, *slow*, *medium*, *fast*, and *faster*) can be used to trade off encoder runtime and coding efficiency. Fig. 11 shows the YUV (using 6:1:1 weighting as defined in [37]) PSNR-based BD bit rate savings (as positive numbers, otherwise computed as in [37]) over relative encoder runtime of different VVenC presets and the VTM [63] over the HEVC test model (HM) reference software [64]. It should be noted that VVenC, VTM-12.0 and HM-16.23 employ MCTF according to the JVET CTC random access configuration [36], [65].
- **Perceptual optimization** can be enabled to improve the subjectively perceived quality by adapting the QP based on human visual sensitivity. For areas where human perception is more sensitive to quantization noise, the QP is decreased which means that more bits are spent to code these areas. In order to estimate the perceived subjective quality, a perceptually weighted metric called xPSNR is used within the VVenC encoder [65]. Since it is similar to the ordinary PSNR measure, it can be used in block-level encoder control. A recent VVC verification test showed that the perceptually optimized first version of VVenC (v0.1) in *medium* preset configuration further reduced the bit rate by about 12% for the same subjective quality measured by MOS when compared to the VTM which does not use any subjective optimization. It should be noted that, in contrast to VVenC, both VTM-10.0 and HM-16.22 do not make use of MCTF. Fig. 12 plots the MOS over the bit rate, pooled from the five UHD SDR verification test sequences for the HM, VTM and VVenC [38]. In the subsequent HD SDR verification test, VVenC v0.3 in *medium* preset also outperformed VTM-11.0 in MOS-based BD bit rate savings on average. In this test, VVenC, VTM and HM had MCTF enabled [39].
- **Rate control** is included to support streaming applications. The first version of VVenC includes a simple 1-pass and a more efficient 2-pass variable bit rate (VBR) rate-control algorithm with signaling of corresponding buffer parameters.
- **Multi-threading** is used to exploit CTU-line and picture-level parallelism. Fig. 11 also shows an additional VVenC configuration with 8 threads, and it can be seen that this leads to further speed improvement. By default, VVenC employs CTU-line parallel encoding using a wavefront-like processing without the normative synchronization of the entropy coder (see Section II.A.3). Enabling entropy coding synchronization for wavefront parallel encoding can bring additional speedup at the cost of slightly reduced coding efficiency.
- **Versatility** is provided to the extent that VVenC supports high-level syntax for open GOP resolution switching with RPR as described in [44] (see Section III.D) and the SCC tools of the Main 10 profile except IBC (see Section III.E).

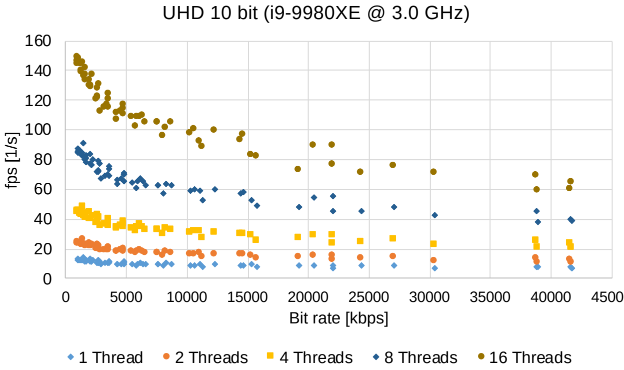

Fig. 13. VVdeC v0.1 performance in fps using multiple threads at different bit rates for the JVET CTC class A (UHD) test sequences.

VVenC has had four version releases at the time of writing of this paper, and is still being further improved. New versions are expected to include further algorithmic improvements for runtime and coding efficiency as well as support for tiles, and IBC as its last currently missing Main 10 profile SCC tool. In addition, support for color formats beyond 4:2:0 and integration of the encoder into open multimedia frameworks such as FFmpeg are also under consideration as future enhancements.

B. Optimized VVC Decoders

During the VVC version 1 development and shortly thereafter, several early VVC decoder software implementations and prototypes for live decoding were demonstrated. Real-time software decoding with up to 60 frames per seconds (fps) of 10-bit 4K video has been achieved for target devices ranging from ARM-based mobile devices to personal computers based on x86 processors [67]–[72]. Within a few months after the VVC standard was finalized, Fraunhofer HHI made their optimized Versatile Video Decoder (i.e., VVdeC) available on GitHub [73], [74]. Tencent also presented an independently developed VVC decoder (i.e., O266dec) [75] which was implemented from scratch and runs on multiple operating systems including Linux, Microsoft Windows, Apple Mac OSX, iOS and Android [76]. Alibaba demonstrated Ali266, a thin VVC decoder that runs efficiently on mobile phones of wide-ranging hardware capabilities [72]. This section focuses on these three decoders, although others have been

Table VII
VVC O266DEC AND VTM DECODER RUNTIMES IN FPS AVERAGED OVER ALL JVET CTC SEQUENCES OF THE RESPECTIVE CLASS.

Test Sequences	VTM (fps)	O266 (fps)			
		1 Thread	2 Threads	4 Threads	8 Threads
Class A (10 bit UHD)	2.31	8.02	15.94	30.53	52.77
Class B (10 bit HD)	9.56	32.38	63.99	118.10	202.82
SCC HD	13.92	55.93	108.33	190.98	296.25

announced as well (e.g., [67], [69], [71]). A more detailed analysis of optimized decoders is provided in [57], which includes the impact of instruction-level parallelism with SIMD instruction sets, profiling and runtime data for 8K video.

Both VVdeC and O266dec are highly optimized and full-featured VVC decoders that conform to the VVC standard. They both aim at enabling live decoding of 4K video on modern x86 processors by incorporating:

- **Single-instruction multiple data (SIMD) architecture optimization** of sample operations using instruction sets such as SSE42, AVX2 and AVX512, where applicable. It is observed that this optimization is specifically beneficial for the inverse transforms, motion compensation interpolation, and in-loop filters [73], [75].
- **Multi-threading** at the picture level, CTU level, task level and sub-CTU level. More detail about multi-threaded implementations is provided in [73] and [75].

In addition, O266dec used bit depth templatization in its design to allow one unified decoder implementation to easily support various input bit depths without performance penalties [75], [76], in contrast to always using a 16-bit internal memory structure as in the VTM reference software.

Fig. 13 shows the performance of VVdeC v0.1 in fps for various bit rates and different numbers of threads on an Intel Core i9-9980XE processor at 3.0 GHz with 18 cores. For each number of threads, the 72 rate points correspond to decoding the six JVET class A (UHD) CTC test sequences encoded using VTM-10.0 [63] with 12 uneven QPs from 21 to 43 and the CTC RA configuration [36]. It can be seen that the decoding speed scales almost linearly when using the multithreading up to the number of physical CPU cores. Furthermore, live decoding at 60 fps up to 10 Mbits/s is feasible using 8 threads and up to 40 Mbits/s (the limit for Level 5.1 Main tier) using 16 threads.

Table VII presents the decoder performance in fps of the O266 and the VTM-10.0 VVC decoder running on an Intel Core i7-9700 processor clocked at 3.0 GHz with eight cores with turbo boost disabled and hyperthreading unsupported. For each class of the JVET CTC test sequences, the fps numbers for decoding bitstreams generated by the VTM-10.0 encoder [63] using the RA configuration and QPs 22, 27, 32, and 37 from the CTC [36] are averaged. It can be seen that the O266 decoder yields 3.5×, 3.4× and 4× speedup with 1 thread, and 23×, 21× and 21× speedup with 8 threads over the VTM-10.0 decoder for UHD, HD 10-bit camera content, and HD 8-bit screen content decoding, respectively. When running on an Intel Core i9-10940X processor at 3.3 GHz with 14 cores, the O266 decoder can decode the 60 Hz UHD content with bit rates up to the 40 Mbits/s maximum bit rate allowed for Level 5.1 Main tier in real time at native frame rate. Benefiting from the templatization feature and specialized 8-bit SIMD optimization, decoding 8-bit UHD and HD video material is on average 10–15% faster than decoding the same contents using 16-bit internal bit-depth as in the VTM reference software. It is observed that decoding screen-captured content is in general faster than decoding camera-captured content of the same resolution. Real-time decoding of typical screen content in HD resolution can be achieved using two threads, which is considered affordable for many applications.

With e-commerce applications such as Taobao Live in mind, Alibaba presented Ali266 in [72], a software VVC decoder optimized specifically for the mobile platform. Written from scratch following the VVC specification text, the work on Ali266 focused on four aspects of software optimization: multi-threading, ARM assembly, cache efficiency, and memory usage. The initial version of Ali266 in [72] had some limitations, for example, it supported only 8-bit decoding and did not support the ALF (see Section II.B.6) and IBC (see Section II.B.7) coding tools. Two types of content were used to measure the decoding speed: the JVET CTC content coded with fixed QPs, and e-commerce content at 720p resolution coded with three bit rates matching real-world needs. For the 4K JVET CTC content, Ali266 achieves 30 fps decoding speed for coded bit rates up to 7 Mbps; for the 1080p CTC content, Ali266 achieves real-time decoding at native frame rate in most cases using two threads. For the e-commerce content, Ali266 achieves real-time decoding using one or two threads on phones with different hardware capabilities, ranging from the most recent iOS and Android models to older models released more than five years previously. Besides decoding speed, memory footprint and robustness were also considered in the development of Ali266; Ali266 occupies 30 MB of memory when decoding 720p video in real time and is robust against erroneous and/or corrupted bitstreams.

C. Analyzers and Conformance Testing Bitstreams

When implementing encoders or decoders to that conform to a video coding standard, specific toolsets become useful. This includes bitstream analyzers as well as conformance testing bitstreams. Typically, commercial products are available for both, but the following focuses on publicly available tools resulting from the VVC standardization activity within the JVET.

1. Bitstream Analyzers

Bitstream analyzers can be used to visualize inter-picture referencing structures, bit allocation, buffer operations, block partitioning and distribution of coding parameters for CUs of different sizes, among many other aspects that can be derived from the syntax. Commercial packages (e.g, as announced with VVC support in [77] and [78]) typically provide a complete solution that decodes a bitstream and provides a large number of visualization and statistics features together with an intuitive user interface. For academic and research purposes, having an analyzer toolchain based on the reference software that allows tailoring to specific needs can be useful. This includes research based on VVC which goes beyond the current specification and requires customization.

The VTM reference software contains some basic tracing functionality which can be used to write out block statistics to files, which can be loaded into a suitable video player tool for overlay on the reconstructed video sequence, or can be used for statistical analysis at a selectable scope, e.g., the block, picture or sequence level [79]. An example implementation for such a visualization is the open-source YUView player [80]. Block statistics can include block partitioning, intra and inter modes as well as motion information. More information on how to use this functionality can be found in the VTM software manual [63]. Further development of publicly available bitstream analyzer software integrated with the VTM is reported in [81] and [82]. (Some updating may be required to make these packages compatible with more recent versions of the VTM.)

2. Conformance Testing Bitstreams

Conformance testing bitstreams are designed to indicate whether decoders meet the normative requirements specified in a video coding standard such as VVC. To achieve its high coding efficiency and provide versatility by an increasing number of high-level functionalities, the VVC standard allows much more flexibility in encoding decisions and bitstream variations compared to previous video coding standards. For VVC decoder products, it is important that they are developed to fully conform to the standard with the aim to avoid update or replacement costs if incompatibilities are found later. This is of particular importance for hardware implementations where the algorithms implemented in a chip cannot be changed after tape out. Commercial solutions (e.g., [83], [84]) have been established as well in the area of test bitstreams, some of them claiming to cover the full product space of syntax combinations. Besides these commercial solutions widely used in the industry, JVET is developing a companion specification for VVC conformance testing, where a sixth draft has been produced as of the time of preparation of this paper [85].

The VVC conformance testing specification also includes a set of test bitstreams together with procedures to test conformance. These bitstreams are available on an ITU-T server which is identified in the draft conformance testing specification document [85]. Together with the bitstream, each conformance stream package contains the MD5 hash of the correctly decoded video and an output picture log (OPL) file, so implementers can verify the correctness of the decoded video from their decoder implementation. The OPL file contains additional information such as the layer ID, picture order count values, width and height of the cropped output picture as well as MD5 hashes for all color components, to enable verifying the decoder output in cases of cropping, RPR and layered coding.

For the VVC Main 10 profile, Draft 6 of the conformance specification lists bitstreams exercising variations for each coding tool from block partitioning to entropy coding. In addition to the coding tools, also high-level functionalities and control flag combinations are defined. For the other profiles such as the Main 4:4:4 10 profile and the Multilayer Main 10 profile, provided bitstreams should test profile-specific features such as additional chroma formats, tools like ACT and the palette mode, as well as multilayer functionality. The conformance specification is defined in a way that the set of conformance bitstreams can always be extended to encourage additional contributions even after the specification is finalized. The VVdeC decoder presented in Section IV.B contains a functionality to test conformance by launching a test script that automatically downloads a set of supported JVET conformance streams, checks the MD5 sums, the correct picture output order as well as the cropping and reports whether the streams passed all the tests.

V. Summary and Outlook

As described in this paper, the new VVC standard provides both a major improvement in video compression and an unprecedented level of application versatility. The new standard supports advanced features such as layered coding capability and bitstream extraction and merging. In addition to that, emerging applications including HDR/WCG video, screen content coding and 360° immersive video have been addressed already in the first version of the standard. Recent studies including formal subjective testing have confirmed the coding efficiency of the new standard for various application scenarios, often reaching approximately 50% bit rate reduction for equivalent subjective video quality when compared with the prior HEVC standard, especially for HD and UHD video resolutions. Moreover, early VVC implementations have begun to emerge that confirm the practicality of decoder implementation and the ability for real-world encoders to realize the potential of the capabilities of the syntax design. Supporting technology in the form of bitstream analyzers and conformance bitstream test sets has also become available.

At the time of writing this paper, the JVET has started working on amendments to VVC version 1 and VSEI version 1. These include operation range extensions for high bit depth beyond 10 bits and high bit rate coding, additional SEI messages, and potentially additional profiles and/or levels. Some additional metadata signaling for further enabling multiview, 3D, and multilayer applications involving auxiliary information such as depth and/or transparency maps as well as to enable random access with higher coding efficiency has already been adopted into a working draft toward a next version of the VSEI standard.

Looking beyond VVC, the JVET continues to actively develop coding technologies that can enhance compression capability beyond VVC. A recent contribution JVET-U0100 brought to the JVET by Qualcomm in January 2021 [86] showed that additional coding performance gain of 11.5% BD bit rate savings in terms of PSNR can be achieved over VVC by adding more coding tools (some of which had been previously proposed but not adopted into VVC, e.g., due to complexity considerations) and extending some current VVC tools (e.g., by removing or relaxing constraints that were imposed to ease the burden on near-term implementations), although with a substantial increase in complexity. The same meeting also saw another contribution demonstrating coding performance gain on top of VVC for screen content [87]. Correspondingly, the JVET set up an exploration experiment [88] to investigate tools that can enhance compression beyond VVC in a more structured manner.

In the last decade, machine learning has demonstrated its superior capability of solving computer vision and image processing problems. Witnessing such success, researchers and engineers are motivated to investigate machine learning for video compression, and some encouraging progress has been demonstrated in recent years. Generally speaking, these works may be classified into two categories: end-to-end learning-based compression schemes, and learning-based coding tools that are embedded into conventional compression schemes, such as HEVC and VVC. The JVET established its first ad-hoc group on neural-network-based video coding tool investigation in early 2018 [89]. Another ad-hoc group was established in 2020 with an expanded scope and an exploration experiment [90] was set up to carry on the exploration on more efficient video coding with machine learning technologies.

ACKNOWLEDGEMENT

The authors thank the experts of ITU-T VCEG, ISO/IEC MPEG, and their ITU-T/ISO/IEC Joint Video Experts Team for their contributions, not only to the new VVC standard, but also to a large archive of innovative contributions available for further study. The archive of JVET documents can be found online at https://www.jvet-experts.org/.

REFERENCES

[1] ITU-T and ISO/IEC JTC 1, *Versatile Video Coding*, Rec. ITU-T H.266 and ISO/IEC 23090-3 (VVC), July 2020.

[2] ITU-T and ISO/IEC JTC 1, *High Efficiency Video Coding*, Rec. ITU-T H.265 and ISO/IEC 23008-2 (HEVC), Apr. 2013 (and subsequent editions).

[3] 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. (TCSVT)*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] ITU-T and ISO/IEC JTC 1, *Advanced Video Coding for generic audio-visual services*, Rec. ITU-T H.264 and ISO/IEC 14496-10 (AVC), May 2003 (and subsequent editions).

[5] Cisco Systems, "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," Cisco Systems White Paper, Dec. 2018 (online at http://web.archive.org/web/20181213105003/https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf).

[6] Cisco Systems, "Cisco Annual Internet Report (2018–2023)," Cisco Systems White Paper, Mar. 2020 (online at http://web.archive.org/web/20200310054239/https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html).

[7] J. Chen, M. Karczewicz, Y. Huang, K. Choi, J. Ohm, and G. J. Sullivan, "The Joint Exploration Model (JEM) for Video Compression with Capability beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 30, No. 5, pp. 1208–1225, May 2020.

[8] B. Bross, K. Andersson, M. Bläser, V. Drugeon, S.-H. Kim, J. Lainema, J. Li, S. Liu, J.-R. Ohm, G. J. Sullivan, and R. Lu, "General Video Coding Technology in Responses to the Joint Call for Proposals on Video Compression with Capability beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 30, No. 5, pp. 1226–1240, May 2020.

[9] Y. Ye, J. M. Boyce and P. Hanhart, "Omnidirectional 360° Video Coding Technology in Responses to the Joint Call for Proposals on Video Compression With Capability Beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 30, No. 5, pp. 1241–1252, May 2020.

[10] E. François, C. A. Segall, A. M. Tourapis, P. Yin and D. Rusanovskyy, "High Dynamic Range Video Coding Technology in Responses to the Joint Call for Proposals on Video Compression With Capability Beyond HEVC," " *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 30, No. 5, pp. 1253–1266, May 2020

[11] V. Baroncini, J.-R. Ohm, and G. J. Sullivan, "Report of results from the Call for Proposals on Video Compression with Capability beyond HEVC," doc. JVET-J1003 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 10th JVET meeting: April 2018.

[12] ITU-T and ISO/IEC JTC 1, *Versatile supplemental enhancement information messages for coded video bitstreams*, Rec. ITU-T H.274 and ISO/IEC 23002-7 (VSEI), July 2020.

[13] K. Grüneberg, M. M. Hannuksela, J. M. Le Fevre, and Y.-K. Wang (eds.), *Potential improvements on Carriage of VVC and EVC in ISOBMFF*, ISO/IEC JTC 1/SC 29 WG 03 output document N0035, Nov. 2020.

[14] K. Grüneberg, Y. Lim, Y. Syed, and P. Wu (eds.), *Text of ISO/IEC 13818-1:2019 DAM 2 Carriage of VVC in MPEG-2 TS*, ISO/IEC JTC 1 SC 29 WG 11 output document N19436, July 2020.

[15] S. Zhao, S. Wenger, Y. Sanchez, and Y.-K. Wang, *RTP Payload Format for Versatile Video Coding (VVC)*, IETF Internet-Draft draft-ietf-avtcore-rtp-vvc-06, Dec. 2020. [Online]. Available: https://tools.ietf.org/html/draft-ietf-avtcore-rtp-vvc-06

[16] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in International Video Coding Standardization After AVC, with an Overview of Versatile Video Coding (VVC)," *Proc. of the IEEE*, to appear. [Online]. Available: https://ieeexplore.ieee.org/document/9328514

[17] Y.-K. Wang, R. Skupin, M. M. Hannuksela, S. Deshpande, Hendry, V. Drugeon, R. Sjöberg, B. Choi, V. Seregin, Y. Sanchez, J. M. Boyce, W. Wan, and G. J. Sullivan, "The High-Level Syntax of the Versatile Video Coding (VVC) Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[18] Hendry, R. Skupin, and W. Wan, "CE12: Summary report on Tile Set Boundary Handling," doc. JVET-N0032 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 14th JVET meeting: March 2019.

[19] M. M. Hannuksela and Y.-K. Wang, "An Overview of Omnidirectional MediA Format (OMAF)," *Proc. of the IEEE*, to appear. [Online]. Available: https://ieeexplore.ieee.org/document/9380215

[20] 3GPP, *Improved video coding support*, 3GPP TR 26.904, Apr. 2011 (and subsequent releases). [Online]. Available: https://www.3gpp.org/DynaReport/26904.htm

[21] 3GPP, *Study on video enhancements in 3GPP multimedia services*, 3GPP TR 26.948, Dec. 2015 (and subsequent releases). [Online]. Available: https://www.3gpp.org/DynaReport/26948.htm

[22] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 26, no. 1, pp. 20–34, Jan. 2016.

[23] G. Tech, Y. Chen, K. Müller, J. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[24] Y.-W. Huang, J. An, H. Huang, X. Li, S.-T. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, "Block Partitioning Structure in the VVC Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[25] J. Pfaff, A. Filippov, S. Liu, X. Zhao, J. Chen, S. De-Luxán-Hernández, V. Rufitskiy, A. K. Ramasubramonian, and G. Van der Auwera, "Intra Prediction and Mode Coding in VVC," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[26] W.-J. Chien, L. Zhang, M. Winken, X. Li, R. Liao, H. Gao, C.-W. Hsu, H. Liu, and C.-C. Chen, "Motion Vector Coding and Block Merging in Versatile Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[27] H. Yang, H. Chen, S. Esenlik, S. Sethuraman, X. Xiu, E. Alshina, and J. Luo, "Subblock based Motion Derivation and Inter Prediction Refinement in Versatile Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[28] P. Hanhart, Y. He, and Y. Ye, "PERP with horizontal geometry padding of reference pictures (Test 3.3)," doc. JVET-L0231 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 12th JVET meeting: October 2018.

[29] X. Zhao, S.-H. Kim, Y. Zhao, H. E. Elgimez, M. Koo, S. Liu, J. Lainema, M. Karczewicz, "Transform Coding in the VVC Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[30] H. Schwarz, M. Coban, M. Karczewicz, T.-D. Chuang, F. Bossen, A. Alshin, J. Lainema, and C. R. Helmrich, "Quantization and Entropy Coding in the Versatile Video Coding (VVC) Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[31] C. Fu, E. Alshina, A. Alshin, Y. Huang, C. Chen, C. Tsai, C. Hsu, S. Lei, J. Park, and W. Han, "Sample Adaptive Offset in the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 22, no. 12, pp. 1755–1764, Dec. 2012.

[32] M. Karczewicz, N. Hu, J. Taquet, C.-Y. Chen, K. Misra, K. Andersson, P. Yin, T. Lu, E. François, and J. Chen, "VVC In-loop Filters," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[33] H. S. Malvar, G. J. Sullivan, and S. Srinivasan, "Lifting-based Reversible Color Transformations for Image Compression," *SPIE Applications of Digital Image Processing XXXI*, Proc. SPIE, San Diego, California, Vol. 7073, paper 7073-07, sequence number 707307, Aug. 2008.

[34] T. Nguyen, X. Xu, F. Henry, R.-L. Liao, M. G. Sarwer, M. Karczewicz, Y.-H. Chao, J. Xu, S. Liu, and G. J. Sullivan, "Overview of the Screen Content Support in VVC: Applications, Coding Tools, and Performance," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[35] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the Emerging HEVC Screen Content Coding Extension," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 26, no. 1, pp. 50–62, Jan. 2016.

[36] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "JVET common test conditions and software reference configurations for SDR video," doc. JVET-T2010 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[37] ITU-T and ISO/IEC JTC 1, *Working practices using objective metrics for evaluation of video coding efficiency experiments*, ITU-T HSTP-VID-WPOM and ISO/IEC TR 23002-8, July 2020.

[38] V. Baroncini and M. Wien, "VVC verification test report for UHD SDR video content," doc. JVET-T2020 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[39] M. Wien and V. Baroncini, "VVC Verification Test Report for High Definition (HD) and 360° Standard Dynamic Range (SDR) Video Content," doc. JVET-V2020 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 22st JVET meeting: April 2021.

[40] ITU-R, "Image parameter values for high dynamic range television for use in production and international program exchange", Rec. ITU-R BT.2100-2, 2018.

[41] A. Segall, E. François, W. Husak, S. Iwamura, and D. Rusanovskyy, "JVET common test conditions and evaluation procedures for HDR/WCG video," doc. JVET-T2011 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[42] T. Biatek, M. Abdoli, T. Guionnet, A. Nasrallah, and M. Raulet, "Future MPEG standards VVC and EVC: 8K broadcast enabler," International Broadcasting Convention, September 2020.

[43] Y. Yan, M. M. Hannuksela, and H. Li, "Seamless switching of H.265/HEVC-coded DASH Representations with open GOP prediction structure," *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada, September 2015.

[44] R. Skupin, C. Bartnik, A. Wieckowski, Y. Sanchez, B. Bross, C. Hellge, and T. Schierl, "Open GOP Resolution Switching in HTTP Adaptive Streaming with VVC," *35th Picture Coding Symposium (PCS)*, Bristol, US, June-July 2021.

[45] ISO/IEC JTC 1, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*, ISO/IEC 23009-1, 2012 (and subsequent editions).

[46] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu, "Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 26, no. 1, pp. 4–19, Jan. 2016.

[47] X. Xu and S. Liu, "Performance Comparison of Screen Content Coding between HEVC and VVC," doc. JVET-S0264 of ITU-T/ISO/IEC/WG11 Joint Video Experts Team (JVET), 20th JVET meeting: July 2020.

[48] HEVC Screen Content Coding Extension Reference Software version 16.21+SCM8.8. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jct-vc/HM/-/tags/HM-16.21+SCM-8.8

[49] Hendry, Y.-K. Wang, and M. Sychev, "A delay analysis for IRAP and GDR," doc. JVET-N0114 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 14th JVET meeting: March 2019.

[50] S. Deshpande, M. M. Hannuksela, K. Kazui, and T. Shierl, "An improved hypothetical reference decoder for HEVC," Proc. SPIE 8666, Visual Information Processing and Communication IV, Feb. 2013.

[51] Y. Ye and J. Boyce (eds), "Algorithm descriptions of projection format conversion and video quality metrics in 360Lib (Version 11)," doc. JVET-S2004 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 19th JVET meeting: June-July 2019.

[52] P. Hanhart, J. Boyce, K. Choi, and J.-L. Lin (eds), "JVET common test conditions and evaluation procedures for 360° video," doc. JVET-L1012 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 12th JVET meeting: October 2018.

[53] JVET 360Lib reference software. [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_360Lib/

[54] Sun Y, Lu A, Yu L, "Weighted-to-spherically-uniform quality evaluation for omnidirectional video," in IEEE Signal Processing Letters, vol. 24, no. 9, pp. 1408-1412, Sept. 2017.

[55] L. Lee, J.-L. Lin, Y. Wang, Y. He, and L. Zhang, "Blending with padded samples for GCMP," doc. JVET-T0118 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[56] *ATSC Standard: Video – HEVC With Amendments No. 1 and No. 2*, Doc. A/341:2018. [Online]. Available: https://www.atsc.org/wp-content/uploads/2017/05/A341-2018-Video-HEVC-2.pdf.

[57] F. Bossen, K. Sühring, A. Wieckowski, and S. Liu, "VVC Complexity and Implementation Analysis," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, same issue.

[58] G. J. Sullivan, "Deployment status of the VVC standard", doc. JVET-V0021 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 22nd JVET meeting: April 2021.

[59] Fraunhofer HHI VVenC software repository. [Online]. Available: https://github.com/fraunhoferhhi/vvenc

[60] KDDI Research, "KDDI Research Develops the world's first 4K H.266 | VVC real-time encoder" (press release), Sep. 2020. [Online]. Available: https://www.kddi-research.jp/english/newsrelease/2020/090101.html.

[61] Ateme, "ATEME and The Explorers to Launch the First OTT Channel Promoting VVC" (press releasez), Nov. 2020. [Online]. https://www.ateme.com/ateme-and-the-explorers-to-launch-the-first-ott-channel-promoting-vvc/.

[62] Bitmovin, "Bitmovin Enables Innovation with New VVC Codec Feature" (press release), Nov. 2020. [Online]. https://bitmovin.com/press-room/bitmovin-enables-innovation-with-new-vvc-codec-feature.

[63] VVC Reference Software version 10.0. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

[64] HEVC Reference Software version 16.22. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jct-vc/HM

[65] K. Sühring, K. Sharman, "Common Test Conditions for HM Video Coding Experiments," doc. JVET-U1100 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 21st JVET meeting: January 2021.

[66] C. R. Helmrich, B. Bross, J. Pfaff, H. Schwarz, D. Marpe, and T. Wiegand, "Information on and analysis of the VVC encoders in the SDR UHD verification test," doc. JVET-T0103 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[67] J. R. Arumugam, S. Kotecha, S. Ramamurthy, A. Chelawat, J. Jayasanker, A. K. Bedgujar, N. M. Thomas, S. Agrawal, G. R. Vijayakumar, and K. Patankar, "AHG16: Early Implementation of VVC software player and Demonstration on Mobile device," doc. JVET-P0307 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 16th JVET meeting: October 2019.

[68] A. Wieckowski, G. Hege, C. Bartnik, C. Lehmann, C. Stoffers, J. Brandenburg, T. Hinz, B. Bross, H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand, "Development of a VVC live software decoder," doc. JVET-P0973 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 16th JVET meeting: October 2019.

[69] F. Bossen, "AHG16: Performance of a reasonably fast VVC software decoder," doc. JVET-S0224 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 19th JVET meeting: June 2020.

[70] B. Zhu, S. Liu, X. Xu, X. Zhang, C. Gu, L. Wang, and W. Feng, "Performance of a VVC software decoder", doc. JVET-T0095 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[71] S. Gudumasu, T. Poirier, F. Urban, F. Hiron, R. Jullian, and P. de Lagrange, "Multi-threaded VTM decoder description and performance analysis," doc. JVET-T0061 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[72] J. Chen, L. Wang, R.-L. Liao, Y. Ye, "VVC software decoder implementation for mobile devices," doc. JVET-U0088 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 21st JVET meeting: January 2021.

[73] A. Wieckowski, G. Hege, C. Bartnik, C. Lehmann, C. Stoffers, B. Bross, and D. Marpe, "Towards a Live Software Decoder Implementation for the Upcoming Versatile Video Coding (VVC) Codec," *2020 IEEE*

International Conference on Image Processing (ICIP), Abu Dhabi, UAE, October 2020, pp. 3124–3128, doi: 10.1109/ICIP40778.2020.9191199.

[74] Fraunhofer HHI VVdeC software repository. [Online]. Available: https://github.com/fraunhoferhhi/vvdec

[75] B. Zhu, S. Liu, Y. Liu, Y. Luo, J. Ye, H. Xu, Y. Huang, H. Jiao, X. Xu, X. Zhang and C. Gu, "A Real-Time H.266/VVC Software Decoder," 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, July 2021, pp. 1–6, doi: 10.1109/ICME51207.2021.9428470.

[76] Y. Li, S. Liu, Y. Chen, Y. Zheng, S. Chen, B. Zhu, and J. Lou, "An optimized H.266/VVC software decoder on mobile platform," [Online]. Available: https://arxiv.org/abs/2103.03612.

[77] Elecard, "Elecard Video Analyzers Now Support VVC" (press release), April 2020. [Online]. Available: https://www.elecard.com/news/elecard-video-analyzers-now-support-vvc.

[78] ViCueSoft VQAnalyzer website. [Online]. Available: https://vicuesoft.com/products/analyzer (accessed Dec. 2020).

[79] J. Sauer, M. Bläser, J. Schneider, M. Wien, and J.-R. Ohm, "Reference software extension for coding block statistics," doc. JVET-K0149 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 11th JVET meeting: July 2018.

[80] YUView Player software repository. [Online]. Available: https://github.com/IENT/YUView.

[81] M. Kränzler, C. Herglotz, and A. Kaup, "Bit Stream Feature Analyzer (BSFA) for Coding Tool Statistics based on VTM-10.0", doc. JVET-T0067 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 20th JVET meeting: October 2020.

[82] Friedrich-Alexander University Erlangen-Nürnberg (FAU) VTM Analyzer repository. [Online]. Available: https://gitlab.lms.tf.fau.de/LMS/vtm-analyzer.

[83] Allegro DVT, "Allegro DVT Unveils the Industry's First VVC Compliance Test Bitstreams" (press release in Design & Reuse), January 2020. [Online]. Available: https://www.design-reuse.com/news/47414/allegro-dvt-vvc-compliance-test-bitstreams.html.

[84] Allegro DVT, "H.266/VVC Elementary Streams", Allegro DVT website. [Online]. Available: https://www.allegrodvt.com/video-ip-compliance-streams/compliance-streams-validation-verification/h266-vvc-standard/ (accessed Jan. 2021).

[85] J. Boyce, E. Alshina, F. Bossen, K. Kawamura, I. Moccagatta, and W. Wan, "Conformance testing for versatile video coding (Draft 6)," doc. JVET-U2008 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 21st JVET meeting: January 2021.

[86] Y.-J. Chang, C.-C. Chen, J. Chen, J. Dong, H. E. Egilmez, N. Hu, H. Huang, M. Karczewicz, J. Li, B. Ray, K. Reuze, V. Seregin, N. Shlyakhov, L. P. Van, H. Wang, Y. Zhang, and Z. Zhang, "Compression efficiency methods beyond VVC," doc. JVET-U0100 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 21st JVET meeting: January 2021.

[87] K. Naser, F. Le Leannec, T. Poirier, and F. Galpin, "Evaluation of Template Matching Prediction for VVC," doc. JVET-U0048 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 21st JVET meeting: January 2021.

[88] V. Seregin, J. Chen, S. Esenlik, F. Le Leannec, L. Li, M. Winken, J. Ström, X. Xiu, and Kai Zhang, "Exploration Experiment on Enhanced Compression beyond VVC capability," doc. JVET-U2024 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 21st JVET meeting: January 2021.

[89] S. Liu, L. Wang, P. Wu, and H. Yang, "JVET AHG report: Neural Networks in Video Coding (AHG9)," doc. JVET-J0009 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 10th JVET meeting: April 2018.

[90] E. Alshina, S. Liu, W. Chen, Y. Li, R.-L. Liao, Z. Ma, and H. Wang, "Exploration Experiments on Neural Network-based Video Coding", doc. JVET-U2023 of ITU-T/ISO/IEC Joint Video Experts Team (JVET), 21st JVET meeting: January 2021.

Benjamin Bross (S'11–M'17) received the Dipl.-Ing. degree in electrical engineering from RWTH Aachen University, Aachen, Germany, in 2008.

In 2009, he joined the Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute, Berlin, Germany, where he is currently head of the Video Coding Systems group and a part-time lecturer at the HTW University of Applied Sciences Berlin. Since 2010, Benjamin is very actively involved in the ITU-T VCEG | ISO/IEC MPEG video coding standardization processes as a technical contributor, coordinator of core experiments and chief editor of the High Efficiency Video Coding (HEVC) standard (ITU-T H.265 | ISO/IEC 23008-2) and the Versatile Video Coding (VVC) standard (ITU-T H.266 | ISO/IEC 23090-3).

Besides giving talks about recent video coding technologies, Benjamin Bross is an author or co-author of several fundamental HEVC-related publications, and an author of two book chapters on HEVC and inter-picture prediction techniques in HEVC. He received the IEEE Best Paper Award at the 2013 IEEE International Conference on Consumer Electronics – Berlin in 2013, the SMPTE Journal Certificate of Merit in 2014 and an Emmy Award at the 69th Engineering Emmy Awards in 2017 as part of the Joint Collaborative Team on Video Coding for its development of HEVC.

Ye-Kui Wang received his BS degree in industrial automation in 1995 from Beijing Institute of Technology, and his PhD degree in information and telecommunication engineering in 2001 from the Graduate School in Beijing, University of Science and Technology of China.

He is currently a Principal Scientist at Bytedance Inc., San Diego, CA, USA. His earlier working experiences and titles include Chief Scientist of Media Coding and Systems at Huawei Technologies, San Diego, CA, USA, Director of Technical Standards at Qualcomm, Principal Member of Research Staff at Nokia Corporation, etc. His research interests include video coding, storage, transport, and multimedia systems.

Dr. Wang has been an active contributor to various multimedia standards, including video codecs, file formats, RTP payload formats, and multimedia streaming and application systems, developed by various standardization organizations including ITU-T VCEG, ISO/IEC MPEG, JVT, JCT-VC, JCT-3V, 3GPP SA4, IETF, AVS, DVB, ATSC, and DECE. He has been chairing the development of OMAF at MPEG, and has been an editor for several standards, including VVC, VSEI, OMAF, all versions of HEVC, VVC file format, HEVC file format, layered HEVC file format, ITU-T H.271, SVC file format, MVC, RFC 6184, RFC 6190, RFC 7798, 3GPP TR 26.906, and 3GPP TR 26.948. He has co-authored about 1000 standardization contributions, over 50 academic papers, and is a listed inventor for more than 300 US patents.

Yan Ye (M'08–SM'13) received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2002. She is currently the Head of Video Standards and Implementations at Alibaba Cloud Intelligence, Alibaba Group, Sunnyvale, CA, USA, where she oversees multimedia standards development, hardware and software video codec implementations, as well as AI-based video research. Prior to Alibaba, she was with the R&D Labs, InterDigital Communications, Image Technology Research, Dolby Laboratories, and Multimedia R&D and Standards, Qualcomm. She has been involved in the development of various video coding and streaming standards, including H.266/VVC, H.265/HEVC, scalable extension of H.264/MPEG-4 AVC, MPEG DASH, and MPEG OMAF. She is an inventor of more than 100 granted U.S. patents, and has published more than 50 articles in peer-reviewed journals and conferences. Her research interests include advanced video coding, processing and streaming algorithms, real-time and immersive video communications, AR/VR, and deep learning-based video coding, processing, and quality assessment. She is an Editor of the VVC Test Model and the 360Lib algorithm description, and was previously a Co-Editor of the scalable extension and the screen content coding extension of the HEVC standard.

Shan Liu (M’01-SM’11) received the B.Eng. degree in electronic engineering from Tsinghua University, the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, respectively. She is a Distinguished Scientist at Tencent and General Manager of Tencent Media Lab. She was formerly Director of Media Technology Division at MediaTek USA. She was also formerly with MERL and Sony, etc. She has been actively contributing to international standards during the last 10+ years and has numerous technical proposals adopted into various standards, such as VVC, HEVC, OMAF, DASH, MMT and PCC, etc. She served co-Editor of H.265/HEVC SCC and H.266/VVC. At the same time, technologies and products she directly contributed to have served hundred million users. Dr. Liu holds more than 200 granted US and global patents and has published more than 100 peer reviewed technical papers. She was in the committee of Industrial Relationship of IEEE Signal Processing Society (2014-2015). She served the VP of Industrial Relations and Development of Asia-Pacific Signal and Information Processing Association (2016-2017) and was named APSIPA Industrial Distinguished Leader in 2018. She is on the Editorial Board of IEEE Transactions on Circuits and Systems for Video Technology (2018-present) and received the Best AE Award in 2019 and 2020, respectively. She has been serving Vice Chair of IEEE Data Compression Standards Committee since 2019. Her research interests include audio-visual, volumetric, immersive and emerging media compression, intelligence, transport and systems.

Jianle Chen (SM’15) received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively. He was formerly with Samsung Electronics, Qualcomm, and Huawei (USA) focusing on the research of video technologies. Since 2006, he has been actively involved in the development of various video coding standards, including the HEVC standard, its scalable, format range and screen content coding extensions, and most recently, the VVC standard in the Joint Video Experts Team (JVET). He has also been a main developer of the recursive partitioning structure with large block size, which is one of the key features of HEVC standard and its potential successors. He is currently a Director of the Multimedia R&D Group, Qualcomm, Inc., San Diego, CA, USA. His research interests include video coding and transmission, point cloud coding, AR/VR, and neural network compression. He was an Editor of the HEVC specification version 2 (the scalable HEVC (SHVC) text specification) and SHVC Test Model. For VVC, he has been the Lead Editor of the Joint Exploration Test Model (JEM) and VVC Test Model (VTM). He is an Editor of the VVC Text Specification.

Gary J. Sullivan (S’83–M’91–SM’01–F’06) received a B.S. in 1982 and M. Eng. in 1983 from the University of Louisville, and a Ph.D. in 1991 from the University of California, Los Angeles. He is a Video and Image Technology Architect at Microsoft Research. He has been a longstanding chairman/co-chairman of various video and image coding standardization activities in ITU-T VCEG, ISO/IEC MPEG, ISO/IEC JPEG, and in their joint collaborative teams since 1996. He has led the development of the Advanced Video Coding (AVC) standard (ITU-T H.264 | ISO/IEC 14496-10) the High Efficiency Video Coding (HEVC) standard (ITU-T H.265 | ISO/IEC 23008-2), the Versatile Video Coding (VVC) standard (ITU-T H.266 | ISO/IEC 23090-3), and various other projects. 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.

Dr. Sullivan is a Fellow of SPIE as well as IEEE. He has received the IEEE Masaru Ibuka Consumer Electronics Award, the IEEE Consumer Electronics Engineering Excellence Award, two IEEE *Trans. CSVT* Best Paper awards, and the SMPTE Digital Processing Medal. The team efforts that he has led have been recognized by three Emmy Awards.

Jens-Rainer Ohm (M’92) holds the chair position at the Institute of Communication Engineering at RWTH Aachen University, Germany since 2000. Currently, he also serves as dean in the Faculty of Electrical Engineering and Information Technology of RWTH. His research and teaching activities cover the areas of multimedia signal processing, analysis, compression, transmission and content description, including 3D and VR video applications, bio signal processing and communication, application of deep learning approaches in the given fields, as well as fundamental topics of signal processing and digital communication systems. Since 1998, he participates 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 2002–2018, the Joint Video Team (JVT) of MPEG and ITU-T SG 16 VCEG 2005–2009, the Joint Collaborative Team on Video Coding (JCT-VC) 2010-2020, and the Joint Collaborative Team on 3D Video Coding Extension Development 2012-2016. He is currently chairing ISO/IEC JTC1/SC29/WG5 a.k.a. Joint Video Experts Team (JVET) in collaboration with ITU-T SG16/Q6. Prof. Ohm has authored textbooks on multimedia signal processing, analysis and coding, on communication engineering and signal transmission, and numerous papers in these fields. He has served on the editorial boards of several journals and program committees of various conferences in the related fields.