VVC PER-TOOL PERFORMANCE EVALUATION COMPARED TO HEVC

Edouard François, Michel Kerdranvat, Rémi Jullian, Christophe Chevance,
Philippe De Lagrange, Fabrice Urban, Tangi Poirier, Ya Chen

InterDigital, France

ABSTRACT

The Versatile Video Coding (VVC) is the most recent video coding standard jointly developed by MPEG (ISO/IEC) and VCEG (ITU-T) in the JVET (Joint Video Experts Team). The VVC Final Draft International Standard was issued in mid-2020. VVC can be considered as the state-of-the-art video coding standard, with an estimated bitrate gain around 40% versus the High Efficiency Video Coding (HEVC). VVC has been developed incrementally based on the HEVC design, with the introduction of multiple new coding tools in all building blocks of the codec architecture. This paper aims at providing an overview of VVC by highlighting the main differences compared to HEVC. It reports a compression performance analysis, based on the coding gain evaluation of each tool for various contents (including contents not used in JVET). The analysis also considers the impact of the tools in terms of encoding and decoding complexity. Global performance measures with regards to HEVC are provided in different encoding configurations and picture formats.

INTRODUCTION

The Versatile Video Coding (VVC) standardization project started by an exploratory phase in mid-2015. This phase was concluded at the end of 2017 by a Call for Proposals, and the standardization phase driven jointly by MPEG (ISO/IEC) and VCEG (ITU-T) in the Joint Video Expert Team (JVET) was launched in April 2018. After around two years development, VVC reached the Final Draft International Standard (FDIS) stage in mid-2020 [3.]. With an estimated bitrate gain of 40% over High Efficiency Video Coding (HEVC) [1. ,2.] for HD and 4K formats [4.], VVC can be considered as the state-of-the-art in video compression. VVC is a hybrid video coding based on a design similar to HEVC. It has been incrementally developed by bringing enhancements to existing (HEVC) coding tools, and by adding numerous new coding tools aimed at increasing the compression performance for a variety of video contents, including Standard Dynamic Range (SDR), High Dynamic Range (HDR), 360° video and computer graphics and screen content. High-level features are also specified in the VVC core design. VVC supports layered coding, giving access to spatial, SNR and temporal scalability. VVC also introduces the concept of self-decodable sub-pictures, allowing region-wise random access, that can be used for instance for viewport dependent streaming of 360° video. VVC is therefore a versatile video coding solution which is able to address a variety of use cases and applications.

The purpose of this paper is to provide an overview of VVC, with HEVC as a reference design, and to report performance evaluations of VVC compared to HEVC. It also provides detailed per-tool performance data of the main new coding elements specified in VVC. The remainder of the paper is structured as follows. A VVC overview is presented in the next section. The two following sections report the per-tool performance evaluation, and the performance comparisons between VVC and HEVC. The last section provides closing remarks.

VVC OVERVIEW

Figure 1 depicts a block diagram of a VVC decoder. The core architecture is very similar to HEVC, with the following main building blocks: entropy decoding of coding modes, coding parameters and prediction residual, inverse quantization and inverse transform of the prediction residual transform coefficients, intra and inter frame predictions, and in-loop filtering of the reconstructed signal obtained by adding the prediction and the decoded prediction residual. The block diagram shows the main enhancements compared to HEVC indicated in red dotted lines rectangles. As it can be observed, all the existing building blocks are impacted. The block diagram also comprises some new building blocks (indicated in red solid lines rectangles) that have been added to the core process. The main new elements compared to HEVC are also summarized per building block in Table 1. An overview of these different elements is provided in the following subsections. The analysis is not intended to be exhaustive, but to emphasize some important design differences between HEVC and VVC.

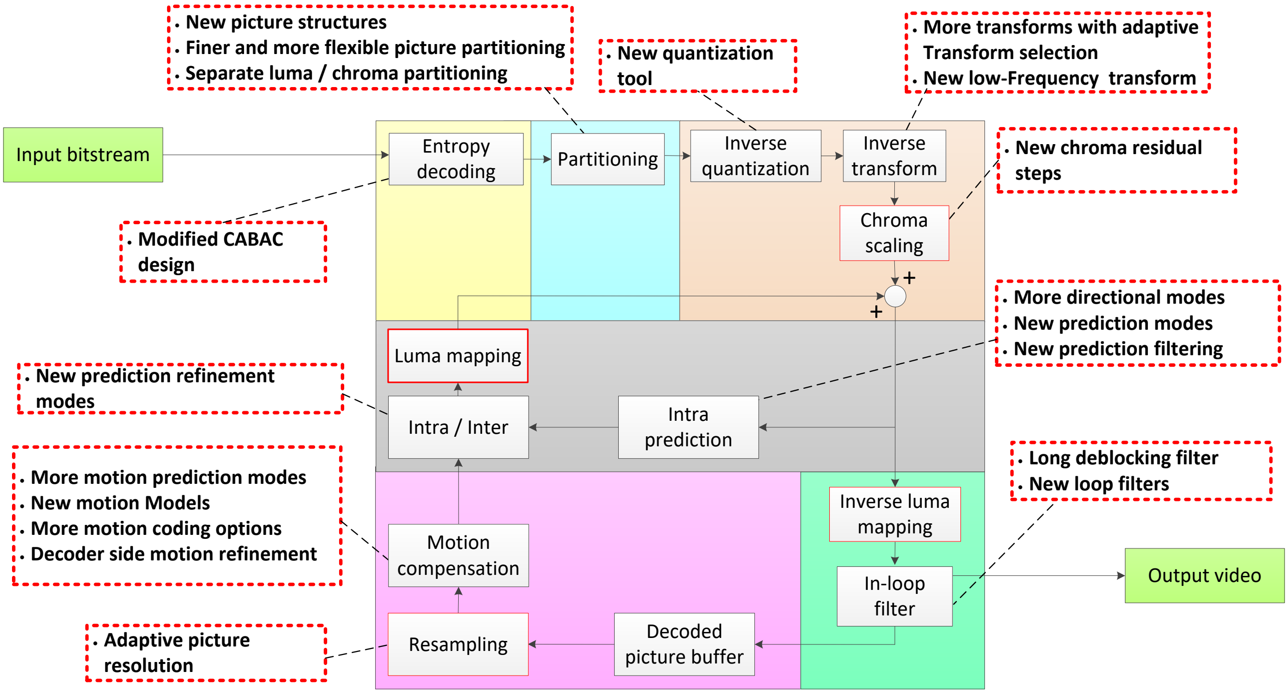

Figure 1 – Block-diagram of a VVC decoder.

Coding block	New Features
Partitioning	• Low-level: Coding Units finer splits: Binary Tree + Ternary Tree • Separate trees for luma/chroma • High-level: picture structures (sub-pictures) • Maximum CTU size 128x128
Intra Prediction	• Conventional modes: 65 angles and Wide angle intra prediction • New modes: Matrix-based intra prediction, Multi-reference lines intra prediction, Intra sub-partitions, Cross-component linear model • Prediction filtering: Position-dependent prediction combination
Inter Prediction	• Motion model: Affine motion, Luma & Chroma 1/16, Switchable interpolation filter • MV prediction: History based MV, Pairwise MV, Sub-block temporal motion prediction • MV coding/refinement: Adaptive MV resolution, Merge MVD, Symmetric MVD, Decoder-side MV refinement • Prediction: Geometric partitioning mode, Combined Intra-Inter prediction, bi-prediction with generalized weights, Wrap around for 360° video • Prediction refinement: Bi-directional optical flow, Prediction refinement with optical flow
Transform	• Multiple type transforms • Low-frequency non-separable transform • Sub-block Transforms
Quantization	• Dependent quantization
Residual Coding	• Improved contextual coding of transform coefficients • Joint coding of chroma residual
Entropy Coding	• Multi-hypothesis probability estimation • Context-adaptive probability window size
Loop Filters	• Adaptive loop filter • Cross-component adaptive loop filter • Longer-tap deblocking filters
Others	• Luma mapping with Chroma residual scaling • Reference picture resampling

Table 1 - Main new elements compared to HEVC.

Picture partitioning

As HEVC, VVC defines the concept of Coding Tree Unit (CTU) and Coding Unit (CU), which are both composed of one to three Coding Tree Blocks (CTBs) and Coding Blocks (CBs), depending on whether the picture is monochrome or not. The CTU is the largest possible CU and is the basic partitioning structure of the picture. The CTU can then be recursively split into smaller CUs. In VVC, the maximum CTU size is 128x128 while in HEVC it is 64x64 (for sake of simplification, in the following, the concept of CTU/CU size is related to the size of the luma CTB/CB attached to the CTU/CU).

VVC introduces new CU partitioning types that are not specified in HEVC. In addition to the quad-tree (QT) split mode of CUs, VVC also supports binary-tree (BT) and ternary-tree (TT) split modes. Binary split divides a CU into two equal-sized sub-CUs, while ternary split divides a CU into three sub-CUs of size 1/4th, 2/4th, 1/4th of the entire CU. The benefits of using larger CTU size and more flexible partitioning are illustrated in Figure 2, which shows the segmentation obtained from the HEVC reference software encoder (named HM) and from the VVC reference software encoder (named VTM). Larger blocks are used in VVC, which reduces the syntax coding cost, and partitions match more closely the objects' boundaries. The concept of Prediction Unit initially defined in HEVC, is more restricted in VVC. In HEVC, PUs can be subdivided from a CU with sharing the same coding mode, but with different coding parameters (e.g. intra prediction mode, uni- or bi-prediction mode). In HEVC the concept is generic, while in VVC it is limited to some particular coding modes (Intra sub-partitions and Geometric partitioning mode, as discussed below).

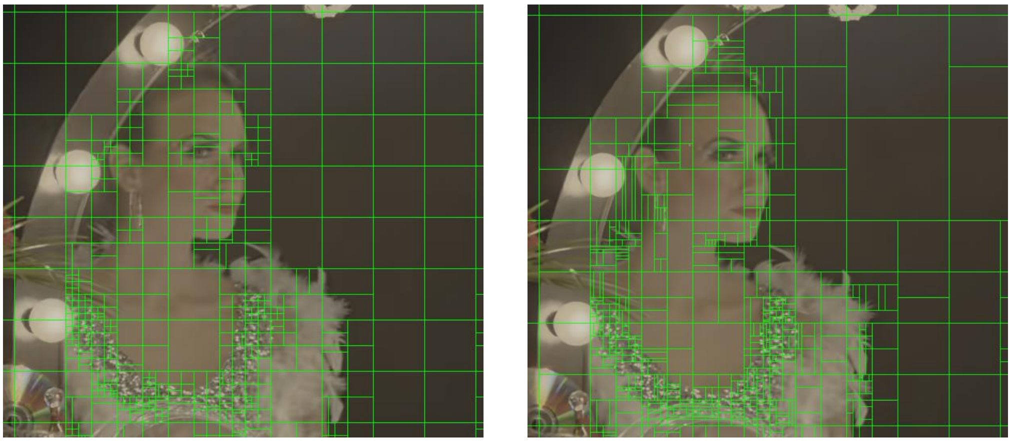

Figure 2 – Illustration of HEVC (left) and VVC (right) partitioning.

VVC also allows performing independent partitioning of the luma and chroma CTBs in a CTU inside an Intra slice or picture. This can benefit the coding cost of the chroma components, as they are generally less textured than the luma component and require less CU splits.

At a higher level, VVC and HEVC both support the concepts of slices and tiles. In addition, VVC introduces the concept of sub-pictures. Sub-pictures are areas located at the same location in the successive pictures of the sequence, that do not depend spatially or temporally on sub-pictures located elsewhere. They are therefore self-decodable and allow region-wise random access.

Intra prediction

As HEVC, VVC spatial prediction uses the two non-directional prediction modes named DC prediction and Planar prediction. They also specify directional spatial prediction modes, 33 for HEVC, 65 for VVC. VVC also includes 28 wide directional angles for non-square blocks.

VVC defines a number of additional intra prediction modes: Matrix-based intra prediction (MIP) modes performing the prediction using trained matrix multiplication of vector made of neighboring samples values (16 modes for 4x4 blocks, 8 modes for 8x4/4x8 blocks, 6 modes for larger blocks); Multi-reference lines intra prediction enabling using one among 3 neighboring lines/rows of the block as reference samples, while HEVC uses only the nearest neighboring line/row; Intra sub-partitions (ISP) that performs the CU prediction progressively by sub-PUs; and Cross-component linear model (CCLM) that performs chroma prediction from co-located luma samples using a linear model. Finally, post-prediction filtering aiming at smoothing the discontinuity at block boundaries has been improved in VVC by adding the Position-dependent prediction combination (PDPC). The intra prediction mode coding has also been enhanced by using 6 Most probable modes (MPMs) instead of 3 in HEVC.

Inter coding

Different aspects of inter prediction have been improved in VVC compared to HEVC.

A local affine motion model with 4 or 6 degrees of freedom is added to the existing translation motion model. The motion accuracy is also increased to 1/16th and the motion compensation uses switchable interpolation filters.

The motion information prediction (motion vectors – noted MV in Table 1, reference picture indexes) is improved by introducing new types of motion vector candidates: History based MV predictors (HMVP) enabling accessing MVs from non-neighbouring blocks using a FIFO list with 5 candidates; Pairwise MV adding the average of two existing candidates as a new MV candidate. Also, Sub-block temporal motion prediction (SbTMVP), which performs more accurate temporal motion prediction of a CU on a 4x4 basis, is added.

The motion vector refinement benefits from new ways of coding the motion vector difference (MVD) (added to the prediction): Adaptive motion vector resolution (AMVR) that controls the accuracy of the coded MVD; Merge with motion vector difference (MMVD) that allows small correction of MV candidates in a specific MV coding mode called merge mode; Symmetric motion vector difference (SMVD) that enables reducing the MVD cost in case of bi-directional prediction. In addition, a Decoder-side motion vector refinement (DMVR), deriving a sub-block refined motion inside the CU at the decoder, is supported.

Several new prediction modes are added: Geometric partition mode (GPM) that enables splitting a CU into two (rectangular and non-rectangular) PUs; Combined inter-intra prediction (CIIP) that combines intra and inter prediction signals using weighted averaging; Bi-prediction with CU-level weight (BCW), allowing more flexible weighting combination than HEVC (not only averaging each prediction with ½ weight). For 360° video, a wrap-around padding solution is implemented to better handle motion compensation at picture borders.

Finally, post-prediction refinement tools are specified in VVC: Bi-directional optical flow (BDOF) that refines the prediction block in case of bi-prediction using the optical flow; Prediction refinement with optical flow (PROF) that applies in case of affine motion. Both tools operate at a 4x4 block granularity.

Figure 3 illustrates the mode selection difference obtained from the HEVC reference encoder and from the VVC reference encoder. The figure depicts the partitioning and CUs coded as Intra are highlighted in orange. It is observed that VVC strongly reduces the areas coded as Intra, which shows the higher performance of inter prediction in VVC.

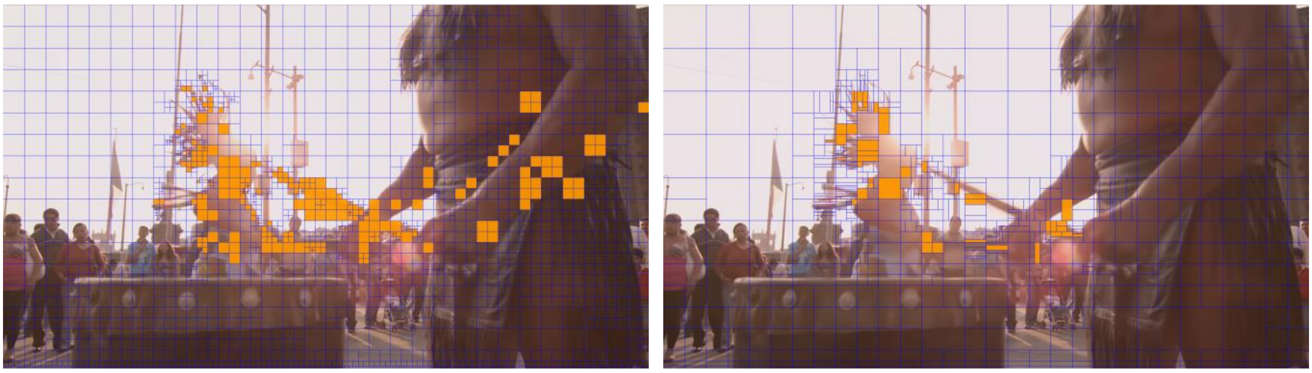

Figure 3 – Illustration of HEVC (left) and VVC (right) selected intra (in orange)/inter modes per CU.

Transform and Quantization

HEVC uses recursive separable DCT-2 from 4x4 to 32x32 square blocks in most cases, except for 4x4 intra blocks where separable DST-7 is used. VVC uses multiple transforms set (MTS) which is an extended set of transform kernels, consisting of DCT-2, DCT-8, and DST-7, but does not support the recursive transform feature. When DCT-2 is used, it applies in both dimensions and can be used from 2x2 to 64x64 block sizes. When DCT-2 is not used, a combination of DCT-8 and DST-7 can be selected for the horizontal and vertical transforms, and for block sizes up to 32x32. VVC also introduces a Low-frequency non-separable transform (LFNST), that applies as an additional transform stage only for intra coded transform blocks. Another new transform tool in VVC is the Sub-block Transforms (SBT), which only performs the transform to a sub-part of the inter CU; the residual signal is zeroed out in the remaining sub-parts. The horizontal and vertical transforms are implicitly inferred from the zeroed-out block shape.

In VVC, a new quantization design is specified, named Dependent quantization (DQ). Instead of using one single scalar quantizer with a given quantization step (derived from the quantization parameter), it adaptively switches between two interleaved scalar quantizers with twice the quantization step. This tool is based on a state machine that adaptively selects the quantizer to be applied to a transform coefficient level considering previous coefficient levels in reconstruction order.

Prediction residual and entropy coding

VVC uses the same basic entropy coding engine design as HEVC, named CABAC. The CABAC design in VVC has been modified by using a Multi-hypothesis probability estimation, and a Context-adaptive probability window size.

For chroma residual coding, VVC introduces a Joint coding of chroma residual (JCCR), which codes one single residual for the two chroma components of a CU. This tool exploits the dependencies between the chroma residuals.

Loop filtering

In addition to the deblocking and the sample adaptive offset filters already present in HEVC, VVC supports two new in-loop filters: Adaptive loop filter (ALF), and Cross-component adaptive loop filter (CCALF), which add offsets obtained from a linear filtering of the neighbouring samples to the reconstructed signal. Adaptive loop filter operates as an intra-component process, while Cross-component adaptive loop filter operates only to chroma components based on reconstructed luma samples. The VVC deblocking filter uses longer filter taps than HEVC, leading possibly to smoother textures.

Other tools

As depicted in Figure **1**, VVC decoder design includes new coding elements, namely Luma mapping with chroma scaling (LMCS), and Reference picture resampling (RPR).

Luma mapping with chroma scaling consists of two parts, luma mapping that applies to the luma prediction signal, with the inverse luma mapping applying before the loop filter; and chroma scaling, which applies to decoded chroma residuals. The purpose of this tool is to better benefit from the actual signal codeword range, in order to adaptively improve the range occupancy.

Reference picture resampling allows adapting dynamically the resolution of the coded picture, for coding efficiency, and for scalability support.

PER-TOOL EVALUATION

During the development of VVC, regular per-tool evaluations have been performed after the issuing of each new version of the draft specification and of its implemented reference version (VTM) [4.], using the common test conditions (CTC) defined by JVET [5.].

In this paper, a similar approach has been followed by measuring the loss on the global performance of switching off a tool, and the per-tool evaluation reported in [4.] has been completed by a) performing the evaluation on an alternate set of test sequences, made of HD and UHD contents, than the JVET set of test sequences ; b) measuring the coding performance using MS-SSIM [6.] and VMAF [7.] as additional objective metrics, in complement to the conventional PSNR.

Test configurations

The evaluations have been performed considering two test configurations, “All intra” (AI) and “Random access” (RA). In “All intra” configuration, the pictures are all coded in intra mode, without any temporal dependency to other pictures. In “Random access” configuration, inter-prediction is enabled with a GOP size of 16 pictures, and the insertion of an intra-picture approximately each 1 second.

As mentioned above, two sets of test sequences were used. The first sequence set is the same as used in JVET but limited to the 5 HD sequences and 6 UHD sequences. The second sequence set is made of 5 HD sequences and 5 UHD sequences, not included in the JVET test set. Table 2 provides the characteristics (picture resolution, bit-depth (BD) and frame rate (fps)) of these sequences. The resulting test set provides a variety of resolutions, bit-depth and frame rate.

	JVET test set			Non-JVET test set		
Resolution	**Sequence Name**	**BD**	**fps**	**Sequence Name**	**BD**	**fps**
3840x2160	Tango2	10	60	Rowing	10	120
	FoodMarket4	10	60	Brest_Sedof	10	60
	Campfire	10	30	Paris_Manege	10	60
	CatRobot	10	60	EBU_Lupo_Boa	10	50
	DaylightRoad2	10	60	EBU_Park_Dancers	10	50
	ParkRunning3	10	50			
1920X1080	MarketPlace	10	60	Birthday	8	60
	RitualDance	10	60	CrowdRun	10	50
	Cactus	8	50	Trafic	8	30
	BasketballDrive	8	50	Tennis	8	24
	BQTerrace	8	60	Walk Path	10	24

Table 2. List of tested JVET and non- JVET test content.

Results

The tested tools or settings performance is summarized in the Table 3, for “All Intra” and “Random Access” configurations. For a complete naming of the tools, a glossary is provided in the end of this paper. The results are reported using the “Bjøntegaard Delta-Rate” (BD) [8. , 9.] measuring an estimated average bit-rate variation between the reference and the tested tool. Results for three objective metrics are reported, PSNR, MS-SSIM [6.] and VMAF [7.]. The BD-PSNR is computed as a weighted average of the Y, U, and V BD-PSNRs (using weight 6,1,1). VMAF and MS-SSIM are measured only on the luma component. Tools are grouped per set as follows: **Set1** – tools operating only to intra CUs, **Set2** – tools operating in both intra and inter CUs, **Set3** – encoder settings operating in both intra and inter CUs, **Set4** – tools operating only to inter CUs.

Figure **4** compares the performance of the tools for the three considered metrics, in Random access configuration, which is relevant for broadcast scenarios. The figure depicts the tools ranked by performance (using PSNR metric), and tools with performance below 0.3% variation are not shown. The figure shows that the per-tool performance is rather consistent among the three considered objective metrics. The tools impacting mostly the chroma component (CCLM, CCALF, CST) have low impact on VMAF and MS-SSIM metrics, that are both only measured on the luma component. As explained above, the PSNR metric mixes the PSNR from the three components.

Figure 5 compares the PSNR performance of the tools for JVET, non-JVET content, and all content, in Random access configuration. It is observed that for most of the tools, the tool performance is very similar between JVET and non-JVET content. The VVC tools that bring most of the gains are Binary- and Ternary-tree partitioning, Adaptive loop filter and Cross-component adaptive loop filter, Cross-component linear model, and usage of CTU 128x128 instead of 64x64 (as typically used in HEVC). These tools provide coding gains from around

2% to 10-12%. The cumulative gain from these tools could be estimated around 50% of the overall gain.

Figure 6 reports the performance versus the VTM8.0 encoding and decoding runtime variations for tools with BD-PSNR performing above 1%. The runtime variation is measured as the ratio of runtime when applying the test setting, compared to the runtime of the default configuration, that is, without applying the test setting. For all the cases, the decoding runtime stands from around 80% to a few more than 100%. On encoder side, the partitioning tools (BT, TT) have a strong impact on the runtime. Removing BT split feature reduces the VTM8.0 encoding runtime by 2. Removing BT and TT split feature reduces the VTM8.0 encoding runtime by 6. Disabling tools related to motion modelling (Affine, Adaptive motion vector resolution) reduces the encoding runtime by a bit less than 20%.

	AI JVET content			AI non-JVET content			RA JVET content			RA non-JVET content		
Abbreviation	PSNR	VMAF	MS-SSIM	PSNR	VMAF	MS-SSIM	PSNR	VMAF	MS-SSIM	PSNR	VMAF	MS-SSIM
CST	2.1%	0.3%	0.1%	2.4%	0.6%	0.4%	0.9%	0.1%	0.0%	0.9%	0.2%	0.3%
MRLP	0.2%	-0.2%	0.1%	0.1%	0.2%	0.2%	0.1%	0.1%	0.1%	0.0%	0.0%	0.0%
ISP	0.3%	0.1%	0.5%	0.4%	0.1%	0.3%	0.2%	0.1%	0.2%	0.2%	0.2%	0.2%
MIP	0.7%	0.6%	0.9%	0.6%	0.8%	0.6%	0.4%	0.4%	0.4%	0.4%	0.4%	0.5%
CCLM	5.2%	2.5%	2.4%	3.6%	1.1%	0.8%	3.7%	1.4%	1.4%	3.0%	0.5%	0.5%
IBC	0.6%	0.9%	0.7%	1.2%	1.3%	1.5%	0.1%	0.2%	0.1%	0.0%	0.0%	0.0%
LFNST	1.3%	1.4%	1.3%	0.8%	0.9%	0.4%	0.8%	1.0%	0.8%	0.5%	0.8%	0.6%
JCCR	0.5%	0.1%	0.8%	0.5%	0.4%	0.3%	0.4%	0.6%	0.6%	0.3%	0.3%	0.3%
MTS	1.2%	0.9%	1.1%	1.6%	1.8%	1.1%	0.8%	0.7%	0.8%	1.0%	1.0%	1.0%
DQ	2.0%	1.3%	1.7%	1.7%	1.3%	1.8%	1.5%	1.2%	1.7%	1.6%	1.1%	1.7%
LMCS	0.9%	-0.1%	1.4%	0.2%	0.1%	0.1%	1.4%	1.6%	1.7%	-0.5%	-0.9%	-0.2%
SAO	0.0%	-0.4%	0.0%	0.1%	-0.1%	0.0%	0.1%	0.1%	0.1%	0.1%	0.2%	0.1%
ALF	4.0%	3.5%	0.8%	4.5%	4.8%	0.7%	8.0%	9.8%	1.8%	5.8%	6.4%	1.2%
CCALF	1.5%	-0.2%	-0.2%	1.9%	-0.1%	-0.1%	2.9%	-0.2%	-0.2%	2.2%	-0.1%	-0.1%
BT_TT	6.9%	6.4%	5.7%	7.0%	5.9%	5.2%	12.4%	11.3%	11.3%	12.8%	11.9%	11.7%
TT	1.0%	1.3%	1.0%	1.1%	1.2%	0.9%	2.5%	2.2%	2.3%	2.5%	2.4%	2.4%
CTU32	3.8%	2.0%	3.0%	3.3%	2.0%	2.1%	15.0%	15.4%	15.8%	7.9%	8.0%	7.5%
CTU64	0.6%	0.1%	0.6%	0.5%	0.3%	0.3%	2.8%	3.1%	3.0%	1.3%	1.3%	1.1%
TU32	1.2%	0.1%	1.3%	1.2%	0.5%	0.7%	1.9%	1.6%	1.9%	1.2%	0.8%	1.1%
MTS_IMP	0.5%	0.5%	0.5%	0.5%	0.6%	0.5%	0.2%	0.3%	0.3%	0.2%	0.3%	0.3%
MTS_EXP	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	-0.1%	-0.2%	-0.1%	-0.1%	-0.3%	-0.1%
MTS_CAND1	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SBT							0.3%	0.4%	0.2%	0.5%	0.5%	0.3%
AFF							3.4%	3.6%	4.1%	1.9%	2.0%	2.3%
SbTMC							0.4%	0.5%	0.5%	0.4%	0.5%	0.5%
AMVR							1.7%	1.6%	1.8%	1.0%	0.8%	0.9%
MMVD							0.6%	0.5%	0.6%	0.4%	0.3%	0.4%
GPM							0.5%	0.3%	0.5%	0.7%	0.6%	0.7%
CIIP							0.2%	0.2%	0.3%	0.2%	0.3%	0.3%
BCW							0.5%	0.3%	0.4%	0.2%	0.0%	0.1%
PROF							0.7%	0.7%	0.7%	0.5%	0.3%	0.4%
BDOF							0.6%	1.1%	0.8%	0.9%	1.4%	1.0%
SMVD							0.3%	0.2%	0.3%	0.2%	0.1%	0.2%
DMVR							1.0%	1.0%	1.1%	0.8%	0.9%	1.0%

Table 3. Tested tools/settings performance in All Intra (AI) and Random Access (RA) configurations.

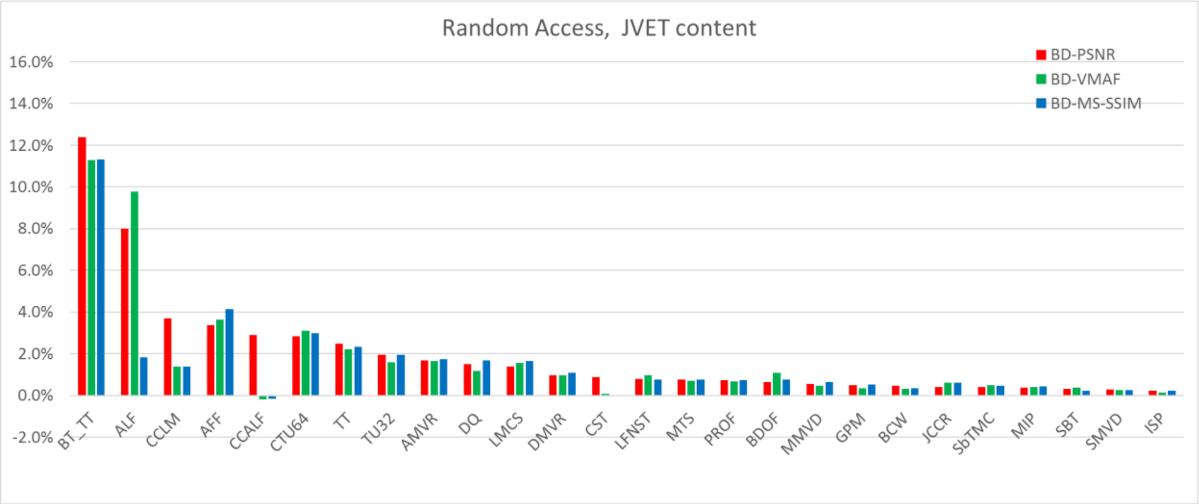

Figure 4 – BD-PSNR, VMAF, MS-SSIM, per tool, Random Access, JVET content.

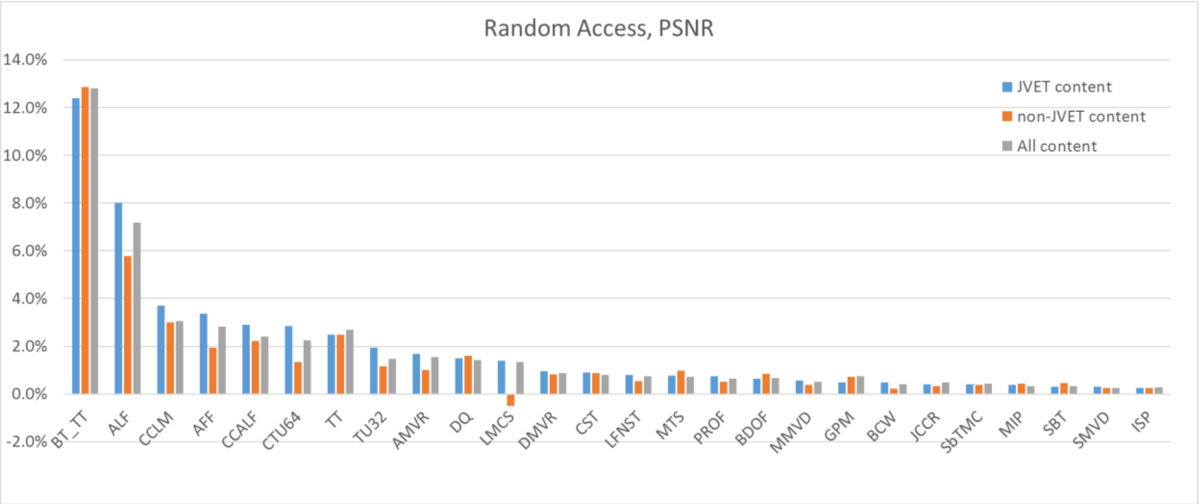

Figure 5 – Tools ranked per BD-PSNR, for Random Access, JVET and non-JVET content.

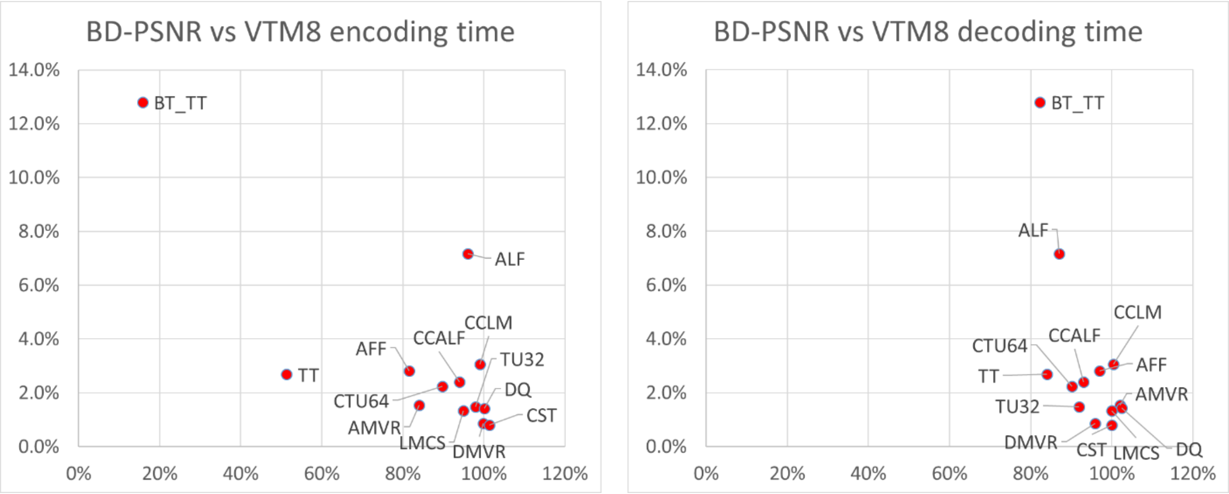

Figure 6 – BD-PSNR versus VTM8.0 encoding (left) and decoding (right) time variations.

COMPRESSION PERFORMANCE COMPARISON VVC VS HEVC

Objective evaluation

Performance comparisons between HEVC (reference software version HM16.19) and VVC (reference software version VTM8.0) have also been made using the same JVET and non-JVET test set. The non-JVET test set has been completed by 5 UHD sequences, to have a wider variety of content. Results are reported for these two sets in the Table 4. A positive number indicates the estimated average bit-rate reduction, for the considered objective metric.

A first observation that can be made from these results is the consistency of BD performance for the three considered objective metrics, especially PSNR and VMAF that show a high correlation. The average BD-PSNR gain from VTM8.0 above HM16.19 is confirmed using the VMAF metric. MS-SSIM has a behaviour a bit different, with BD performance in general below the two other metrics.

A second observation from Table 4 is that a performance difference around 2-3% in All Intra, and 5-6% in Random Access between JVET and non-JVET content, is observed. The difference can be naturally explained by the fact that VVC has been developed based on the JVET test content, and therefore a drop when using alternate content can be expected. However, non-normative algorithms of VTM encoder can be improved to better cope with all types of content as only the decoder is normative.

	All Intra					Random Access				
	PSNR	VMAF	MS-SSIM	EncTime	DecTime	PSNR	VMAF	MS-SSIM	EncTime	DecTime
JVET content										
class A	29.7%	30.3%	28.4%	2081%	234%	41.5%	43.6%	40.6%	914%	267%
class B	24.4%	24.0%	24.8%	2906%	263%	38.3%	38.9%	33.6%	965%	257%
overall	**27.3%**	**27.4%**	**26.8%**	**2494%**	**249%**	**40.1%**	**41.4%**	**37.4%**	**939%**	**262%**
non-JVET content										
class A	26.0%	26.1%	23.7%	1987%	229%	35.6%	35.9%	31.9%	923%	195%
class B	23.0%	22.8%	22.4%	2878%	271%	33.0%	33.4%	31.6%	1115%	295%
overall	**25.0%**	**25.0%**	**23.3%**	**2433%**	**250%**	**34.7%**	**35.1%**	**31.8%**	**1019%**	**245%**

Table 4. VTM8 vs HM16.18 performance.

Figure **7** and Figure **8** depict the BD performance per sequence, in Random Access, of each test sequence of the JVET test set, and non-JVET test set, respectively. The sequences are named UHDxx or HDyy depending on their picture resolution. JVET set comprises UHD01 to 06, and HD01 to 05. Non-JVET set comprises UHD07 to 16, and HD06 to 10. A BD-PSNR peak gain of around 45% is observed both in both test sets. It is also observed that several non-JVET test sequences lead to a BD-PSNR gain around or below 30%, which is not the case for the JVET test set. This lower gain seems to happen for sequences with high-frequency texture close to a white noise, which is obviously quite challenging for any codec. Noise cannot be properly predicted, and signal statistics cannot be properly exploited for the entropy coding engine.

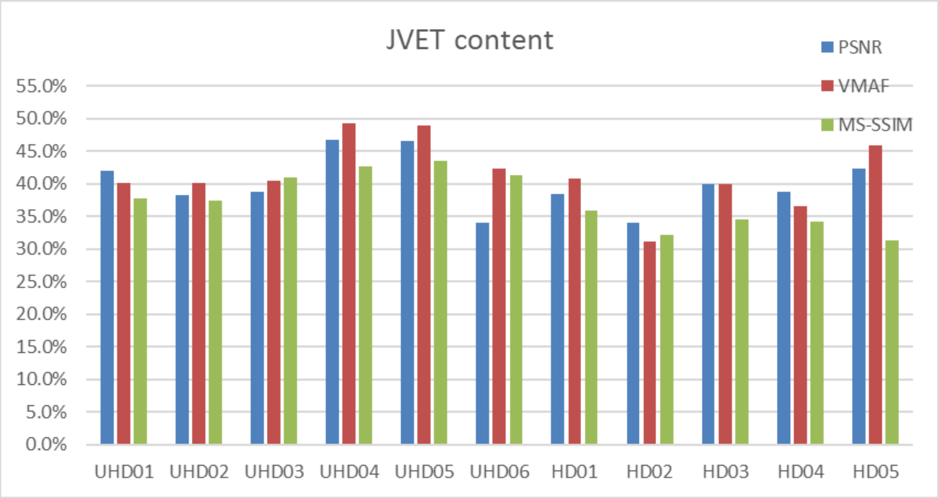

Figure 7 – VTM versus HM performance, per sequence and per metric, for JVET content.

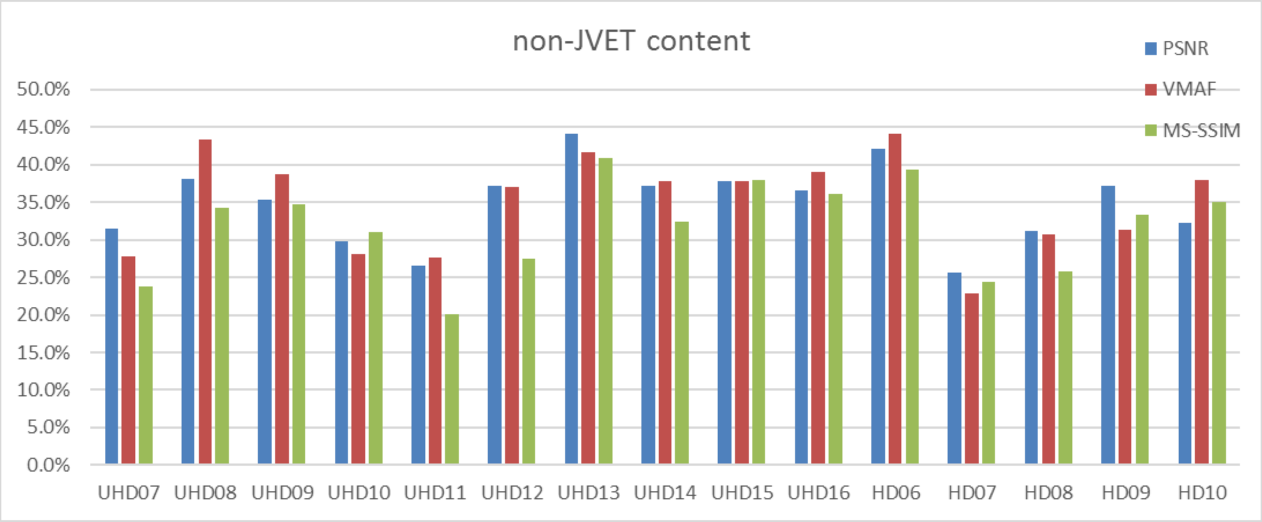

Figure 8 – VTM versus HM performance, per sequence and per metric, for non-JVET content.

Subjective evaluation

Partial subjective evaluations have also been made with expert viewers. Visual comparisons were made between VTM8.0 bitstreams coded at QP 37, HM16.19 bitstreams at the VTM8.0 bitrate and at 166% the VTM8.0 bitrate. The HM at 166% the VTM bitrate corresponds to a VTM bit rate 40% lower than the HM one. The goal of using such settings was to confirm the gains using objective metrics reported above.

Two main observations were reported from the visual comparisons. VTM has been judged as clearly outperforming HM when using same bitrates for both. The quality observed between HM and VTM at 40% lower bitrate has been considered by the viewers as generally equivalent. However, behaviour from both codecs can result in different visual impacts. VVC has a trend to smooth more the textures than HM, which results in possible texture loss for static areas, but which is very beneficial when considering content with rather flat or moving areas such as waves on water or human faces.

CONCLUDING REMARKS

This VVC overview and reported performance evaluation indicates that VVC surpasses HEVC by around 40% in compression efficiency, with high consistency among different objective metrics (PSNR, VMAF, MS-SSIM). The gap of 5% reported with non-JVET sequences is not surprising and in the range of what had been observed when HEVC was introduced with regards to AVC. This gap is expected to be filled in as encoding algorithms, which are non-normative, are still in the learning curve. Furthermore, the efficiency and complexity of each individual tool is measured on a wide set of test sequences. The evaluation disclosed here was focused on SDR content. Similar evaluations have been made on for HDR content (using BT.2100 PQ or HLG format) [10.], with very close trends to those reported in this paper.

VVC can be considered as the state-of-the-art video coding standard. It supports in addition new features of adaptive spatial resolution (using Reference picture resampling) from picture to picture, scalability (using layered coding) and spatial random access (using sub-pictures). Such features coupled with its high compression efficiency make this standard a versatile solution able to address a variety of use cases, applications and content types.

REFERENCES

1. High Efficiency Video Coding, Recommendation ITU-T H.265, 2018.
2. Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding, ISO/IEC 23008-2, 2018.
3. B. Bross, J. Chen, S. Liu, Y.-K. Wang, Versatile Video Coding (Draft 9), document JVET-R2001, 19th JVET meeting, Geneva, Switzerland, July 2020.
4. W.-J. Chien, J. Boyce, Y.-W. Chen, R. Chernyak, K. Choi, R. Hashimoto, Y.-W. Huang, H. Jang, R.-L. Liao, S. Liu, JVET AHG report: Tool reporting procedure and testing (AHG13), JVET-R0013, 18th JVET meeting, Teleconference, April 2020.
5. F. Bossen, J. Boyce, X. Li, V. Seregin, K. Sühring, JVET common test conditions and software reference configurations for SDR video, document JVET-N1010, 14th JVET meeting, Geneva, Switzerland, March 2019.
6. Z. Wang, E. P. Simoncelli, 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.
7. M. Orduna, C. Díaz, L. Muñoz, P. Pérez, I. Benito, N. García, Video Multimethod Assessment Fusion (VMAF) on 360VR Contents, IEEE Transactions on Consumer Electronics, Vol.66, Issue 1, Feb. 2020

8. Gisle Bjontegaard, “Calculation of Average PSNR Differences between RD curves”, ITU-T SG16/Q6 VCEG 13th meeting, Austin, Texas, USA, April 2001, Doc. VCEG-M33 (available at http://wftp3.itu.int/av-arch/video-site/0104_Aus/).

9. Gisle Bjontegaard, “Improvements of the BD-PSNR model”, ITU-T SG16/Q6 VCEG 35th meeting, Berlin, Germany, 16–18 July, 2008, Doc. VCEG-AI11 (available at http://wftp3.itu.int/av-arch/video-site/0807_Ber/).

10. A. Segall, E. François, W. Husak, S. Iwamura, D. Rusanovskyy, JVET AHG report: Coding of HDR/WCG material (AHG7), JVET-R0007, 18th JVET meeting, Teleconference, April 2020.

TOOLS GLOSSARY

	Abbreviation	Tool
Set1: **tools operating only to intra CUs**	CST	Chroma separate tree
	MRLP	Multi-reference line prediction
	ISP	Intra sub-partitioning
	MIP	Matrix based intra prediction
	CCLM	Cross-component linear model
	IBC	Intra block copy
	LFNST	Low frequency non-separable transform
Set2: **tools operating in both intra and inter CUs**	JCCR	Joint coding of chrominance residuals
	MTS	multiple transform set
	DQ	Dependent quantization
	LMCS	Luma Mapping with Chroma Scaling
	SAO	Sampled adaptive offset
	ALF	Adaptive loop filter
	CCALF	Cross component adaptive loop filter
Set3: **encoder settings operating in both intra and inter CUs**	BT_TT	BT+TT (QT-only)
	TT	TT (QT+BT only)
	CTU64	Max CTU size 64x64
	TU32	Max TU size 32x32
	MTS_IMP	Implicit MTS
	MTS_EXP	Explicit MTS 3
	MTS_CAND1	MTSIntraMaxCand1
	AFF	Affine motion model
Set4: **tools operating only to inter CUs**	SbTMC	subblock-based temporal merging candidates
	AMVR	Adaptive motion vector resolution
	GPM	Geometry partition
	BDOF	Bi-directional optical flow
	CIIP	Combined intra/inter prediction
	MMVD	Merge with MVD
	BCW	Bi-prediction with CU weights
	DMVR	Decoder motion vector refinement
	SBT	Sub-block Transform
	SMVD	Symmetric motion vector difference
	PROF	Prediction refinement using optical flow