Performance Comparison of VVC, AV1 and HEVC on 8-bit and 10-bit content

Pankaj Topiwala, Madhu Krishnan, and Wei Dai {pankajtva at. gmail dot. com}
FastVDO LLC
3097 Cortona Drive, Melbourne, FL, USA

ABSTRACT

This paper presents a study comparing the coding efficiency performance of three video codecs: (a) the Versatile Video Coding (VVC) Bench Mark Set 1 (BMS1); (b) AV1 codec of the Alliance for Open Media (AOM); and (c) the HEVC Main Profile Reference Software. Two approaches to coding were used: (i) constant quality (QP); and (ii) target bit rate (VBR). Constant quality encoding is performed with all the three codecs for an unbiased comparison of the core coding tools. Whereas, target bitrate coding is done with the AV1 codec to study the compression efficiency achieved with rate control, which can and does have a significant impact. Performance is tabulated for on two fronts: (1) objective performance based on PSNR's and (2) informal subjective assessment. Our general conclusion derived from the assessment of objective metrics and subjective evaluation is that VVC (BMS1) appears to be superior to AV1 and HEVC under both constant quality and target bitrate coding constraints. AV1 shows superior coding gains with respect to HEVC under target bitrate coding, but in general has increased computational complexity and henceforth an encode time factor of 20 – 30 over HEVC.

1 INTRODUCTION

High Efficiency Video Coding (HEVC) is a video coding standard for video compression developed by the Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC/ITU [1]. The HEVC standard provides significant coding efficiency gain compared to previous standards, including H.264/MPEG-4 AVC [2] (Average bitrate savings of 50% for equivalent perceptual quality). The cost for this significant coding efficiency gain is the substantial increase in the computational complexity at the encoder side. Thus the development of a real time encoder that can achieve the compression efficiency of the reference software requires significant work [3]. The latest version of the reference software (HM) is available at https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware.

Versatile Video Coding (VVC) is a video coding standard for video compression developed by the Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 [4]. The bench mark model of VVC, known as BMS1, uses a subset of the tools in Joint Exploration Model (JEM). Tools that either had high decoder implementation complexity, or that had low demonstrated coding efficiency improvement were not adopted in BMS1. BMS1 provides an additional average bitrate savings of 20-25% for equivalent perceptual quality on top of HEVC. The latest version of the reference software (BMS1) is available at https://jvet.hhi.fraunhofer.de/svn/svn_VVCSoftware_BMS

The AV1 video codec is a high quality open source video codec developed by the Alliance for Open Media, a Joint Development Foundation project formed to define and develop media codecs, media formats, and related technologies to address marketplace demand for an open standard for video compression and delivery over the web [5]. A stable initial release of AV1 was deployed in March 2018. Built on baseline VP9 by the addition of new coding tools, coding efficiency gains have been reported for large enough data sets [6]. A variety of performance comparisons of AV1 to other codecs have been published or presented at SPIE 2017 [7-9, 11]. The latest version of the codec is available at https://aomedia.org/. We will use [6] from AOM, and [11] from HHI, from PCS2018, as key references for our paper.

2 STRUCTURE OF AV1 CODEC, AND COMPARISON TO HEVC, VVC CODECS

2.1. Basic Structure of AV1 Codec

The AV1 codec follows the general structure of a hybrid motion-compensated video codec that has been in use since at least H.261 (1988). The basic block structure is 128x128 pixels (expanded from 64x64 pixel size of VP9), which can be subdivided further for purposes of prediction and transform, right down to 4x4 size. The block partitioning structure is similar to the QTBT structure of HEVC New to AV1, a square block can be subdivided into 4 (rect.) or 3 (2 square, one rect.) blocks in a variety of ways in addition to the block division structure used in VP9. Blocks can be predicted in either Intra (56 directional modes, 2 enhanced non-directional predictors), Inter (single, compound, wedge), or mixed. Transforms include DCT, DST types and identity up to size 64x64. Though the codec does not explicitly contain B-frames, a method called compound prediction effectively offers the same functionality (although it appears no direct analog of hierarchical B-frames, such as in HEVC, exists). Multi-symbol arithmetic coding is used. AV1 supports tiles, though not slices (as in AVC and HEVC), which are not needed in TCP/IP communication, but useful over RTP.

2.2. Tool Comparison to HEVC and VVC Codecs

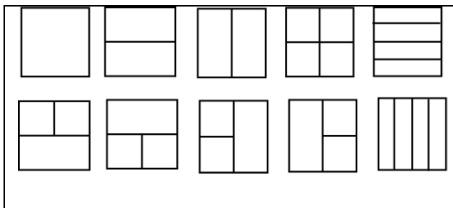
AV1 shares the same basic structure of a hybrid motion-compensated design as all recent codecs in standardization development. The following table summarizes the comparative structure of AV1, HEVC, and VVC (BMS1). In toolset, AV1 falls somewhere between HEVC and VVC BMS model. As we will see, its performance is comparable to HEVC.

	AOM AV1 (1.0)	**H.265/HEVC HM 16.18**	**H.266/VVC BMS 1.0 (draft)**
Block Structure	10-way split (AV1), like QTBT Largest block size 128x128 (superblock).	Quadtree CTU size up to 64x64	(QTBT) + Ternary Tree (TT) CTU size up to 256x256
Intra Prediction	56 intra directional modes 5 non-directional modes Recursive filt. based intra prediction Chroma from Luma Color palette based intra prediction Intra block copy	35 intra prediction modes.	65 intra prediction modes with improved intra mode coding Cross-component linear model (CCLM) prediction
Inter prediction	Single and compound prediction (similar to P and B) (VP9) Extended reference frames (3 to 7) Dynamic spatial and temporal motion vector referencing Overlapped block motion compensation Warped motion compensation Advanced compound prediction	Hierarchical weighted prediction (P, B frames) PU level motion vector prediction Motion vector difference 1/4 pel MV accuracy Block motion comp. Translation motion prediction	Hierarchical weighted prediction (P, B frames) Sub-CU based motion vector prediction Adaptive motion vector precision Affine motion prediction Decoder-side motion vector refinement
Transform	Transform blocks 4x4 up to 64x64 DCT, ADST (VP9), Flipped ADST, DST-I	Transform block size 8x8, 16x16, 32x32 DCT-II and DST-VII	Transform block sizes 4x4 up to 64x64 Adaptive multiple core transforms Mode dependent non-separable secondary transforms (4x4)
Loop filter	Constrained directional enh. filter Loop restoration filters Frame super resolution Film grain synthesis	Deblocking filter, SAO	Deblocking filter, SAO, Adaptive loop filter
Entropy Coding	Multi-symbol entropy coding Level map coefficient coding	CABAC	Modified CABAC (with Context modelling for transform coefficient levels)

Table 1. Tool Comparison Resume in a table (AOM AV1, HEVC HM, JVET VVC BMS1).

2.3. Other Recently Published Comparisons

In this section, we briefly review some recently published comparisons of AV1 with other codecs, specifically [6] from AOM team members, and [11] from HHI, both of which were presented at the Picture Coding Symposium, in June, 2018, in San Francisco (we are grateful for courtesy preprints from these groups). Paper [6], led by Google, provides a very valuable review of the key tools in the AV1 codec, in comparison to the previous VP9. Figure 1 captures for example the changes in the block partitioning structure from VP9, which we find reminiscent of the QTBT (quadtree-binary tree) structure of HEVC. The key differences are: that some elements of two levels of decomposition in QTBT are captured in one shot, and that only square blocks can be further decomposed. It also provides some overall performance comparisons (fig. 1), where AV1 is 22% more efficient than x265 (representing HEVC), 28% more than VP9. Figure 2 provides a performance comparison by HHI (AV1 ~ HEVC), but has encoder runtime 20X longer; they also find AV1 is ~30% more efficient than VP9. In this paper, we derive our own comparisons, to add to this growing body of literature.



AV1 Average Perf. Gains in BD-Rate (%)

Codec	Y	Cb	Cr
VP9	-28.07	-30.1	-31.8
x265	-22.75	-39.18	-40.17

Figure 1. Comparisons of AV1 to VP9 in block partitioning (a), and in performance, to VP9 and x265, from [6]. Note that from the point of view of commercial codecs in common use, AV1 has mainly x265 and VP9 to compare with.

	VP9	HM	AV1
HM	-25.00%		
AV1	-22.90%	2.30%	
JEM	-48.70%	-31.60%	-32%

	VP9	HM	AV1
HM	2.66		
AV1	55.82	20.95	
JEM	22.58	8.48	0.40

Figure 2. (a) Performance comparison of VP9, AV1, HEVC (HM), and JEM, quoted from [11]. (b) encoder speed comparison [11], where the AV1 encoder performs on par with HEVC, but with 20X more complexity/run time, while JEM is better in perf./speed. Key differences between the AOM [6] and HHI [11] tests is that AOM used x265, while HHI used the HM reference SW. Another important factor is that the test data were different in the two cases. In particular, [6] used screen-content sequences in the test, whereas [11] did not.

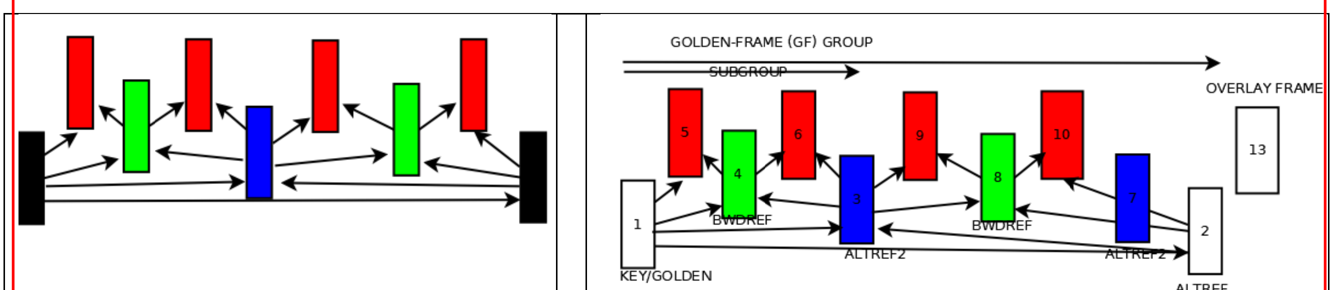


Fig. 3. Example hierarchical prediction/coding structures in (a) HEVC/VVC from [12], and (b) as approximated in AV1, from [6]. While AV1 does not explicitly have hierarchical B-frames (or indeed B-frames per se), and thus the figures are not identical, these can be approximated with the use of Alfref and Bwedref frames. This is suitably done in a 2-pass mode, and we thus test 2-pass exclusively for AV1. This is slower, but suitable for VOD and applications other than live broadcast (even that is possible with some latency). Note that (b) is only a rough approximation to figure 10 in [6], with no guarantees; our point is that it is complicated.

3 FASTVDO EXPERIMENTAL SETUP

The simulations are performed using: a) Rocks cluster distribution (an open source Linux cluster distribution for high performance computing) with 34 cores (5 AMD Phenom II X6 1055T's and 1 AMD Phenom II X4 955) b) two quad core (i7 6700) and one 6 core (i7 8700) windows systems with RAM capacity of 16GB each.

3.1 Test Sequences

Thirteen 10-bit and 8-bit YCbCr 4:2:0 video sequences (from JVET test sequence set) at different resolutions, each 10 second long, are chosen as the test set. The test set captures all the scenarios commonly found in video content (fast/slow motion, camera panning, scene fades, scene cuts and variation in illumination). Table 1 lists the sequences, corresponding resolutions, frame rates, frame counts and bitdepth.

Class	Sequence name	Resolution	Frame count	Frame rate	Bit depth
A2	CatRobot1	4K	300	60	10
A2	ParkRunning3	4K	300	50	10
B	MarketPlace	1080p	600	60	10
B	RitualDance	1080p	600	60	10
B	Cactus	1080p	500	50	8
B	BasketballDrive	1080p	500	50	8
B	BQTerrace	1080p	600	60	8
C	BQMall	832x480	600	60	8
C	PartyScene	832x480	500	50	8
D	BQSquare	416x240	600	60	8
D	BlowingBubbles	416x240	500	50	8
E	FourPeople	720p	600	60	8
E	KristenAndSara	720p	600	60	8

Table 2: Test Data Set for FastVDO Comparison, drawn from the JVET common test conditions (CTC).

3.2 Encoder Configuration

The software versions used are: a) HM16.18 (HEVC) b) AV1 version 1.0 and c) BMS 1.0 (VVC). The encoders are all configured for random access coding (where intra predicted keyframes are inserted at regular intervals, usually 1 second intervals, to enable playback from specific points) with internal bitdepth set as 10. The hierarchical group of pictures (GOP) size is set as 16 frames for BMS and HM. This means that an inter predicted keyframe is inserted every 16 frames. All frames in between the intra keyframe and inter keyframe are bi-predicted using a hierarchical structure. AV1 does not use the concept of GOP, but instead uses alternative reference frames (ARF) and golden frames to achieve bi-prediction. The ARF and golden frame distance is set as 16 to imitate the GOP structure. Constant quality (QP) encoding was performed with the HM and BMS encoders, whereas both QP and target bitrate (VBR) encoding was done for AV1. For AV1, moreover, the cpu-used parameter was set as 0 and the number of passes set as 2. More details on the command line arguments used can be found in the following sections.

3.2.1 Constant Quality Coding (QP)

HM16.18: Each sequence is encoded at four specific quality parameters (QP's); 22, 27, 32, 37. Example command line arguments for HM encoding & decoding in a Windows system:

Encode: TAppEncoder.exe -c encoder_randomaccess_main.cfg -i <input.yuv> -q <QP> -ip <intra period> -f <number of frames> -fr <frame rate> -b <encoded file>

Decode: TAppDecoder.exe -b <encoded file> -o <output reconstructed file>

BMS 1.0: Each sequence is encoded at four specific quality parameters (QP's); 22, 27, 32, 37. Example command line arguments for BMS1 encoding & decoding in a Windows system:

Encode: EncoderApp.exe -c encoder_randomaccess_main.cfg -i <input.yuv> -q <QP> -ip <intra period> -f <number of frames> -fr <frame rate> -b <encoded file>

Decode: DecoderApp.exe -b <encoded file> -o <output reconstructed file>

AV1 1.0: Each sequence is encoded at four specific quality parameters (QP's); 23, 31, 39, 47. Example command line arguments for AV1 encoding & decoding in a Windows system:

aomenc.exe --cpu-used=0 --tune=psnr -b 10 --input-bit-depth=8 --threads=0 --profile=0 --width=1920 --height=1080 --fps=50/1 --lag-in-frames=19 --min-q=31 --max-q=39 --auto-alt-ref=1 --passes=2 --kf-max-dist=48 --kf-min-dist=48 --min-gf-interval=16 --max-gf-interval=16 --drop-frame=0 --static-thresh=0 --bias-pct=50 --minsection-pct=0 --maxsection-pct=2000 --arnr-maxframes=7 --arnr-strength=5 --sharpness=0 --undershoot-pct=100 --overshoot-pct=100 --frame-parallel=0 --tile-columns=0 --end-usage=q --cq-level=31 input.yuv -o <encoded file>

aomdec.exe --rawvideo -o <output reconstructed file> <encoded file>

3.2.2 Target Bitrate Coding (VBR)

3.2.2.1. AV1. In VBR mode the AV1 encoder is configured to get bitrates close to those obtained in the QP mode for the other codecs, so that the results can be compared. Example command line arguments for AV1 encoding & decoding in a Windows system is given below. We highlight that we use –cpu-used=0, rather than =1, as used by HHI (a faster mode):

aomenc.exe **--cpu-used=0** -b 10 --input-bit-depth=8 --threads=0 --profile=0 --width=1920 --height=1080 --fps=50/1 --lag-in-frames=19 --min-q=0 --max-q=63 --auto-alt-ref=1 --passes=2 --kf-max-dist=150 --kf-min-dist=0 --drop-frame=0 --static-thresh=0 --bias-pct=50 --minsection-pct=0 --maxsection-pct=2000 --arnr-maxframes=7 --arnr-strength=5 --sharpness=0 --undershoot-pct=100 --overshoot-pct=100 --frame-parallel=0 --tile-columns=0 --end-usage=vbr --target-bitrate=<in_kbps> input.yuv -o <encoded file>

aomdec.exe --rawvideo -o <output reconstructed file> <encoded file>

3.2.2.2. x265. Example command line arguments for x265, in this instance for 10-bit input video.
x265.exe -p placebo --pass 2 --input-depth 10 --profile main10 --fps 50 -f 50 --bitrate 2000 --max-crf 51 --min-crf 0 --no-wpp --tune psnr --minkeyint 48 --keyint 48 --input-res 1920x1080 --input input.yuv -o output.bin

TAppDecoder.exe -b <output.bin> -o <recon.yuv>

4 RESULTS & SUMMARY

The modified Bjøntegaard Delta Rate (BDRATE) [10] metric - which is a measure of the integral of the rate difference between two rate-distortion (RD) curves, is used for objective evaluation. Negative values of BDRATE indicate a bitrate gain in percentage. The results from HM16.18 constant quality (QP) encoding is used as the reference. RD curves are also plotted for selected sequences. Cropped regions of the selected sequences are shown at full resolution as part of the subjective evaluation.

4.1 Constant Quality Coding (QP)

In Table 3 the BDRATES obtained with AV1 and BMS1 (VVC) are tabulated (1s test for AV1, full test for BMS). Overall PSNR is used for BDRATE calculations and RD plots. We note that constQ testing is important in that tests core toolsets, indicating the potential of each codec, and not rate control, which is an encoder only technology that is available to any encoder. Of course, in real applications, rate control is almost always used, so that testing for target rate is more directly indicative of performance in practice. From our point if view, therefore, it is useful to test by each method to obtain an overall picture.

	OVER HM-16.18		
	Y	U	V
Class A2	23.49%	42.23%	29.31%
Class B	16.88%	26.86%	26.51%
Class C	21.13%	31.50%	37.79%
Class D	16.14%	22.46%	24.25%
Class E	24.33%	35.21%	41.33%
Overall	20.39%	31.65%	31.84%

	BMS 1.0 over HM 16.18			
	Y	U	V	EncT
Class A2	-27.76%	-35.13%	-30.39%	825%
Class B	-21.84%	-36.10%	-36.53%	850%
Class C	-18.52%	-28.61%	-31.15%	1012%
Class D	-17.97%	-26.69%	-28.24%	930%
Class E	-17.95%	-27.00%	-29.02%	803%
Overall	**-20.81%**	**-30.71%**	**-31.07%**	**884%**

Table 3: BDRATES for constant quality mode; a) AV1 vs HM16.18 (for 1s), 22X runtime; b) BMS1 (VVC) vs HM16.18 (full 10s). The BMS vs HM results are extracted from extensive JVET results [13]. Thus, in constantQ testing, HEVC HM16.18 sits roughly halfway between AV1 and JVET BMS 1.0. We remark that in practice, rate control is widely used. But constantQ testing remains important in that it tests the performance of the core tool sets, apart from rate control.

The rate distortion curves for two example sequences (10-bit 4K CatRobot, 8-bit 1080p Cactus) are given in Figure 4.

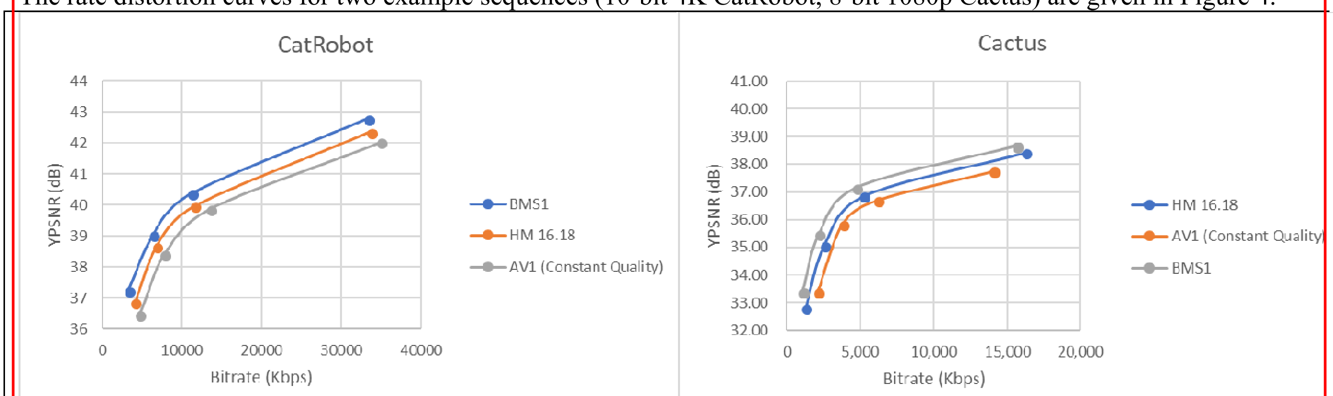


Figure 4: Example RD curves in constant quality mode, for two example sequences, where HM outperforms AV1.

From the BDRATE tables and curves it can be seen that BMS1 (VVC) outperforms HEVC, which in turn modestly outperforms AV1, in constant quality mode. The average encoding time factors for AV1 and BMS1 (VVC) are about 20x and 9x respectively, relative to HEVC. That is, AV1 is significantly more complex to encode than others.

Figure 2 shows a frame from the original CatRobot (3840x2160) sequence (scaled to fit the page) and cropped sections (original size) of the reconstructed frame encoded using the three encoders in constant quality mode. Increased levels of blurring and edge artifacts are visible in the reconstructed frame using AV1 and HEVC.

Figure 5a): Original frame from CatRobot sequence. We will zoom in on the robot, and the card in hand.

Figure 5b): Cropped section of a frame from CatRobot encoding using, from left-to-right, VVC BMS1, HEVC HM16.18, and AV1, respectively, in constant Q mode, showing decreasing subjective visual quality (see fingers, text).

4.2 Target Bitrate Coding (VBR)

In Tables 4, 5, the BDRATES obtained with AV1 over HM16.18 are presented (with 1s and full 10s testing). Since HM does not have an effective rate control, the constant-Q HM bitrates were used to drive the target bitrates for AV1.

	OVER HM 16.18		
	Y	U	V
Class A2	-10.43%	-18.14%	-17.05%
Class B	-7.53%	-17.59%	-16.91%
Class C	-6.49%	-14.69%	-15.18%
Class D	-5.19%	-14.11%	-14.03%
Class E	-5.29%	-12.96%	-12.87%
Overall	-6.99%	-15.50%	-15.21%

Table 4: BDRATES (AV1 vs HM16.18) for target bitrate mode, 1s test. We find AV1 about 7% more efficient than HEVC HM for VBR, but at an encoder runtime of nearly 30X over the HM reference software (let alone x265).

	OVER HM 16.18		
	Y	U	V
Class B	-6.68%	-17.72%	-15.07%
Class C	-5.73%	-12.88%	-14.45%
Class D	-5.57%	-14.06%	-11.91%
Class E	-6.55%	-12.49%	-11.95%
Overall	-6.13%	-14.29%	-13.34%

Table 5. BDRATES (AV1 vs HM16.18) for target bitrate mode, in a full 10s sequence test. To save computation, since we are using the slowest mode of AV1, the Class A2 (4K) was left out. Nevertheless, the consistency of the results between the 1s and 10s tests strongly supports the conclusion that AV1 slightly outperforms HM16.18. Moreover, we expect better results at 4K. This also shows that quick results for 1s tests are still indicative, and consistent with the 10s tests. Runtimes also remain consistent.

The rate distortion curves for two example sequences (CatRobot, Cactus) are given in Figure 6.

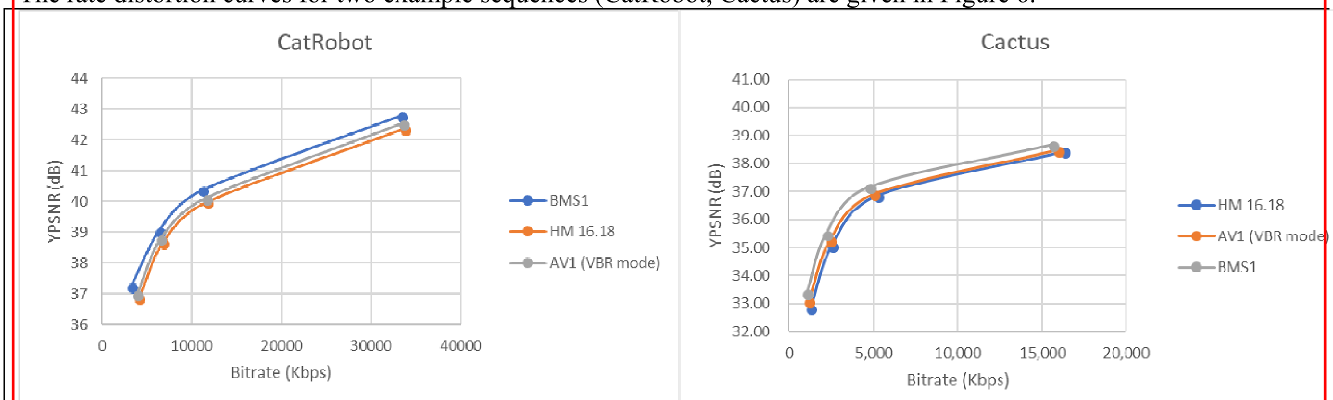


Figure 6. Performance comparison of BMS1, HM, and AV1, in target bitrate mode on two example sequences.

From the BDRATE tables and curves it can be inferred that AV1 is comparable (or in fact slightly superior) in coding efficiency with HM16.18 (HEVC) in target bitrate mode. Since HM itself does not have an effective rate control, this test was conducted as follows. We took the constant Qp results from HM tested early, and then set rate control on AV1 to match the rates. Figure 7 shows a cropped section of the reconstructed frame from CatRobot encoded using AV1 in target bitrate mode. The visual quality of this region is comparable to or slightly better than that illustrated in Figure 5b (center) for HEVC (HM16.18). The average encoding time factor for AV1 is about 29x relative to HEVC in this mode.

Figure 7. Cropped section of a frame from CatRobot sequence reconstructed after encoding with AV1 under target bitrate mode (VBR). Note that the subjective visual performance is now on par with HEVC.

	OVER x265 v2.4			
	Y	U	V	
Class A2	-20.81%	-36.25%	-33.37%	
Class B	-17.47%	-36.14%	-33.42%	
Class C	-14.62%	-27.86%	-27.76%	
Class D	-13.50%	-27.56%	-27.05%	
Class E	-13.34%	-24.63%	-24.53%	
Overall	-15.95%	-30.49%	-29.23%	

Table 6. Performance of AV1 over x265 in our VBR test (for 1s), achieving about 16% performance gain over x265, and nearly 21% for the 4K Class A2 data. These results are consistent with the results of AOM members in [6], noted in figure 1. Slight differences can be due to using different test sequences, and limited 1s test here. Note that we did not test screen content sequences.

	OVER x265 v2.4			
	Y	U	V	
Class B	-19.34%	-25.35%	-24.98%	
Class C	-13.61%	-20.05%	-22.15%	
Class D	-13.14%	-25.50%	-19.84%	
Class E	-17.12%	-23.71%	-28.24%	
Overall	-15.80%	-23.65%	-23.80%	

Table 7. Performance of AV1 over x265 in our VBR test (full 10s), excluding the 4K A2 sequences for expedience, showing an average of 15.8% on the Classes B-E, quite consistent with the 1s test; thus, even a 1s test is indicative of performance. Furthermore, the fact the Class B results improved for the 10s suggests that the 1s test for the 4K A2 may also be an underestimate. Runtimes not compared.

5 CONCLUSIONS

It can be concluded that the VVC reference software (e.g., BMS1) performs better in terms of coding efficiency when compared to HEVC reference software (HM16.18) and AV1 (under both constant quality and target bitrate mode). AV1 is slightly better in coding efficiency than HEVC HM, and even more against x265, when adhering to a target bitrate. The HEVC HM encoding is much faster compared to VVC (~9x for BMS1) and AV1 (~23-29X), but has performance better in constQ, slightly worse in target bitrate. Our performance results are slightly in contrast to, but still compatible with, those of both [6] and [11]. Compared to [11], we find that in VBR mode, AV1 is about 7% more efficient than HEVC HM on our test sequences. This is likely due both to a change in encoder settings (we use cpu-used=0, HHI uses 1, a faster mode), and different test data. Compared to [6], since [6] only tests with x265 as a representative of HEVC, whereas we (and HHI in [11]) test with HM ref. sw, a more powerful encoder than x265, we again get different results. However, note that in tables 6, 7, we also test AV1 against x265 in target bitrate, and obtain results consistent with those of [6], first noting that our test data are quite different, and we do not test on screen content sequences, for which x265 has no supported tools (though such tools are in the HEVC standard, and in the HM reference software). We further note that testing on screen content data is in fact reasonable, as it is an important use case (esp. for YouTube), though somewhat less reasonable for a comparison point. Nevertheless, we observe that from the point of commercially deployed codecs, x265 remains the only comparator for AV1, even though HM performs better. We thus conclude that these three sets of test results, [6], [11], and ours, are consistent. The differences can be reasonably accounted for by encoder settings and test data. Our runtimes are also consistent to the HHI results in [11], noting config. changes.

6 REFERENCES

[1] 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. VideoTechnol., vol. 22, pp. 1648-1667, 2012

[2] J.-R. Ohm , G. J. Sullivan , H. Schwarz , T. K. Tan and T. Wiegand, "Comparison of the coding efficiency of video coding standards-Including High Efficiency video coding (HEVC)", IEEE Trans. Circuits Syst. Video Technol., vol. 22, pp. 1669-1684, 2012

[3] F. Bossen , B. Bross , K. Sühring and D. Flynn, "HEVC complexity and implementation analysis", IEEE Trans. Circuits Syst.Video Technol., vol. 22, pp. 1684-1695, 2012

[4] B. Bross, J. Chen, S. Liu, "Versatile Video Coding (Draft 2)", ISO/IEC doc. JVET-K1001, Ljubljana, SI, 10–18 July.

[5] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu and R. Bultje, "The latest open-source video codec VP9-an overview and preliminary results", PCS, pp. 390-393, 2013

[6] Y. Chen et al, "An overview of core coding tools in the AV1 video codec". In: Picture Coding Symposium (PCS), 24–27 June 2018, San Francisco.

[7] D. Grois, T. Nguyen, D. Marpe, "Coding Efficiency Comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC Encoders," Picture Coding Symposium, Nuremburg, DE, Dec., 2016.

[8] P. Topiwala, W. Dai, M. Krishnan, "Performance Comparison of AV1, HEVC, and JVET Video Codecs on 360 (Spherical) Video," SPIE Int'l Symposium, San Diego, CA, Aug., 2017.

[9] D. Grois, T. Nguyen, D. Marpe, "Performance comparison of AV1, JEM, VP9, and HEVC encoders," SPIE Int'l Symposium, San Diego, CA, Aug., 2017.

[10] G. Bjøntegaard, "Calculation of average psnr differences between rd-curves," VCEG-M33, 13th VCEG meeting, Austin, Texas, March 2001.

[11] T. Nguyen, D. Marpe, "Future Video Coding Technologies: A Performance Evaluation of AV1, JEM, VP9, and HM," PCS 2018, San Francisco, June, 2018.

[12] HHI Blog. https://www.hhi.fraunhofer.de/en/departments/vca/research-groups/image-video-coding/research-topics/hierarchical-prediction-structures.html

[13] F. Bossen et al, "AHG report: Test model software devel. (AHG3)," JVET-K0003, Ljubljana, SI, 10-18 July, 2018.