ANALYSIS OF THE EMERGING AOMEDIA AV1 VIDEO CODING FORMAT FOR OTT USE-CASES

Sriram Sethuraman, Cherma Rajan A., Kaustubh Patankar

Ittiam Systems (P) Ltd., Bangalore, India

ABSTRACT

To achieve improved quality of service and efficiencies with higher resolution, high dynamic range (HDR), and wide color gamut (WCG) in online media delivery, there is a need for advanced video compression standards. Alliance for Open Media (AOM), a joint development foundation, is targeting AOMedia Video 1 (AV1) as a royalty-free video coding format that is likely to be finalized before the end of 2017. Built on top of Google's VP9 codec, AV1 has brought in new coding tools from other open source royalty-free codecs such as Google's VP10, Cisco's Thor, and Mozilla/Xiph.org's Daala. Multiple new royalty-free coding tools have been contributed by members of AOM. Tools that have been legally vetted and technically provide good coding gain to decoding complexity trade-off have been consolidated as default tools, while many other tools are still available as experimental tools. In this paper, we provide an analysis of the coding gains offered by the default and experimental tools and the corresponding decoding complexity increase for the over-the-top (OTT) adaptive bit-rate streaming delivery use-cases. Also, the coding gains are compared against the compression performance of x264, x265, and libVP9 open-source codecs to highlight the potential bit savings possible with AV1 when migrating from these previous generation options using multiple video quality metrics such as PSNR, SSIM, and Netflix's VMAF metric. These results indicate that AV1 offers competitive compression performance over H.265 without significantly increasing the decoding complexity. Though the encoding complexity at this stage is several factors higher, multiple encoding presets exist that trade encoding complexity for reduced compression gain.

1. INTRODUCTION

The ability to watch personalized content anytime and anywhere at ever increasing resolution, frame-rate, and quality has resulted in online media delivery gaining momentum over traditional broadcast delivery. In spite of the progressively increasing broadband bandwidths to subscribers, the cost of unicast delivery to the online media delivery service providers of a large array of titles in ultra-high-definition (UHD) HDR-WCG necessitate higher video compression. While the H.265/HEVC standard offers in excess of 30% bit savings over the well-entrenched H.264/AVC standard [11], its adoption is mired in unclear royalty payments to organizations with essential patents. In September 2015, AOM embarked on developing a royalty-free standard that achieves a compression efficiency that far exceeds that of H.265/HEVC [1]. The codebase used as the starting point for developing the first version of the standard was Google's VP10 that had already extended Google's libvpx codec with new coding tools [2]. Aspects from other royalty-free codecs such as Thor [7] and Daala [8] were integrated into this codebase. In addition, many new tools have been experimentally integrated into it. The tools are legally studied to ensure that they are good candidates for a royalty-free standard. As of April 2017, several of these tools that have offered good compression gains without disproportionate increase in decoding complexity have been enabled by default, while the remaining experimental tools are being refined. This paper presents an analysis of the compression efficiency offered by the different tools in the default set and in the experimental set. These are compared against the popular open-source implementations of H.264, H.265, and VP9, namely, x264, x265, and libvpx [3, 4, 5]. Though some comparison in this regard has been done in [6], since newer tools have been consolidated recently, and also conditions like intra-period are different, this analysis is expected to provide different results compared to that.

A brief overview of the key new tools are presented in section 2. In section 3, intra-coding improvements in AV1 are presented and compared against VP9. In section 4, coding improvements for a typical recipe used for OTT delivery are considered. In section 5, the decoding complexity increase brought about by different tool combinations are studied.

2. NEW CODING TOOLS IN AV1

Table 1 provides a summary of the new coding tools in AV1 when compared to VP9. These tools are grouped into categories that indicate the domain of applicability for the coding tool such as intra/inter prediction, transform coding, or in-loop filtering. Tools that are already enabled by default have been *italicised*. The other tools remain as experimental tools. It should be noted that this is not intended to be an exhaustive list of all the tools, but to call attention to the key tools. Certain tools that resulted in error during decoding have not been included.

Category	**Tool name**	**Brief description of Tool or its benefit**
Super Block	Up to 128x128 luma samples	Effective for coding homogeneous motion and/or texture regions in UHD
Intra Prediction	*Extended neighbor availability*	Use of top-right superblock neighbor reference samples
	Extended Intra	65 angular intra prediction modes
	Alt_intra	A new prediction mode suitable for smooth regions
	Chroma from luma	Deriving a prediction for chroma intra residuals from decoded luma intra residuals
	Filter_intra	Interpolate the reference samples before prediction to reduce the impact of quantization noise
Transform	Extended Transform	Improve energy compaction efficiency with the ability to choose different horizontal and vertical transforms from a set of 4 different transform types
	Recursive Transform	Ability to split a prediction unit adaptively based on residual properties
	Super Transform	Ability of transform to cover multiple prediction units

	Rectangular Transform	Effective when residuals across non-square prediction units of a coding unit have different properties
Entropy coding	*Multisymbol*	Non-binary symbol entropy coding to help make entropy coding/decoding faster
	Adapt (ec-adapt)	Adapts symbol probabilities on-the-fly
	New token set	A double-sided Pareto probability distribution used to generate tokens
In-loop Filtering	*Directional De-ringing Filter (DDF)*	Remove ringing artifacts due to transform and quantization
	Conditional Low Pass Filter (CLPF)	Remove artifacts introduced by quantization and interpolation filter
	Loop Restoration Filter	Remove blur artifacts due to block processing through Wiener restoration filter or self-guided restoration filter
Inter Prediction	8 Partition types + Wedge prediction (ext_inter)	More flexibility to partition close to underlying texture and motion
	Overlapped Block Motion Compensation	Reduce discontinuities at block edges due to motion compensation across the block boundary using different motion vectors
	Extended Reference Frames	Improve temporal prediction and reduce intra coding need
	Dual (interpolation) filter	Ability to choose a different interpolation filter for sub-pixel motion compensation in horizontal and vertical directions
	Global Motion / Warped motion	Captures camera induced motion seen in background regions
	REF_MV	Better methods for coding the motion vector predictors through implicit list of spatial and temporal neighbor MVs
Others	*Extended Tiles*	Allow configurable rectangular tiling of a frame with option of no dependence across tile rows within a column to improve parallelism in both encoder and decoder
	Delta quantization step	Ability to have arbitrary adaptation of quantizers within a frame; improves perceptual quality, allows fine-grain boosting, and allows tighter rate-control within a frame

Table 1: Brief Description of new coding tools in AV1 (category-wise grouping)

While a majority of the tools are aimed at improving the compression efficiency, some tools such as those in the “Others” category are aimed at a certain functionality of the encoder/decoder such as improving the degrees of parallelism or the flexibility in bit allocation.

3. PERFORMANCE OF NEW INTRA CODING TOOLS

Since intra coded blocks consume a significant percentage of the bit-rate and decide the compression efficiency in hard-to-code scenes, in this section, new coding tools related to intra-coding are studied in this section. Table 2 indicates the configuration of the different encoders used for this comparison.

x264.exe --fps 50 --frames 100 --profile high --preset veryslow --keyint 1 --qp <22,26,30,24> --aq-mode 0 --tune=psnr
x265.exe --fps 50.0 --frames 100 --profile main --preset veryslow --keyint 1 --qp <20,23,26,30> --aq-mode 0 --no-wpp --tune=psnr
vpxenc --codec=vp9 --passes=1 --best --limit=100 --i420 --profile=0 fps=50000/1000 --end-usage=q --cq-level=<30,36,42,48> --aq-mode=0 --ivf --tile-rows=0 --tile-columns=0 --kf-max-dist=1
aomenc --codec=av1 --psnr -v --passes=1 --good --cpu-used=0--limit=100 --i420 --profile=0 fps=50000/1000 --end-usage=q --cq-level=<30,36,42,48> --aq-mode=0 --ivf --tile-rows=0 --tile-columns=0 --kf-max-dist=1

Table-2: Configuration of the encoders used for analysing the intra coding performance

To measure the intra coding performance, one frame each from 100 different scenes that span a wide range of spatial complexities were used and encoded at 1080p, 720p, and 360p. Both PSNR and SSIM were measured for the luminance component and the Bjontegaard delta bit-rates (BDRATE) [9] were measured as an indicator of the bit savings possible over the previous standards.

Figure-1 summarizes the bit savings (or increase) for VP9, x265, and the AV1-cfg7 cases over x264 across the 3 resolutions considered. It is clear that the average intra coding performance of AV1 is on par or better than x265, while giving a 5-10% average bits reduction over VP9.

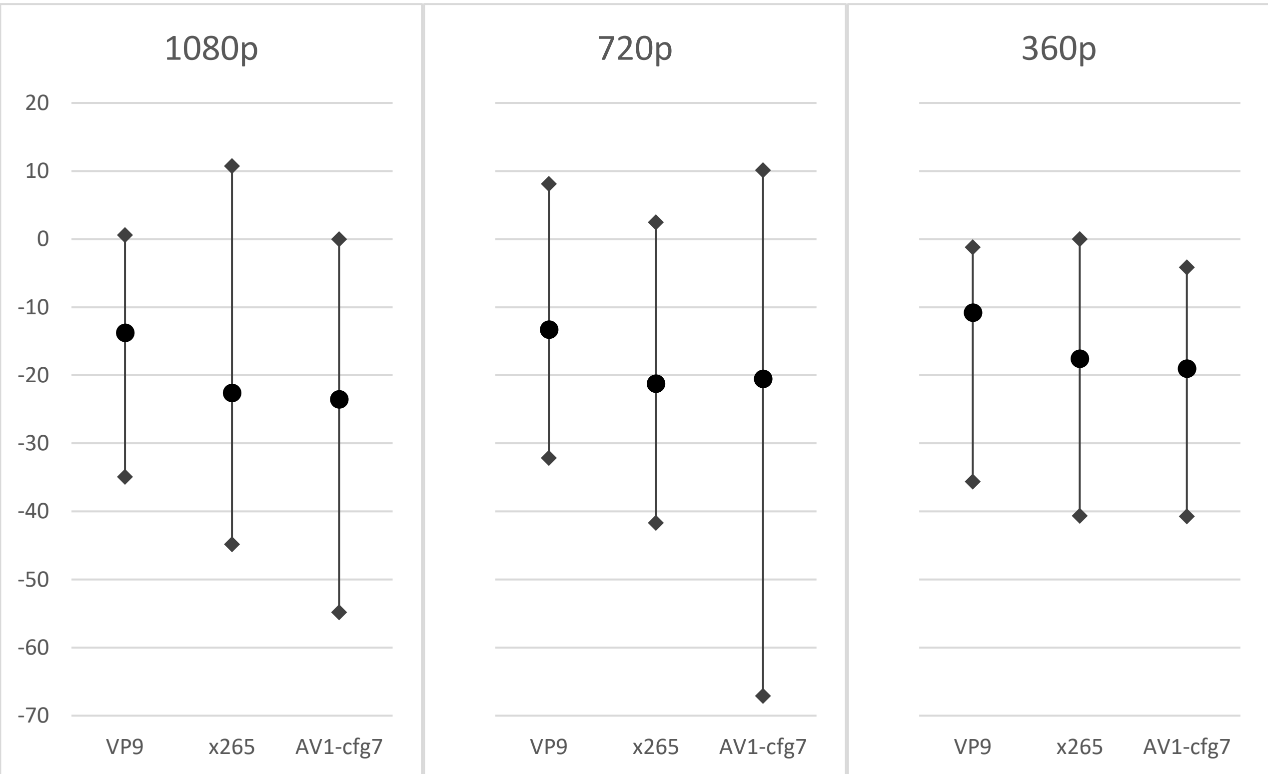

Figure-1: Intra-coding: Avg., Min., and Max. BDRATE (SSIM) gain/loss compared to x264

Table 3 shows the BDRATE gains (or losses) against VP9 when using PSNR and SSIM as the quality metrics. The table also shows 8 configurations for AV1, where a new coding tool is progressively added one at a time. While SSIM-BDRATE shows consistent gains over VP9 at all the 3 resolutions, the PSNR-BDRATE for 360p shows certain losses.

		BDRATE(SSIM-Y) %			BDRATE (PSNR-Y) %		
Configuration name	Configuration	1080p	720p	360p	1080p	720p	360p
X264	-	16.33	15.5	12.40	14.20	11.2	12.90
X265	-	-4.83	-5.10	-4.10	-7.60	-11.70	1.60
AV1-Cfg-1	Default tools –CLPF -DDF –ec-adapt -new-token-set –ext-intra	-2.57	-1.96	-1.60	-5.17	-4.37	4.07
AV1-Cfg-2	Cfg-1 + Extended Intra	-3.71	-3.45	-3.04	-6.52	-6.00	2.47
AV1-Cfg-3	Cfg-2 + Extended Transform	-5.93	-5.44	-9.06	-8.54	-8.00	-2.85
AV1-Cfg-4	Cfg-3 + Loop Restoration	-6.58	-6.10	-5.54	-9.39	-8.77	-1.25
AV1-Cfg-5	Cfg-4 + CLPF + DDF	-6.59	-6.12	-6.13	-9.44	-8.84	-0.87
AV1-Cfg-6	Cfg-5 + ec-adapt + new-token-set	-7.76	-7.14	-7.02	-9.95	-9.21	-0.94
AV1-Cfg-7	Cfg-6 + Filter Intra	-8.01	-7.39	-7.19	-10.25	-9.53	-1.15

Table-3: BDRATE gain/loss for intra-coding toolsets compared to VP9

4. PERFORMANCE FOR OTT USE-CASES

In this section, AV1's default toolsets (AV1-Default) and a set of experimental tools added over the default tools (AV1-Exp) were compared against x264. Specifically, the experimental tools added over the default tools were chroma-from-luma, filter_intra, loop restoration filter, and ext_inter. Table 4 below provides the configurations used for the different encoders. 4 different typical test sequences at 1080p@50fps, each with 100 frames, were used for the comparison. A 2 second intra-period was used to correspond to typical OTT minimum segment durations.

x264.exe --preset veryslow --profile high --input-res 1920x1080 --fps 50.0 --qp <36,38,40,42> --rc-lookahead 100 --keyint 100 --frames 500 --threads 1 --tune psnr (modified to enable mb-tree in cqp mode)
./x265 --preset veryslow --profile high --input-res 1920x1080 --fps 50.0 --qp <36,38,40,42> --rc-lookahead 100 --keyint 100 --frames 500 --threads 1 --tune psnr (modified to enable cu-tree in cqp mode)
vpxenc --codec=vp9 --passes=2 --good --cpu-used=0 --limit=500 --i420 --profile=0 -w 1920 -h 1080 --fps=50000/1000 --end-usage=q --cq-level=<50,53,56,59> --aq-mode=0 --ivf --auto-alt-ref=1 --resize-allowed=0 --threads=1 --tile-rows=0 --tile-columns=0 --kf-min-dist=0 --kf-max-dist=100
aomenc --codec=av1 --passes=2 --fpf=./stat_file.stat --good --cpu-used=0 --limit=500 --i420 --profile=0 -w 1920 –h 1080 --fps=50000/1000 --end-usage=q --cq-level=<50,53,56,59> --aq-mode=0 --ivf --auto-alt-ref=1 --resize-allowed=0 --threads=1 --tile-rows=0 --tile-columns=0 --kf-min-dist=0 --kf-max-dist=100

Table-4: Configuration of the encoders used for analysing the OTT use-case

Table-5 below summarizes the BDRATE gain (or loss) over x264 using 3 different objective quality metrics, namely, PSNR-Y, SSIM-Y, and VMAF-Y. It can be seen that AV1 default gets very close to H.265 and the experimental tools have started to pull ahead of H.265. Some of the experimental tools could not be enabled in combination without having decoding errors and hence not included in the current round of comparisons.

Quality metric	Sequences	BDRATE with respect to x264			
		x265	VP9	AV1-Default	AV1-Exp
PSNR-Y	Cactus	-46.3%	-35.2%	-45.8%	-47.6%
	BasketballDrive	-56.9%	-54.4%	-60.0%	-61.2%
	CrowdRun	-32.6%	-25.0%	-32.8%	-33.9%
	ParkJoy	-34.3%	-23.8%	-33.3%	-34.9%
SSIM-Y	Cactus	-36.1%	-34.1%	-40.9%	-42.9%
	BasketballDrive	-48.1%	-54.2%	-57.2%	-58.1%
	CrowdRun	-7.4%	-13.6%	-14.1%	-15.3%
	ParkJoy	-3.8%	-7.2%	-10.8%	-12.6%
VMAF-Y	Cactus	-55.4%	-37.5%	-51.1%	-50.1%
	BasketballDrive	-61.7%	-56.1%	-64.0%	-63.5%
	CrowdRun	-37.2%	-18.8%	-32.3%	-29.3%
	ParkJoy	-39.9%	-23.3%	-36.5%	-34.2%

Table 5: BDRATE gain over x264 for OTT use-case for 1080p@50fps sequences

Figures 2(a) and 2(b) illustrate the rate-distortion performance of the different codecs considered for the 1080p50 BasketballDrive and CrowdRun sequences. It can be seen that AV1 has ~0.5dB PSNR gain over x265 for the BasketballDrive sequence and 0.2-0.3dB PSNR gain over x265 for the Crowdrun sequence.

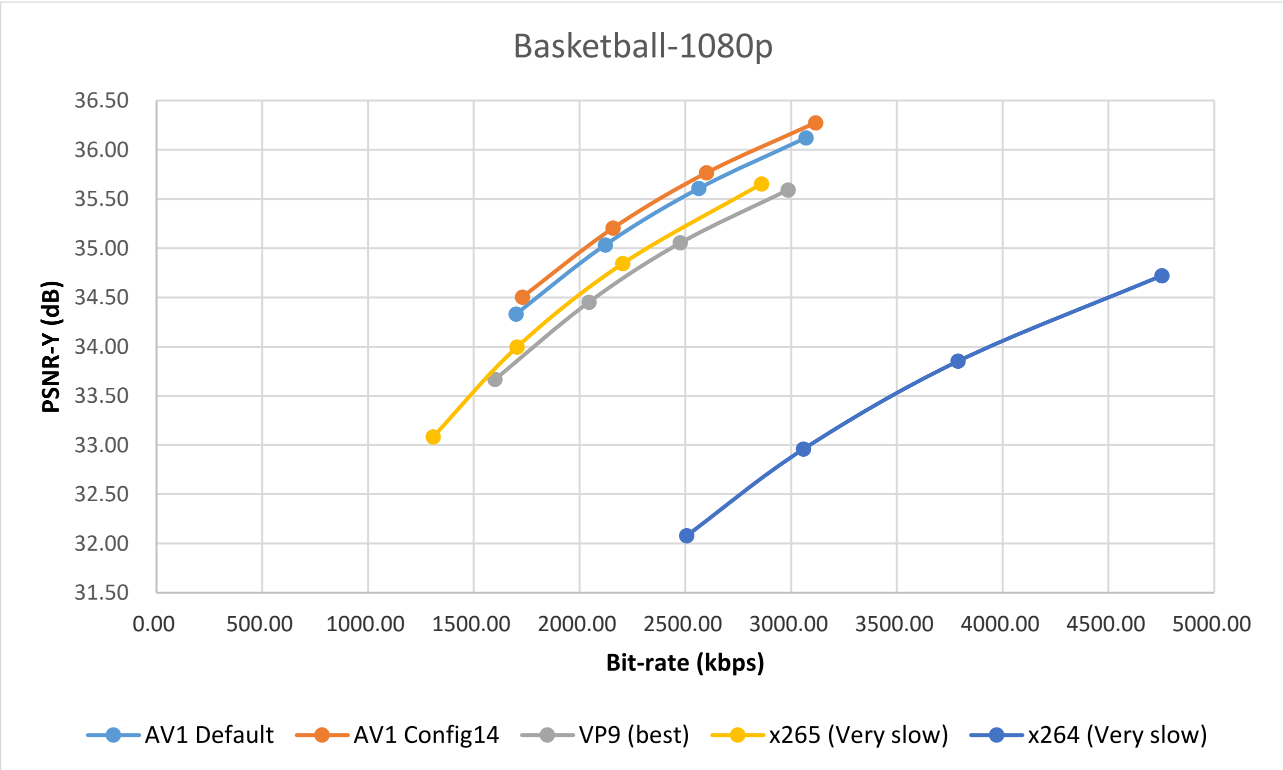

Figure 2a: Rate-Distortion plots for Basketball-1080p@50fps sequence

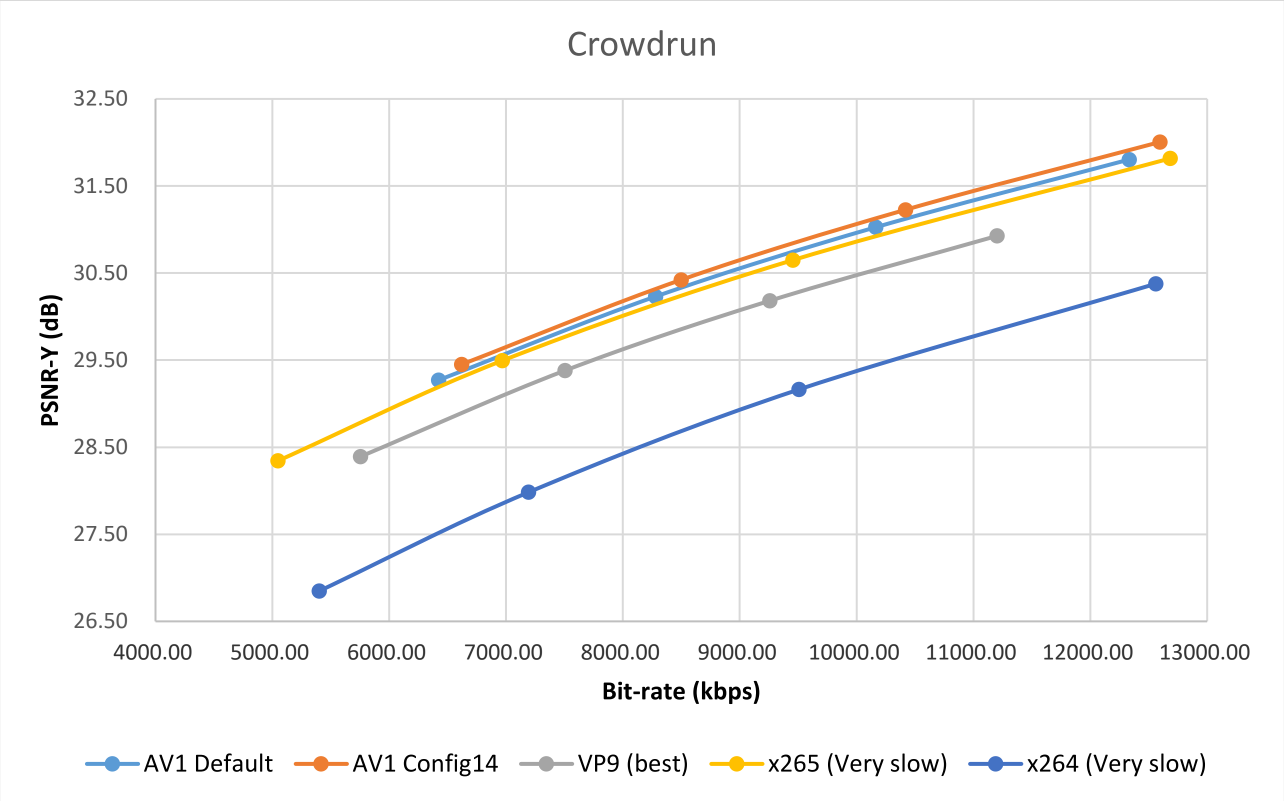

Figure 2b: Rate-Distortion plots for Crowdrun-1080p@50fps sequence

5. ANALYSIS OF DECODING COMPLEXITY INCREASE

The total decoding time across the 4 sequences coded at cq-level=53 was measured in single-thread mode on an AVX2 enabled x86 CPU clocked at 3.2GHz to understand the speed hit due to the new tools. In addition, the percentage of time spent in the different aspects of decoding were measured. Table 6 shows these results.

	VP9	AV1-Default	AV1-Exp
Total Decoding time (s)	5.74	15.584	20.19
	% of total decoding time		
Parse (non-coeffs)	11.96	11.01	10.56
Motion compensation	41.96	25.33	17.2
Coeff Decode + Inv Q	7.6	8.12	4.55
Inv Transf. + Recon	6.79	2.25	3.86
MV derivation	0.27	12.22	8.1
Deblock	21.47	18.62	17.95
CLPF + DDRF		11.61	8.25
Loop restoration			16.9
Rest	9.94	10.85	12.62

Table 6: Decoding complexity comparisons against VP9 decoder

When compared to VP9 (which only had a marginally higher decoding complexity over H.264), AV1 default tools have roughly 3x decoding time and the AV1-experimental tools have a roughly 4x decoding time. However, given that the decoder may not be fully optimized for x86 SIMD, the actual complexity is expected to be lower than what has been observed.

6. CONCLUSIONS

AV1 offers a wide range of new coding tools. The toolsets seem to have surpassed the compression efficiency of x265, VP9, and x264 for both intra coding and OTT use-cases. Though the encoding is several factors slower than VP9 and precluded a more exhaustive study that included UHD, it is anticipated that new encoding algorithms can arrive at faster methods that lose very little compression efficiency. Also, for the OTT use-case, the encoding speeds are not that critical. The decoding complexity has only increased by a factor of 3-4x when compared to VP9. Since the AV1 decoder may not be fully leveraging SIMD optimizations, the final decoding complexity is expected to be lower than the results in this paper.

REFERENCES

1. Alliance for Open Media Press release, http://aomedia.org/press-releases/alliance-to-deliver-next-generation-open-media-formats/, August 2015.
2. AV1 code repository, https://aomedia.googlesource.com/aom/
3. X264 code repository, http://www.videolan.org/developers/x264.html
4. X265 code repository, https://bitbucket.org/multicoreware/x265/wiki/Home
5. Libvpx code repository, https://github.com/webmproject/libvpx/
6. Dan Grois, et al., Coding Efficiency Comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC Encoders, Picture Coding Symposium, 2016.
7. Cisco's Thor Video Codec, IETF Internet Draft, https://tools.ietf.org/html/draft-fuldseth-netvc-thor-03
8. Mozilla's Daala Video Codec, https://xiph.org/daala/
9. Calculation of average PSNR differences between RD-curves, VCEG-M33, March 2001.
10. Useful AV1 related links, https://nwgat.ninja/test-driving-aomedias-av1-codec/
11. Jan De Cock, et al., A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications, Proc. SPIE 9971, Applications of Digital Image Processing XXXIX, Sep. 2016.