

Received March 12, 2021, accepted April 10, 2021, date of publication May 3, 2021, date of current version May 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077116

Comparative Rate-Distortion-Complexity Analysis of VVC and HEVC Video Codecs

ALEXANDRE MERCAT¹, (Member, IEEE), ARTTU MÄKINEN¹, (Member, IEEE), JOOSE SAINIO,
ARI LEMMETTI¹, (Member, IEEE), MARKO VIITANEN¹, (Member, IEEE),
AND JARNO VANNE¹, (Member, IEEE)

Ultra Video Group, Tampere University, 33014 Tampere, Finland

Corresponding author: Alexandre Mercat (alexandre.mercat@tuni.fi)

This work was supported in part by the Academy of Finland (decision no. 301820).

ABSTRACT Versatile Video Coding (VVC/H.266) is the next-generation international video coding standard and a successor to the widespread High Efficiency Video Coding (HEVC/H.265). This paper analyzes the rate-distortion-complexity characteristics of the VVC reference software (VTM10.0) by using HEVC reference software (HM16.22) as an anchor. In this independent study, the rate-distortion performance of VTM was benchmarked against HM with the objective PSNR, SSIM, and VMAF quality metrics and the associated encoder and decoder complexities were profiled at function level using Intel VTune Profiler on Intel Xeon E5-2699 v4 22-core processors. For a fair comparison, all our experiments were conducted under the VTM common test conditions (CTC) that define 10-bit configurations of the VTM codec for the addressed All Intra (AI), Random Access (RA), and Low Delay B (LB) conditions. The VTM CTC test set was also extended with complementary 4K UHD sequences to elaborate RD characteristics with higher resolutions. According to our evaluations, VTM improves the average coding efficiency over HM, depending on quality metric, by 23.0-23.9% under the AI condition, 33.1-36.6% under the RA condition, and 26.7-29.5% under the LB condition. However, the coding gain of VTM comes with 34.0×, 8.8×, and 7.5× encoding complexity over that of HM under the AI, RA, and LB conditions, respectively. The corresponding overhead of the VTM decoder stays steady at 1.8× across all conditions. This study also pinpoints the most complex parts of the VTM codec and discusses practical implementation aspects of prospective real-time VVC encoders and decoders.

INDEX TERMS Common test conditions (CTC), HEVC test model (HM), high efficiency video coding (HEVC), objective quality analysis, performance profiling, rate-distortion-complexity (RDC), UVG dataset, versatile video coding (VVC), video codec, VVC test model (VTM).

I. INTRODUCTION

Our society is surrounded by a myriad of media applications where digital video is of the essence. According to Cisco, the global IP video traffic will increase fourfold from 2017 and account for 82% of all IP traffic by 2022 [1]. Moreover, Comcast estimates that the prevailing COVID-19 crisis has increased *Voice over Internet Protocol (VoIP)* and videoconferencing by 210–285% and other video consumption by 20–40% over that of the pre-pandemic period [2]. This snowballing growth is mainly driven by the omnipresent connectivity and proliferation of advanced multimedia solutions that support emerging bandwidth-greedy formats like 4K/8K

Ultra High Definition (UHD) or 360-degree omnidirectional videos.

Over the past three decades, ISO/IEC MPEG and ITU-T VCEG have addressed the exponential growth of digital video consumption by publishing a series of international video coding standards. The latest two established MPEG/ITU-T standards, *Advanced Video Coding (AVC/H.264)* [3] and *High Efficiency Video Coding (HEVC/H.265)* [4], were ratified in 2003 and 2013, respectively. As of now, AVC holds its position as the mainstream standard in existing applications but HEVC is gradually gaining market share in the state-of-the-art devices and services [5].

However, even HEVC is not able to meet the prospective industry needs, and future application scenarios call for more efficient compression for media storage and transmission [6]. Therefore, VCEG and MPEG again joined forces

The associate editor coordinating the review of this manuscript and approving it for publication was Gulistan Raja¹.

and formed the *Joint Video Exploration Team (JVET)* in Oct. 2015 to investigate coding techniques beyond the capabilities of HEVC. After a two-year exploratory phase, JVET was able to provide adequate evidence for the need of the new video coding standard that was named *Versatile Video Coding (VVC/H.266)*. In Oct. 2017, the JVET was reformed as *Joint Video Experts Team* with the goal of doubling the coding efficiency of VVC over that of HEVC for the same visual quality. The first version of the VVC standard was approved by ITU-T in July 2020 [7] and published by ISO/IEC as ITU-T H.266 | ISO/IEC 23090-3 in August 2020 [8].

The VVC reference software is called *VVC test model (VTM)* [9]. It is the successor to *Joint Exploration Model (JEM)* that JVET used as an experimental software in the exploration phase [10]. JEM was based on the HEVC reference software called *HEVC test model (HM)* [11]. It attained around 30% better coding efficiency than HM but at a cost of 9–36× computational complexity [10]. Therefore, JEM was replaced by VTM that originally contained a minimum set of coding tools and was gradually expanded thereafter. The latest version of VTM supports all normative coding tools of VVC and therefore serves as the most appropriate public reference for VVC. The evaluation of VTM is recommended to be performed under the four VTM *common test conditions (CTC)* [12]: *All Intra (AI)*, *Low Delay P (LP)*, *Low Delay B (LB)*, and *Random Access (RA)*.

This paper provides a comprehensive *rate-distortion-complexity (RDC)* comparison between the VVC and HEVC video codecs. In practice, the results were obtained by benchmarking the reference encoders and decoders of HM version 16.22 (*HM16.22*) and VTM version 10.0 (*VTM10.0*) under the AI, RA, and LB conditions. For a fair comparison, both VTM and HM were configured according to VTM CTC [12]. The *rate-distortion (RD)* performances are reported in terms of *Bjontegaard delta bitrates (BD-rates)* [13], [14] for identical visual quality measured with three different objective quality metrics: *Peak Signal-to-Noise Ratio (PSNR)*, *Structural SIMilarity (SSIM)* [15], and *Video Multimethod Assessment Fusion (VMAF)* [16]. The computational complexities are detailed at function level using Intel VTune Profiler [17] on Intel Xeon E5-2699 v4 22-core processors. The applied test set contains all natural VTM CTC sequences as well as eight versatile 4K120p sequences from our own *Ultra Video Group (UVG)* dataset [18].

Over the past two years, a couple of works have already compared the features of VVC and HEVC, but most of them address VTM8.0 or earlier versions [19]–[25], i.e., before the VVC standard was approved, which makes them outdated. This study is also far more extensive than the most recent one [26], especially in terms of different quality metrics, number of test sequences, and comprehensiveness in complexity analysis and classification of coding tools. Altogether, our results were compiled from over 1300 encoding and decoding runs that took approximately 1650 days of CPU time. This way, we are able to provide the video coding community with a reliable and comprehensive codec comparison. The selected

evaluation methodology follows our independent academic approach with HEVC and AVC in 2012 [27] and thereby continues the series of our baseline comparison studies.

The remainder of this paper is organized as follows. Section II investigates the comprehensiveness and up-to-datedness of the existing comparisons between VVC and HEVC. Section III presents the main differences between HEVC and VVC coding tools. Section IV describes the experimental setup and objective assessment criteria used in our comparative RDC analysis. Section V analyzes the RD characteristics of the VTM codec by reporting its coding efficiency over that of the HM codec. The absolute function-level complexities and relative complexity overheads of VTM are reported over HM by addressing the encoders in Section VI and the decoders in Section VII. Section VIII discusses practical aspects of real-time HEVC and VVC coding. Finally, Section IX concludes the paper.

II. PRIOR COMPARISONS BETWEEN VVC AND HEVC

Table 1 highlights the key differences between the existing comparisons and ours. Related works can be classified into three categories: 1) RD comparisons [19], [20]; 2) complexity comparisons [21], [22]; and 3) RDC comparisons [23]–[26].

TABLE 1. Existing comparisons of VTM and HM encoders.

Ref.	Encoder		Quality				Complexity			Condition			Sequence	
	VTM	HM	PSNR	SSIM	VMAF	Subjective	Encoder	Decoder	Func. level	AI	RA	LP/LB	Total	≥2160p
[19]	5.0	16.18	×								×		9	2
[20]	5.0	16.20	×	×	×	×					×		7	7
[21]	8.0	16.18					×				×	×	8	2
[22]	6.0	16.?					×	×	×		×	×	6	1
[23]	5.0	16.9	×				×		×		×		19	0
[24]	2.0	16.16	×				×				×		28	8
[25]	4.0	16.19	×				×			×	×		28	8
[26]	11.0	16.22	×				×	×		×	×	×	22	6
Our	10.0	16.22	×	×	×		×	×	×	×	×	×	30	14

A. RATE-DISTORTION COMPARISONS

The existing RD comparisons are focused on the older VTM5.0 [19], [20] under the RA condition. In [19], only the PSNR metric and nine sequences were used, but the comparison also included encoders other than VTM and HM.

The evaluation in [20] dealt with seven UHD and down-sampled HD sequences. PSNR, SSIM, and VMAF BD-rates were provided as well as subjective results based on a five-point grading scale.

B. COMPLEXITY COMPARISONS

The main contribution of [21] was memory profiling results of VTM8.0, containing the shares of memory accesses per each tool category. In addition, a more in-depth analysis of inter prediction was conducted since it causes the most memory accesses. The results were obtained using the 17 first

frames of eight sequences under the RA and *Low Delay (LD)* conditions (not specified whether LP or LB).

In [22], VTM6.0 was used to quantify the average shares of different encoding and decoding tool categories. HM served as an anchor. The analysis was performed with six VTM CTC sequences under the AI, RA, and LD conditions. Additionally, the average memory bandwidth requirements of the VTM codec were evaluated with three CTC 1080p sequences under the LD condition.

C. RATE-DISTORTION-COMPLEXITY COMPARISONS

In [23], VTM5.0 was compared with HM16.9 using 19 sequences, but only PSNR BD-rate results under the RA condition were provided. The complexity of VTM was considered with and without the *Single Instruction Multiple Data (SIMD)* optimizations that were shown to reduce the encoding time by roughly a third. A more thorough complexity analysis was also performed by providing the relative time for each VTM tool category in comparison with HM.

In [24], VTM2.0 was compared with HM16.16 under the RA condition, but only PSNR BD-rate and encoder complexity results were reported.

In [25], only PSNR BD-rate and encoder complexity results of VTM4.0 were provided under the AI and RA conditions. However, the analysis also included other encoders.

Similarly, the comparison made by JVET [26] only included PSNR BD-rate results, but under the AI, RA, LB, and LP conditions. In addition, both encoder and decoder complexity results were given but only in terms of overall coding time.

To the best of our knowledge, our study is the most comprehensive RDC analysis between VTM and HM containing three different quality metrics (PSNR, SSIM, and VMAF), 30 test sequences (from 240p to 2160p), three diverse test conditions (AI, RA, and LB), and VTM encoder and decoder complexity profiling at function level.

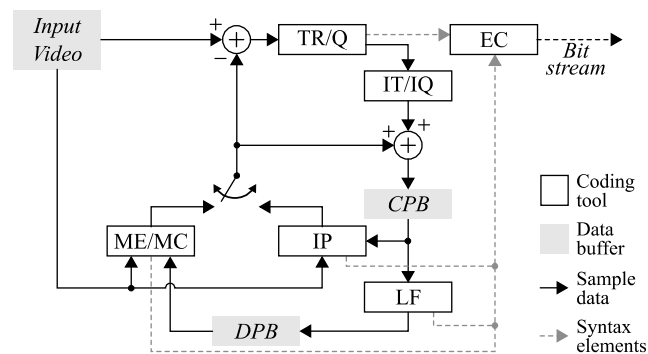


FIGURE 1. Simplified block diagram of a VVC encoder.

III. COMPARISON OF VVC AND HEVC CODING TOOLS

Fig. 1 depicts an overview of the VVC encoder architecture. Both VVC and HEVC encoding processes are based on the well-known block-based hybrid video coding scheme that is

composed of five stages: *intra prediction (IP)*, *motion estimation and compensation (ME/MC)* a.k.a. inter prediction, *forward/inverse transform and quantization (TR/Q)*, *entropy coding (EC)*, and *loop filtering (LF)*.

Table 2 summarizes the main coding tools of HEVC and VVC. Generally speaking, VVC has adopted many new coding tools in each coding stage. Please refer to VVC algorithm description [29] and specification [30] by JVET for further information.

IV. EXPERIMENTAL SETUP

All our experiments were performed under the VTM CTC [12]. The benchmarked codecs were VTM10.0 [9] and HM16.22 [11] Main 10 profile that were the latest available versions during our experiments.

A. TEST SEQUENCES

Table 3 details our test set that features a broad range of sequence parameters (spatial resolution, frame rate, and bit depth) and content (motion, texture, and illumination). It includes all 22 natural full-length 8-bit and 10-bit YUV420 test sequences specified “mandatory” in the VTM CTC (*classes A – E*) [12]. In addition, it was extended with eight 4K120p sequences from our UVG dataset [18] for more exhaustive RD analysis with future media formats. Our RD analysis is based on the entire test set but, to save profiling time, the complexity profiling was only conducted on the sequences of each VTM CTC class with the *highest (H)* and the *lowest (L)* complexities. The selection of these sequences was based on their overall encoding complexities averaged across all test runs.

B. CODING CONFIGURATIONS AND CONDITIONS

The VTM and HM encoders were configured to 10-bit mode. They adopted the AI, RA, and LB conditions with the base *quantization parameter (QP)* values of 22, 27, 32, and 37 from the VTM CTC [12]. The respective 10-bit configuration files are available online on per sequence basis at [9] for VTM and at [11] for HM.

Under the AI condition, all frames were encoded as I-frames in display order without any QP offsets. The complexity of the VTM intra coding was reduced by encoding only every eighth frame (I0, I8, ...) as per the VTM CTC. For a fair comparison, the same subsampling ratio was used with HM.

Under the RA condition, both VTM10.0 and HM16.22 encoders exploited a five-layer hierarchical coding structure with the *group of pictures (GOP)* size of 16. Table 4 details the coding order of the frames with the associated layers (L1 ...L5) and QP offsets (−3 ... +6). However, the QP offsets of the B-frames are subject to vary as a function of a scaling coefficient specified for each layer. The intra refresh period depends on the frame rate of the sequence and is rounded to multiples of the GOP size so that the time between successive I-frames is approximately 1 second, as defined by the VTM CTC. The interval between I-frames is filled

TABLE 2. Main coding tools of HEVC and VVC [28].

	HEVC	VVC
Block partitioning	<ul style="list-style-type: none"> ♦ Coding Tree Unit (CTU) quadtree (QT) structure ♦ From 64×64 to 8×8 Coding Unit (CU) size 	<ul style="list-style-type: none"> ♦ CTU quadtree structure with nested multi-type tree (QT+MTT) ♦ From 128×128 to 4×4 CU size ♦ Chroma separate tree (CST) ♦ Local dual tree ♦ Virtual pipeline data units (VPDUs)
Intra prediction	<ul style="list-style-type: none"> ♦ DC, planar ♦ 33 directional prediction modes ♦ Linear interpolation 	<ul style="list-style-type: none"> ♦ DC, planar ♦ 65 directional prediction modes ♦ Wide-angle prediction modes ♦ 4-tap interpolation filters (IFs) using 2 sets of filters ♦ Position-dependent prediction combination (PDCP) ♦ Multiple reference lines (MRL) ♦ Matrix-based intra-picture prediction (MIP) ♦ Cross-component linear model (CCLM) ♦ Intra sub-partitions (ISP)
Inter prediction	<ul style="list-style-type: none"> ♦ Merge mode ♦ Advanced motion vector prediction (MVP) ♦ 8-tap IFs for luma ♦ 4-tap IFs for chroma 	<ul style="list-style-type: none"> ♦ Extended merge mode and MVP with <ul style="list-style-type: none"> ▪ History-based MVP (HMVP) ▪ Pair-wise average MVP candidate ▪ Subblock-based temporal MVP (SBTMVP) ▪ Merge with motion vector difference (MMVD) ▪ Symmetric motion vector difference (SMVD) ♦ Adaptive motion vector resolution (AMVR) ♦ 8-tap or 6-tap IFs for luma ♦ 4-tap IFs for chroma ♦ Geometric partitioning mode (GPM) ♦ Bi-prediction with CU-level weight (BCW) ♦ Combined Intra/Inter prediction (CIIP) ♦ Decoder-side motion vector refinement (DMVR) ♦ Bi-directional optical flow (BDOF) ♦ Affine motion ♦ Prediction refinement with optical flow (PROF)
Forward/inverse transform and quantization	<ul style="list-style-type: none"> ♦ Square transform (up to 32×32) ♦ Discrete cosine transform and discrete sine transform <ul style="list-style-type: none"> ▪ DCT-II and DST-VII ♦ Sign data hiding (SDH) 	<ul style="list-style-type: none"> ♦ Square and rectangular transform (up to 64×64) ♦ Multiple transform selection (MTS) <ul style="list-style-type: none"> ▪ DCT-II, DST-VII, and DCT-VIII ♦ Non-separable secondary transform (LFNST) ♦ Subblock transform (SBT) ♦ Adaptive chroma QP offset ♦ SDH ♦ Dependent quantization (DQ) ♦ Joint coding of chroma residuals (JCCR)
Entropy coding	<ul style="list-style-type: none"> ♦ Context-adaptive binary arithmetic coding (CABAC) ♦ Coefficient group ♦ Reverse diagonal, horizontal and vertical coefficient scan 	<ul style="list-style-type: none"> ♦ CABAC with multi-hypothesis probability estimates ♦ Additional coefficient group size ♦ Reverse diagonal coefficient scan only ♦ Improved probability model sections for absolute transform coefficient levels
Loop filtering	<ul style="list-style-type: none"> ♦ Deblocking filter (DF) ♦ Sample adaptive offset (SAO) 	<ul style="list-style-type: none"> ♦ Luma mapping with chroma scaling (LMCS) ♦ Deblocking boundary handling modifications ♦ Deblocking long filter ♦ Luma-adaptive deblocking ♦ Sample adaptive offset (SAO) ♦ Adaptive loop filter (ALF) ♦ Cross-Component ALF (CC-ALF)
Parallelization	<ul style="list-style-type: none"> ♦ Slices ♦ Tiles ♦ WPP with CTU row delay of two CTUs 	<ul style="list-style-type: none"> ♦ Slices ♦ Tiles ♦ Subpictures ♦ WPP with CTU row delay of one CTU

with B-frames. Each B-frame has 2-5 reference frames depending on its layer in a GOP.

Under the LB condition, both VTM and HM encoders used a three-layer hierarchical coding structure (L1 ... L3) with a GOP size of eight, as shown in Table 4. Only the first frame of the sequence is an I-frame, and the others are B-frames (I0, B1, B2, ...) with four reference frames. All frames were

coded in display order. The LB GOP also used the QP offset scaling.

C. QUALITY METRICS

The coding efficiencies of the VTM and HM codecs were compared using the well-known BD-rate evaluation

TABLE 3. Test sequences.

Class	Format	Sequence	Frame count	Frame rate	Bit depth	Test set RD	Cplx
A1	3840×2160 (2160p)	Campfire	300	30 fps	10	×	× ^H
		FoodMarket4	300	60 fps	10	×	× ^L
		Tango2	294	60 fps	10	×	×
A2	3840×2160 (2160p)	CatRobot	300	60 fps	10	×	×
		DaylightRoad2	300	60 fps	10	×	×
		ParkRunning3	300	50 fps	10	×	×
B	1920×1080 (1080p)	BasketballDrive	500	50 fps	8	×	× ^H
		BQTerrace	600	60 fps	8	×	×
		Cactus	500	50 fps	8	×	×
		MarketPlace	600	60 fps	10	×	×
		RitualDance	600	60 fps	10	×	× ^L
C	832×480 (480p)	BasketballDrill	500	50 fps	8	×	×
		BQMall	600	60 fps	8	×	× ^L
		PartyScene	500	50 fps	8	×	×
		RaceHorses	300	30 fps	8	×	× ^H
D	416×240 (240p)	BasketballPass	500	50 fps	8	×	× ^L
		BlowingBubbles	500	50 fps	8	×	×
		BQSquare	600	60 fps	8	×	×
		RaceHorses	300	30 fps	8	×	× ^H
E	1280×720 (720p)	FourPeople	600	60 fps	8	×	× ^H
		Johnny	600	60 fps	8	×	× ^L
		KristenAndSara	600	60 fps	8	×	×
UVG Dataset	3840×2160 (2160p)	Beauty	600	120 fps	10	×	×
		Bosphorus	600	120 fps	10	×	×
		HoneyBee	600	120 fps	10	×	×
		Jockey	600	120 fps	10	×	×
		Lips	600	120 fps	10	×	×
		ReadySetGo	600	120 fps	10	×	×
		ShakeNDry	300	120 fps	10	×	×
YachtRide	600	120 fps	10	×	×		

H = The highest complexity in the class
L = The lowest complexity in the class

TABLE 4. Hierarchical coding structures of the RA and LB conditions.

RA	Type	1	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	
	Cod. order	0	5	4	6	3	8	7	9	2	12	11	13	10	15	14	16	1
	Layer	L5	L4	L5	L3	L5	L4	L5	L2	L5	L4	L5	L3	L5	L4	L5	L1	
LB	Type	1	B	B	B	B	B	B	B									
	Cod. order	0	1	2	3	4	5	6	7	8								
	Layer	L3	L2	L3	L2	L3	L2	L3	L1									
	QP offset	-1	+5	+4	+5	+4	+5	+4	+5	+1	+6	+5	+6	+4	+6	+5	+6	

method [13], [14] that computes average bitrate differences for the same quality. In this paper, HM was used as an anchor for the BD-rate calculations, so negative values imply better coding efficiency for VTM over that of HM. In practice, the average difference between the RD curves of VTM and HM was interpolated per sequence with piecewise cubic interpolation through RD points of four base QP values: 22, 27, 32, and 37 (see Fig. 2). In our comparison, BD-rate is computed with three objective image quality metrics: 1) PSNR, 2) SSIM [15], and 3) VMAF [16]. VMAF score partially depends on surrounding frames, so it is not reported for the AI condition where only every eighth frame was encoded as defined in VTM CTC. Although subjective quality assessments such as the mean opinion score (MOS) tend to be considered as the most reliable indicators of perceived

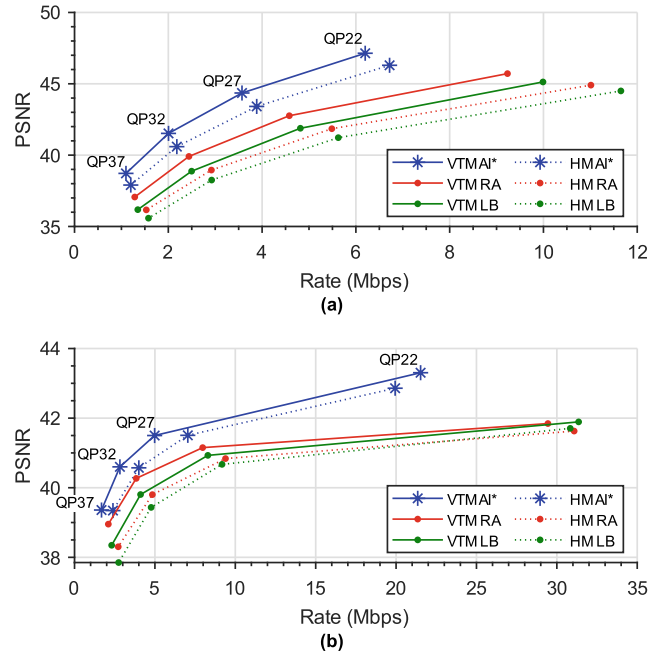


FIGURE 2. PSNR RD curves. (a) RitualDance (1920 × 1080). (b) Tango (4096 × 2160). (*) The actual AI coding rate is eight times as high due to sequence subsampling.

TABLE 5. Profiling platform for complexity analysis.

Processor	Intel Xeon E5-2699 v4 22-core (3.60 GHz)
Cache	55 MB Intel® Smart Cache
SSD	Intel 660p 2TB (NVME M.2)
RAM	64GB 2400MHz DDR4 ECC
Compiler	Microsoft Visual Studio 2019
Operating system	64-bit Microsoft Windows 10 Professional

media quality, they are cumbersome to organize. Therefore, our evaluation is focused on automatic and repeatable objective quality measures.

D. COMPLEXITY PROFILING SETUP

Our complexity profiling environment was composed of two identical Intel Xeon E5-2699 v4 22-core processors detailed in Table 5. The profiling was performed with Intel VTune Profiler [17], which is able to quantify the complexity of each encoder and decoder function in CPU cycles. For a reliable complexity analysis, a codec under test was the only software running at the time. Furthermore, the function-level profiling does not only monitor the number of function calls but also their internal complexities.

The complexity distributions were reported by categorizing all functions into the main encoding and decoding stages according to their functionality and function call hierarchy. However, a part of the functions cannot be assigned to a single category because they are called by different functions or they do not unambiguously belong to any specific category. Therefore, they were allocated to several categories

TABLE 6. BD-rates of VTM over HM for the same PSNR, SSIM, and VMAF values under the AI, RA, and LB conditions.

Class	Format	Sequence	All Intra (AI)		Random Access (RA)			Low Delay B (LB)		
			PSNR BD-rate	SSIM BD-rate	PSNR BD-rate	SSIM BD-rate	VMAF BD-rate	PSNR BD-rate	SSIM BD-rate	VMAF BD-rate
A1	3840×2160 (2160p)	Campfire	-30.9%	-37.0%	-37.1%	-43.8%	-42.9%	-34.7%	-39.9%	-39.4%
		FoodMarket4	-28.0%	-28.9%	-37.6%	-42.8%	-41.7%	-28.3%	-33.7%	-30.3%
		4096×2160 Tango2	-29.4%	-30.4%	-39.5%	-44.9%	-42.2%	-27.4%	-31.8%	-27.8%
A2	3840×2160 (2160p)	CatRobot	-28.9%	-30.5%	-45.8%	-48.4%	-51.8%	-37.3%	-39.3%	-38.0%
		DaylightRoad2	-27.5%	-25.0%	-44.5%	-48.6%	-51.2%	-32.7%	-36.4%	-31.5%
		ParkRunning3	-18.6%	-23.6%	-33.1%	-41.9%	-44.8%	-27.4%	-33.8%	-32.5%
B	1920×1080 (1080p)	BasketballDrive	-24.8%	-26.5%	-36.8%	-40.7%	-37.3%	-31.0%	-34.7%	-28.2%
		BQTerrace	-20.3%	-20.1%	-37.1%	-44.7%	-46.8%	-30.4%	-36.3%	-36.2%
		Cactus	-22.8%	-23.9%	-38.2%	-41.7%	-40.5%	-32.4%	-34.7%	-34.0%
		MarketPlace	-20.5%	-22.6%	-35.9%	-40.2%	-41.5%	-27.9%	-31.1%	-29.5%
		RitualDance	-23.8%	-25.3%	-32.0%	-37.2%	-31.9%	-25.8%	-28.9%	-24.0%
C	832×480 (480p)	BasketballDrill	-33.1%	-34.0%	-33.6%	-38.0%	-28.7%	-31.2%	-32.5%	-24.2%
		BQMall	-22.0%	-23.7%	-31.2%	-35.2%	-33.6%	-24.7%	-26.3%	-24.7%
		PartyScene	-16.2%	-18.9%	-28.4%	-32.0%	-31.1%	-23.3%	-23.6%	-26.0%
		RaceHorses	-17.4%	-15.5%	-26.9%	-27.7%	-28.4%	-21.7%	-21.0%	-22.7%
D	416×240 (240p)	BasketballPass	-20.9%	-21.7%	-26.2%	-29.9%	-26.1%	-21.7%	-22.8%	-21.2%
		BlowingBubbles	-15.7%	-15.9%	-24.2%	-26.7%	-27.3%	-20.2%	-18.3%	-22.9%
		BQSquare	-14.9%	-16.9%	-34.0%	-32.5%	-39.9%	-31.0%	-31.8%	-31.9%
		RaceHorses	-18.6%	-17.3%	-24.3%	-24.5%	-25.2%	-20.6%	-19.1%	-20.1%
E	1280×720 (720p)	FourPeople	-25.0%	-26.5%	-32.4%	-36.0%	-31.3%	-25.7%	-29.6%	-23.2%
		Johnny	-26.5%	-27.4%	-33.7%	-36.6%	-29.7%	-30.3%	-33.5%	-27.3%
		KristenAndSara	-25.3%	-25.8%	-33.2%	-35.7%	-33.1%	-28.4%	-30.8%	-27.5%
UVG Dataset	3840×2160 (2160p)	Beauty	-13.5%	-10.9%	-25.2%	-21.5%	-16.5%	-19.2%	-14.2%	-10.6%
		Bosphorus	-27.8%	-32.1%	-28.6%	-35.3%	-30.7%	-21.6%	-26.9%	-16.6%
		HoneyBee	-22.4%	-19.3%	-41.9%	-48.7%	-44.1%	-35.4%	-39.4%	-40.5%
		Jockey	-25.1%	-19.9%	-47.7%	-52.2%	-43.0%	-33.9%	-38.2%	-29.2%
		Lips	-13.8%	-10.9%	-29.0%	-25.4%	-20.9%	-23.5%	-18.5%	-17.9%
		ReadySetGo	-27.5%	-29.9%	-34.4%	-38.7%	-36.3%	-28.8%	-33.3%	-29.8%
		ShakeNDry	-22.8%	-22.2%	-23.8%	-29.1%	-17.7%	-18.9%	-24.8%	-12.7%
YachtRide	-31.2%	-37.7%	-30.5%	-38.5%	-27.9%	-24.2%	-30.1%	-20.5%		
Average		Class A	-27.2%	-29.2%	-39.6%	-45.1%	-45.8%	-31.3%	-35.8%	-33.2%
		Class B	-22.4%	-23.7%	-36.0%	-40.9%	-39.6%	-29.5%	-33.1%	-30.4%
		Class C	-22.2%	-23.0%	-30.0%	-33.2%	-30.5%	-25.2%	-25.8%	-24.4%
		Class D	-17.5%	-18.0%	-27.2%	-28.4%	-29.6%	-23.4%	-23.0%	-24.0%
		Class E	-25.6%	-26.6%	-33.1%	-36.1%	-31.4%	-28.1%	-31.3%	-26.0%
		UVG Dataset	-23.0%	-22.9%	-32.7%	-36.2%	-29.6%	-25.7%	-28.2%	-22.2%
Total Average			-23.0%	-23.9%	-33.1%	-36.6%	-34.4%	-27.2%	-29.5%	-26.7%

by calculating their relative shares from call hierarchy trees created by VTune Profiler.

The codecs were benchmarked with the SIMD optimizations that were enabled in the default configurations. This approach favors VTM over HM because SIMD-optimized functions account for a larger relative share in VTM, e.g., enabling them in VTM5.0 decreases the encoding time by a third [23]. Nevertheless, our approach follows that of JVET [26]. Furthermore, executing our massive test set took more than 1650 days of CPU time even with the chosen optimizations, so it was considered reasonable to keep them on.

V. COMPARATIVE RD ANALYSIS OF VTM AND HM

Table 6 tabulates the BD-rates of VTM over HM for our entire test set under the AI, RA, and LB conditions. The BD-rate results are given with the PSNR, SSIM, and VMAF metrics.

A. RD COMPARISON OF THE VTM AND HM CODECS

VTM is shown to achieve an average BD-rate improvement of 23.0% and 23.9% with the PSNR and SSIM metrics under

the AI condition, respectively. The corresponding sequence-specific variations are 13.5%–33.1% and 10.9%–37.7%. Under the RA condition, the average BD-rate increases to 33.1% (23.8%–47.7%) for PSNR and 36.6% (21.5%–52.2%) for SSIM. The results are also consistent with VMAF: 34.4% (16.5%–51.8%). Correspondingly, the coding gains under the LB condition are 27.2% (18.9%–37.3%) for PSNR, 29.5% (14.2%–39.9%) for SSIM, and 26.7% (10.6%–40.5%) for VMAF. One should note that the bit rate savings of VTM are more limited with the *Beauty* and *Lips* sequences than with other 2160p sequences because their noisy dominant black backgrounds introduce non-redundancy that is difficult to compress.

Our previous study [27] reported 23%, 35%, and 40% PSNR BD-rate gains for HEVC over AVC under the AI, RA, and LB conditions, respectively. The results here, with a more versatile test set though, verify that VVC continues to improve coding efficiency close to the rate of its predecessors. The relative progress is consistent under the AI condition but around 2 and 13 percentage points lower under the RA and LB conditions, respectively.

Fig. 2(a) and Fig. 2(b) plot the PSNR RD curves of the VTM and HM codecs for the *RitualDance* and *Tango* sequences, respectively. Solid curves represent the VTM results and dotted curves HM results under the AI, RA, and LB conditions marked in blue, red, and green, respectively. The corresponding BD-rates are highlighted in gray in Table 6. Only the PSNR RD curves are presented since the SSIM and VMAF curves behave similarly. In addition, the reported AI coding rate is for the subsampled sequence, i.e., the actual bit rate in AI coding is eight times as high.

The RD curves plotted for the *RitualDance* sequence represent most of the cases where VTM improves both coding efficiency and quality on each QP value. In addition, the relationship between the AI, RA, and LB conditions tends to remain the same in all cases. However, occasional irregularities were found. For example, with the *Tango* sequence VTM outputs more bits than HM at QP 22 under the AI and LB conditions.

B. COMPARISON AS A FUNCTION OF RESOLUTION

The results in Table 6 confirm that VVC excels at higher resolutions. Indeed, new features of VVC, such as larger CTUs, are particularly introduced to provide bit rate savings for high-resolution sequences.

TABLE 7. BD-rate of VTM over HM as a function of resolution.

Format	All Intra (AI)		Random Access (RA)			Low Delay B (LB)			
	PSNR	SSIM	PSNR	SSIM	VMAF	PSNR	SSIM	VMAF	
	BD-rate (%)		BD-rate (%)			BD-rate (%)			
CatRobot	2160p	-28.9	-30.5	-45.8	-48.4	-51.8	-37.3	-39.3	-38.0
	1440p	-27.9	-30.1	-43.1	-45.8	-45.8	-35.9	-37.7	-36.2
	1080p	-26.7	-28.8	-41.9	-44.0	-44.8	-35.1	-36.7	-36.0
	720p	-24.4	-26.1	-39.4	-41.0	-43.0	-33.3	-34.2	-35.3
	480p	-22.1	-23.4	-36.8	-38.2	-41.1	-31.1	-31.6	-34.2
	240p	-19.7	-19.9	-30.9	-31.8	-36.4	-26.3	-26.3	-31.1
ReadySetGo	2160p	-27.5	-29.9	-34.4	-38.7	-36.3	-28.8	-33.3	-29.8
	1440p	-27.0	-28.8	-32.8	-36.8	-34.0	-27.7	-31.3	-27.1
	1080p	-26.4	-28.3	-32.1	-36.2	-33.5	-27.0	-30.2	-26.3
	720p	-24.5	-26.8	-29.9	-33.7	-32.5	-25.3	-28.1	-24.5
	480p	-22.3	-25.0	-28.0	-31.7	-31.2	-23.5	-26.0	-23.6
	240p	-19.2	-21.4	-24.5	-27.1	-29.6	-20.9	-22.9	-22.1

This aspect is more carefully considered in Table 7 that reports the BD-rates of VTM over HM as a function of resolution. The benchmarked *CatRobot* and *ReadySetGo* sequences were downsampled from the original 2160p resolution to 2560×1440 (1440p), 1080p, 720p, 480p, and 240p formats by using the bilinear interpolation filter in *FFmpeg* [31].

Our previous study [27] showed that increasing resolution favors HEVC over AVC and the same trend continues between VVC and HEVC. In most cases, there is a logarithmic relationship between BD-rate values and resolution. The largest deviation to this relation can be found between 2160p and 1440p resolutions, where the gains of VTM are higher.

This particularly holds for the VMAF results under the RA condition.

VI. ENCODING COMPLEXITY ANALYSIS

Our complexity analysis is carried out by dividing the encoder functions into six encoding tool categories (see Section III): 1) *Entropy coding (EC)*; 2) *Forward/inverse transform and quantization (TR/Q)*; 3) *Intra prediction (IP)*; 4) *Motion estimation and compensation (ME/MC)*; 5) *Loop filtering (LF)*; and 6) *Miscellaneous (Misc.)*.

This categorization divides VTM and HM into logical and consistent entities that cannot be further divided into meaningful subcategories because VTM has many tools not found in HM. Additionally, the functions of VTM and HM do not always follow the single-responsibility principle which complicates categorization. The Misc. category contains the functions, such as high-level control-logic and memory management, that cannot be allocated to any other category.

A. COMPLEXITY ANALYSIS OF THE VTM ENCODER

Table 8 tabulates the complexity results of the VTM encoder for the base QP values under the AI, RA, and LB conditions. The results include the relative complexity shares between the six encoding tool categories and the absolute complexities in *thousand cycles per pixel (kcpp)*. This allows us to fairly compare sequences with different frame rates and resolutions. In each column, the highest and the lowest relative shares per QP are colored red and green, respectively.

Overall, the absolute complexity is inversely proportional to the QP value and depends on both the resolution and the content of the sequence. Furthermore, the cycle count per pixel is inversely proportional to the resolution because the smaller resolutions tend to be split into smaller CUs whose processing time is relatively higher. The new QT/MTT partitioning scheme was shown to have the largest effect on the intra coding complexity [32], and the same can be assumed for inter coding because the QT/MTT partitioning increases the plurality of blocks and thereby complexity. However, the block partitioning overhead is distributed among the encoding tools and cannot be extracted from Table 8.

Fig. 3(a) depicts the absolute complexity in cycle counts per pixel for each encoding tool category of VTM. The results are averaged across the profiled sequences and given for each base QP value under the AI, RA, and LB conditions.

Under the AI condition, the absolute complexity and relative share of the EC correlate with those of TR/Q as a function of the QP value. Incrementing the QP value decreases the number of non-zero coefficients after quantization and thereby the number of encoded symbols.

Correspondingly, increasing the QP value indirectly degrades the absolute complexity of IP. The higher the number of zero coefficients, the more all-zero blocks are chosen. All-zero blocks trigger the termination mechanisms that reduce the number of QT/MTT [29] splitting options. The relative share of IP still increases with the QP value, since

TABLE 8. Complexity breakdown of the VTM encoder under the AI, RA, and LB conditions.

Class (Format)	Sequence (Complexity)	QP	All Intra (AI)							Random Access (RA)							Low Delay B (LB)						
			EC (%)	TR/Q (%)	IP (%)	ME/MC (%)	LF (%)	Misc (%)	Overall (kepp)	EC (%)	TR/Q (%)	IP (%)	ME/MC (%)	LF (%)	Misc (%)	Overall (kepp)	EC (%)	TR/Q (%)	IP (%)	ME/MC (%)	LF (%)	Misc (%)	Overall (kepp)
A1/A2 (2160p)	Campfire (H)	22	22.3	39.3	32.3	0.0	1.2	5.0	595	15.3	26.5	15.0	38.3	0.7	4.3	1123	16.0	25.6	6.5	46.3	0.9	4.8	931
		27	20.5	37.8	34.8	0.0	1.8	5.1	473	11.5	21.3	15.6	45.8	1.3	4.4	558	10.8	20.1	7.2	55.4	1.7	4.9	467
		32	17.6	33.2	39.9	0.0	3.3	6.0	365	9.6	17.9	15.5	50.7	2.0	4.3	355	9.0	16.6	7.3	60.0	2.5	4.7	296
	FoodMarket (L)	37	16.0	28.4	42.1	0.0	6.5	7.0	184	7.8	15.8	14.7	54.5	3.1	4.1	233	7.4	15.0	7.1	62.3	3.6	4.5	185
		22	18.5	33.7	40.3	0.0	2.1	5.4	473	7.9	15.6	13.5	58.2	1.5	3.4	470	6.4	13.3	5.6	69.7	1.6	3.4	477
		27	17.3	31.9	40.8	0.0	4.2	5.8	279	6.0	12.5	12.3	63.3	2.6	3.3	281	4.8	10.7	5.4	73.5	2.6	3.1	279
B (1080p)	BasketballDrive (H)	32	15.9	29.5	41.4	0.0	6.8	6.5	170	4.7	10.7	10.7	66.4	4.3	3.3	180	3.6	8.9	4.5	75.2	4.6	3.1	179
		37	13.2	27.8	38.4	0.0	12.8	7.7	89	3.6	9.6	9.3	65.6	7.9	3.9	113	2.7	7.6	3.9	74.2	8.0	3.5	120
		22	21.0	38.6	34.5	0.0	1.1	4.8	1142	13.0	22.3	13.9	45.6	1.2	4.0	789	11.7	19.9	5.5	57.3	1.3	4.3	787
	RitualDance (L)	27	19.3	34.9	38.9	0.0	1.7	5.2	778	10.1	17.5	13.7	53.0	1.9	3.9	460	8.5	14.5	5.5	65.1	2.2	4.2	430
		32	18.0	30.6	42.8	0.0	2.9	5.7	446	8.1	14.9	13.0	57.6	2.8	3.7	287	6.6	11.8	5.2	69.7	3.0	3.8	287
		37	16.2	29.6	42.4	0.0	5.7	6.1	227	6.5	13.1	11.7	60.8	4.5	3.4	190	5.3	10.5	4.8	71.0	4.9	3.5	196
E (720p)	FourPeople (H)	22	20.9	33.7	38.4	0.0	1.7	5.3	787	10.8	17.3	13.9	52.3	1.7	4.0	549	9.3	14.4	5.9	64.2	1.8	4.4	551
		27	19.6	32.8	39.6	0.0	2.6	5.3	501	9.0	15.5	13.6	55.7	2.2	3.8	359	7.7	12.8	6.0	67.2	2.4	3.9	358
		32	18.4	30.5	41.8	0.0	3.7	5.7	345	7.2	13.9	12.3	59.6	3.4	3.5	239	5.9	11.1	5.3	70.5	3.8	3.5	251
	Johnny (L)	37	16.3	29.3	41.9	0.0	6.4	6.1	198	5.9	12.9	11.5	61.0	5.4	3.4	173	4.8	10.2	5.1	71.0	5.5	3.4	197
		22	20.4	35.6	37.1	0.0	1.9	5.0	762	8.3	14.8	9.6	58.7	5.2	3.3	269	6.5	12.5	3.8	67.9	5.8	3.5	269
		27	19.8	33.5	38.3	0.0	3.2	5.2	462	5.5	10.3	8.2	64.8	8.0	3.2	190	3.5	7.1	2.9	74.3	9.1	3.1	191
C (480p)	RaceHorses (H)	32	18.3	30.4	41.3	0.0	4.5	5.5	310	4.9	9.5	9.2	60.7	11.6	4.0	109	3.1	6.2	3.2	72.0	11.8	3.6	117
		37	16.4	29.8	40.3	0.0	7.8	5.8	187	3.3	6.9	6.8	60.7	17.9	4.5	71	1.7	4.1	2.2	70.1	18.0	4.0	74
		22	22.7	37.0	34.1	0.0	1.2	4.9	1445	16.6	22.0	14.5	41.4	1.3	4.2	1244	15.9	19.4	6.0	51.9	1.8	5.0	1241
	BQMall (L)	27	21.7	35.3	36.1	0.0	1.7	5.1	1103	12.4	18.4	12.9	50.3	2.1	3.9	1244	11.7	15.9	5.5	59.1	3.0	4.7	1184
		32	20.7	33.7	38.1	0.0	2.3	5.2	838	9.6	16.0	12.1	55.9	2.9	3.6	977	8.9	14.1	5.1	63.7	4.2	4.1	844
		37	19.1	32.7	39.7	0.0	3.2	5.3	584	7.6	14.0	11.4	59.9	3.8	3.3	603	7.0	12.2	4.8	66.3	5.9	3.7	534
D (240p)	RaceHorses (H)	22	21.9	37.4	34.4	0.0	1.4	5.0	1410	13.5	20.7	12.1	46.6	3.1	4.0	622	12.0	17.6	5.0	57.5	3.5	4.4	621
		27	21.2	34.9	36.9	0.0	1.9	5.1	1005	10.8	16.8	11.7	52.4	4.6	3.8	622	8.9	13.6	4.7	63.4	5.3	4.0	621
		32	20.1	32.8	39.3	0.0	2.5	5.3	749	8.9	14.6	11.1	56.4	5.3	3.6	434	7.0	11.4	4.4	66.9	6.7	3.7	471
	BasketballPass (L)	37	18.6	31.4	40.9	0.0	3.7	5.4	513	7.2	13.0	10.7	59.4	6.3	3.5	281	5.4	9.7	4.1	70.2	7.2	3.4	306
		22	23.4	36.0	33.5	0.0	2.1	5.1	1538	15.6	20.6	13.8	42.2	3.5	4.3	1878	14.9	18.1	5.7	51.3	4.9	5.0	1586
		27	22.1	34.9	35.3	0.0	2.7	5.1	1259	12.6	17.7	12.9	48.6	4.0	4.1	1381	11.5	15.1	5.3	55.5	7.9	4.8	1118
BasketballPass (L)	32	20.5	33.0	37.4	0.0	3.5	5.5	1064	10.0	15.6	12.0	54.4	4.2	3.8	1000	8.5	13.0	4.8	61.2	8.3	4.1	894	
	37	18.6	30.5	40.0	0.0	5.4	5.6	743	7.9	13.6	11.4	58.4	5.2	3.5	633	6.8	11.2	4.6	65.1	8.4	3.8	536	
	22	22.3	34.7	35.3	0.0	2.6	5.1	1432	15.7	21.0	13.9	40.3	4.6	4.5	1506	14.2	17.6	5.8	51.4	6.0	4.9	1298	
	27	20.9	33.5	37.0	0.0	3.5	5.1	1114	12.7	18.4	13.4	45.3	5.9	4.2	1101	11.3	14.9	5.4	56.2	7.5	4.6	1008	
BasketballPass (L)	32	19.4	31.1	39.3	0.0	4.9	5.3	839	10.5	16.7	12.8	50.1	5.8	4.0	776	8.7	13.0	5.0	59.8	9.3	4.1	810	
	37	17.8	29.4	39.8	0.0	7.5	5.5	559	8.7	15.5	12.3	53.1	6.6	3.8	493	7.1	11.5	4.9	64.0	8.5	4.0	504	

the QP value directly affects the absolute complexities of EC and TR/Q.

Unlike the other categories, the absolute complexity of LF stays practically the same for each QP value and thus the relative share increases with the QP value. The absolute complexity of LF is related to the resolution and is independent of the content.

The small shares of ME/MC in AI coding stem from unnecessarily initializations in VTM and could be optimized out.

Under the RA and LB conditions, the absolute coding complexities are practically on a par with each other. However, when compared with AI coding, their overall absolute complexity ratios are between 0.5x and 4x. The range is smaller with higher QP values. The largest complexity increase falls on the *Campfire* sequence, which is mainly intra coded. Conversely, the largest decrease is found in the

class E sequences because inter prediction dominates their coding.

EC and TR/Q are coupled as in the AI case, but the role of IP is much smaller, because it is skipped with B-frames whenever the result of ME/MC is accurate enough.

The absolute complexity of IP is three times as high in the RA case because only the first frame is intra coded in the LB case. Additionally, ME is not as effective for the lower layer frames in RA coding since there are large temporal gaps between compared frames.

The classes with smaller sequences have higher relative complexity in ME/MC because the search is performed for smaller CUs and in turn for a larger number of CUs. Conversely, when ignoring the effect of the content, the higher resolution sequences tend to have larger relative shares of ME/MC, because the movements in pixels are larger.

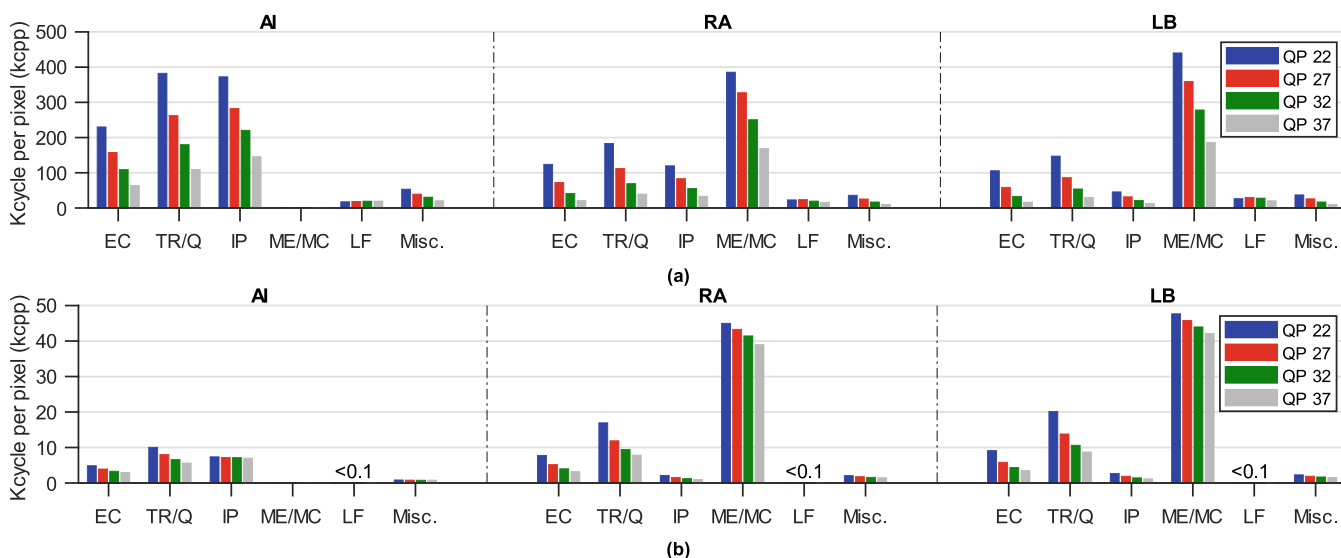


FIGURE 3. Absolute complexities in cycles per pixel under the AI, RA, and LB conditions. (a) VTM encoder. (b) HM encoder. Results averaged across the sequences for each QP value.

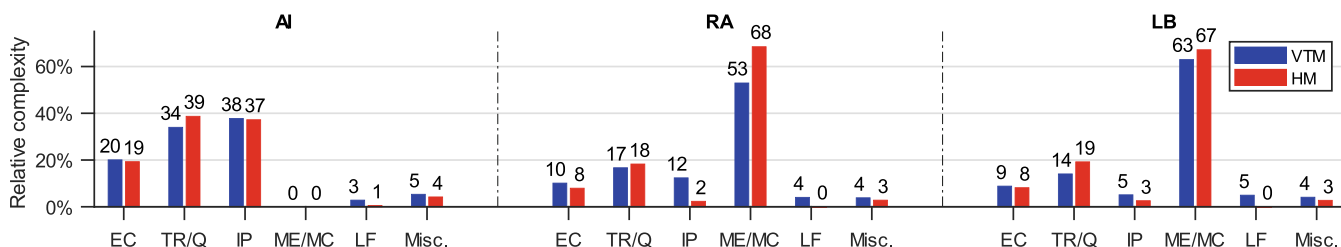


FIGURE 4. Relative complexities of the VTM and HM encoders under the AI, RA, and LB conditions. Results averaged across QP values and sequences.

B. COMPLEXITY COMPARISON OF THE VTM AND HM ENCODERS

Fig. 3(b) replicates the bar diagrams of Fig. 3(a) for the HM encoder with three main observations: 1) the complexity overhead of VTM is much evident under the AI condition; 2) the QP value has much higher impact on the complexity of VTM; and 3) the QP value has particularly higher effect on the complexity of IP and ME/MC in VTM, mostly due to pronounced role of early termination mechanisms [29].

Fig. 4 shows the average complexity shares of the encoding tool categories in VTM and HM across all base QP values and sequences. Under all test conditions, the introduction of ALF [29] has increased the absolute complexity of LF over hundredfold. Despite that new transform types elevate the absolute complexity of TR/Q by an order of magnitude in VTM, the relative share of TR/Q is still smaller than in HM. Similarly, the relative share of ME/MC decreases in VTM even though the numerous new inter coding tools bring around tenfold complexity. IP is the only category whose relative share decreases in AI coding but increases otherwise. Nevertheless, the absolute complexity of IP is 20- to 30-fold in VTM, depending on the condition. This becomes apparent

when comparing the complexity shares of IP under the RA condition. In fact, about 15% of the total RA coding overhead of VTM comes from IP.

Fig. 5(a)–(c) present the absolute encoding complexities of VTM (in blue) and HM (in red) as a function of resolution under the AI, RA, and LB conditions, respectively. The dashed lines are plotted using the average complexities of the sequences for QP 32. The results are similar for the other QP values. The lines are annotated with average complexity ratios between VTM and HM.

Our results show that the complexities of both encoders have a linear relationship with the resolution. The complexity drop with the 720p resolution is due to the content specificity of the class E sequences. The colored regions around the lines reflect the deviations of the individual complexity results. An almost unnoticeable red region around the HM curve indicates that the complexity of VTM varies more than that of HM. For 2160p sequences, the complexity of VTM varies around the average by $\pm 10\%$ under the AI condition, $\pm 30\%$ under the RA condition, and $\pm 20\%$ under the LB condition. This variation comes mainly from the termination mechanisms of the QT/MTT splitting process [29] which correlates with the content.

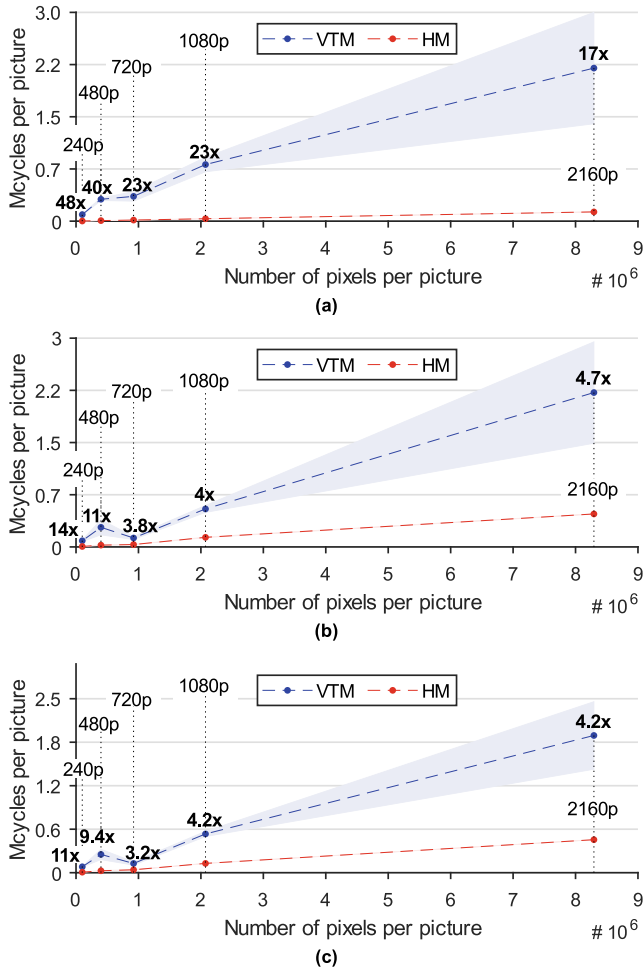


FIGURE 5. Encoding complexity as a function of resolution for QP 32. Results averaged across the sequences of the same resolution. Blue and red areas show the variation between minimum and maximum complexities. (a) AI condition. (b) RA condition. (c) LB condition.

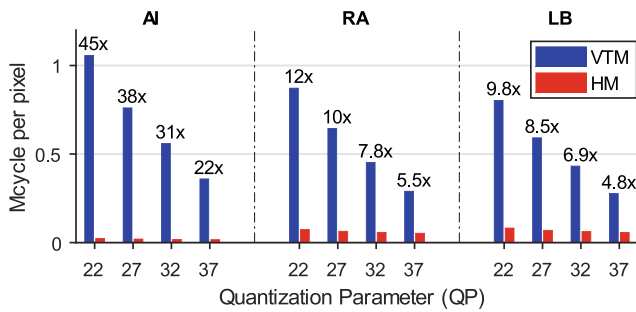


FIGURE 6. Absolute complexities in cycles per pixel and ratios of the VTM and HM encoders under the AI, RA, and LB conditions averaged across sequences.

Fig. 6 depicts the absolute encoding complexities of VTM and HM for each QP value under the AI, RA, and LB conditions. The results are averaged across all sequences. The blue bars are annotated with the average complexity ratios between VTM and HM. The encoder complexities and the gap between them decrease as the QP value increases.

On average, the complexity of VTM is $34.1\times$ that of HM under the AI condition with a QP-specific variation of $21.8\times-45.4\times$. The respective metrics are $8.8\times$ ($5.5\times-11.8\times$) under the RA condition and $7.5\times$ ($4.8\times-9.8\times$) under the LB condition. The highest gap between VTM and HM exists in AI coding because VTM introduces many new directional IP modes and intra coding tools such as CCLM, PDPC, MRL, and ISP (see Table 2). All in all, a comparison with our previous study [27] reveals that the complexity increase is much higher than that between HEVC and AVC.

VII. DECODING COMPLEXITY ANALYSIS

The decoder functions are respectively divided into six decoding tool categories: 1) *Entropy decoding (ED)*; 2) *Inverse quantization and transform (IQ/IT)*; 3) *Intra prediction (IP)*; 4) *Motion compensation (MC)*; 5) *Loop filtering (LF)*; and 6) *Miscellaneous (Misc.)*.

A. COMPLEXITY ANALYSIS OF THE VTM DECODER

Table 9 tabulates the complexity results of the VTM decoder for the same sequences as with the encoder. As in Table 8, the lowest and the highest relative shares per QP are colored in green and red in each column, respectively. Additionally, Fig. 7(a) depicts the absolute complexities of the decoding tool categories in *cycles per pixel (cpp)* for each base QP value under the AI, RA, and LB conditions.

In general, the QP value has the largest effect on the decoding complexity, but the impact is still smaller than with the encoder. Decoding is also dependent on both resolution and content. In particular, the class E sequences can be decoded with relatively low complexity.

Under the AI condition, ED is the most complex part of the decoder with small QP values, because most of the decoded symbols come from the quantized residual. As the QP value raises, the absolute complexity shrinks with the number of residual symbols. Additionally, the sequences with the highest relative EC complexities at the encoder side have also the highest relative shares in ED. However, contrary to the encoder, the complexity of IQ/IT does not correlate with that of ED, because of the absence of the *Rate Distortion Optimized Quantization (RDOQ)* [33] in the decoder.

The highest and lowest relative shares of IP also correlate with those of the encoder, although not as strongly. The complexity of IP depends on the number of CUs in the final CTU structure. With higher QP values, the CTU tends to be split into fewer CUs, explaining why the absolute complexity of IP decreases as the QP value increases.

LF is the least complex part of the encoder but it turns out to be the most compute-intensive part of the decoder with higher QP values. This is explained by the fact that, apart from ALF, the LF algorithms themselves require little iteration during encoding and thus almost the same operations are executed by the encoder and the decoder.

As for the encoder, the MC category should be ignored when considering the overall complexity.

TABLE 9. Complexity breakdown of the VTM decoder under the AI, RA, and LB conditions.

Class (Format)	Sequence (Complexity)	QP	All Intra (AI)							Random Access (RA)							Low Delay B (LB)						
			ED (%)	IQ/IT (%)	IP (%)	MC (%)	LF (%)	Misc (%)	Overall (Cpp)	ED (%)	IQ/IT (%)	IP (%)	MC (%)	LF (%)	Misc (%)	Overall (Cpp)	ED (%)	IQ/IT (%)	IP (%)	MC (%)	LF (%)	Misc (%)	Overall (Cpp)
A1/A2 (2160p)	Campfire (H)	22	32.7	7.4	12.0	1.0	28.9	17.8	531	20.4	7.6	7.7	15.0	30.3	18.9	340	23.2	7.7	6.5	13.7	30.3	18.7	360
		27	25.3	8.8	14.1	1.1	32.8	17.9	393	12.4	8.3	7.6	19.8	31.3	20.6	222	12.1	9.7	6.8	17.8	33.7	19.9	219
		32	17.2	10.2	16.1	1.2	36.6	18.8	276	10.5	7.2	7.3	25.3	32.4	17.3	177	10.2	8.3	6.9	20.3	32.8	21.6	176
		37	15.5	10.4	17.0	1.4	34.2	21.4	225	8.7	5.7	7.0	28.9	31.7	17.9	155	9.3	7.0	6.6	21.2	32.5	23.4	150
	FoodMarket (L)	22	19.3	15.0	14.2	1.1	35.5	14.9	354	7.8	9.3	2.7	36.3	28.8	15.0	213	7.9	12.1	1.9	32.3	32.0	13.7	224
		27	16.8	14.9	15.1	1.2	37.3	14.7	290	6.5	7.5	2.5	42.1	26.3	15.2	188	6.9	10.7	1.9	34.8	31.0	14.6	186
		32	13.5	14.9	15.9	1.2	39.5	15.0	252	5.6	6.0	2.3	45.7	25.0	15.5	169	6.4	9.3	1.9	35.1	31.6	15.7	161
		37	12.1	15.9	17.9	1.3	36.5	16.3	209	5.7	4.7	2.1	48.7	22.8	16.0	154	6.6	8.0	1.7	35.5	31.3	16.9	141
B (1080p)	BasketballDrive (H)	22	32.2	9.8	12.2	1.0	27.9	16.9	565	15.7	6.1	4.4	27.2	29.9	16.6	280	15.7	8.7	2.9	24.2	31.2	17.2	312
		27	22.8	11.6	14.0	1.0	33.6	16.9	400	10.4	5.5	3.9	35.0	28.9	16.3	210	11.2	8.4	2.9	28.8	31.2	17.5	224
		32	18.2	10.9	16.7	1.4	35.6	17.2	321	8.2	4.2	3.3	40.6	27.0	16.7	178	8.9	7.0	2.6	31.1	30.9	19.5	181
		37	17.3	11.8	16.0	1.1	33.5	20.3	250	6.9	3.5	3.0	45.1	25.6	16.0	158	7.9	6.3	2.3	32.7	29.9	20.8	155
	RitualDance (L)	22	25.4	12.2	15.3	1.1	31.4	14.6	432	12.0	6.9	4.5	31.7	29.6	15.2	243	12.0	8.4	3.6	28.5	33.2	14.3	266
		27	22.4	13.0	15.4	1.2	33.9	14.2	353	9.8	6.4	4.5	36.5	26.8	16.0	198	10.2	8.3	3.6	31.0	31.9	15.0	206
		32	18.2	13.2	16.1	1.1	36.0	15.4	287	8.2	5.4	4.3	40.1	25.0	17.0	170	9.3	7.7	3.6	31.4	31.7	16.3	167
		37	16.4	13.1	19.6	1.6	33.0	16.3	228	7.2	5.0	3.8	43.6	22.8	17.6	148	8.5	7.9	3.3	31.3	31.4	17.6	139
E (720p)	FourPeople (H)	22	26.6	11.8	15.3	0.9	29.7	15.6	488	10.2	4.6	2.5	37.6	25.6	19.4	150	12.0	6.0	1.4	29.3	30.0	21.3	143
		27	25.3	10.7	13.8	1.2	32.9	16.1	407	9.0	2.1	2.2	44.7	20.5	21.4	124	10.4	3.8	1.2	31.6	29.8	23.2	111
		32	23.0	8.6	15.4	1.1	34.7	17.1	341	7.3	1.8	1.9	47.8	18.8	22.5	115	9.0	2.2	1.4	32.3	29.1	26.1	95
		37	19.2	11.8	15.8	1.2	34.1	17.8	276	6.9	1.3	1.7	47.0	20.7	22.4	111	9.0	1.6	1.7	31.4	28.5	27.9	85
	Johnny (L)	22	24.9	12.6	13.3	1.1	30.7	17.4	369	9.2	2.8	1.5	41.3	22.7	22.5	137	10.8	4.4	1.0	32.0	25.2	26.5	125
		27	20.9	10.2	16.4	1.4	32.1	19.0	307	6.9	1.5	1.9	50.4	17.7	21.7	115	7.3	2.1	0.8	38.0	23.5	28.2	104
		32	17.9	9.8	17.2	1.5	34.5	19.1	258	5.9	0.9	1.5	52.8	15.4	23.4	109	6.5	1.3	0.9	37.7	23.8	29.8	92
		37	14.4	11.7	17.6	1.4	34.5	20.4	217	5.8	0.9	1.7	53.9	15.7	22.0	105	7.6	1.0	1.0	35.7	23.5	31.2	84
C (480p)	RaceHorses (H)	22	34.8	12.6	10.9	1.0	22.6	18.0	689	22.3	6.5	4.9	22.8	27.9	15.5	420	24.1	7.6	3.5	20.6	27.8	16.5	453
		27	30.2	8.0	13.1	1.1	25.9	21.7	568	17.3	4.8	5.3	27.8	28.8	16.1	299	16.4	7.3	4.2	24.6	31.6	15.9	309
		32	24.5	11.7	15.3	2.6	28.1	17.8	463	11.7	4.4	5.1	34.3	28.7	15.7	230	14.1	5.7	3.4	28.1	33.3	15.4	227
		37	24.9	11.8	15.0	2.5	27.8	18.0	372	11.2	3.4	4.2	37.2	26.8	17.3	183	10.6	5.3	3.0	31.4	33.5	16.2	177
	BQMall (L)	22	32.5	8.9	14.2	1.2	25.8	17.3	674	16.1	5.2	3.6	32.7	26.5	15.9	257	18.3	6.5	2.1	27.1	30.2	15.8	285
		27	31.0	9.5	14.3	1.4	27.9	15.9	539	14.2	2.9	2.7	38.6	25.6	16.0	198	14.4	5.6	2.0	31.7	29.6	16.7	206
		32	26.6	10.5	14.5	1.6	29.7	17.2	435	11.2	2.7	2.4	42.9	25.1	15.6	166	12.4	3.9	2.1	33.6	30.5	17.5	159
		37	23.9	9.4	16.5	1.3	28.5	20.3	351	9.1	1.6	2.2	46.3	23.0	17.8	144	9.4	3.5	2.0	36.5	31.0	17.7	133
D (240p)	RaceHorses (H)	22	39.1	11.9	9.7	0.9	22.3	16.1	840	22.8	4.8	4.9	26.2	25.0	16.3	454	23.7	5.5	3.8	22.0	27.3	17.6	477
		27	33.2	9.3	12.5	1.2	19.7	24.0	686	17.0	3.7	4.5	30.3	27.5	17.0	345	18.1	6.3	3.0	25.8	31.1	15.7	344
		32	29.7	8.5	12.9	1.2	25.2	22.3	573	13.4	4.4	4.8	31.2	27.6	18.7	267	14.1	4.2	3.7	28.7	31.1	18.2	251
		37	23.2	11.5	15.0	1.6	23.2	25.3	447	12.4	2.3	3.2	38.6	24.9	18.7	211	11.4	3.3	3.8	29.4	36.0	16.1	201
	BasketballPass (L)	22	30.5	7.8	15.2	1.1	23.9	21.5	734	21.8	5.2	5.1	26.7	25.9	15.2	363	23.0	5.8	2.8	22.0	28.3	18.1	394
		27	29.5	6.7	15.2	2.5	22.3	23.6	594	15.4	4.4	4.8	29.9	27.9	17.4	280	16.8	6.1	4.4	24.8	29.3	18.6	300
		32	28.5	9.8	14.2	0.6	26.1	20.8	480	15.0	2.5	4.5	34.0	26.1	17.9	222	15.3	4.0	3.9	28.0	31.4	17.5	227
		37	18.8	11.7	19.8	2.5	25.9	21.2	389	9.6	2.0	4.4	37.6	26.1	20.2	182	10.2	7.1	2.4	31.9	32.1	16.3	187

Under the RA and LB conditions, the absolute complexities of all common categories apart from Misc. are smaller than those of AI coding because the RA and LB coding efficiencies are higher and the decoder has less symbols to process. ED remains the most complex part of the decoder with small QP values.

As justified with the encoder, the sequences with higher relative IP share have lower relative MC share and vice versa. Conversely to the encoder, both the absolute and relative complexities of MC are slightly higher in the RA case because it introduces new biprediction tools that are more complex than the unidirectional prediction tools used in LB coding. As in AI coding, LF is the most complex part of the decoder in the LB case, whereas MC or LF has the highest complexity in the RA case, depending on the QP value.

B. COMPLEXITY COMPARISON OF THE VTM AND HM DECODERS

Fig. 7(b) replicates the bar diagrams of Fig. 7(a) for the HM decoder. The main observation is that the QP value has a similar effect on complexity for all decoding tool categories of VTM and HM.

Fig. 8 presents the complexity shares of the decoder tool categories in VTM and HM. The results are averaged across the base QP values and sequences. The complexity distributions between the different decoding tool categories remain similar between VTM and HM, except for Misc. and LF. The absolute complexities of the Misc. categories are close to each other because they are mainly composed of the similar writing operations of the decoded file. The absolute complexity increase of LF in VTM is due to the introduction of the new

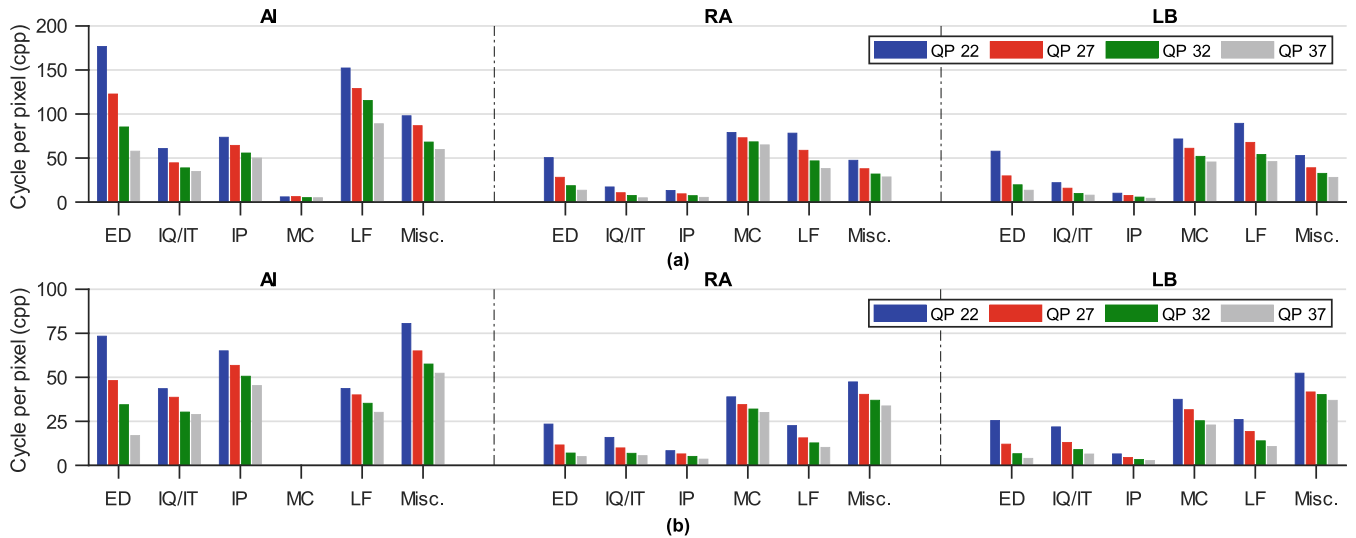


FIGURE 7. Absolute complexities in cycles per pixel under the AI, RA, and LB conditions. (a) VTM decoder. (b) HM decoder. Results averaged across the sequences for each QP value.

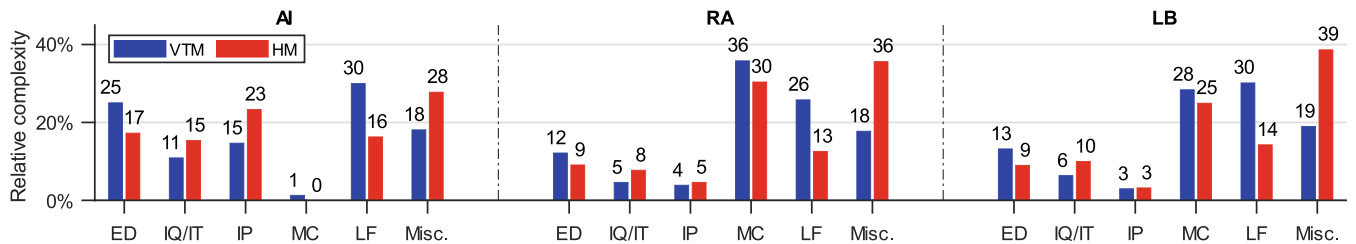


FIGURE 8. Relative complexities of the VTM and HM decoders under the AI, RA, and LB conditions. Results averaged across QP values and sequences.

ALF and LMCS filters, which also have a significant impact on the overall decoding complexity. In general, the differences between the VTM and HM decoders are consistent across the different conditions.

Fig. 9(a)–(c) show the absolute decoding complexities of VTM (in blue) and HM (in red) as a function of resolution under the AI, RA, and LB conditions, respectively. The notation is the same as that with the encoders in Fig. 5. Results show a linear relation between complexity and resolution in each case. However, as for the encoder side, the absolute complexity slightly drops for the 720p sequences due to the specificity of the class E. The blue and red areas denote a low variation between the minimum and maximum complexities. These results attest to the stability of the absolute complexity of the VTM and HM decoders.

Fig. 10 presents the absolute complexities of the VTM and HM decoders for each QP value under the AI, RA, and LB conditions. The results are averaged across all sequences. The blue bars are annotated with the average complexity ratios between VTM and HM. The complexity gap decreases as the QP value increases and the degradation is more significant under the AI condition due to lower coding efficiencies. On the other hand, the results show that the complexity

overhead of VTM is very stable, around $1.8\times$ under all test conditions.

With regard to our previous study [27], the ED complexity gap between the VVC and HEVC decoders is smaller than that with HEVC and AVC decoders. Otherwise, the decoder complexities behave similarly between these standards.

VIII. REAL-TIME HEVC AND VVC VIDEO CODING

In general, the complexity requirements of video coding are polarized between offline and real-time media applications. *Video on demand (VoD)* services such as YouTube, Netflix, TikTok, Amazon Prime Video, Hulu, and Bilibili prefer coding efficiency to speed since a majority of their operating expenditure comes from video delivery rather than compression. Even though numerous compression formats are supported, each of them needs to be encoded only once. There are also many powerful cloud services hosted, e.g., by AWS Elemental, Coconut, Qencod, and Zencod for third-party offline coding.

On the other end of the spectrum are live streaming, communication, and broadcasting applications such as Microsoft Teams, Zoom, Twitch, Google Hangouts Meet, Skype, Facebook Live, Instagram Live Stories, and Periscope, for which

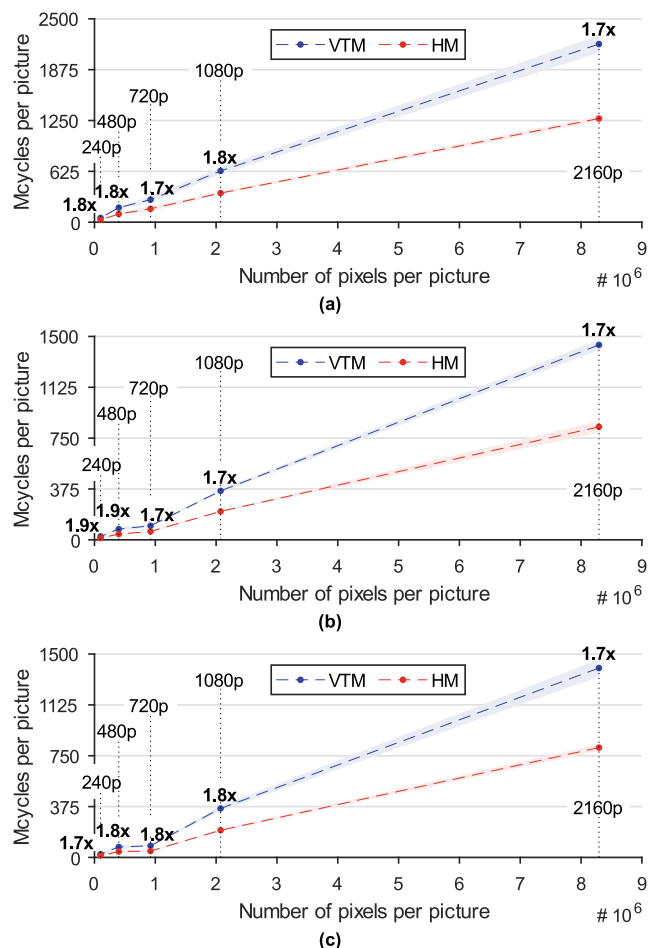


FIGURE 9. Decoding complexity as a function of resolution for QP 32. Results averaged across the sequences of the same resolution. Blue and red areas show the variation between minimum and maximum complexities. (a) AI condition. (b) RA condition. (c) LB condition.

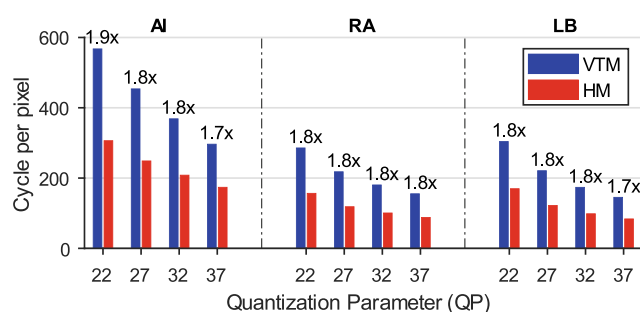


FIGURE 10. Absolute complexities in cycles per pixel and ratios of the VTM and HM decoders under the AI, RA, and LB conditions averaged across sequences.

coding speed is the most valuable attribute. In addition, codec latency has become a crucial factor in many vision-based applications gaining ground, e.g., in autonomous driving, robotics, and smart manufacturing. Advanced network technologies such as 5G and WiFi 6 will further broaden the range of real-time video applications in the future. Here, our primary focus is to envisage forthcoming practical VVC codec

implementations from the perspective of existing solutions and our RDC results.

A. EXISTING REAL-TIME HEVC VIDEO CODECS

Since HEVC was standardized, real-time HEVC codecs have been released by many companies such as MainConcept (*MainConcept HEVC*), Huawei (*HW265*), Tencent (*Tencent V265*), Nanjing Yunyan (*sz265*), and ByteDance (*Bytedance V265*) [34]. In addition, many commercial [35]–[37] and academic [38]–[40] hardware implementations for up to 4K120p format have been published. Nowadays, a hardware HEVC codec is an integral component in many state-of-the-art video cameras, smartphones, tablets, TVs, PCs, and gaming consoles. For instance, Qualcomm systems-on-chip has been equipped with 4K30p UHD HEVC encoders and 4K60p UHD 10-bit HEVC decoders since the release of Snapdragon 820 in 2016.

There also exist a couple of noteworthy practical open-source HEVC encoders and decoders out of which only x265 encoder [41], our Kvazaar encoder [42], and OpenHEVC decoder [43] are under active academic research and development. In these implementations, real-time HEVC coding speed has been reached by implementing hardware optimizations through handcrafted assembly functions, vectorization, and by exploiting high-level parallelism [44].

B. PROSPECTIVE REAL-TIME VVC VIDEO CODECS

The emergence of real-time VVC decoders is the key to global adoption of the VVC standard. Therefore, close attention has been paid to hardware-friendliness during the VVC standardization. If the complexity gap of 1.8x reported here between VTM and HM decoders also holds for practical HEVC and VVC decoders, the rapid advances in processing technologies are alone able to overcome that overhead [45]. Until now, a couple VVC software decoders have been released and we believe there are many others already on the horizon. A proof-of-concept 4K real-time hardware VVC decoder was already shown [46] and still unreleased OpenVVC was used in the UHD video demonstration [47]. A VLC player plugin called O266dec [48], [49] and Fraunhofer VVdeC [50], [51] are two practical VVC decoder software implementations that include multiple levels of parallelization.

The VVC standard only defines the decoding process so there are several degrees of freedom to optimize nonnormative VVC encoding tools. Even though tackling the reported 7-9 times complexity of VVC encoding is a challenging task, there are many approaches to simplify or create close approximations of the nonnormative VVC encoding tools. These design decisions tend to be taken at the cost of RD loss over VTM, but it is also the case with practical HEVC encoders. Hence, we anticipate that the RD performance of VVC encoders will gradually improve and coding gain reported here will thereby turn into reality in practical encoders in the long run. So far, MulticoreWare has formed a new multi-company consortium to develop x266 for open-source VVC

encoding [52] and Fraunhofer HHI is developing VVenC [53], [54]. VVenC is an optimized implementation of the VTM encoder, but further optimizations are still needed before reaching real-time VVC encoding performance with it.

The trend in the recent years has been to increase coding speed with parallelism and intelligent coding techniques [55] so we believe that the following three implementation approaches will gain traction in practical software VVC encoders: 1) state-of-the-art *machine learning* (ML) techniques that dynamically adapt to video content and predict the advantageous coding decisions beforehand; 2) parallel processing with novel vector extensions, such as AVX2 and AVX-512; and 3) multi-level threading strategies on the latest high-end multicore processors. Further speedup and lower power dissipation can be obtained by offloading the compute-intensive coding tools to custom hardware accelerators or implementing the entire VVC encoder on FPGA or ASIC. In addition, ML approaches may particularly benefit from GPUs. Anyway, implementing a real-time VVC encoder with a reasonable coding efficiency, implementation cost, and power budget requires novel encoder optimizations and powerful computing platforms.

IX. CONCLUSION

This paper presented a comparative rate-distortion-complexity analysis between the reference video codecs of VVC (VTM10.0) and HEVC (HM16.22). To the best of our knowledge, this is the first independent wide-scale RDC study between VTM and HM containing three different quality metrics (PSNR, SSIM, and VMAF), 30 versatile test sequences (from 240p to 2160p), and three diverse test conditions (AI, RA, and LB). In addition, complexity hotspots of the VTM encoder and decoder were highlighted by in-depth profiling at cycle level. For a fair comparison, the VTM and HM codecs were benchmarked under the same VTM CTC test conditions.

TABLE 10. Summary of VVC and HEVC reference codec comparison.

VTM vs. HM	AI	RA	LB
PSNR BD-rate	-23.0%	-33.1%	-27.2 %
SSIM BD-rate	-23.9%	-36.6%	-29.5 %
VMAF BD-rate	–	-34.4%	-26.7 %
Encoding complexity	34.0×	8.8×	7.5×
Decoding complexity	1.8×	1.8×	1.8×

Table 10 summarizes our main results that serve as a baseline for future VVC codec implementations. On average, VTM improves AI coding efficiency over that of HM by around 23% but at the cost of over 34× encoding complexity. The respective metrics are 35% and 8.8× for the RA case and 28% and 7.5× for the LB case.

The first-generation practical encoders are facing the challenge of tackling the encountered 7–9 times complexity growth with acceptable RD trade-offs. In the course of time, the next-generation fully-fledged encoders will gradually

be able to take better advantage of the coding gains of VVC through novel ML techniques, parallelization, hardware acceleration, and more powerful processing technology. On the other hand, the VTM decoding overhead of 1.8× over HM is already well compensated even by the current state-of-the-art mobile computing platforms. The existing implementations already serve as a clear evidence for VVC practicality and foster the deployment of VVC in the next-generation media applications worldwide.

REFERENCES

- [1] Cisco Systems. (Dec. 2018). *Cisco Visual Networking Index: Forecast and Trends 2017-2022*. [Online]. Available: <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>.
- [2] Comcast Corporation. *COVID-19 Network Update*. Accessed: Apr. 29, 2021. [Online]. Available: <https://corporate.comcast.com/covid-19/network/may-20-2020>
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [5] *Bitmovin Video Developer Report 2019*, Bitmovin, San Francisco, CA, USA, 2019.
- [6] *Requirements for a Future Video Coding Standard v5*, document N17074, The Moving Picture Experts Group (MPEG), Turin, Italy, Jul. 2017.
- [7] International Telecommunication Union (ITU). *New Versatile Video Coding Standard to Enable Next-Generation Video Compression*. Accessed: Apr. 29, 2021. [Online]. Available: <https://www.itu.int/en/mediacentre/Pages/pr13-2020-New-Versatile-Video-coding-standard-video-compression.aspx>
- [8] *Versatile Video Coding*, Standard Recommendation ITU-T Rec. H.266 and ISO/IEC 23090-3 (VVC), ITU-T and ISO/IEC JTC 1, Jul. 2020.
- [9] *VVC Reference Software Version 10.0*. Accessed: Apr. 29, 2021. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSsoftware_VTM/-/tree/VTM-10.0
- [10] J. Chen, M. Karczewicz, Y.-W. Huang, K. Choi, J.-R. Ohm, and G. J. Sullivan, "The joint exploration model (JEM) for video compression with capability beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1208–1225, May 2020.
- [11] *HEVC Reference Software Version 16.20*. Accessed: Apr. 29, 2021. [Online]. Available: <https://vcgit.hhi.fraunhofer.de/jct-vc/HM/-/tags/HM-16.20>
- [12] F. Bossen, J. Boyce, K. Stuehring, X. Li, and V. Seregin, *VTM Common Test Conditions and Software Reference Configurations for SDR Video*, document JVET-T2010, Teleconference, Oct. 2020.
- [13] G. Bjontegaard, *Improvements of the BD-PSNR Model*, document VCEG-A11, Berlin, Germany, Jul. 2008.
- [14] *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.
- [15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [16] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara. (Jun. 2016). *Toward a Practical Perceptual Video Quality Metric*. [Online]. Available: <http://techblog.net?ix.com/2016/06/toward-practical-perceptual-video.html>
- [17] Intel Corporation. *Intel VTune Performance Analyzer*. Accessed: Apr. 29, 2021. [Online]. Available: <https://software.intel.com/content/www/us/en/develop/home.html>
- [18] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Proc. 11th ACM Multimedia Syst. Conf.*, Istanbul, Turkey, May 2020, pp. 297–302.
- [19] P. Topiwala, M. Krishnan, and W. Dai, "Performance comparison of VVC, AV1 and EVC," in *Proc. Appl. Digit. Image Process.*, Sep. 2019, pp. 290–301.

- [20] P. Philippe, J. Fournier, W. Hamidouche, and J. Y. Aubie, *AHG4: Subjective comparison of VVC and HEVC*, document JVET-00451, Gothenburg, Sweden, Jul. 2019.
- [21] A. Cerveira, L. Agostini, B. Zatt, and F. Sampaio, "Memory assessment of versatile video coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 1186–1190.
- [22] F. Pakdaman, M. A. Adelimanesh, M. Gabbouj, and M. R. Hashemi, "Complexity analysis of next-generation VVC encoding and decoding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 3134–3138.
- [23] I. Siqueira, G. Correa, and M. Grellert, "Rate-distortion and complexity comparison of HEVC and VVC video encoders," in *Proc. IEEE 11th Latin Amer. Symp. Circuits Syst. (LASCAS)*, San Jose, Costa Rica, Feb. 2020, pp. 1–4.
- [24] D. García-Lucas, G. Cebrián-Márquez, and P. Cuenca, "Rate-distortion/complexity analysis of HEVC, VVC and AV1 video codecs," *Multimedia Tools Appl.*, vol. 79, nos. 39–40, pp. 29621–29638, Aug. 2020.
- [25] T. Laude, Y. G. Adhisantoso, J. Voges, M. Munderloh, and J. Ostermann, "A comprehensive video codec comparison," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–16, Nov. 2019.
- [26] F. Bossen, X. Li, K. Sühning, K. Sharman, V. Seregin, and A. Tourapis, *JVET AHG Report: Test Model Software Development (AHG3)*, document JVET-U0003, Jan. 2021.
- [27] J. Vanne, M. Viitanen, T. D. Hämäläinen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1885–1898, Dec. 2012.
- [28] 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. IEEE*, early access, Jan. 19, 2021. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9328514>, doi: 10.1109/JPROC.2020.3043399.
- [29] J. Chen, Y. Ye, and S. Kim, *Algorithm Description for Versatile Video Coding and Test Model 10 (VTM 10)*, document JVET-S2002, Teleconference, Jul. 2020.
- [30] J. Chen, Y. Ye, and S. Kim, *Versatile Video Coding Editorial Refinements on Draft 10*, document JVET-T2001, Teleconference, Oct. 2020.
- [31] *FFmpeg*. Accessed: Apr. 29, 2021. [Online]. Available: <https://ffmpeg.org>
- [32] A. Tissier, A. Mercat, T. Amestoy, W. Hamidouche, J. Vanne, and D. Menard, "Complexity reduction opportunities in the future VVC intra encoder," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Sep. 2019, pp. 1–6.
- [33] J. Stankowski, C. Korzeniewski, M. Domanski, and T. Grajek, "Rate-distortion optimized quantization in HEVC: Performance limitations," in *Proc. Picture Coding Symp. (PCS)*, May 2015, pp. 85–89.
- [34] MSU Video Group. *HEVC/AV1 Video Codecs Comparison 2019*. Accessed: Apr. 29, 2021. [Online]. Available: http://compression.ru/video/codec_comparison/hevc_2019/
- [35] System-on-Chip Technologies. *H.265 HD Video Encoder IP Core*. Accessed: Apr. 29, 2021. [Online]. Available: <https://www.soctechnologies.com/ip-cores/ip-core-h265-encoder>
- [36] Socionext. *H.265(HEVC) 4K/60p Multi Format Codec MB86M30*. Accessed: Apr. 29, 2021. [Online]. Available: <http://www.socionext.com/en/products/assp/h264h265/MB86M30/>
- [37] Xilinx. *NGCodec HEVC Encoder*. Accessed: Apr. 29, 2021. [Online]. Available: <https://www.xilinx.com/video/fpga/ngcodec-hevc-encoder.html>
- [38] Y. Omori, K. Nakamura, T. Onishi, D. Kobayashi, T. Osawa, and H. Iwasaki, "4K 120fps HEVC temporal scalable encoder with super low delay," in *Proc. 26th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Genoa, Italy, Nov. 2019, pp. 410–413.
- [39] Y. Omori, T. Onishi, H. Iwasaki, and A. Shimizu, "A 120 fps high frame rate real-time HEVC video encoder with parallel configuration scalable to 4K," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 4, no. 4, pp. 491–499, Oct. 2018.
- [40] T. Onishi, T. Sano, Y. Nishida, K. Yokohari, J. Su, K. Nakamura, K. Nitta, K. Kawashima, J. Okamoto, N. Ono, R. Kusaba, A. Sagata, H. Iwasaki, M. Ikeda, and A. Shimizu, "Single-chip 4K 60fps 4:2:2 HEVC video encoder LSI with 8K scalability," in *Proc. Symp. VLSI Circuits (VLSI Circuits)*, Jun. 2015, pp. C54–C55.
- [41] MulticoreWare. *X265 HEVC Encoder / H.265 Video Codec*. Accessed: Apr. 29, 2021. [Online]. Available: <https://bitbucket.org/multicoreware/x265/downloads>
- [42] Ultra Video Group. *Kvazaar Open-Source HEVC Encoder*. Accessed: Apr. 29, 2021. [Online]. Available: <https://github.com/ultravideo/kvazaar>
- [43] W. Hamidouche, M. Raulat, and O. Deforges, "4K real-time and parallel software video decoder for multilayer HEVC extensions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 169–180, Jan. 2016.
- [44] A. Lemmetti, M. Viitanen, A. Mercat, and J. Vanne, "Kvazaar 2.0: Fast and efficient open-source HEVC inter encoder," in *Proc. 11th ACM Multimedia Syst. Conf.*, New York, NY, USA, May 2020, pp. 237–242.
- [45] K. Rupp. *42 Years of Microprocessor Trend Data*. Accessed: Apr. 29, 2021. [Online]. Available: <https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/>
- [46] S. Gudumasu, S. Bandyopadhyay, and Y. He, "Software-based versatile video coding decoder parallelization," in *Proc. 11th ACM Multimedia Syst. Conf.*, Istanbul, Turkey, May 2020, pp. 202–212.
- [47] S. A. Ateme. (Jul. 2020). *End-to-end UHD Satellite Broadcast Transmission Using Versatile Video Coding (VVC)*. [Online]. Available: <https://www.ateme.com/end-to-end-uhd-satellite-broadcast-transmission-using-versatile-video-coding-vvc/>
- [48] Tencent Cloud. *Tencent O266dec Plugin (0.0.1)*. Accessed: Apr. 29, 2021. [Online]. Available: <https://github.com/TencentCloud/O266player>
- [49] B. Zhu, S. Liu, X. Xu, X. Zhang, C. Gu, L. Wang, and W. Feng, *Performance of a VVC Software Decoder*, document JVET-T0095, Teleconference, Oct. 2020.
- [50] *Fraunhofer Versatile Video Decoder (VVDeC)*. Accessed: Apr. 29, 2021. [Online]. Available: <https://github.com/fraunhoferhhi/vvdec>
- [51] 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 *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 3124–3128.
- [52] MulticoreWare. *Leading Next-Gen Video Technologies With Development of Open Source x266 (VVC) Encoding and x266 Consortium*. Accessed: Apr. 29, 2021. [Online]. Available: <https://multicorewareinc.com/video/#x266>
- [53] *Fraunhofer Versatile Video Encoder (VVeC)*. Accessed: Apr. 29, 2021. [Online]. Available: <https://github.com/fraunhoferhhi/vvenc>
- [54] 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 *Proc. IEEE 22nd Int. Workshop Multimedia Signal Process. (MMSp)*, Tampere, Finland, Sep. 2020, pp. 1–6.
- [55] D. Liu, Z. Chen, S. Liu, and F. Wu, "Deep learning-based technology in responses to the joint call for proposals on video compression with capability beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1267–1280, May 2020.



ALEXANDRE MERCAT (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Institut National des Sciences Appliquées (INSA) of Rennes, Rennes, France, in 2015 and 2018, respectively.

He has been a Postdoctoral Researcher with Computing Sciences, Tampere University (TAU), Tampere, Finland, since 2018. His research interests include implementation of image and signal processing applications in many core embedded systems, real-time implementations of the new generation video coding standards, complexity-aware video coding, machine learning, approximate computing, power consumption, and digital systems design. He received the Best Open Dataset and Software Paper Award from ACM MMSys'20 Conference.



ARTTU MÄKINEN (Member, IEEE) received the M.Sc. degree in electrical engineering from Tampere University (TAU), Tampere, Finland.

He worked as a Researcher/Research Assistant with the Faculty of Information Technology and Communication Sciences, TUNI, from 2019 to 2021. His research interests include video compression, performance analysis, and video coding standards.



JOOSE SAINIO received the M.Sc. degree in information technology from the Tampere University of Technology, Tampere, Finland, in 2018. He is currently pursuing the Ph.D. degree with UVG.

He has been a part of UVG, since 2016. His research interests include HEVC/VVC video coding, in particular enabling real-time encoding. He has experience in both hardware acceleration and more traditional optimization methods.

Additionally, he has some familiarity with perceptual video coding and rate control.



ARI LEMMETTI (Member, IEEE) received the B.Sc. degree in information technology from Tampere University, Tampere, Finland, in 2019. He is currently pursuing the M.Sc. degree.

He is also a Research Assistant with Tampere University, and a member of the Ultra Video Group, since 2014. His research interests include HEVC and VVC video compression, rate-distortion-complexity optimization, and parallel computing. He received the Best Open Dataset and Software Paper Award from the ACM MMSys'20 Conference.



MARKO VIITANEN (Member, IEEE) received the M.Sc. degree in information technology from the Tampere University of Technology, Tampere, Finland, in 2017, where he is currently pursuing the Ph.D. degree.

He is also working as a Doctoral Researcher with TAU. His research interests include HEVC/VVC video coding, 360/VR video capturing and compression, and customized transmission systems.



JARNO VANNE (Member, IEEE) received the M.Sc. degree in information technology and the Ph.D. degree in computing and electrical engineering from the Tampere University of Technology (TUT), Tampere, Finland, in 2002 and 2011, respectively.

He is currently an Associate Professor with the Unit of Computing Sciences, Tampere University, Tampere. He is also the Founder and a Leader of the Ultra Video Group that is also the leading academic video coding group in Finland. He has been the project manager for 17 international/national research projects. He is the author of over 70 peer-reviewed scientific publications. His research interests include HEVC/VVC video coding, ML-powered video coding, immersive 3D/360 media processing for extended reality (XR), volumetric video capture and coding, vision-based environment perception in autonomous vehicles and drones, hybrid human-machine vision, remote machine control over 5G, telepresence, hardware accelerated video coding, video annotation, and virtual traffic simulation environments.

...