



**HAL**  
open science

## Versatile Video Coding Standard: A Review from Coding Tools to Consumers Deployment

Wassim Hamidouche, Thibaud Biatek, Mohsen Abdoli, Edouard Francois, Fernando Pescador, Milos Radosavljevic, Daniel Menard, Mickael Raulet

### ► To cite this version:

Wassim Hamidouche, Thibaud Biatek, Mohsen Abdoli, Edouard Francois, Fernando Pescador, et al.. Versatile Video Coding Standard: A Review from Coding Tools to Consumers Deployment. IEEE consumer electronics magazine, 2022, 11 (5), pp.10-24. 10.1109/MCE.2022.3144545 . hal-03540142

**HAL Id: hal-03540142**

**<https://hal.science/hal-03540142>**

Submitted on 31 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Versatile Video Coding Standard: A Review from Coding Tools to Consumers Deployment

Wassim Hamidouche *Member, IEEE*, Thibaud Biatek, Mohsen Abdoli, Edouard François, Fernando Pescador *Senior Member, IEEE*, Miloš Radosavljević, Daniel Menard and Mickael Raullet *Senior Member, IEEE*

**Abstract**—The amount of video content and the number of applications based on multimedia information increase each day. The development of new video coding standards is a challenge to increase the compression rate and other important features with a reasonable increase in the computational load. The joint video experts team (JVET) of the ITU-T video coding experts group (VCEG) and the ISO/IEC moving picture experts group (MPEG) have worked together to develop the versatile video coding (VVC) standard, finalized in July 2020 as the international standard 23090-3 (MPEG-I Part 3). This paper overviews some interesting consumer electronic use cases, the compression tools described in the standard, the current available real time implementations and the first industrial trials done with this standard.

**Index Terms**—Encoding/decoding, Application/implementation, Versatile Video Coding, Real Time video codecs.

## I. INTRODUCTION

THE last two decades have witnessed exciting developments in Consumer Electronics Applications. In this framework, the multimedia applications, and more specifically those in charge of video encoding, broadcasting, storage and decoding, play a key role. Video content represents today around 82% of the global Internet traffic according to a study recently conducted by Cisco [1] and video streaming represents 58% of the Internet traffic [2]. All these new trends will increase the part of video traffic, storage requirement and especially its energy footprint. For instance, video streaming contributes today to 1% of the global greenhouse gas emissions, which represent the emissions of a country like Spain [3]. It is expected that in 2025, CO<sub>2</sub> emissions induced by video streaming will reach the global CO<sub>2</sub> emissions of cars [3].

The impressive consumption of multimedia contents in different consumer electronic products (mobile devices, smart TVs, video consoles, immersive and 360° video or augmented and virtual reality devices) requires more efficient video coding algorithms to reduce the bandwidth and the storage capabilities while increasing the video quality. Nowadays, the mass market products demand videos with higher resolutions (greater than 4K) with higher quality (HDR or 10-

bit resolution) and higher frame rates (100/120 frames per second). All of these features must be integrated in devices with low resources and limited batteries. Therefore, a balance between complexity of the algorithms and efficiency in the implementations is a challenge in the development of new consumer electronic devices.

Taking into account this situation, the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and the JCT group within ISO/IEC started working in 2010 [4] on the development of more efficient video coding standard. An example of the success of this collaboration was the high efficiency video coding (HEVC) [5] standard. This latter [5] reduces the bit-rate of the previous video standard advanced video coding (AVC) [6] in a 50% for similar visual quality [7]. Presently, Versatile Video Coding [8] is the most recent video standard and therefore the one that defines the current state-of-the-art. The challenge of this new video standard is to ensure that ubiquitous, embedded, resource-constrained systems are able to process in real-time to the requirements imposed by the increasingly complex and computationally demanding consumer electronics video applications.

Versatile video coding (VVC) [8], [9] has important improvements compared to its predecessors, although it is also based on the conventional hybrid prediction/transform video coding design scheme. VVC has achieved up to 50% [10], [11] bitrate reduction compared to HEVC by implementing a set of new tools and features distributed over the main modules of the traditional hybrid prediction/transform coding scheme. On the other hand, the complexity of both encoder and decoder has been increased [12] as is explained in further sections of this paper.

The VVC standard has been published and, at present, several research institutions and companies are working on efficient implementations that will be included in new consumer electronics devices very soon. This paper reports in further sections some efficient implementations and trials done recently in real scenarios.

The remainder of this paper is organized as follows. Section II describes some use cases and the integration of the standard with other standards included in the video ecosystem. Sections III and IV outline the basic tools of the VVC standard and the complexity of the algorithm, respectively. In Section V, some state-of-the-art implementations targeting consumer electronic devices are reported and, first commercial trials are then presented in Section VI. Finally, Section VII concludes the paper.

Wassim Hamidouche and Daniel Menard are with Univ. Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France (e-mail: [wassim.hamidouche@insa-rennes.fr](mailto:wassim.hamidouche@insa-rennes.fr) and [daniel.menard@insa-rennes.fr](mailto:daniel.menard@insa-rennes.fr)).

Thibaud Biatek, Mohsen Abdoli and Mickael Raullet are with ATEME, Rennes, France.

Edouard François and Miloš Radosavljević are with InterDigital, Cesson-Sévigné, France.

Fernando Pescador is with CITSEM at Universidad Politécnica de Madrid, Madrid, Spain (e-mail: [fernando.pescador@upm.es](mailto:fernando.pescador@upm.es)).

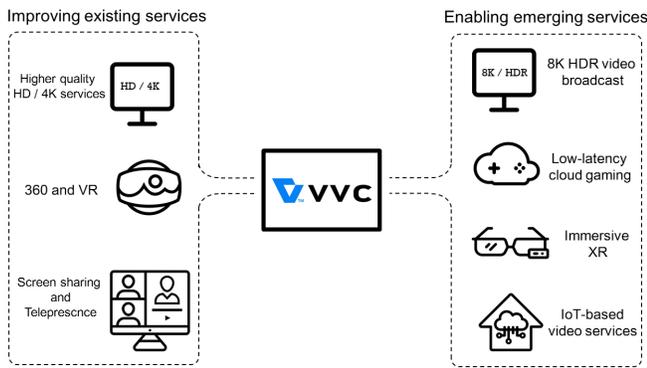


Fig. 1: Potentials of VVC from two different points of view: improving existing services and enabling emerging ones.

## II. USE-CASES AND APPLICATION STANDARDS INTEGRATION

The need for more efficient codecs has arisen from different sectors of the video delivery ecosystem, as the choice of codec plays a critical role in their success during the coming years. This includes different applications on different transport mediums and VVC is consistently being considered as one of the main options. The VVC standard covers a significantly wider range of applications, compared to previous video codecs. This aspect is likely to have a positive impact on the deployment cost and interoperability issue of solutions based on VVC. Thanks to its versatility and high capacity of addressing the upcoming compression challenges, VVC can be used both for improving existing video communication applications and enabling new ones relying on emerging technologies, which are illustrated in Figure 1.

To properly address market needs and be deployed at scale, VVC shall be referenced and adopted by application-oriented standards developing organization (SDO) specifications. Organizations such as digital video broadcast (DVB), 3rd generation partnership project (3GPP) or advanced television systems committee standards (ATSC) are defining receivers' capabilities for broadcast and broadband applications and are thus critical to foster VVC adoption in the ecosystem. Apart from its intrinsic performance (complexity and compression), the successful adoption of a new video codec also relies on its licensing structure.

DVB, which is a set of international open standards for digital television, is currently working to include next generation video coding solutions in the DVB specification. In late 2020, before starting the standardization activities, DVB organized a workshop on new video codecs. During this workshop, five potential codecs were presented and discussed as candidates to address DVB customers' needs: VVC, essential video coding (EVC), alliance for open media (AOM) video (AV1), low complexity enhancement video coding (LC-EVC) and audio video coding standard (AVS3). Since then, the commercial and technical work on inclusion of new video codec in the DVB toolbox has started and is in progress, with a new TS-101-154 specification expected in 2022.

DVB, which is a set of international open standards for

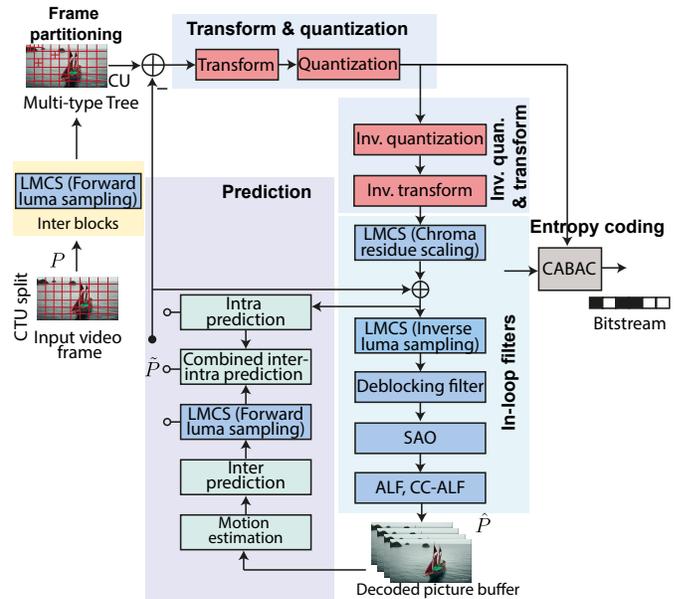


Fig. 2: VVC encoder block diagram.

digital television, is currently working to include next generation video coding solutions in the DVB specification. It is expected that DVB releases a new version of the TS-101-154 specification in early 2022 including support of VVC in the video codec toolbox for application up to 8K-TV [13].

Similarly, 3GPP, which specifies mobile technologies from physical layer to application layer (e.g 4G and 5G), is also investigating the adoption of new codecs for 5G applications. Currently, three video codecs are being characterized, namely VVC, AV1 and EVC. In TR26.955 [14], these codecs are investigated for several scenarios, such as HD-Streaming, 4K-TV, Screen-Content, Messaging, Social-Sharing and Online-Gaming.

To limit the risk of reproducing the same licensing uncertainty as HEVC, VVC has taken a different approach. First, the media coding industry forum (MC-IF) has been created to deal with all non-technical issues related to VVC such as licensing and commercial development. Second, the specification of supplemental enhancement information (SEI) messages has been shifted to a dedicated specification called VSEI (Versatile SEI), published as ITU-T H.274 or ISO/IEC-23002-7. Finally, VVC has defined in its high level syntax (HLS) a structure, named general constrained information (GCI), enabling to switch tools off in a normative way in case licensing of specific intellectual property (IP) would be an issue [8].

Finally, the integration of the VVC with all these standards and initiatives will allow its use in different consumer electronic devices such as mobile phones, set-top-boxes and TVs.

## III. VVC CODING TOOLS

In this section, we give a brief description of the VVC coding tools to understand the improvements regarding its predecessors. Figure 2 illustrates the block diagram of a VVC encoder. This latter relies on a conventional hybrid prediction/transform coding. The VVC encoder is composed

of seven main blocks including 1) luma forward mapping, 2) picture partitioning, 3) prediction, 4) transform/quantization, 5) inverse transform/quantization, 6) in-loop filters, and 7) entropy coding. These seven blocks are briefly described in this section. The luma forward mapping step is normative and relies on the luma mapping with chroma scaling (LMCS) tool, which is described in the in-loop filters section. For more exhaustive description of VVC high level syntax [15] or VVC coding tools and profiles, the reader may refer to the overview paper [9].

### A. Picture partitioning

The first step of the picture partitioning block splits the picture into blocks of equal size, named coding tree unit (CTU). The maximum CTU size is  $128 \times 128$  samples in VVC and is composed of one or three coding tree blocks (CTBs) depending on whether the video signal is monochrome or contains three-color components. The CTUs are then processed in raster scan order from top left to bottom right. In order to adapt the prediction block size to the local activity of the samples, each CTU is then recursively split into smaller rectangular coding units (CUs), according to the multi-type tree (MTT) partitioning scheme. The MTT partitioning [16] is an extension of the quad-tree (QT) partitioning adopted in HEVC. The blocks resulting from the partitioning process are named CUs and may have a size between  $64 \times 64$  and  $4 \times 4$ . In Intra slice, the luma and chroma components can be recursively split according to their own coding trees (separate luma and chroma coding trees). As in HEVC, QT divides a CU in four equal sub-CUs. In addition, VVC allows rectangular shape for CU with its novel splits binary-tree (BT) and ternary-tree (TT). The BT divides a CU in two sub-CUs while the TT divides a CU in three sub-CUs with the ratio 1:2:1. Both BT and TT can split a CU horizontally or vertically.

### B. Intra coding tools

Intra coding principal takes advantage of spatial correlation existing in local image texture. To decorrelate them, a series of coding tools are provided in VVC which are tied with partitioning and a set of Intra prediction modes (IPMs) [17]. For the block partitioning, in addition to the principals explained in previous section, a new tool called dual-tree is introduced which allows separate partitioning trees for luma and chroma channel types. As opposed to the single-tree partitioning used in inter coding, the intra-specific dual-tree tool offers a higher level of freedom for coding decisions of chroma blocks.

The set of IPMs in VVC has extended to 67 modes, compared to 35 in HEVC. This set consists of two modes of DC and planar for modeling homogeneous textures, as well as 65 directional modes for modeling angular textures. The IPMs are coded in VVC through an most probable modes (MPMs) list of six IPMs, while a three MPMs list is used in HEVC. For square shape blocks, these directional modes cover the same range of directional angles, with twice precision as in HEVC. Moreover, thanks to a new tool in VVC called wide-angle Intra prediction (WAIP), the set of 65 angular directional IPMs are adaptively shifted for non-square blocks. This tool assigns

additional directional modes to the longer side of non-square blocks. By doing so, prediction directions with angle greater than  $45^\circ$  relative to the pure horizontal or vertical modes are also possible.

Intra prediction references are improved in VVC to more efficiently manage noisy and less correlated neighboring samples. A reference selection tool, called multiple reference lines (MRL) has been introduced that allows the encoder to choose among three reference lines and explicitly signal the best one. Once the reference line is selected, the reference sample smoothing and the reference sample interpolation are two mechanisms in VVC to denoise reference samples. The choice between these two mechanisms is made implicitly (*i.e* without signaling) and based on block characteristics.

position-dependent prediction combination (PDPC) is yet another new tool in VVC that implicitly combines the prediction signal of a block with its unfiltered and filtered boundaries. Although PDPC is applied differently on DC, planar, horizontal/vertical and other directional modes, its general functionality can be formulated as follows:

$$\tilde{P}(x, y) = \lfloor (\omega_L R_L + \omega_T R_T + (64 - \omega_L - \omega_T) \tilde{P}(x, y) + 32) / 2^6 \rfloor, \quad (1)$$

where,  $\tilde{P}(x, y)$  is the predicted sample at position of coordinates  $(x, y)$ ,  $R_T$  and  $R_L$  are, respectively, reference samples aligned from the top and left to this position. Weights of these reference samples, expressed as  $\omega_T$  and  $\omega_L$ , respectively, are determined based on the position  $(x, y)$  as well as the selected intra mode. Finally, in this equation, the operator  $\lfloor x \rfloor$  is the lower integer value of a real  $x$  used with a division by  $2^6$  to normalize the modified prediction value into its range.

Intra sub-partitions (ISP) tool allows splitting an intra block into two or four sub-blocks, each having its separate residual block, while sharing one single intra mode. The initial motivation behind this tool is to allow short-distance intra prediction of block with non-stationary texture, by sequential processing of sub-partitions. On thin non-square blocks, this scheme can result in sub-partitions coded with 1-D residual blocks, providing the closest possible reference lines for intra prediction.

The matrix-based Intra prediction (MIP) in VVC is a new tool designed by an AI-based data-driven method. MIP modes replace the functionality of conventional IPMs, by matrix multiplication of reference lines, instead of their directional projection. The process of predicting a block with MIP consists in three main steps: 1) averaging with sub-sampling, 2) matrix-vector multiplication and 3) linear interpolation. The data-driven aspect of MIP is expressed in the second step, where a set of matrices are pre-trained and hard-coded in the VVC specification. This set is designed to provide distinct matrices for different combinations of block size and internal MIP mode.

Cross-component linear model (CCLM) is a new tool in VVC for exploiting local correlations between luma and chroma channels. This tool is based on a similar concept in the HEVC range extension standard, where the inter-channel correlation is modeled in the residual domain. In VVC, this

modeling is carried out in the reconstructed pixel domain, in the form of:

$$\tilde{P}_c(x, y) = \alpha \hat{P}(x, y) + \beta, \quad (2)$$

where  $\tilde{P}_c(x, y)$  is the predicted chroma value at position  $(x, y)$  based on the co-located reconstructed luma value  $\hat{P}(x, y)$ . The model parameters  $\alpha$  and  $\beta$  are explicitly derived based on the relation between neighboring luma and chroma samples.

### C. Inter coding tools

Inter coding relies on inter-prediction of motion and texture data, from previously reconstructed pictures stored in the decoded picture buffer (DPB). A simplified block diagram of the VVC inter-prediction process is provided in Figure 3. The process first involves motion prediction, based on a list of motion data candidates. The motion prediction can be corrected by residual motion information signaled in the bitstream. The reconstructed motion vectors are then used to perform one or two motion compensations, whether the coding block is uni- or bi-predicted. When bi-prediction is performed, a blending process is then applied to mix the two motion compensated blocks. Finally, a prediction enhancement step is performed as a post-prediction filtering. The resulting predicted signal can be further corrected by a residual block signaled in the bitstream.

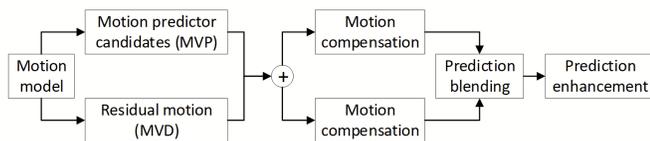


Fig. 3: Block diagram of the VVC inter-prediction process.

Two different motion models are supported in VVC, translational model and affine model, controlled at CU-level. The affine model can rely on 4 or 6 degrees of freedom. When a CU is coded with affine motion, it is split into  $4 \times 4$  sub-blocks, whose motion vectors (MVs) are derived from the affine parameters of the CU. These parameters are deduced from 2 or 3 control point motion vectors positioned in the top-left, top-right and possibly bottom left corners of the CU, depending on the number of affine parameters.

In VVC, the motion vectors accuracy is of 1/16th sample for luma component, and 1/32nd sample for chroma component (for 4:2:0 chroma format). An inter CU can be coded according to three main modes: 1) skip mode, that only specifies a motion data predictor (motion vector and reference picture index) among a set of motion data candidates, and does not add any motion residual or texture residual to the predicted motion and texture block; 2) merge mode, that additionally signals a texture residual; 3) advanced motion vector prediction (AMVP) mode, that is a superset of merge mode where the whole motion information is signaled. Each of these modes uses a list of motion information candidates. In addition, the lists construction differs for the translational and affine modes. As in HEVC, the lists are built from neighboring spatial motion information of the current CU (spatial

MVs), motion information from reference pictures (TMVP). In addition, VVC specifies two new types of motion candidates, history-based motion vector prediction (HMVP), and pairwise MV. HMVP stores in a FIFO buffer up to 5 MVs, which allow accessing MVs not present in the spatial or temporal neighborhood of the CU. Pairwise MV is an average of the two first candidates in the list. VVC also supports a new mode, sub-block temporal motion prediction (SBTMVP) that performs temporal motion field prediction of a CU on a  $4 \times 4$  block granularity. Residual motion signaling applies in AMVP mode and consists in coding a corrective motion information added to the motion information prediction. In VVC, new residual motion coding tools are supported compared to HEVC. Adaptive motion vector resolution (AMVR) allows adapting at CU-level the coding precision of the motion vector difference (MVD). Symmetrical motion vector difference (SMVD) applies in the case of bi-directional prediction and consists in coding a single corrective MV applying symmetrically to MV of each direction. In merge mode case, merge with motion vector differential (MMVD) mode allows slightly correcting the predicted motion information. Finally, in case of bi-directional prediction, it is also possible to refine the MVs of a CU, with a  $4 \times 4$  granularity using the decoder-side MV refinement (DMVR) mode, that performs a motion search of limited range per  $4 \times 4$  sub-block.

The motion compensation is based on separable linear 8-tap filters / 16 phases for luma, and 4-tap filters / 32 phases for chroma. In VVC, it is also possible to adaptively change the coded picture resolution, using the reference picture resampling (RPR) tool. Four different sets of interpolation filters are used depending on the motion model, block size, and scaling ratio between the current picture and the reference picture (maximum downsampling ratio 2 and upsampling ratio 8). RPR offers new capabilities for bit-rate control to adapt to network bandwidth variations and is also applicable to scalability and adaptive resolution coding. For 360° video content, using equi-rectangular projection format, horizontal wrap around motion compensation allows limiting seam artifacts by performing wrapping instead of padding of the samples located at the reference picture vertical borders. Regarding the motion prediction blending, several new modes compared to HEVC are supported in VVC. Bi-prediction with CU-level weights (BCW) is an enhanced version of the bi-prediction blending of HEVC, performing a weighted averaging of the two prediction signals  $\hat{P}_0$  and  $\hat{P}_1$  according to the following formula:

$$\tilde{P} = \lfloor [(8-w)\hat{P}_0 + w\hat{P}_1 + 4]/2^3 \rfloor, \quad (3)$$

where the weight  $w$  is selected among a pre-defined set of weights. VVC also supports geometric partitioning mode (GPM), that splits a CU into non-rectangular sub-partitions, each partition embedding a translational MV.

Combined inter-intra prediction (CIIP) generates a mixed version of temporal prediction  $\hat{P}_{Inter}$  and spatial prediction  $\hat{P}_{Intra}$  samples according to the following formula:

$$\tilde{P} = (w\hat{P}_{Inter} + (1-w)\hat{P}_{Intra})/4, \quad (4)$$

where the mixing weight  $w$  depends on the top and left CUs coding mode (intra or inter). The final prediction enhancement

step is a new coding step in VVC that is not present in HEVC. This step consists in slightly adjusting the prediction signal, according to two possible modes both relying on the optical flow principles. Bi-directional optical flow (BDOF) applies in case of bi-directional prediction and consists in correcting the samples value based on the MV and spatio-temporal gradients derived from the two reference pictures. Prediction refinement with optical flow (PROF) applies to CU coded with affine model and performs a per-sample correction that takes into account the difference between the true per-sample motion field derived from the affine model parameters, and the approximated  $4 \times 4$  motion field used in the actual affine motion compensation step.

#### D. Transform and quantization

The transform module in VVC is composed of two blocks namely multiple transform selection (MTS) and low frequency non-separable transform (LFNST) that perform separable and non-separable transforms, respectively [18].

MTS: The MTS block in VVC involves three transform types including the discrete cosine transform (DCT)-II, DCT-VIII and discrete sine transform (DST)-VII. The kernels of DCT-II  $C_2$ , DST-VII  $S_7$  and DCT-VIII  $C_8$  are derived from (5), (6) and (7), respectively.

$$C_{2i,j}^N = \gamma_i \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(i-1)(2j-1)}{2N}\right), \quad (5)$$

$$\text{with } \gamma_i = \begin{cases} \sqrt{\frac{1}{2}} & i = 1, \\ 1 & i \in \{2, \dots, N\}. \end{cases} \quad (6)$$

$$S_{7i,j}^N = \sqrt{\frac{4}{2N+1}} \sin\left(\frac{\pi(2i-1)j}{2N+1}\right).$$

$$C_{8i,j}^N = \sqrt{\frac{4}{2N+1}} \cos\left(\frac{\pi(2i-1)(2j-1)}{2(2N+1)}\right), \quad (7)$$

with  $(i, j) \in \{1, 2, \dots, N\}^2$  and  $N$  is the transform size.

The MTS concept selects, for Luma blocks of size lower than 64, a set of transforms that minimizes the rate distortion cost among five transform sets and the skip configuration. However, only DCT-II is considered for chroma components and Luma blocks of size 64. The *sps\_mts\_enabled\_flag* flag defined at the sequence parameter set (SPS) enables to activate the MTS at the encoder side. Two other flags are defined at the SPS level to signal whether implicit or explicit MTS signalling is used for Intra and Inter coded blocks, respectively. For the explicit signalling, used by default in the reference software, a syntax element signals the selected horizontal and vertical transforms. To reduce the computational cost of large block-size transforms, the effective height  $M'$  and width  $N'$  of the coding block (CB) are reduced depending of the CB size and transform type

$$N' = \begin{cases} \min(N, 16) & trTypeHor > 0, \\ \min(N, 32) & \text{otherwise.} \end{cases} \quad (8)$$

$$M' = \begin{cases} \min(M, 16) & trTypeVer > 0, \\ \min(M, 32) & \text{otherwise.} \end{cases} \quad (9)$$

In (8) and (9),  $M'$  and  $N'$  are the effective width and height sizes, *trTypeHor* and *trTypeVer* are respectively the types of vertical and horizontal transforms (0: DCT-II, 1: DCT-VIII and 2: DST-VII), and the  $\min(a, b)$  function returns the minimum between  $a$  and  $b$ . The sample value beyond the limits of the effective  $N$  and  $M$  are considered to be zero, thus reducing the computational cost of the 64-size DCT-II and 32-size DCT-VIII/DST-VII transforms. This concept is called zeroing in the VVC specification.

LFNST: The LFNST has been adopted in VVC since the VVC test model (VTM) version 5. The LFNST relies on matrix multiplication applied between the forward primary transform and the quantisation at the encoder side:

$$\vec{Z} = T \cdot \vec{Y}, \quad (10)$$

where the vector  $\vec{Y}$  includes the coefficients of the residual block rearranged in a vector and the matrix  $T$  contains the coefficients transform kernel. The inverse LFNST is expressed in (11).

$$\vec{Y} = T^T \cdot \vec{Z}. \quad (11)$$

Four sets of two LFNST kernels of sizes  $16 \times 16$  and  $64 \times 64$  are applied on 16 coefficients of small blocks (min (width, height)  $< 8$ ) and 64 coefficients of larger blocks (min (width, height)  $> 4$ ), respectively. The VVC specification defines four different transform sets selected depending on the Intra prediction mode and each set defines two transform kernels. The used kernel within a set is signaled in the bitstream. The transform index within a set is coded with a Truncated Rice code with rice parameter  $p = 0$  and *cMax* = 2 (TRp) and only the first bin is context coded. The LFNST is applied on Intra CU for both Intra and Inter slices and concerns Luma and Chroma components. Finally, LFNST is enabled only when DCT-II is used as a primary transform.

#### E. In-loop filters

The picture partitioning and the quantization steps used in VVC may cause coding artifacts such as block discontinuities, ringing artifacts, mosquito noise, or texture and edge smoothing. Four in-loop filters are thus defined in VVC to alleviate these artifacts and enhance the overall coding efficiency [19]. The VVC in-loop filters are deblocking filter (DBF), sample adaptive offset (SAO), adaptive loop filter (ALF) and cross-component adaptive loop filtering (CC-ALF). In addition, the LMCS is a novel tool introduced in VVC that performs both luma mapping to the luma prediction signal in inter mode and chroma scaling to residuals after inverse transform and inverse quantisation. The DBF is applied on block boundaries to reduce the blocking artifacts. The SAO filter is then applied on the deblocked samples. The SAO filter first classifies the reconstructed samples into different categories. Then, for each category, an offset value retrieved by the entropy decoder is added to each sample of the category. The SAO is particularly efficient to alleviate ringing artifacts and correct the local average intensity changes. The last in-loop filters, ALF and CC-ALF, perform block-based linear filtering and adaptive clipping. The ALF performs adaptive filtering to minimize the

mean squared error (MSE) between original and reconstructed samples relying on Wiener filtering. A  $7 \times 7$  diamond filter shape is applied on luma components. The filter coefficients are derived from a  $4 \times 4$  block classification based on local sample gradients. According to the computed class, filter coefficients are selected from a set of filters which are whether fixed or signaled in the bitstream at the level of the adaptation parameter set (APS). Geometric transformations such as 90-degree rotation, diagonal or vertical flip may also be applied to the filter coefficients according to the block class. For chroma samples ALF, a  $5 \times 5$  diamond filter shape is first applied. The chroma filter coefficients can only be signaled in the APS. The CC-ALF uses co-located Luma samples to generate a correction for chroma samples. The CC-ALF is applied only on chroma samples and it is performed in parallel with the ALF.

#### IV. COMPLEXITY AND CODING PERFORMANCE

In order to assess the benefits of the VVC coding tools described in previous sections, a “tool off” test has been performed, consisting, for each individual tool, in evaluating the coding cost variations between an encoding setting with all tools enabled, compared to an encoding setting with all tools enabled except the tested tool. A set of 42 UHD sequences, not included in the common test sequences used during the VVC development process, has been used. This set includes sequences of various texture complexity, motion amplitude and local variations, and frame rates (from 30 to 60 frames per second). The evaluation has been performed in random access (RA) configuration, with group of pictures (GOP) size of 32, and one intra frame inserted every 1 second, using the VVC reference software (VTM11.0). The evaluation focuses on the main new tools supported by VVC and not present in HEVC (except the partitioning and the entropy coding parts which are not considered in this evaluation). For motion coding, only tools incrementally added to the HEVC design are considered. SAO, which has the same design in HEVC and VVC, has also been evaluated. The Bjøntegaard delta bit rate (BD-rate) metric [20] is used as estimation of the bitrate variations, using as objective quality metrics PSNR, VMAF [21] and MS-SSIM [22]. For PSNR metric, the BD-rate variations are computed for the PSNR of each color component (Y, U, V), then a weighted BD-rate value is computed from the BD-rate of each component (using weight 6/8 for luma, and 1/8 for each chroma component). VMAF and MS-SSIM are only computed on the luma (Y) component. A positive value of BD-rate variation indicates a bit rate increase when disabling the tool. Encoding and decoding runtimes variations are also reported. These latest figures are provided as indicative data, but must be considered with a lot of care, since the VTM decoder implementation is far from a real product implementation and from an efficient implementation. Detailed results are reported in Table I. Tools are grouped per category (<sup>†</sup> Intra tools, <sup>⊕</sup> Transform tools, <sup>⊗</sup> In-loop filter tools, <sup>\*</sup> MV coding, <sup>⊙</sup> Subblock Motion Compensation, <sup>•</sup> Prediction blending, <sup>◊</sup> Prediction enhancement). When all these tools are disabled, BD-rate variations are 43.18% for

TABLE I: Per-tool VVC performance. The BD-rate performance are reported by disabled tool in terms of PSNR, VMAF and SSIM. The relative encoding time (EncT) and decoding time (DecT) variations are reported with respect to the anchor.

Tools	BD-rate			Complexity	
	PSNRYUV	VMAF	MS-SSIM	EncT	DecT
MIP <sup>†</sup>	0.31%	0.57%	0.46%	96.0%	100.0%
MRL <sup>†</sup>	0.12%	0.16%	0.13%	100.0%	100.0%
Lmchroma <sup>†</sup>	5.07%	1.47%	1.31%	99.0%	100.0%
ISP <sup>†</sup>	0.36%	0.32%	0.33%	96.0%	100.0%
MTS <sup>⊕</sup>	0.81%	1.07%	1.03%	93.7%	99.2%
SBT <sup>⊕</sup>	0.32%	0.52%	0.27%	95.0%	100.0%
LFNST <sup>⊕</sup>	0.57%	0.97%	0.73%	96.1%	99.8%
JCCR <sup>⊕</sup>	0.27%	0.38%	0.32%	99.0%	100.0%
DQ <sup>⊕</sup>	2.15%	1.74%	2.20%	99.1%	100.0%
DBF <sup>⊗</sup>	0.44%	0.71%	0.20%	100.8%	85.0%
SAO <sup>⊗</sup>	0.08%	0.15%	0.02%	99.9%	98.1%
ALF+CCALF <sup>⊗</sup>	7.64%	6.09%	0.66%	95.1%	90.0%
LMCS <sup>⊗</sup>	0.15%	-1.01%	0.85%	97.6%	100.0%
TMVP <sup>*</sup>	1.29%	1.56%	1.43%	99.4%	100.0%
AMVR <sup>*</sup>	1.29%	0.87%	1.16%	82.6%	100.0%
MMVD <sup>*</sup>	0.28%	0.19%	0.23%	90.3%	100.0%
SMVD <sup>*</sup>	0.22%	0.19%	0.17%	96.2%	100.0%
Affine <sup>⊙</sup>	2.31%	2.32%	2.61%	80.8%	97.0%
SBTMVP <sup>⊙</sup>	0.34%	0.53%	0.41%	101.1%	99.0%
BCW <sup>•</sup>	0.25%	0.17%	0.15%	93.7%	98.0%
GPM <sup>•</sup>	0.66%	0.73%	0.63%	95.1%	100.0%
CIIP <sup>•</sup>	0.08%	0.17%	0.16%	96.7%	100.0%
DMVR <sup>◊</sup>	0.86%	0.93%	1.00%	99.9%	95.8%
BDOF <sup>◊</sup>	0.64%	1.29%	0.74%	96.7%	98.0%
PROF <sup>◊</sup>	0.33%	0.41%	0.31%	98.3%	99.0%

<sup>†</sup> Intra tools, <sup>⊕</sup> transform tools, <sup>⊗</sup> in-loop filter tools, <sup>\*</sup> MV coding, <sup>⊙</sup> Subblock Motion Comp, <sup>•</sup> motion compensation, <sup>◊</sup> prediction enhancement.

PSNRYUV, 30.22% for VMAF and 27.67% for MS-SSIM, with encoding and decoding runtime variations of 19% and 49% compared to the setting with all tools enabled (anchor). This demonstrates the substantial coding performance brought by the new VVC tools. It is also observed that, for most of the evaluated tools, the reported results are very consistent with the per-tool evaluation reported by JVET document [23], which demonstrates that tools performance was not optimized for the JVET test sequences. LMCS presents lower gains in this paper than in [23], which tends to show that this tool requires very accurate and content-dependent tuning. Gains from SAO are small in terms of objective metrics, but this tool was considered by JVET as particular relevant for subjective quality. Also, its implementation is extremely low-cost, which makes this tool very useful when ALF is disabled.

In Figure 4, the tools listed in Table I are grouped per category, and relative contribution of each category to the total PSNRYUV BD-rate coding gain brought by all the tools is depicted. Similarly, the relative runtime ratios are depicted in Figure 4b and Figure 4c for encoder and decoder, respectively. All tools’ categories turn out to have an important contribution to the overall coding gain. Most computing demanding

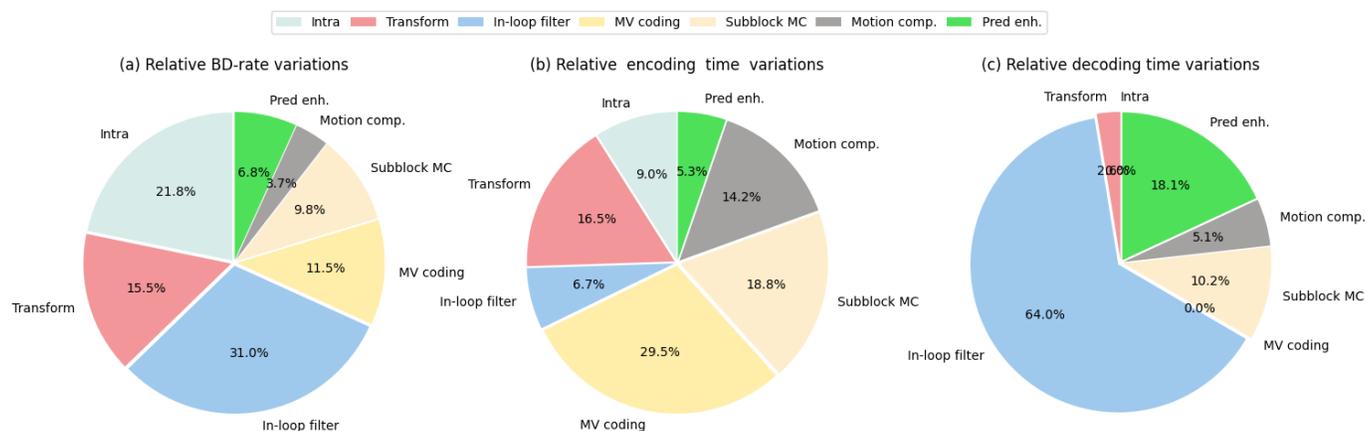


Fig. 4: Relative contributions of the main VVC tools' categories in RA coding configuration.

parts in decoder are loop filtering and inter coding (motion compensation, sub-block MC and prediction enhancement). Intra and transform categories have negligible impacts on the VTM decoding time. At VTM encoding side, the most demanding part is the inter part, which represents 2/3rd of the encoding time increase. Another substantial part of the encoding runtime is consumed by the partitioning, which is not assessed and reported in this paper. The decoding runtime impact of these tools remains limited in this particular RA coding configuration [24].

## V. REAL-TIME IMPLEMENTATIONS

As of now, VVC benefits from both industrial and open-source implementations, contributing to the emergence of an end-to-end value chain. For example, manufacturers, universities and research-institutes have announced the availability of fast VVC implementations [25]. Among existing solutions, the INSA Rennes real-time OpenVVC decoder, the Fraunhofer Heinrich Hertz Institute VVdeC decoder [26], VVenC encoder [27] and the ATEME TitanLive encoding platform have been developed during the last months.

### A. Real-Time VVC Decoding with OpenVVC

OpenVVC is the world-first real time software VVC decoder developed from scratch in C programming language. The OpenVVC project is intended to provide consumers with an open source library enabling UHD real time VVC decoding capability<sup>1</sup>. The VVC main profile tools are supported by the OpenVVC decoder and the most complex operations are optimized in single instruction multiple data (SIMD) for both Intel x86 and ARM Neon platforms. The decoder is parallel-friendly supporting high level parallelism such as parallel decoding of slices, tiles, wavefront and frames (frame-based parallelism). These latter leverage multi-core platforms to further speedup the decoding process and reduce the decoded frame latency. Finally, the decoder is compatible with the well known video players such as FFplay, GPAC and VLC.

<sup>1</sup><https://github.com/OpenVVC/OpenVVC>

### B. Real-Time VVC Decoding with VVdeC

It is important to consider that all the improvements added to VVC in order to achieve better coding performance come with a cost in terms of computational complexity increasing, which has been determined to be around 10x in the encoder and 2x in the decoder compared to HEVC. The Fraunhofer Heinrich Hertz Institute has been working on Versatile Video deCoder [28] (VVdeC) since October 2020, aiming to provide a publicly available optimized VVC software decoder<sup>2</sup>.

This implementation supports multicore architectures and it is optimized for x86 platforms. This decoder takes advantage of functional (multi-threading) and data parallelization (SIMD instructions) to achieve the maximum performance. Compared to VTM decoder, this optimized implementation has reached 50% to 90% improvement in terms of decoding time reduction over a x86 platform [29].

### C. Real-Time VVC Encoding with VVenC

VVenC [27] is an open source fast implementation of a VVC encoder<sup>3</sup> developed by the Fraunhofer Heinrich Hertz Institute. VVenC is developed in C++ programming language and includes low level optimizations through SIMD instructions targeting Intel x86 platform. The encoder is parallel-friendly enabling to process a set of pictures in parallel on multi-core platforms. Five presets are defined by the encoder offering a wide range trade-off between coding efficiency (quality) and speed (complexity). Perceptual optimization is also integrated to improve subjective video quality, based on the XPSNR visual model [30]. Finally, frame-level single-pass and two-pass rate control with variable bit-rate (VBR) encoding are supported by the encoder.

### D. Real-Time VVC Encoding and Packaging with TitanLive

The ATEME TitanLive solution provides software-based implementation of a wide variety of standards for audio/video

<sup>2</sup><https://github.com/fraunhoferhhi/vvdec>

<sup>3</sup><https://github.com/fraunhoferhhi/vvenc>

coding, packaging and transport. This solution is currently used worldwide for broadcast and over-the-top (OTT) head-end deployments. In order to support VVC, a number of components were upgraded.

As further described in [31], VVC and HEVC present some structural similarities making an upgrade from HEVC to VVC feasible in a cost effective manner. In order to do so, the VVC syntax has been implemented with support for the tools already implemented in HEVC, disabling the other ones in the APS. Then, the HEVC tools have been upgraded to comply with VVC specification and some tools offering a good complexity-vs-gains trade-offs were implemented. Relying on the same core coding engine enabled us to leverage the existing optimized function (assembly, intrinsic) to achieve VVC real-time encoding with interesting gains over HEVC, from 10% to 15% depending on the video content.

The packager has been upgraded as well to support VVC encapsulation into MPEG2-TS and ISOBMFF. Since the final draft international standard (FDIS) has not been issued yet for MPEG2-TS and ISOBMFF binds, some draft amendment versions (DAM) were implemented [32], [33], assuming that these versions are close to the FDIS ones.

## VI. FIRST COMMERCIAL TRIALS

Using the previously described tools some world-first VVC in field trials [34], [35] have been implemented as described in the following subsections.

### A. World-First OTA Broadcasting with VVC

1) *Overview:* The trial depicted in Figure 5a took place in June 2020 and is the result of a collaboration between the following entities:

- ATEME provided the encoding and packaging units.
- SES provided the satellite transponder used for the experiment as well as the gateways needed at transmission and reception sides.
- VideoLabs provided the media player (VLC) including demuxing and playback.
- IETR provided the VVC real-time decoding library used by VLC player.

As illustrated, the UHD source provided by The Explorers is encoded with VVC, and encapsulated in MPEG-TS using ATEME video processing platform. The provided video bitstreams were received by SES and sent to the modulator gateways feeding the ASTRA 2E transponder (Europe coverage). The signal is demodulated and forwarded on IP to a VLC player that displays the video thanks to the real-time OpenVVC decoder developed by IETR.

2) *VVC Encoding and Encapsulation:* The ATEME encoding engine used in this experiment followed the VVC draft specification [36] and produced a bitstream decodable by the VTM software (tag version 6.1). The 2160p-10b-SDR video input was encoded offline using RA GOP structure with a 1sec RAP period and a 20Mbps constant bitrate (CBR) and deltaQP is enabled.

The produced elementary stream was encapsulated in MPEG-TS following draft specification incorporating VVC

amendments [37]. Hence, a stream embedding stream\_type 0x32 with video descriptor 57 was generated and ready to be delivered over existing broadcast infrastructure.

3) *Satellite Transmission:* The MPEG-TS provided by ATEME was rate-adapted and played out on Transponder 2.014 on SES prime UK position 28.2 East. The transponder used for this transmission is a transponder that has been used previously for SES 8K test transmissions and therefore the parameters and link budget were slightly unusual for transmissions at this position (Freq: 11.973, Pol: Vertical, SR 31 MS/s, DVB-S2, 8PSK 9/10). The uplink was done in Betzdorf, Luxembourg which is the location of SES headquarters.

Reception was done using a DVB-S2 to IP Gateway (in our instance the Kathrein EXIP 4124 which is a SAT>IP server). The Gateway was statically tuned to the corresponding transponder and setup to forward the relevant PIDs of the received TS encapsulated in RTP multicast (RFC 2250) to the local network. On that network a powerful PC was used to decode the 4K-UHD stream in real-time and displayed on several TVs. The PC framerate was set to the framerate of the source content.

4) *Player and Decoder:* The OpenVVC described in Section V-A was used as decoding library for this trial. The VLC player wraps both OpenVVC and libavformat demuxer for this trial. Libavformat demuxer is modified to handle MPEG-TS streams carrying VVC and the extracted NALUs are processed by OpenVVC. The synchronization and picture presentation is managed by VLC player.

### B. World-First OTT Delivery with VVC

1) *Overview:* The trial depicted in Figure 5b took place in June 2020 and is the result of a collaboration between the following entities:

- ATEME provided the encoding unit.
- Telecom Paris provided the DASH packager (MP4Box) and the player (MP4Client) from GPAC.
- IETR provided the VVC real-time decoding library used by MP4Client.

As illustrated, the UHD source provided by The Explorers is encoded with VVC, and formatted into ISOBMFF mp4 files using ATEME video processing platform. The mp4 files are encapsulated into DASH with multiple representations using Telecom Paris MP4Box software. The generated DASH is pushed on an origin server publicly accessible on the internet. The MP4Client demultiplexes and plays the content thanks to the real-time OpenVVC decoder developed by IETR.

2) *VVC Encoding:* The ATEME encoding engine used in this experiment followed the VVC draft specification [36] and produced a bitstream decodable by the VTM software (tag version 6.1). The video input was encoded offline using RA closed-GOP structure with a 1sec RAP period, producing the following bitrate ladder:

- 540p @ 1.6 Mbps
- 720p @ 3.4 Mbps
- 1080p @ 5.8 Mbps
- 2160p @ 16.8 Mbps

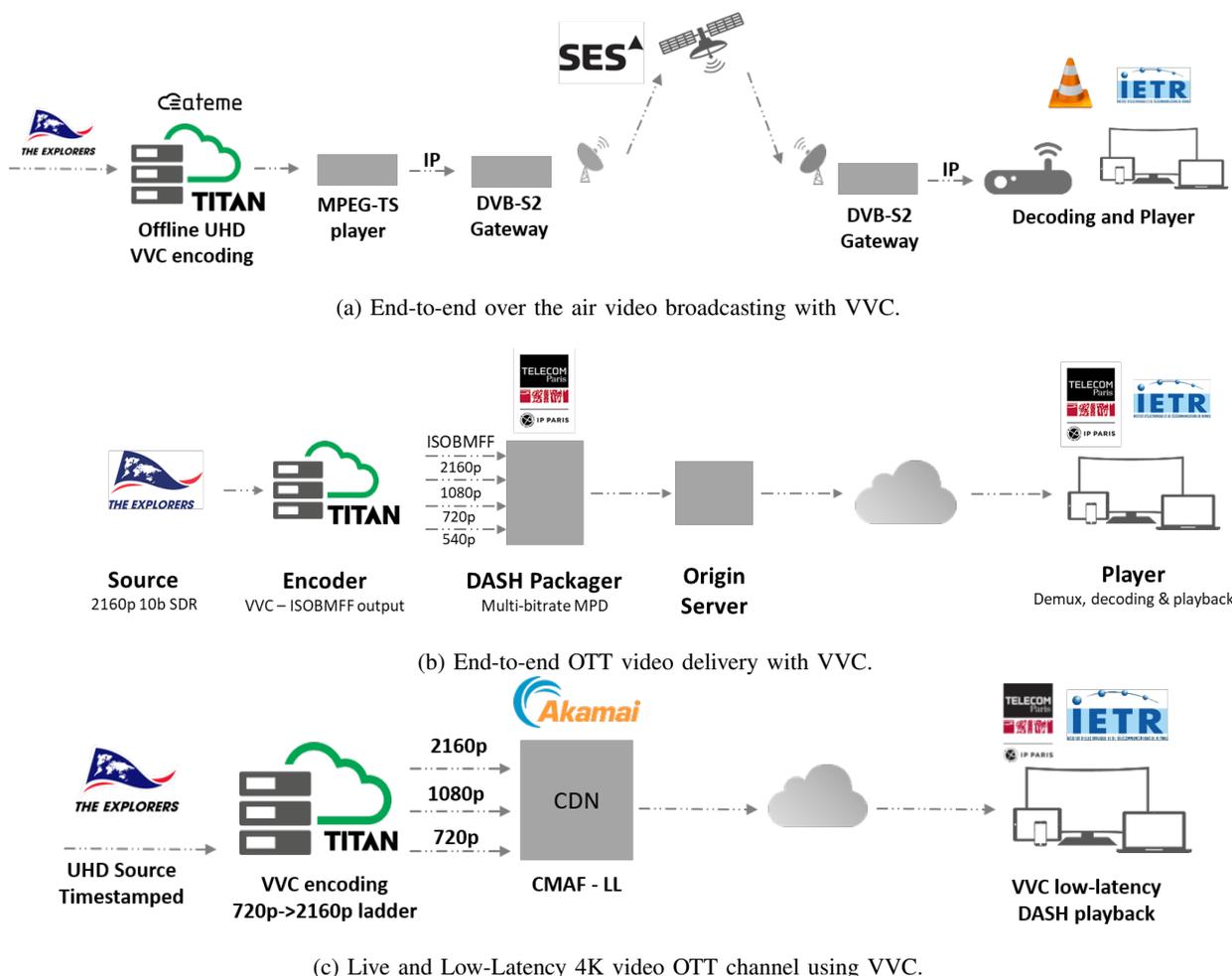


Fig. 5: Illustration of different end-to-end demonstrations with the VVC standard on different transmission mediums.

3) *Client and Player*: Support for VVC transport was added to GPAC [38] as follows:

- ISOBMFF demultiplexing for 'vvc1' and 'vvi1' sample description entries has been added
- ISOBMFF multiplexing for 'vvc1' and 'vvi1' sample description entries has been added
- Inspection of files containing VVC tracks has been added (partial support, no bitstream parsing of VVC has been added yet).

The tools used in this demo were based on GPAC 1.0. In this version, the DASH segmenter is independent from the media packaging format and did not require any modification. It will however require further update once the "codecs" MIME parameter for VVC have been defined; for the purpose of the demo, the "codecs" MIME parameter for VVC is set to "vvc1" only. Since the demo relies on GPAC 1.0, the packager can output to both MPEG-DASH and HLS formats.

Similarly, the DASH access engine in GPAC is independent from the media packaging format and did not require any modification. The OpenVVC decoder is integrated through a patched version of libavcodec. The playback chain has been tested under Windows, Linux and Mac OSX platforms. Since the experiment, the VVC support has been merged in the

GPAC's main code repository in the master branch<sup>4</sup>.

### C. World-First 4K Live OTTChannel with VVC

The trial depicted in Figure 5c took place in September 2020 and is the result of a collaboration between the following entities:

- ATEME provided the encoding unit.
- Telecom Paris provided the DASH packager (MP4Box) and the player (MP4Client) from GPAC.
- IETR provided the VVC real-time decoding library used by MP4Client.
- Akamai provided content delivery network (CDN) infrastructure supporting HTTP chunk transfer encoding to enable low-latency.

In this experiment, an end-to-end live 4K TV channel was demonstrated during a period of 1 month. The input video, provided by The Explorers was timestamped with UTC time and live-encoded using the TitanLive platform. The VVC encoding was carried out with low-latency CMAF packaging, issuing 100ms chunks within a 2000ms segments, pushed to the Akamai CDN thanks to HTTP chunk transfer encoding. The GPAC player described in the previous section was used

<sup>4</sup><https://github.com/gpac/gpac/tree/master>

and highlighted the low-latency delivery when compared to UTC time at the receiver side (typically measuring a 2s glass-to-glass).

## VII. CONCLUSION

In this paper, we have addressed several important aspects of the latest video coding standard VVC from market use-cases, coding tools description, per-tool coding efficiency and complexity assessments, to the description of real time implementations of VVC codecs used in early commercial trials. Real-time implementations of VVC codecs as well as its adoption by application standards are essential to ensure a wide adoption and a successful deployment of the VVC standard. The current status of the developed real time VVC codecs and the demonstrated end-to-end VVC transmissions over broadcast and OTT communication mediums clearly show that the VVC technology is mature enough and ready for real deployment on consumer electronic products. Our prediction is that VVC will be integrated in most of the consumer electronics devices in a near future.

## REFERENCES

- [1] U. Cisco, "Cisco annual internet report (2018–2023) white paper," 2020.
- [2] N. Arulrajah and E. Marketing, "The world is in a state of flux. is your website traffic, too?" March 2021.
- [3] M. Efoui-Hess, "Climate crisis: The un- sustainable use of online video. The practical case for digital sobriety," July 2019.
- [4] "Jct-vc - joint collaborative team on video coding," <https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jctvc.aspx>, 2010.
- [5] G. J. Sullivan, J. M. Boyce, Y. Chen, J. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (hevc)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, December 2013.
- [6] G. J. Sullivan and T. Wiegand, "Video compression - from concepts to the h.264/avc standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, January 2005.
- [7] T. K. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J.-R. Ohm, and G. J. Sullivan, "Video quality evaluation methodology and verification testing of hevc compression performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 76–90, 2016.
- [8] 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)," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463–1493, 2021.
- [9] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.
- [10] N. Sidaty, W. Hamidouche, O. Déforges, P. Philippe, and J. Fournier, "Compression Performance of the Versatile Video Coding: HD and UHD Visual Quality Monitoring," in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.
- [11] C. Bonnineau, W. Hamidouche, J. Fournier, N. Sidaty, J.-F. Travers, and O. Deforges, "Perceptual quality assessment of hevc and vvc standards for 8k video resolution," *arXiv preprint arXiv:2109.06555*, 2021.
- [12] F. Bossen, K. Sühling, A. Wiecekowski, and S. Liu, "Vvc complexity and software implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 3765–3778, 2021.
- [13] DVB, "Press release: Dvb prepares the way for advanced 4k and 8k broadcast and broadband television," July 2021.
- [14] "TR26.955: Video codec characteristics for 5G-based services and applications," *3GPP*.
- [15] 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 Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3779–3800, 2021.
- [16] Y.-W. Huang, J. An, H. Huang, X. Li, S. Hsiang, K. Zhang, H. Gao, J. Ma, , and O. Chubach, "Block partitioning structure in the vvc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 3818–3833, 2021.
- [17] J. Pfaff, A. Filippov, S. Liu, X. Zhao, J. Chen, S. De-Luxán-Hernández, T. Wiegand, V. Ruffitskiy, A. Ramasubramonian, and G. Van der Auwera, "Intra prediction and mode coding in vvc," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 3834 – 3847, 2021.
- [18] X. Zhao, S.-H. Kim, Y. Zhao, H. E. Egilmez, M. Koo, S. Liu, J. Lainema, and M. Karczewicz, "Transform coding in the vvc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 3834 – 3847, 2021.
- [19] 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 Transactions on Circuits and Systems for Video Technology*, pp. 3907 – 3925, 2021.
- [20] "Working practices using objective metrics for evaluation of video coding efficiency experiments," *document ITU-T HSTP-VID-WPOM and ISO/IEC DTR 23002-8, ITU-T and ISO/IEC JTC 1*, 2020.
- [21] M. Orduna, C. Díaz, L. Muñoz, P. Pérez, I. Benito, and N. García, "Video multimethod assessment fusion (vmaf) on 360vr contents," *IEEE Transactions on Consumer Electronics*, Vol.66, Issue 1, Feb. 2020.
- [22] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," *Proc. 37th IEEE Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, USA*, Nov. 2003.
- [23] W.-J. et al., "JVET AHG report: Tool reporting procedure and testing (AHG13)," in *Document JVET-T0013, Teleconference*, October 2020.
- [24] E. François, M. Kerdranvat, R. Jullian, C. Chevance, P. de Lagrange, F. Urban, T. Poirier, and Y. Chen, "Vvc per-tool performance evaluation compared to hevc," *IBC 2020, Amsterdam, NL*, Sept. 2020.
- [25] G.-J. Sullivan, "Deployment status of the VVC standard," in *Document JVET-V0021, Online meeting*, March 2021.
- [26] A. Wiecekowski, 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," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3124–3128.
- [27] J. Brandenburg, A. Wiecekowski, T. Hinz, A. Henkel, V. George, I. Zupancic, C. Stoffers, B. Bross, H. Schwarz, and D. Marpe, "Towards fast and efficient vvc encoding," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6.
- [28] A. Wiecekowski and B. Bross, "Vdec fraunhofer versatile video decoder," *Heinrich Hertz Institute (HHI) White Paper*, 2020.
- [29] A. W. et al, "Towards a live software decoder implementation for the upcoming versatile video coding (vvc) codec," *IEEE International Conference on Image Processing*, pp. 3124–3128, 2020.
- [30] C. R. Helmrich, M. Siekmann, S. Becker, S. Bosse, D. Marpe, and T. Wiegand, "Xpsnr: A low-complexity extension of the perceptually weighted peak signal-to-noise ratio for high-resolution video quality assessment," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2727–2731.
- [31] T. Biatek, M. Abdoli, T. Guionnet, A. Nasrallah, and M. Raulet, "Future MPEG standards VVC and EVC: 8K Broadcast Enabler," in *International Broadcasting Convention (IBC)*, September 2020.
- [32] "Text of ISO/IEC 13818-1:2019 DAM 2 Carriage of VVC in MPEG-2 TS," *MPEG document w19436*, July 2020.
- [33] "Text of ISO/IEC 14496-15:2019 DAM 2 Carriage of VVC and EVC in ISO/BMFF," *MPEG document w19454*, July 2020.
- [34] T. Biatek, M. Abdoli, T. Guionnet, M. Raulet, T. Wrede, J. Outters, T. Christophory, H. Bauzée-Luyssen, S. Latapie, J.-B. Kempf, P.-L. Cabarat, and W. Hamidouche, "End-to-End UHD Satellite Broadcast Transmission using VVC," *MPEG document m54377*, 2020.
- [35] T. Biatek, M. Abdoli, T. Guionnet, M. Raulet, J. L. Feuvre, P.-L. Cabarat, and W. Hamidouche, "End-to-End OTT Streaming using DASH/VVC," *MPEG document m54379*, 2020.
- [36] B. Bross, J. Chen, and S. Liu, "Versatile Video Coding (Draft 6)," *JVET document Q2001*, 2019.
- [37] "ISO/IEC 13818-1:2019 CDAM2 Carriage of VVC in MPEG-2 TS," *MPEG document N19048*, 2020.
- [38] J. Le Feuvre, C. Concolato, and J.-C. Moissinac, "GPAC: open source multimedia framework," in *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07*. Augsburg, Germany: ACM Press, 2007, p. 1009.