A Comparison of JEM and AV1 with HEVC: Coding Tools, Coding Efficiency and Complexity

Thorsten Laude, Yeremia Gunawan Adhisantoso, Jan Voges, Marco Munderloh, and Jörn Ostermann
Institut für Informationsverarbeitung, Leibniz Universität Hannover, Germany
Email: {laude, gunawan, voges, munderloh, office}@tnt.uni-hannover.de

Abstract—**The current state-of-the-art for standardized video codecs is *High Efficiency Video Coding (HEVC)* which was developed jointly by ISO/IEC and ITU-T. Recently, the development of two contenders for the next generation of standardized video codecs began: ISO/IEC and ITU-T advance the development of the *Joint Exploration Model (JEM)*, a possible successor of HEVC, while the Alliance for Open Media pushes forward the video codec *AV1*. It is asserted by both groups that their codecs achieve superior coding efficiency over the state-of-the-art. In this paper, we discuss the distinguishing features of JEM and AV1 and evaluate their coding efficiency and computational complexity under well-defined and balanced test conditions. Our main findings are that JEM considerably outperforms HM and AV1 in terms of coding efficiency while AV1 cannot transform increased complexity into competitiveness in terms of coding efficiency with neither of the competitors except for the all-intra configuration.**

I. INTRODUCTION

For several decades, the market for standardized video codecs was dominated by the standardization groups ISO, IEC and ITU-T: MPEG-1, MPEG-2/H.262, H.263, MPEG-4 Visual, AVC (also referred to as MPEG-4 Part 10 and H.264) are some standards in this line. In 2013, the steady improvement of video coding algorithms resulted in High Efficiency Video Coding (HEVC) which was standardized as MPEG-H Part 2 by ISO/IEC and as H.265 by ITU-T. A reference implementation of HEVC is available with the HM software. Compared to its predecessor standard AVC, HEVC considerably increases the coding efficiency. Depending on the selected configuration, HEVC achieves a 40-60% bit rate reduction while maintaining the same visual quality [1].

More recently, new participants entered the market for video codecs. Among the proposed codecs are VP8, VP9, Daala and Thor. The participants responsible for these codecs joined their efforts in the Alliance for Open Media (AOM) to develop the video codec AV1. Furthermore, AV1 is a contender for standardization by the Internet Engineering Task Force (IETF) as Internet Video Codec (NetVC). The finalization of the standardization process was scheduled for 2017 but delayed to the end of 2018 [2].

Concurrently, ISO/IEC and ITU-T established the Joint Video Exploration Team (JVET) in October 2015 to explore technologies for a potential HEVC successor. For this purpose, a reference software called Joint Exploration Model (JEM) was developed which includes a variety of novel coding tools.

Given these three codecs — or to be more precise codec implementations — (HM as state-of-the-art and JEM and AV1 as contenders), it is of great interest to assess and compare their performance. This comparison can be performed in terms of coding efficiency but also in terms of computationally complexity (run times for the encoding and decoding, as well as memory requirements). For JEM and HM, a straightforward comparability is given because both codecs share the same foundation (with JEM being an extension of HM) and Common Test Conditions are defined to configure both codecs similarly [3]. To include AV1 in a fair comparison is more challenging because its structure and working principle are fundamentally different compared to HM and JEM. This also explains why existing comparisons of HEVC with VP8, VP9 or AV1 in the literature come to different conclusions [4], [5].

In this paper, we compare JEM and AV1 with HEVC under well-defined and balanced (i.e. on a par for all codecs) conditions. First, we analyze the novel coding tools in JEM and AV1 in Section II. In Section III and in Section IV, we compare the performance of the three codecs in terms of coding efficiency and complexity, respectively. Section V concludes the paper.

II. CODING TOOLS

In this section, we briefly introduce the distinguishing features of JEM and AV1 compared to HEVC.

A. Distinguishing features of JEM

JEM extends the underlying HEVC framework by modifications of existing tools and by adding new coding tools. In what follows, we briefly address the most important modifications. An exhaustive review can be found in [6].

1) Block partitioning: One of the most beneficial innovations of HEVC is the flexible quad-tree partitioning of the coding tree units (CTUs) into coding units (CUs), prediction units (PUs) and transform units (TUs) of different sizes. This partitioning scheme is replaced in JEM by a quad-tree plus binary-tree (QTBT) block structure. CTUs (whose maximal size is increased from 64×64 to 128×128 to reflect increasing spatial video resolutions) are partitioned using a quad-tree followed by a binary tree. Thereby, CUs can be square or rectangular. This more flexible partitioning allows CUs, PUs and TUs to have the same size which circumvents the signaling overhead of having three independent partitioning instances.

2) Intra prediction: In JEM, the number of angular modes is extended to 65. Additionally, the preciseness of the fractional pel filters for directional modes is increased by using

4-tap instead of 2-tap filters. Boundary filters are applied for more directional mode to reduce the occurrence of abrupt boundaries. The position-dependent intra prediction combination (PDPC) which combines the usage of filtered reference samples and unfiltered reference samples is used to improve the planar mode. Typically, there remains some redundancy between the luma component and the chroma components. To exploit this redundancy, a cross-component linear model (CCLM) similar to e.g. [7] is adopted in JEM. With this algorithm, chroma blocks are predicted based on the corresponding luma blocks.

3) Inter prediction: Sub-CU motion vector prediction (using Alternative Temporal Motion Vector Prediction, ATMVP, and Spatial-temporal Motion Vector Prediction, STMVP) allows splitting larger CUs into smaller sub-CUs and predicting a more accurate motion vector field for these sub-CUs via additional merge candidates. The on CU-level activatable Overlapped Block Motion Compensation (OBMC) uses the motion information of neighboring sub-CUs in addition to the motion information of the currently coded sub-CU to predict multiple signals for the current sub-CU which are then combined by a weighted average. Conceptually (despite the adaptivity) this can also be found in H.263. To cope with illumination changes between the current CU and the reference block, a Local Illumination Compensation (LIC) is defined. With LIC, the illumination is adjusted using a linear model whose parameters are derived by a least-squares approach. To improve the prediction of content with non-translative motion, JEM supports affine motion compensation. Multiple techniques are employed to improve the motion vector accuracy: The available motion information after block-wise motion compensation can be improved using Bi-directional Optical Flow (BIO), a Decoder-side Motion Vector Refinement (DMVR) is applied in case of bi-prediction, and Pattern Matched Motion Vector Derivation (PMMVD) is used to derive motion information for merged blocks at the decoder. The CU-level Locally Adaptive Motion Vector Resolution (LAMVR) enables the signaling of motion vector differences with full-pel, quarter-pel, and four-pel precision. Additionally, the precision of the internal motion vector storage is increased to $1/16$ pel (and $1/32$ pel for chroma).

4) Transform coding: The transform coding technique of HEVC is quite consistent for different block sizes, different modes and different contents. For almost every case, a discrete cosine transform (DCT-II) is used. Intra-coded 4×4 TUs constitute the only deviation as they are coded with a discrete sine transform (DST-VII). In contrast to that, JEM can rely on a greater variety of selectable core transforms from the DCT and DST families (DCT-II, DCT-V, DCT-VIII, DST-I and DST-VII). Depending on the selected mode (intra or inter), and in case of intra depending on the selected direction, a sub-set of the available core transforms is formed and one transform from this sub-set is selected via rate-distortion (RD) optimization. This technique is referred to as Adaptive Multiple Transform (AMT). For big blocks (width or height is equal to or larger than 64), the high-frequency coefficients are automatically zeroed out as no meaningful information is expected from them for signals which are encoded at this block size. In addition to the higher variety of core transforms, JEM provides multiple other novel transform techniques over HEVC: A Mode-Dependent Non-Separable Secondary Transform (MDNSST) is applied between the core transform and the quantization. Its purpose is to reduce remaining dependencies after the separable core transforms which only address horizontal and vertical dependencies. It is known that the Karhunen-Loève transform (KLT) is the only orthogonal transform which can achieve uncorrelated transform coefficients with the extra benefit of efficient energy compaction. At first glance, the drawback of the KLT is that it is signal-dependent. It would be necessary to signal the transform matrix for a given block as part of the bit stream. As this is unfeasible due to the considerable signaling overhead, the KLT cannot be employed directly. To circumvent this drawback, the KLT is realized in JEM (here referred to as Signal-Dependent Transform or SDT) in such a way that the transform matrix is calculated based on the most similar region within the already reconstructed signal.

5) In-loop filtering: Adaptive Loop Filters (ALF) [8], [9] were studied intermediately during the standardization process of HEVC but were dismissed prior to the finalization of the standard. With JEM, they return to the codec design. Basically, Wiener filters are derived to optimize the reconstructed signal towards the original signal during the in-loop filtering stage. Another new in-loop filter in the JEM architecture is a bilateral filter which smooths the reconstructed signal with a weighted average calculation on neighboring sample values. ALF and the bilateral filter are applied in addition to Sample Adaptive Offset and to the deblocking filter. The order of filtering is: Bilateral – SAO – deblocking – ALF.

6) Entropy coding: The CABAC technique is enhanced by a multiple-hypothesis probability estimation model and by an altered context modeling for the transform coefficients. Furthermore, the context model states of already coded pictures can be used as initialization for the state of the currently coded picture.

B. Distinguishing features of AV1

AV1 originates from the combination of multiple codecs (VP9, Daala and Thor) which were developed by members of the Alliance for Open Media. In this section, we briefly review the distinguishing features of AV1. Additional information can be found in [10].

1) Block partitioning: Similar to JEM, AV1 relies on an enhanced quad-tree partitioning structure. Pictures are partitioned into superblocks (equivalent to CTUs) with a maximum size of 128×128. Superblocks can be recursively partitioned into either square or rectangular shaped blocks down to a minimum size of 4×4. The tree-based partitioning is extended by a wedge mode in which a rectangular block can be partitioned by a wedge into non-rectangular parts for which different predictors are used. Thereby, the partitioning can be better adapted to object boundaries. The wedges can be selected from a wedge code book. The wedge mode enables the combination of inter and intra prediction within a block.

2) Intra prediction: For intra prediction, AV1 provides the following modes: a generic directional predictor, a Paeth predictor and a smooth predictor. The generic directional predictor resembles the angular intra prediction as it is realized in JEM and HEVC. It consists in an angular prediction in one of 56 different directions using a 2-tap linear interpolation with a precision of $1/256$ pel. The Paeth predictor and the smooth predictor of AV1 are conceptually similar to the planar mode in JEM and HEVC. The Paeth predictor performs a prediction based on three pixels in neighboring blocks to the left, top and top-left side. The smooth predictor is based on the weighted averaging of neighboring pixels from the left and top neighboring blocks and of interpolated pixels at the bottom and right of the current pixel.

3) Inter prediction: The inter prediction in AV1 has access to up to six reference pictures of which one or two can be chosen per block. Motion vectors can be predicted at 8×8 block level by Dynamic Reference Motion Vector Prediction. Similar to JEM, AV1 specifies an Overlapped Block Motion Compensation (OBMC) to refine the prediction at block boundaries by utilizing neighboring predictors. AV1 supports multiple global motion compensation models: a rotation-zoom model with four parameters, an affine model with six parameters, and a perspective model with eight parameters. Warping can be applied by horizontal and vertical shearing using 8-tap filters. For high spatial resolutions, a technique called Guided Restoration is applied. With this technique, the video signal is downscaled and encoded at a lower resolution. At the decoder, the signal is upscaled to its original spatial resolution. To reduce the distortion as a result of the downscaling, Wiener filters for the upscaling are derived at the encoder and transmitted to the decoder.

4) Transform coding: AV1 supports multiple transforms: DCT, Asymmetric DST (ADST), flipped ADST, and Identity. The identity transform is similar in spirit to the transform skip mode of JEM and AV1 and beneficial for example for screen content coding. The vertical and the horizontal transform can be selected independently from the set of four available transforms. In total, 16 transform combinations are possible this way. AV1 includes both, linear and non-linear quantization matrices for the quantization.

5) In-loop filtering: For the in-loop filtering, AV1 combines the constrained low-pass filter from the Thor codec with the directional deringing filter from the Daala codec into the Combined Constrained Directional Enhancement (CDEF). It is stated that this filter merging increases the quality of the filtered picture while at the same time reducing the complexity compared to two separate filtering processes.

6) Entropy coding: The entropy coding in AV1 is based on the combination of a Multi-symbol Arithmetic Range Coder with Symbol Adaptive Coding. Thereby, a multi-symbol alphabet is encoded with up to 15-bit probabilities and an alphabet size of up to 16 symbols. With this entropy coder, multiple binary symbols are combined into non-binary symbols. It is stated that the efficiency is increased compared to a binary entropy encoder especially for lower bit rates due to a reduced signaling overhead.

III. CODING EFFICIENCY

In this section, we compare the coding efficiency of the three codecs. Encoders can be tuned for various metrics which can be objective (e.g. PSNR) or which can approximate properties of the human visual system. Furthermore, JEM and HM support only tuning for PSNR while AV1 supports other metrics in addition to PSNR. We think that subjective encoder tunings are highly application-specific. Therefore, to achieve a fair and meaningful comparison of the codecs themselves (and not of particular implementations for particular applications), we tune all encoders to PSNR. The experiments were performed using HM-16.16, JEM-7.0 and AV1 0.1.0-5913 on a heterogeneous cluster consisting of Intel Xeon CPU E5-2680 v3 and Intel Xeon CPU E5-2690 v2 CPUs. HM and JEM were configured following the Common Test Conditions (CTC) [3]. The three encoder configurations all-intra (AI), low-delay B (LDB), and random access (RA) were used in combination with four quantization parameters (22, 27, 32, 37). Following the CTC, 28 test sequences in seven classes were encoded. The CTC for JEM allow a temporal subsampling for the all-intra configuration (only every eighth picture is coded) to save computational time. This was omitted for our experiments, i.e. every picture was coded. For each encoder configuration, AV1 was configured in such a way that the CTC configurations were matched as closely as possible. For this purpose, we followed the recommended encoder configuration for *best quality VBR encoding* [11] while additionally enabling PSNR tuning and adjusting the key-frame distances to match those of the JEM CTC. The reader is referred to [11] for more details on the encoder configuration. Additionally, the AV1 quantization parameter was adjusted for each data point to approach the PSNR of the corresponding HM data point. Three comparisons were conducted: JEM vs. HM, JEM vs. AV1 and AV1 vs. HM. The Bjøntegaard-Delta (BD)-rate as defined in [12] was computed for these three comparisons. It is worth noting that the BD-rate is only meaningful since the encoders were tuned for PSNR. The resulting BD rates are listed in Table I. For all-intra, it is observed that AV1 is competitive to HM with an average BD-rate gain of 4.1% while JEM outperforms HM and AV1 with 19.7% and 16.2% BD-rate gains on average, respectively. For configurations with motion compensation, JEM achieved the highest coding efficiency with average gains of 27.7% (RA)/23.2% (LDB) over HM and 44.6% (RA)/38.2% (LDB). AV1 does not reach the coding efficiency of HM with average losses of -38.3% (RA) and -34.3% (LDB).

IV. COMPLEXITY

In this section, we compare the complexity of the three codecs in terms of computational time and memory consumption. All codecs operated single-threaded to allow a fair comparison.

TABLE I

BD-rates. It is observed that JEM achieves the best coding efficiency for all encoder configurations. For encoder configurations which enable motion compensated prediction, HM achieves the the second place while AV1 is the runner-up for all-intra.

		All-intra			**Random Access**			**Low-delay B**		
Class	**Sequence**	JEM vs. HM	JEM vs. AV1	AV1 vs. HM	JEM vs. HM	JEM vs. AV1	AV1 vs. HM	JEM vs. HM	JEM vs. AV1	AV1 vs. HM
A1 (4K)	Tango2	-22.7%	-18.4%	-4.3%	-34.8%	-37.0%	7.76%	-26.4%	-24.65%	0.2%
	Drums100	-18.2%	-17.0%	-1.7%	-30.4%	-39.1%	18.21%	-23.1%	-27.58%	7.9%
	Campfire	-25.2%	-20.2%	-6.4%	-28.9%	-24.4%	-3.07%	-22.5%	-22.96%	2.2%
	ToddlerFountain2	-15.4%	-12.4%	-3.4%	-14.8%	-7.8%	-6.89%	-13.7%	-7.97%	-5.9%
A2 (4K)	CatRobot	-23.3%	-20.1%	-3.7%	-38.6%	-50.0%	27.07%	-31.0%	-33.44%	6.9%
	TrafficFlow	-26.5%	-20.9%	-6.6%	-28.9%	-51.6%	52.82%	-23.6%	-59.48%	91.8%
	DaylightRoad2	-19.6%	-14.9%	-4.6%	-39.1%	-47.6%	23.50%	-27.8%	-31.33%	9.4%
	Rollercoaster2	-24.7%	-20.4%	-5.6%	-38.7%	-45.9%	16.46%	-32.6%	-35.84%	6.5%
B (1080p)	Kimono	-18.0%	-19.2%	1.1%	-22.2%	-45.6%	43.77%	-15.7%	-30.6%	21.44%
	ParkScene	-17.0%	-16.0%	-1.2%	-21.1%	-51.7%	64.92%	-15.0%	-36.0%	33.42%
	Cactus	-19.1%	-18.5%	-0.9%	-32.0%	-55.0%	52.40%	-25.8%	-45.3%	35.69%
	BasketballDrive	-19.8%	-18.9%	-1.4%	-30.3%	-43.8%	25.00%	-24.5%	-30.6%	8.85%
	BQTerrace	-15.1%	-13.8%	-1.9%	-29.4%	-42.6%	29.87%	-23.1%	-37.9%	23.73%
C (WVGA)	BasketballDrill	-29.4%	-26.6%	-4.1%	-29.1%	-56.0%	60.94%	-24.1%	-42.7%	32.83%
	BQMall	-18.5%	-14.0%	-5.2%	-28.1%	-51.5%	51.42%	-21.8%	-33.0%	17.17%
	PartyScene	-13.9%	-9.1%	-5.3%	-25.1%	-53.3%	62.14%	-20.7%	-37.8%	27.43%
	RaceHorses	-16.5%	-15.6%	-1.1%	-23.1%	-31.1%	12.89%	-19.1%	-24.9%	7.76%
D (WQVGA)	BasketballPass	-17.8%	-13.7%	-4.8%	-24.4%	-40.7%	28.26%	-19.6%	-27.2%	10.84%
	BQSquare	-12.8%	-8.9%	-4.3%	-32.9%	-60.3%	69.35%	-30.5%	-42.7%	21.22%
	BlowingBubbles	-14.4%	-10.2%	-4.7%	-23.5%	-53.2%	67.64%	-20.1%	-36.8%	27.55%
	RaceHorses	-17.9%	-17.1%	-1.2%	-23.3%	-39.1%	25.85%	-19.4%	-29.8%	14.86%
E (720p)	FourPeople	-21.9%	-20.7%	-1.7%	-28.5%	-59.3%	76.15%	-22.8%	-54.3%	69.04%
	Johnny	-22.0%	-22.0%	-0.4%	-29.5%	-61.1%	81.26%	-27.9%	-61.4%	86.70%
	KristenAndSara	-22.2%	-22.0%	-0.4%	-29.4%	-60.6%	79.22%	-27.2%	-56.1%	66.33%
F	BasketballDrillText	-28.5%	-24.4%	-5.8%	-28.8%	-55.1%	58.69%	-25.2%	-42.5%	30.39%
	ChinaSpeed	-16.7%	-8.7%	-8.8%	-19.2%	-34.0%	22.34%	-18.7%	-35.2%	25.42%
	SlideEditing	-15.1%	2.1%	-16.9%	-15.3%	-7.8%	-8.31%	-17.3%	-76.6%	254.08%
	SlideShow	-20.6%	-11.9%	-10.3%	-25.2%	-44.2%	33.84%	-30.5%	-43.7%	25.91%
Mean	Class A1	-20.4%	-17.0%	-4.0%	-27.2%	-27.1%	4.0%	-21.4%	-20.8%	1.1%
	Class A2	-23.5%	-19.1%	-5.1%	-36.4%	-48.8%	30.0%	-28.7%	-40.0%	28.7%
	Class B	-17.8%	-17.3%	-0.9%	-27.0%	-47.7%	43.2%	-20.8%	-36.1%	24.6%
	Class C	-19.6%	-16.3%	-4.0%	-26.3%	-48.0%	46.9%	-21.4%	-34.6%	21.3%
	Class D	-15.7%	-12.5%	-3.7%	-26.1%	-48.3%	47.8%	-22.4%	-34.1%	18.6%
	Class E	-22.0%	-21.6%	-0.8%	-29.2%	-60.3%	78.9%	-26.0%	-57.3%	74.0%
	Class F	-20.2%	-10.7%	-10.5%	-22.1%	-35.3%	26.6%	-22.9%	-49.5%	84.0%
	Overall	**-19.7%**	**-16.2%**	**-4.1%**	**-27.7%**	**-44.6%**	**38.3%**	**-23.2%**	**-38.2%**	**34.3%**

TABLE II

Encoder run times of JEM and AV1 normalized to the encoder run time of HM.

	AI		**RA**		**LDB**	
Sequence	JEM	AV1	JEM	AV1	JEM	AV1
Class A1	34.8	10.9	11.9	37.4	10.5	28.7
Class A2	29.7	10.8	8.8	46.7	7.3	32.0
Class B	38.7	8.4	8.9	29.0	7.8	18.1
Class C	47.6	9.4	11.5	33.4	9.9	21.3
Class D	53.6	10.5	11.7	37.1	10.1	25.2
Class E	28.2	5.9	5.2	28.5	3.4	16.2
Class F	34.9	6.5	9.0	16.1	7.0	11.0
Overall	**38.6**	**9.0**	**9.7**	**32.6**	**8.2**	**21.9**

TABLE III

Decoder run times of JEM and AV1 normalized to the decoder run time of HM.

	AI		**RA**		**LDB**	
Sequence	JEM	AV1	JEM	AV1	JEM	AV1
Class A1	2.1	0.7	5.6	1.3	4.1	1.2
Class A2	2.0	0.7	6.2	1.4	4.8	1.3
Class B	2.2	0.4	6.2	0.8	4.2	0.6
Class C	2.7	0.5	8.0	1.1	6.1	0.8
Class D	4.7	0.5	11.6	1.0	9.9	0.8
Class E	2.5	0.4	6.2	0.8	4.3	0.6
Class F	2.5	0.4	5.3	0.7	4.2	0.6
Overall	**2.7**	**0.5**	**7.0**	**1.0**	**5.4**	**0.9**

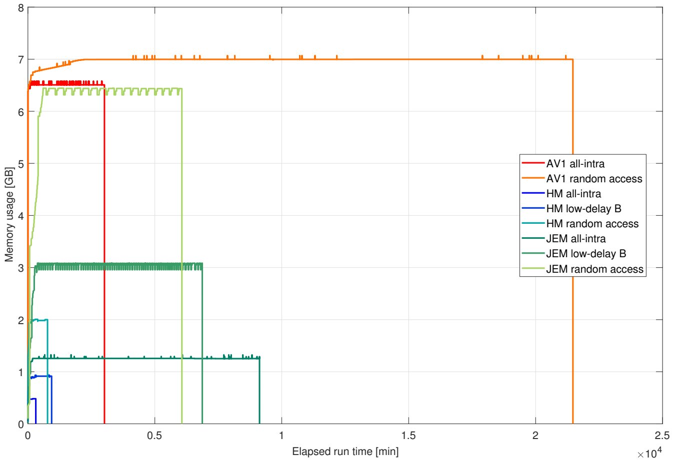

Fig. 1. Memory usage of the encoders over time.

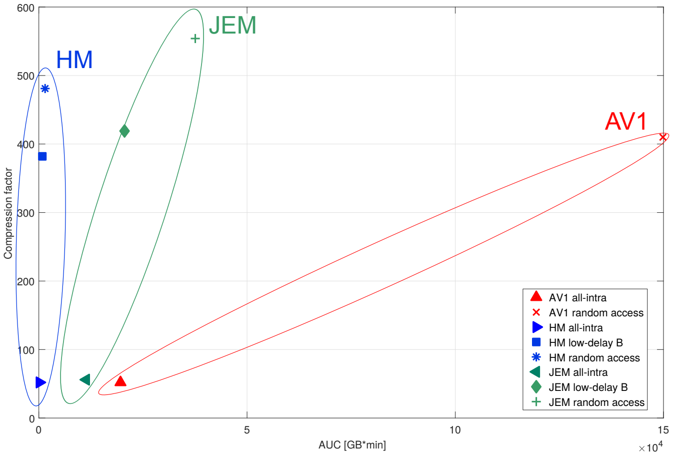

Fig. 2. Compression factor over computational complexity measured as Area Under Curve (AUC) in Fig. 1.

To produce Fig. 1 and Fig. 2, we encoded the Drums100 sequence with HM, JEM and AV1. In Fig. 1, the memory usage of the different encoders is visualized for the considered encoder configurations. It is observed that HM originates the lowest complexity followed by JEM followed by AV1. Encoding a video poses a trade-off between achieved coding efficiency and expended computational complexity to reach this coding efficiency. We measure how the considered encoders perform for this trade-off in Fig. 2 were the compression factor (i.e. the bit rate of the uncompressed video divided by the bit rate of the compressed video) is plotted over the Area Under Curve (AUC) of the curves in Fig. 1 (i.e. the product of run time and memory usage). In this schematic representation, the most desirable positions for encoders is at the top-left (high compression factor, low complexity) while the least desirable position is at the bottom-right (low compression factor, high complexity). It is observed that JEM increases the coding efficiency at the cost of increased complexity compared to HM. Furthermore, AV1 increases the complexity significantly without achieving an increase in coding efficiency. In Table II and in Table III, we compare the encoder and decoder run times normalized to the corresponding HM run times. It is observed that JEM increases the encoder run times considerably compared to HM, especially for all-intra, with factors of 38.6 (AI), 9.7 (RA) and 8.2 (LDB). AV1 also increases the encoder run times considerably, especially for configurations with motion compensation, with factors of 9.0 (AI), 32.6 (RA) and 21.9 (LDB). On the decoder side it is observed that AV1 results in lower or equal complexity than HM for all configurations while JEM increases the complexity (especially for inter prediction since some inter coding tools shifted complexity to the decoder). The worst-case measurement error due to the different CPU types is 2.8%. This is negligible since the average values in Tab. II are between 820% and 3860%. It is worth noting that neither of the codec implementations is a commercial implementation. It is expected that commercial implementations will be more efficiently implemented.

V. CONCLUSIONS

In this paper, we compared two contenders for the next generation of video codecs (JEM and AV1) with the HEVC reference implementation HM. For this purpose, we analyzed the distinguishing features of the new codecs in comparison with HEVC. An objective evaluation was performed with respect to coding efficiency and complexity. Our main findings are that JEM considerably outperforms HM and AV1 in terms of coding efficiency while AV1 cannot transform increased complexity into competitiveness with neither of the competitors except for the all-intra configuration.

REFERENCES

[1] Jan De Cock, Aditya Mavlankar, Anush Moorthy, et al., “A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications,” sep 2016, p. 997116, International Society for Optics and Photonics.

[2] IETF 100, “NetVC WG: Internet Video Codec. https://datatracker.ietf.org/meeting/100/materials/slides-100-netvc-chair-slides/,” 2017.

[3] Joint Video Exploration Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, “JVET-G1010: JVET common test conditions and software reference configurations. 7th Meeting: Torino, IT, 1321 July,” 2017.

[4] Dan Grois, Tung Nguyen, and Detlev Marpe, “Coding efficiency comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC encoders,” in *2016 Picture Coding Symposium (PCS)*. 2016, pp. 1–5, IEEE.

[5] Debargha Mukherjee, Jim Bankoski, Adrian Grange, et al., “The latest open-source video codec VP9 - An overview and preliminary results,” in *2013 Picture Coding Symposium (PCS)*. dec 2013, pp. 390–393, IEEE.

[6] Joint Video Exploration Team JVET of ITU-T SG16 WP3 and ISO/IEC, “JVET-G1001: Algorithm Description of Joint Exploration Test Model 7 (JEM 7).,” 2017.

[7] Ali Khairat, Tung Nguyen, Mischa Siekmann, et al., “Adaptive Cross-component Prediction for 4 : 4 : 4 High Efficiency Video Coding,” in *ICIP*, 2014, pp. 3734–3738.

[8] Ching-Yeh Chen, Chia-Yang Tsai, Yu-Wen Huang, et al., “The adaptive loop filtering techniques in the HEVC standard,” oct 2012, vol. 8499, p. 849913, International Society for Optics and Photonics.

[9] Chia-Yang Tsai, Ching-Yeh Chen, Tomoo Yamakage, et al., “Adaptive Loop Filtering for Video Coding,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 934–945, dec 2013.

[10] Pascal Massimino, “Paris Video Tech 4 : AOM - AV1. How does it work?,” https://parisvideotech.com/wp-content/uploads/2017/07/AOM-AV1-Video-Tech-meet-up.pdf, 2017, Online.

[11] “Webm project : an open web media project, vp8 encode parameter guide. https://www.webmproject.org/docs/encoder-parameters/,” Online.

[12] Gisle Bjøntegaard, “VCEG-AI11: Improvements of the BD-PSNR model. ITU-T SG 16 Q 6. 35th Meeting, Berlin, Germany,” 2008.