PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications

Jan De Cock, Aditya Mavlankar, Anush Moorthy, Anne Aaron

Jan De Cock, Aditya Mavlankar, Anush Moorthy, Anne Aaron, "A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications," Proc. SPIE 9971, Applications of Digital Image Processing XXXIX, 997116 (27 September 2016); doi: 10.1117/12.2238495

SPIE.

Event: SPIE Optical Engineering + Applications, 2016, San Diego, California, United States

A Large-Scale Video Codec Comparison of x264, x265 and libvpx for Practical VOD Applications

Jan De Cock, Aditya Mavlankar, Anush Moorthy, and Anne Aaron

Netflix – Video Algorithms, 121 Albright Way, Los Gatos, CA, United States

ABSTRACT

Over the last years, we have seen exciting improvements in video compression technology, due to the introduction of HEVC and royalty-free coding specifications such as VP9. The potential compression gains of HEVC over H.264/AVC have been demonstrated in different studies, and are usually based on the HM reference software. For VP9, substantial gains over H.264/AVC have been reported in some publications, whereas others reported less optimistic results. Differences in configurations between these publications make it more difficult to assess the true potential of VP9.

Practical open-source encoder implementations such as x265 and libvpx (VP9) have matured, and are now showing high compression gains over x264. In this paper, we demonstrate the potential of these encoder implementations, with settings optimized for non-real-time random access, as used in a video-on-demand encoding pipeline. We report results from a large-scale video codec comparison test, which includes x264, x265 and libvpx. A test set consisting of a variety of titles with varying spatio-temporal characteristics from our catalog is used, resulting in tens of millions of encoded frames, hence larger than test sets previously used in the literature. Results are reported in terms of PSNR, SSIM, MS-SSIM, VIF and the recently introduced VMAF quality metric. BD-rate calculations show that using x265 and libvpx vs. x264 can lead to significant bitrate savings for the same quality. x265 outperforms libvpx in most cases, but the performance gap narrows (or even reverses) at the higher resolutions.

Keywords: H.264/AVC, HEVC, VP9, x264, x265, libvpx, video codec, video compression, video-on-demand, video quality metrics

1. INTRODUCTION

Video-on-demand (VOD) streaming services account for a significant portion of internet traffic today. According to the Sandvine June 2016 report, Netflix alone is responsible for 35% of peak downstream traffic on fixed networks in North America.[1] The aggregated bandwidth share of the top four video streaming services (Netflix, YouTube, Amazon Video and Hulu) is almost 60% of downloaded bytes during peak times. As VOD services expand globally, it is expected that video content will continue to grow and dominate global internet traffic, including mobile networks.

Using more efficient video coding technologies can positively impact network congestion while ensuring good quality of experience for consumers. The goal of this study is to evaluate compression performance of practical video codecs to assess the impact on VOD streaming services. To achieve this goal, the study incorporates the following conditions:

- The evaluation is applied on video content from professional TV shows and movies, typical of the content streamed by services such as Netflix, Amazon Video and Hulu.

Further author information: (Send correspondence to Jan De Cock)
Jan De Cock: E-mail: jdecock@netflix.com
Aditya Mavlankar: E-mail: amavlankar@netflix.com
Anush Moorthy: E-mail: amoorthy@netflix.com
Anne Aaron: E-mail: aaaron@netflix.com

Applications of Digital Image Processing XXXIX, edited by Andrew G. Tescher, Proc. of SPIE
Vol. 9971, 997116 · © 2016 SPIE · CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2238495

- The results are generated using codec implementations that are open-source and readily available to the community.
- Codec parameters are chosen to closely reflect the settings used in VOD encoding pipelines. The video streams do not require real-time encoding or low-delay GOP structures. The rate control algorithms internal to the codec implementation have a significant impact on the quality of the streams.
- VOD services typically support stream switching to adapt to the viewer's bandwidth. Although the original source content is of high quality, a wide range of output resolutions and bitrates are covered by this study.

H.264/AVC is the video compression format standardized jointly by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC JTC1 Moving Picture Experts Group (MPEG) in 2003.[2,3] This standard is widely used for VOD services and has extensive decoder support on web browsers, TVs, cellular phones, tablets, set-top boxes, and other consumer devices. Two newer video compression formats – HEVC and VP9 – employ more sophisticated encoding tools (for example, larger coding units, larger transforms, more flexible intra modes, sample adaptive offset coding, etc.) and have the potential to reduce the required streaming bandwidth while maintaining the same visual quality.

HEVC is the successor to H.264/AVC, jointly developed by VCEG and MPEG in the Joint Collaborative Team on Video Coding (JCT-VC), and standardized in 2013.[4,5] Using the video sequences and testing conditions defined by JCT-VC,[6] an average bitrate savings of about 50% is observed for HEVC compared to H.264/AVC at the same quality.[7] The video clips were encoded using the H.264/AVC JM and HEVC HM reference software with fixed Quantization Parameter (QP) settings. The main focus was to investigate coding efficiency achievable by the bitstream syntax.[7] Numerous companies have announced HEVC decoder support in consumer devices such as cellular phones, game consoles, media players and graphics cards.[8]

VP9 was developed by Google as a royalty-free video compression format.[9,10] The codec implementation, libvpx, was released as open source software.[11] YouTube has used VP9 for encoding videos that account for billions of hours of viewing. According to YouTube, VP9 achieves 50% bitrate savings over legacy streams for HD and 4K. It enabled users with limited bandwidth to watch higher resolution streams, for example, upgrade from 240p low-definition to 360p standard definition.[12] Google announced that VP9 decoding support can be found in the Chrome web browser, Android devices and various TVs and game consoles.[12]

Although significant compression gains were reported for HEVC and VP9 over H.264/AVC,[7,12] the magnitude of bitrate savings do not necessarily carry over to a VOD service streaming high quality entertainment content. The experimental set-up used by Ohm et al.[7] for HEVC evaluation covered a limited set of test clips not fully representative of TV shows and movies. For example, it did not include animation, which is highly viewed by kids and adults (for example, children's cartoons, Japanese anime, adult animated sitcoms, etc.). Other conditions in the study which are impractical for VOD services are encoding with the HM reference encoder and constant QP rate control. In Robertson,[12] YouTube's reported results on VP9 bitrate savings over legacy streams are representative of streams generated from a practical encoding pipeline. However, since YouTube processes a significant amount of user-generated videos, the encode settings may be biased towards saving processing cycles. A VOD service providing viewers with high-value entertainment content can afford to spend more encoding cycles to ensure the quality of the streams. Furthermore, user-generated videos typically differ in signal characteristics from professionally captured and edited TV shows and movies.

The goal of this paper is not to assess the compression efficiency achievable by the bitstream syntax of H.264/AVC, HEVC and VP9, but instead to measure the compression performance achievable by the current codec implementations deployed into a practical production system. We assess the entirety of the codec, including algorithms related to mode decisions, frame type selection, motion estimation and rate allocation. In this study we compare three open-source codec implementations, corresponding to the three video coding formats:

- x264:[13,14] This is a well-established codec and considered the leading open-source implementation for H.264/AVC encoding. It is widely used by web services, television broadcasters and ISPs.
- x265:[15,16] x265 is the x264 code ported and adapted for encoding HEVC. It was the front runner in the 2015 Moscow State University HEVC codec comparison study[17] as described in more detail in Sec. 2.

- libvpx:[18,19] libvpx is the software package developed by Google to support VP9 (and its predecessor, VP8) encoding and decoding. Aside from libvpx, we are not aware of other open-source software implementations for VP9 encoding.

It is important to note that while x264 has been under development for more than a decade, x265 and libvpx are less mature technologies and have greater potential for future enhancements. In this paper we only report current results. Analyzing the weak points of the current codec implementations and assessing future attainable gains are outside the scope of this paper.

2. RELATED WORK

In this section we present an overview of literature that quantifies compression efficiency gains of HEVC and VP9 over previous codecs. We focus on results with experimental conditions closest to random access or entertainment applications, as opposed to real-time or interactive encoding scenarios.

In the entertainment application scenario (dyadic high-delay hierarchical prediction structures and 1-second I-frame interval), Ohm et al.[7] reported about 35.4% bitrate savings for HEVC over H.264/AVC for the same PSNR using eight 10-second test clips at 480p and 1080p resolutions. Subjective tests for nine 10-second test clips at 240p, 480p and 1080p resolutions showed 49.3% bitrate savings compared to H.264/AVC. For both cases, the video clips were encoded using the JM or HM reference software with fixed QP settings.

Hanhart et al.[20] ran subjective assessment tests on three 4K video sequences, one of which was derived from a computer-generated animated movie. The clips were encoded with 1-second I-frame intervals, at constant QP settings, using the JM or HM reference software. For the PSNR metric, HEVC resulted in 44.4% bitrate savings over H.264/AVC but subjective evaluation resulted in a more impressive 66.5% bitrate savings.

Mukherjee et al.[10] and Grois et al.[21] were two of the initial studies that aimed to compare HEVC and VP9 compression performance. Mukherjee et al.[10] evaluated x264 constant rate factor (CRF) encoding, HEVC (HM reference software) fixed QP encoding and VP9 (libvpx software) constant quality encoding. No regular I-frames were inserted. For a set of 29 publicly-available CIF and SIF sequences, VP9 showed an average of 19.6% bitrate savings compared to x264 and very close performance to HEVC. For the set of 16 HD sequences, VP9 resulted in 30.4% bitrate savings over x264 and required 2.5% more bits than HEVC for the same PSNR. Grois et al.[21] on the other hand reported less favorable results for VP9 with libvpx. Using a subset of the test sequences from the JCT-VC common test conditions,[6] results showed that VP9 required 79.4% more bits than HEVC and 8.4% more bits than x264 for the same PSNR. Řeřábek and Ebrahimi[22] focused on subjective quality evaluation of H.264/AVC JM, HEVC HM and VP9 libvpx video encodes from four 4K video sequences. The results report VP9 compression performance at par with H.264/AVC (1.6% bitrate savings for the same PSNR and 5.1% more bits for the same subjective score) and significantly worse than HEVC, with HEVC requiring 35.6% less bits for the same PSNR and 49.4% less bits for the same subjective scores. One caveat of these tests[10,21,22] is that practical codec implementations (libvpx or x264) were mixed in with reference software implementations of the standard (H.264/AVC JM or HEVC HM) so it is unclear whether the evaluation is being performed on what can be achieved by the bitstream syntax or on what can be achieved by the codec implementations.

The Moscow State University (MSU) Image and Graphics Lab conducted comparisons on HEVC codec implementations and included x264 and VP9 libvpx results in the study.[17] Twenty HD sequences were used with various testing configurations related to encoding speed. The SSIM metric was applied for quality comparison. We concentrate on the Ripping category results which had no speed requirement and is most relevant to the VOD use case. Among the HEVC codecs tested, x265 demonstrated the best compression efficiency. libvpx was a close second, outperforming the other HEVC codecs. For the Ripping Server configuration, x265 showed a bitrate savings of 26% and VP9 a bitrate savings of 23% compared to x264 at the same SSIM.

As can be seen from the review, related literature on comparisons of H.264/AVC, HEVC and VP9 report conflicting conclusions on relative efficiency. The results are highly dependent on the test sequences, codec settings and quality metrics used in the study. In this paper we generate more comprehensive results using a large dataset of videos, a wide range of bitrates and resolutions, two configurations covering two optimization criteria, multiple objective quality metrics, and the most recent stable versions of the codecs. The evaluation is performed on an encoding scenario that is most relevant to a VOD service streaming high-quality TV shows and movies.

3. TEST METHODOLOGY

This section describes the content, codec parameters, and quality measurement used in our large-scale test. To limit the scope of the test, the three codecs under study were tested using 8-bit profiles, in a two-pass variable bitrate (VBR) rate-controlled setting. Bitrates were set as described in 3.2. Each codec was tested using two sets of codec parameters. Whereas the first configuration targets optimization for PSNR, the second configuration targets perceptual quality optimization, as detailed in Sec. 3.3.

3.1 TEST CONTENT

Our test content consists of a wide selection of clips extracted from our catalog, and includes licensed, original, and open-source content. In total, 500 titles were selected, including 100 titles which have a native 4K resolution. The non-4K titles have a 1080p source resolution, with 16:9 aspect ratio. All sources have a 1:1 pixel aspect ratio, and frame rates are 23.97, 24, 25, 29.97, or 30 fps.

Licensed and original titles include movies and episodes of popular shows such as *House of Cards*, *Orange is the New Black*, *Stranger Things*, *Daredevil*, and *Narcos*.

From each title, ten clips were randomly extracted, leading to 5000 clips overall. We avoided extracting clips from the beginning and end of the titles, to exclude any opening scenes (e.g. Netflix logo) and rolling credits in the selected clips. All clips are 12 seconds in length.

In Fig. 1, we show scatter plots of the spatial and temporal information[23] for all 5000 clips. Since scene/shot changes can occur within the selected clips, we adjust the formulas for spatial and temporal information to use the median, instead of the maximum value of the standard deviation, leading to the following formulas:

$$SI = median\{\sigma(Sobel(F_n))\}$$

$$TI = median\{\sigma(M_n(i,j))\}$$

where the difference $M_n(i,j)$ between successive frames $F_n(i,j)$ is calculated as follows:

$$M_n(i,j) = F_n(i,j) - F_{n-1}(i,j) \ .$$

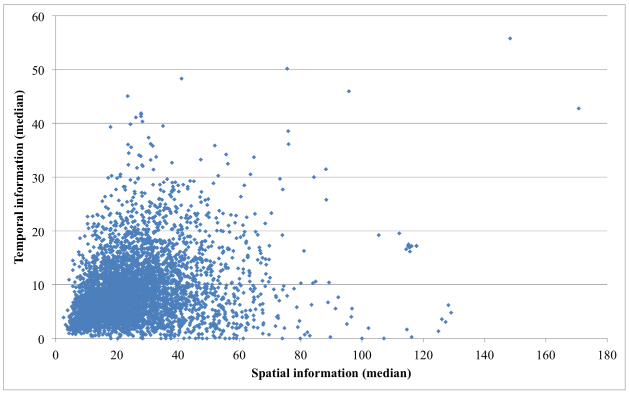
Fig. 1a shows a depiction of all (SI,TI)-combinations, including outliers with extremely high spatial (e.g. containing noise patterns) and temporal content. We additionally zoom in on the region bounded by (SI,TI) pairs with $SI \leq 60$ and $TI \leq 30$ (Fig. 1b), which excludes the outliers, but still includes sequences with very high SI and TI values, according to the above definitions. The plot indicates a wide spread of content, including slow and high-motion scenes, and clips with low and high spatial complexity. Although more clips are found in the lower SI/TI region, we believe these 5000 clips are a representative sample of a VOD streaming catalog.

3.2 RESOLUTION, GOP AND BITRATE SETTINGS

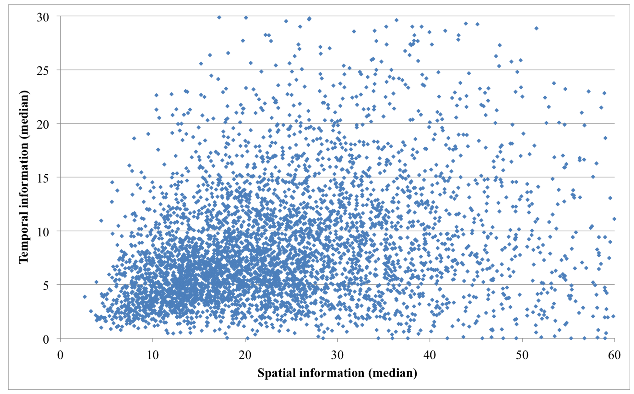
All clips were encoded at the SD and HD resolutions of 640x360, 1280x720, and 1920x1080. An output 16:9 display aspect ratio was maintained in all cases. For a subset of the 4K sources which were not perfectly 16:9 in display aspect ratio, padding was applied for simplicity in the output resolutions.

Since we are focusing on VOD applications, adaptive streaming constraints are enforced. In particular, for random access purposes, a fixed keyframe (IDR) interval of exactly four seconds is used. Each codec is allowed to use a flexible (e.g. hierarchical) GOP structure in between key frames. For x264 and x265, the maximum number of successive B-frames (up to 16) is allowed, in a 'non-strict' hierarchy with hierarchical QP cascading. For libvpx, alt-ref frames can be flexibly set, with a recommended 'lag-in-frames' value of 16.

To determine the bitrates in a content-adaptive way, the following ('three-pass') procedure was used: (i) In a first step, each clip is encoded using eight CRF (x264/x265) or constant QP (libvpx) values; (ii) In a second



(a) (SI,TI)-combinations for all clips.



(b) (SI,TI)-combinations for clips with $SI \leq 60$, $TI \leq 30$.

Figure 1: Scatter plots showing spatial and temporal information for the 5000 encoded clips.

step, the output bitrates of step 1 are used to drive the two-pass VBR encode. This procedure ensures that we avoid over-allocation of the bitrate for easy clips and under-allocation for complex clips, and still benefit from 2-pass rate control. For the x264 and x265 encodes, eight CRF (QP) values were used, to cover a wide span of bitrates:

$$QP_{AVC} = QP_{HEVC} = \{15, 19, 23, 27, 31, 35, 39, 43\} \ .$$

For libvpx, larger increments of the QP are needed to cover the same quality range as x264 and x265. The following eight constant QP values were used, to match the x264/x265 quality range as closely as possible:

$$QP_{VP9} = \{14, 21, 28, 35, 42, 49, 56, 63\} \ .$$

These QP values cover a wide spread of bitrates, and guarantee a large overlap in BD-rate calculation. To provide us with more insight on particular quality ranges, additional Bjontegaard-Delta-rate (BD-rate)[24] measurements are performed on low, medium, and high bitrate 'subranges', as detailed in Sect. 3.5.

3.3 CODEC PARAMETERS

The most recent versions of the three codecs (at the time of writing) were used: x264 release 2705,[14] x265 stable release 2.0,[16] and libvpx tag 1.6.0.[11] Settings were chosen that reflect the current capabilities of each codec, by using 'high-complexity' settings, and enabling the fullest range of coding modes available. Although certain settings can be made more exhaustive (e.g. a larger set of reference frames, or an even larger motion estimation (ME) search range for x264 and x265), we increased the complexity of settings to a point where diminishing returns are observed, and where the overall results and observations of this comparison test are only marginally affected.

The settings for x264 and x265 use the 'placebo' settings as a starting point. Motion estimation in both x264 and x265 is performed according to the simplified uneven cross multi hexagon grid search (UMH) algorithm.[25] Four reference frames are used for both codecs. For the motion estimation search range, a compromise was made to increase the range from a default of 24 (placebo preset in x264) or 92 (placebo preset in x265) to $frame\_width/8$. Further increase of the ME search range beyond these values results in limited gains, but significantly increases computational complexity. libvpx streams are created using the 'best' setting, with adaptive alt-ref placement and default (recommended) settings.

To exclude possible negative impact of multi-threading approaches on coding efficiency, all codecs were run in single-threaded mode, with one slice/tile per frame, and (for x265) wavefront parallel processing disabled. This significantly increases the runtime of each encode, but avoids potential rate-distortion losses.

For each codec, two configurations were evaluated:

- PSNR-tuned (PT) Configuration: optimizing for *PSNR*, disabling adaptive quantization and (if available) psychovisual options.
- Visual Quality-tuned (VQ) Configuration: optimizing for *visual quality*, enabling adaptive quantization and recommended psychovisual options.

In this comparison, we do not set video buffer verifier (VBV) constraints as defined in the Hypothetical Reference Decoder sections of the H.264/AVC and HEVC specifications.[2,4] As of writing, the VP9 specification does not describe similar well-defined VBV constraints.

A summary of codec settings can be found in Table 1-Table 3.

Table 1: Codec settings for x264.

	Configuration PT	Configuration VQ
profile	High	High
preset	placebo	placebo
keyint	4s	4s
min-keyint	4s	4s
ME algorithm	UMH	UMH
ME search range	$width/8$	$width/8$
ref	4	4
partitions	all	all
threads	1	1
subme	9	10
aq-mode	0	1
aq-strength	0.0	1.0
psy-rd	0.0	1.0

Table 2: Codec settings for x265.

	Configuration PT	Configuration VQ
profile	Main	Main
preset	placebo	placebo
keyint	4s	4s
min-keyint	4s	4s
ME algorithm	UMH	UMH
ME search range	$width/8$	$width/8$
subme	5	5
ref	4	4
pools	none	none
wpp	0	0
aq-mode	0	2
aq-strength	0.0	1.0
psy-rd	0.0	2.0
psy-rdoq	0.0	1.0

3.4 QUALITY MEASUREMENT

In order to compare the different codecs in this study, we utilize a variety of full-reference quality metrics, ranging from traditional metrics such as PSNR to perceptual quality metrics such as SSIM and the recently introduced

*Note that tuning for SSIM is not supported in libvpx v1.6.

Table 3: Codec settings for libvpx.

	Configuration PT	Configuration VQ
profile	0	0
complexity mode	best	best
end-usage	vbr	vbr
kf-max-dist	4s	4s
kf-min-dist	4s	4s
lag-in-frames	16	16
auto-alt-ref	1	1
arnr-maxframes	7	7
arnr-strength	5	5
arnr-type	3	3
static-thresh	0	0
frame-parallel	0	0
threads	1	1
tile-columns	0	0
tune	psnr	-*
aq-mode	0	1

'meta-metric' VMAF. While PSNR has traditionally been used in studies of this nature, especially for codec standardization, and the PSNR-based BD-rate[24] has been utilized as a measure of performance,[17,21,22] PSNR has often been criticized for its poor correlation with human perception.[26] Further, the codecs in this study have specific toolsets that optimize for psychovisual perception such as the adaptive quantization mode described above. Finally, while there is a general consensus that PSNR correlates poorly with visual perception, there does not seem to be a single quality metric that is capable of universally replacing PSNR. Thus, in order to facilitate objective comparison, apart from PSNR, we utilize a series of popular, full-reference, visual quality metrics. The quality metrics used in this study are listed below.

1. Mean PSNR: PSNR Computed per-frame and averaged across frames. While a popular choice, such averaging of PSNR tends to hide quality fluctuations since it favors frames with lower MSE[†].

2. $PSNR_{\overline{MSE}}$: PSNR computed after averaging the mean-squared-error (MSE) across frames. While this measure is not typically used, we believe that such computation ameliorates the 'masking' of quality variations seen in the mean PSNR and truly reflects the quality using the MSE as a quality measure.

3. Structural Similarity Index (SSIM):[27] A popular image quality metric which utilizes local luminance, contrast and structure measures to predict quality.

4. Multi-scale Structural Similarity Index (MS-SSIM):[28] A multi-scale extension of the SSIM metric that has been shown to perform well in terms of correlation with human perception.

5. Visual Information Fidelity (VIF):[29] Another image quality metric that computes the loss of information between source and distorted videos in the wavelet domain after performing a perceptually motivated divisive normalization process.

6. Video Multimethod Assessment Fusion (VMAF):[30] A video quality measure that operates on video frames, VMAF is a recent 'meta-metric' that fuses a series of high-performance metrics, using a machine learning model in order to leverage the best features of each of these fundamental quality metrics. VMAF is available

[†]By definition, $Mean\ PSNR = 10 \times (\sum_i log_{10}(\frac{255^2}{MSE_i})) = 10 \times log_{10}(\prod_i(\frac{255^2}{MSE_i})))$, which implies that lower MSEs have an undue influence on the final score. Computing the logarithm after averaging the MSE, as in $PSNR_{\overline{MSE}}$, is more sensitive to high MSE values and low-quality frames.

as an open source release to the community[31] and the interested reader is directed to the Netflix tech blog for a thorough overview.[32]

We note two limitations of the quality metrics used in these study. First, we only apply luminance-based calculations. Second, there exist other sophisticated high performing video quality metrics that operate on spatio-temporal regions such as VQM-VFD[33] and MOVIE.[34] However, these measures are computationally impractical for an evaluation of the scale undertaken here. We believe that the metrics used in this study provide a fine balance between computational tractability and performance.

3.5 BD-RATE CALCULATION

BD-rate calculations are used to provide aggregate results over the set of test clips, and are calculated based on each of the quality metrics mentioned in Sect. 3.4. Besides the overall BD-rate, which spans eight encodes per R-D curve, BD-rate measurements are provided for three bitrate subranges, i.e. low, medium, and high bitrates.

- Overall: BD-rate calculation over all 8 rate points, corresponding to the QP values mentioned above;
- Low quality range: BD-rate calculation where the quality range is determined by the four lowest rate points of each curve;
- Medium quality range: BD-rate calculation where the quality range is determined by the the middle four rate points of each curve;
- High quality range: BD-rate calculation where the quality range is determined by the highest four rate points of each curve.

A piecewise cubic hermite interpolated polynomial (PCHIP)[35] is used to interpolate rate-distortion curves, and to calculate the BD-rate across N points ($N = 4$ or $N = 8$). The interpolation procedure corresponds to that used in JCT-VC and JVET (Joint Video Exploration Team) standardization.[6,36]

The four BD-rate measurements are calculated for both configurations and all resolutions. Since the chosen QPs do not guarantee a perfect match in quality, depending on the content type and clip, the overlap in bitrate subranges can change. In all four cases, the BD-rate difference is calculated only in the quality (sub)range that is common to the three curves. For any comparison where no overlap is present, the measurement is rejected. Given the chosen QP values, the 'overall' BD-rate calculation will show a large overlap for all clips.

An illustration of the four (sub)ranges used in the BD-rate calculations is given in Fig. 2, for a 720p sequence. For example, for the 'low quality range' calculation, the calculation starts at the first libvpx rate point (the maximum of the three minima of each curve), and ends at the fourth x264 rate point (the minimum of the three maxima for the lowest four points of each curve). The other ranges – medium, high, and overall – are obtained in a similar way, as illustrated in Fig. 2.

Note that in Fig. 2 there is a range at low bitrates where the x264 and x265 curves overlap in quality, but the data is not included in the BD-rate calculation since the first libvpx point only starts at around 37.7 dB. Similarly, for the high bitrate end, we disregard the x265 and libvpx curves where PSNR is greater than 48.5 dB. We believe this is an acceptable trade-off in order to generate BD-rate numbers that have consistent ranges across the three pairwise comparisons. Furthermore, the very low and very high bitrate ranges are less relevant from the perspective of practical VOD applications.

3.6 ENCODING PIPELINE

The tests were run at scale in a cloud-based encoding infrastructure.[37] The experiments leveraged under-utilized compute resources on our reserved cloud instances.

The source videos were processed through a video inspection stage to ensure that the source files were free of corruption, metadata issues, and serious visual artifacts. Source formats supported were Interoperable Master Format (IMF), ProRes, DPX and MPEG (for older titles). FFmpeg version 3.0[38,39] was used for decoding,

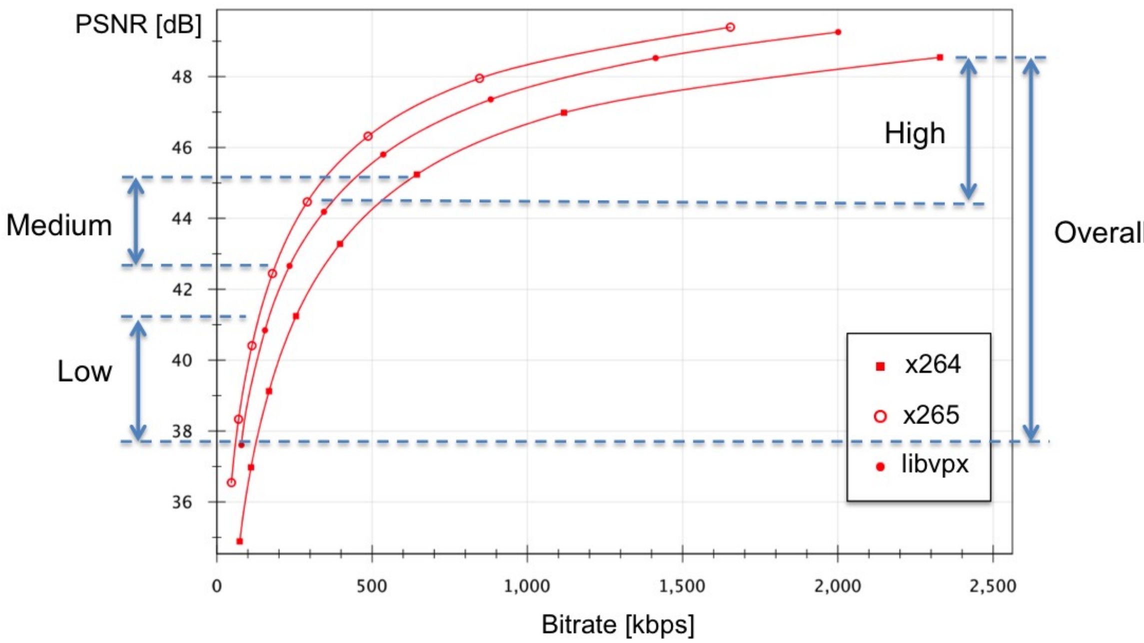


Figure 2: Illustration of the four used BD-rate calculation ranges (overall, low, medium, and high) for a 720p clip with (SI,TI)=(10.5,3.9).

employing internal decoder plug-ins when necessary. Lanczos resizing was applied to generate output resolutions smaller than the source resolution. After each encode, validation software was run to verify the correctness of the output bitstream. The quality metrics were calculated on the cloud instances after each encoded video was generated. More information about the encoding pipeline can be found in Aaron and Ronca[37] and Aaron et al.[40]

4. RESULTS AND DISCUSSION

In this section, we provide results for the 5000 encoded clips, along with average BD-rate numbers based on the six metrics discussed above. We also make a distinction between animation (cartoons, anime, etc.) and non-animation titles and report separate numbers in Sec. 4.3.

4.1 AVERAGE RESULTS FOR PSNR-TUNED CONFIGURATION

Configuration PT aims to achieve the highest possible PSNR, by tuning the codecs towards PSNR, and by disabling adaptive quantization (AQ) and psychovisual settings. The settings chosen for x264 and x265 in this configuration are equivalent to PSNR tuning, while for libvpx the 'tune=psnr' option is explicitly set.

Table 4 shows the average BD-rate results for the three combinations of codecs, both for the whole quality range and split up in the low, medium and high quality ranges as described above. These results show a significant BD-rate reduction for both x265 and libvpx, with gains that increase for higher resolutions. At 360p, gains are achieved of 30.8% and 22.6%, respectively, increasing to 43.4% and 43.5% at 1080p. Comparing x265 and libvpx, we see that x265 performs noticeably better at lower resolutions. The gap between both decreases for 1080p, particularly in the high bitrate range.

When looking at the results based on $PSNR_{\overline{MSE}}$ (Table 5), the gap between x265 and libvpx widens. This can be attributed to larger frame-by-frame variations in the VP9 streams, leading to a lower $PSNR_{\overline{MSE}}$ for the same average PSNR. An example of the frame-by-frame variation of PSNR values is shown in Fig. 3, for a static documentary sequence at 360p, where all three streams have a bitrate of approximately 220 kbps. We can see that, after a shot change around frame 140, the libvpx rate control undershoots, leading to a steep quality drop and a frame quality of 35 dB, while x264 and x265 keep quality around 40 dB. Additionally, the alt-ref frames lead to peaks in the PSNR value, which drops for subsequent frames. A less pronounced effect is found in the x264 and x265 streams due to hierarchical QP cascading.

For completeness, we add the results for SSIM, MS-SSIM, VIF, and VMAF in Tables 6 to 9. Note that tuning for PSNR can hurt these metrics, so they should be interpreted more carefully for Configuration PT.

In certain cases, a non-intuitive result can be seen, e.g. from the high bitrate range of Table 7 and Table 8 (see last column). Although x265 demonstrates (slightly) larger bitrate savings over x264 (35.1%) than libvpx (35.0%) for MS-SSIM at 1080p, x265 has an overhead of 3.2% when compared to libvpx (the last column, last row entry). This seemingly inconsistent behavior is due to arithmetic averaging of percentage gains. If we would use *geometric* averaging instead, this 'non-transitive' behavior would disappear. Using geometric averaging, the corresponding three results become -36.3%, -36.9% and +0.9%, respectively, showing slightly larger gains for libvpx in this example (PT configuration, MS-SSIM, 1080p, High range). For VIF, the results become -31.9%, -31.4%, and -0.8%, reflecting a slight gain for x265 instead. Since arithmetic averaging is conventionally used to average BD-rate gains across sequences, we report these numbers.

Table 4: Average BD-rate results (PSNR-based) for Configuration PT.

		Overall	Low	Medium	High
x265 vs. x264	360p	-30.8%	-34.1%	-29.8%	-28.2%
	720p	-40.7%	-46.6%	-39.6%	-35.3%
	1080p	-43.4%	-52.4%	-45.2%	-34.2%
libvpx vs. x264	360p	-22.6%	-25.2%	-21.6%	-21.0%
	720p	-36.4%	-41.9%	-34.5%	-31.6%
	1080p	-43.5%	-51.7%	-42.7%	-34.3%
x265 vs. libvpx	360p	-9.8%	-10.7%	-9.6%	-8.1%
	720p	-6.7%	-7.5%	-6.7%	-3.7%
	1080p	-2.5%	-3.5%	-2.8%	2.0%

Table 5: Average BD-rate results ($PSNR_{\overline{MSE}}$-based) for Configuration PT.

		Overall	Low	Medium	High
x265 vs. x264	360p	-30.9%	-34.1%	-29.8%	-28.2%
	720p	-40.7%	-46.7%	-39.5%	-35.0%
	1080p	-43.2%	-52.4%	-45.0%	-32.8%
libvpx vs. x264	360p	-15.7%	-17.6%	-14.2%	-15.3%
	720p	-30.8%	-36.0%	-28.4%	-26.8%
	1080p	-38.7%	-47.1%	-37.9%	-29.4%
x265 vs. libvpx	360p	-17.1%	-18.8%	-17.2%	-13.9%
	720p	-14.0%	-16.0%	-13.8%	-8.7%
	1080p	-9.8%	-12.0%	-9.4%	-1.8%

4.2 RESULTS FOR VISUAL-QUALITY-TUNED CONFIGURATION

Configuration VQ enables options which enhance subjective quality, by using adaptive quantization and psycho-visual options. For this configuration, we primarily focus on metrics which better capture perceptual quality, such as VIF and VMAF. PSNR-based BD-rate values are less meaningful, since enabling the AQ modes will typically harm PSNR. (MS-)SSIM values can also give a better indication of visual quality than PSNR, although the used psycho-visual settings in x264 and x265 can also hurt these metrics.

For the VIF metric (Table 14), overall gains of up to 47.0% and 39.7% are observed (at 1080p) for x265 and libvpx, respectively. The largest gap between both codecs is found in terms of VMAF (Table 15), with an overall BD-rate reduction of 17.8% for x265 at 1080p. Still, libvpx achieves an overall BD-rate reduction of 42.6%.

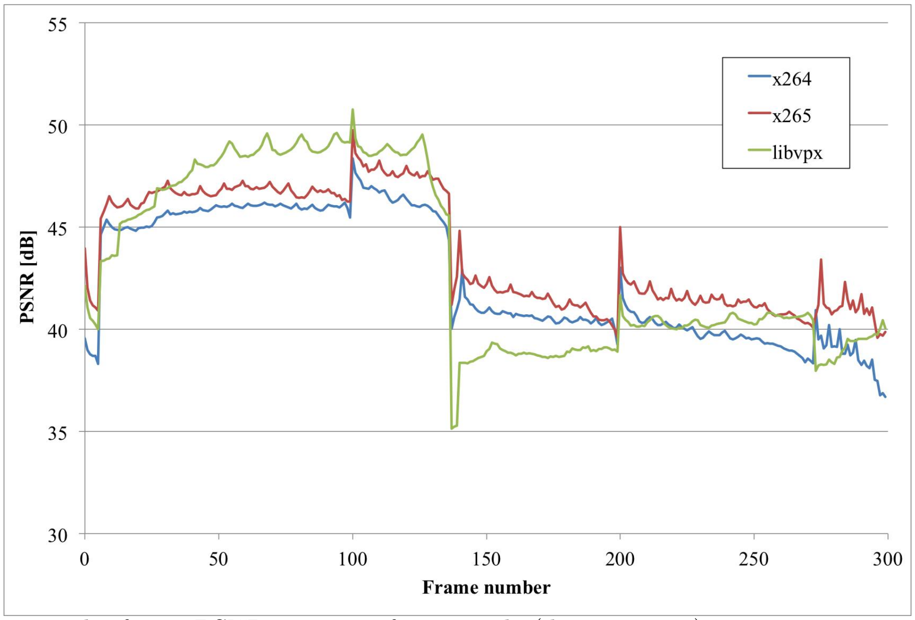


Figure 3: Frame-by-frame PSNR variation for example (documentary) 360p sequence at 220 kbps.

Table 6: Average BD-rate results (SSIM-based) for Configuration PT.

		Overall	Low	Medium	High
x265 vs. x264	360p	-31.4%	-33.2%	-29.4%	-29.1%
	720p	-45.5%	-48.7%	-42.8%	-37.9%
	1080p	-50.7%	-55.1%	-49.3%	-37.0%
libvpx vs. x264	360p	-19.2%	-19.9%	-18.3%	-20.7%
	720p	-40.9%	-44.1%	-38.5%	-34.3%
	1080p	-50.4%	-55.2%	-48.2%	-37.4%
x265 vs. libvpx	360p	-14.0%	-15.2%	-12.4%	-9.0%
	720p	-7.2%	-7.2%	-5.3%	-2.7%
	1080p	-1.9%	-1.4%	0.3%	4.9%

Table 7: Average BD-rate results (MS-SSIM-based) for Configuration PT.

		Overall	Low	Medium	High
x265 vs. x264	360p	-34.3%	-36.1%	-31.8%	-30.1%
	720p	-44.6%	-48.2%	-41.7%	-36.7%
	1080p	-48.1%	-53.8%	-47.1%	-35.1%
libvpx vs. x264	360p	-25.6%	-27.0%	-24.2%	-24.4%
	720p	-40.2%	-43.8%	-37.4%	-33.7%
	1080p	-47.3%	-53.1%	-45.1%	-35.0%
x265 vs. libvpx	360p	-10.4%	-10.8%	-8.6%	-5.6%
	720p	-6.6%	-6.8%	-5.0%	-1.7%
	1080p	-3.1%	-3.0%	-1.1%	3.2%

For completeness, we report the BD-rate results based on PSNR and $PSNR_{\overline{MSE}}$ in Table 10 and Table 11, respectively. For the PSNR-based comparison, we observe overall gains of up to 50.2% for x265, and up to 48.9% for libvpx when compared to x264. But as mentioned above, Configuration VQ, is not aimed to optimize for PSNR.

Table 8: Average BD-rate results (VIF-based) for Configuration PT.

		Overall	Low	Medium	High
x265 vs. x264	360p	-30.4%	-33.5%	-29.2%	-26.8%
	720p	-39.1%	-45.0%	-38.2%	-32.2%
	1080p	-41.4%	-50.4%	-43.5%	-30.9%
libvpx vs. x264	360p	-22.2%	-24.3%	-21.8%	-20.1%
	720p	-34.4%	-39.7%	-33.2%	-28.5%
	1080p	-40.2%	-48.7%	-40.5%	-30.2%
x265 vs. libvpx	360p	-9.4%	-10.6%	-8.4%	-7.3%
	720p	-6.7%	-7.8%	-6.0%	-3.4%
	1080p	-3.3%	-4.7%	-3.1%	0.8%

Table 9: Average BD-rate results (VMAF-based) for Configuration PT.

		Overall	Low	Medium	High
x265 vs. x264	360p	-30.6%	-32.5%	-28.7%	-25.9%
	720p	-39.9%	-43.9%	-36.4%	-26.0%
	1080p	-44.1%	-49.3%	-41.2%	-18.2%
libvpx vs. x264	360p	-22.9%	-23.5%	-22.0%	-27.9%
	720p	-34.6%	-37.8%	-31.0%	-30.9%
	1080p	-41.8%	-46.5%	-37.4%	-29.5%
x265 vs. libvpx	360p	-9.1%	-10.7%	-7.2%	8.1%
	720p	-7.6%	-9.2%	-6.4%	13.7%
	1080p	-4.9%	-6.7%	-4.4%	20.8%

Table 10: Average BD-rate results (PSNR-based) for Configuration VQ.

		Overall	Low	Medium	High
x265 vs. x264	360p	-38.3%	-40.1%	-36.3%	-39.8%
	720p	-48.1%	-50.3%	-45.3%	-50.2%
	1080p	-50.2%	-56.0%	-50.0%	-47.4%
libvpx vs. x264	360p	-27.6%	-29.3%	-24.9%	-29.7%
	720p	-43.3%	-44.8%	-39.9%	-46.0%
	1080p	-48.9%	-53.3%	-48.4%	-46.8%
x265 vs. libvpx	360p	-13.6%	-13.8%	-14.1%	-12.8%
	720p	-5.7%	-6.8%	-5.4%	-3.3%
	1080p	0.0%	-1.3%	2.9%	2.9%

4.3 Results split up for animation and non-animation titles

In our results, we found a non-negligible difference in compression performance between animation and non-animation content, depending on the codec. Given the high proportion of animation titles streamed on VOD platforms, it is worth reporting separate numbers for both categories. In our tests, we included 60 animation titles (all of which are non-4K sources), corresponding to 12% of all encoded clips.

The (PSNR-based and VMAF-based) results are summarized in Table 16 and Table 17 for Configuration PT, and in Table 18 and Table 19 for Configuration VQ. For animation content, libvpx shows gains over x264 of 11% to 30% on average, which is substantially lower than for non-animation content, where it achieves gains of up to 45%. [‡]

[‡]Note that libvpx supports an additional mode, which allows tuning for 'screen capture content'. Although this mode has not been tested in this comparison, and we only use a single set of parameters for each codec, it could provide

Table 11: Average BD-rate results ($PSNR_{\overline{MSE}}$-based) for Configuration VQ.

		Overall	Low	Medium	High
x265 vs. x264	360p	-38.7%	-40.7%	-36.7%	-40.0%
	720p	-48.4%	-50.9%	-45.4%	-50.1%
	1080p	-50.2%	-56.4%	-49.8%	-46.6%
libvpx vs. x264	360p	-22.4%	-23.7%	-19.2%	-25.4%
	720p	-39.2%	-40.2%	-35.3%	-42.9%
	1080p	-45.4%	-49.7%	-44.9%	-43.6%
x265 vs. libvpx	360p	-19.7%	-20.7%	-20.3%	-17.4%
	720p	-11.9%	-14.4%	-11.7%	-6.9%
	1080p	-5.6%	-8.5%	-2.2%	0.6%

Table 12: Average BD-rate results (SSIM-based) for Configuration VQ.

		Overall	Low	Medium	High
x265 vs. x264	360p	-35.0%	-36.5%	-32.5%	-37.1%
	720p	-41.4%	-45.2%	-37.4%	-33.7%
	1080p	-45.6%	-51.5%	-40.7%	-29.2%
libvpx vs. x264	360p	-22.0%	-22.7%	-19.7%	-27.0%
	720p	-38.3%	-41.8%	-35.0%	-29.5%
	1080p	-47.2%	-52.0%	-44.2%	-29.8%
x265 vs. libvpx	360p	-15.0%	-15.9%	-14.2%	-11.3%
	720p	-2.0%	-1.9%	0.7%	-1.6%
	1080p	7.0%	7.0%	15.3%	7.3%

Table 13: Average BD-rate results (MS-SSIM-based) for Configuration VQ.

		Overall	Low	Medium	High
x265 vs. x264	360p	-31.3%	-33.5%	-28.1%	-29.2%
	720p	-41.1%	-44.5%	-37.4%	-36.2%
	1080p	-45.0%	-51.2%	-41.8%	-33.0%
libvpx vs. x264	360p	-21.6%	-23.5%	-18.8%	-21.0%
	720p	-37.9%	-40.9%	-34.7%	-32.5%
	1080p	-45.1%	-50.4%	-43.2%	-32.0%
x265 vs. libvpx	360p	-10.9%	-11.2%	-9.8%	-8.1%
	720p	-1.8%	-1.7%	0.6%	-0.8%
	1080p	3.8%	4.9%	10.7%	4.3%

For non-animation content, the difference between x265 and libvpx decreases, in particular when looking at the PSNR results. For the VMAF results, gains over x265 are achieved in the high bitrate range, while for animation titles x265 obtains an overall reduction of roughly 15%.

Similar to the previous section, for Configuration VQ, libvpx performs worse in terms of VMAF (with the exception of the high bitrate range at 1080p), but achieves gains close to those for x265 in terms of PSNR, in particular at 1080p.

additional gains for libvpx. Similarly, further tweaking of e.g. psy-rd values could be used for animation content coding in x265.

Table 14: Average BD-rate results (VIF-based) for Configuration VQ.

		Overall	Low	Medium	High
x265 vs. x264	360p	-34.5%	-37.2%	-32.3%	-34.3%
	720p	-44.4%	-49.4%	-41.9%	-42.0%
	1080p	-47.0%	-56.1%	-47.2%	-40.5%
libvpx vs. x264	360p	-21.9%	-24.8%	-19.7%	-21.2%
	720p	-35.0%	-40.3%	-32.5%	-31.7%
	1080p	-39.7%	-48.8%	-39.6%	-32.5%
x265 vs. libvpx	360p	-15.0%	-15.1%	-14.6%	-15.3%
	720p	-12.9%	-13.2%	-12.1%	-12.9%
	1080p	-11.1%	-11.9%	-10.3%	-10.4%

Table 15: Average BD-rate results (VMAF-based) for Configuration VQ.

		Overall	Low	Medium	High
x265 vs. x264	360p	-35.4%	-37.7%	-32.7%	-34.0%
	720p	-46.4%	-50.6%	-41.1%	-37.6%
	1080p	-53.3%	-57.7%	-48.0%	-42.0%
libvpx vs. x264	360p	-16.6%	-18.5%	-12.9%	-21.3%
	720p	-31.6%	-36.0%	-24.7%	-26.2%
	1080p	-42.6%	-46.5%	-35.8%	-36.5%
x265 vs. libvpx	360p	-21.8%	-22.7%	-21.9%	-13.3%
	720p	-20.7%	-21.6%	-20.7%	-10.9%
	1080p	-17.8%	-19.2%	-17.2%	-1.2%

Table 16: Average BD-rate results (PSNR-based) for Configuration PT (animation and non-animation).

		Animation				Non-animation			
		Overall	Low	Medium	High	Overall	Low	Medium	High
x265 vs. x264	360p	-27.1%	-29.3%	-26.4%	-25.5%	-31.4%	-34.8%	-30.3%	-28.6%
	720p	-36.3%	-40.3%	-34.6%	-33.5%	-41.4%	-47.6%	-40.4%	-35.5%
	1080p	-39.9%	-46.3%	-38.9%	-33.8%	-44.0%	-53.4%	-46.2%	-34.3%
libvpx vs. x264	360p	-11.5%	-13.8%	-10.6%	-9.6%	-24.3%	-26.9%	-23.2%	-22.7%
	720p	-23.3%	-28.2%	-21.2%	-19.4%	-38.4%	-44.0%	-36.4%	-33.4%
	1080p	-30.6%	-37.4%	-28.3%	-24.1%	-45.4%	-53.8%	-44.9%	-35.9%
x265 vs. libvpx	360p	-16.7%	-16.6%	-16.6%	-16.5%	-8.8%	-9.8%	-8.6%	-6.9%
	720p	-15.7%	-15.1%	-15.4%	-15.6%	-5.3%	-6.3%	-5.5%	-2.0%
	1080p	-12.6%	-12.8%	-12.8%	-11.0%	-0.7%	-1.9%	-1.4%	3.9%

5. CONCLUSIONS

In this paper we have compared the compression performance of x264, x265 and libvpx over a large set of video clips representative of TV shows and movies. Previous related work only covers a limited set of test sequences, PSNR or SSIM as objective quality metrics, or encoding configurations that may not be practical for an encoding pipeline deployed in production. We focus on the VOD streaming scenario which can accommodate more complex GOP structures and non-real-time encoding. We compare codec implementations that are available as open source software.

For the different encoding configurations, output parameters and quality metrics applied, x265 and libvpx demonstrate superior encoding performance compared to x264. For Configuration PT, which is optimized for PSNR, x265 shows bitrate savings of 28% to 52%, compared to x264 at the same PSNR. For libvpx, the bitrate

Table 17: Average BD-rate results (VMAF-based) for Configuration PT (animation and non-animation).

		Animation				Non-animation			
		Overall	Low	Medium	High	Overall	Low	Medium	High
x265 vs. x264	360p	-26.5%	-28.0%	-24.5%	-22.7%	-31.2%	-33.1%	-29.3%	-26.3%
	720p	-35.6%	-38.5%	-31.2%	-24.7%	-40.5%	-44.7%	-37.2%	-26.2%
	1080p	-41.0%	-44.4%	-35.9%	-24.3%	-44.6%	-50.2%	-42.0%	-17.3%
libvpx vs. x264	360p	-12.2%	-13.5%	-10.3%	-11.8%	-24.5%	-25.0%	-23.8%	-30.3%
	720p	-22.2%	-25.2%	-16.3%	-14.0%	-36.5%	-39.7%	-33.2%	-33.4%
	1080p	-29.5%	-33.0%	-21.7%	-16.3%	-43.6%	-48.5%	-39.7%	-31.5%
x265 vs. libvpx	360p	-15.2%	-15.5%	-14.4%	-8.9%	-8.2%	-10.0%	-6.1%	10.6%
	720p	-16.0%	-16.3%	-16.2%	-8.7%	-6.2%	-8.1%	-4.9%	17.0%
	1080p	-15.5%	-15.7%	-16.3%	-6.7%	-3.0%	-5.1%	-2.6%	25.0%

Table 18: Average BD-rate results (PSNR-based) for Configuration VQ (animation and non-animation).

		Animation				Non-animation			
		Overall	Low	Medium	High	Overall	Low	Medium	High
x265 vs. x264	360p	-34.2%	-35.9%	-33.3%	-34.1%	-39.0%	-40.7%	-36.8%	-40.7%
	720p	-43.0%	-44.9%	-41.2%	-43.9%	-48.9%	-51.1%	-45.9%	-51.1%
	1080p	-46.7%	-49.9%	-45.8%	-45.3%	-50.7%	-56.9%	-50.6%	-47.7%
libvpx vs. x264	360p	-14.3%	-17.0%	-12.8%	-13.7%	-29.6%	-31.1%	-26.7%	-32.1%
	720p	-25.3%	-28.2%	-22.7%	-25.6%	-46.0%	-47.2%	-42.4%	-49.0%
	1080p	-31.8%	-35.4%	-28.8%	-31.2%	-51.4%	-56.0%	-51.3%	-49.0%
x265 vs. libvpx	360p	-22.0%	-21.3%	-22.3%	-22.3%	-12.4%	-12.7%	-12.9%	-11.5%
	720p	-22.0%	-20.5%	-21.7%	-22.3%	-3.3%	-4.8%	-3.0%	-0.6%
	1080p	-19.8%	-18.6%	-21.3%	-18.5%	3.0%	1.2%	6.5%	6.0%

Table 19: Average BD-rate results (VMAF-based) for Configuration VQ (animation and non-animation).

		Animation				Non-animation			
		Overall	Low	Medium	High	Overall	Low	Medium	High
x265 vs. x264	360p	-30.1%	-31.4%	-28.3%	-30.5%	-36.1%	-38.6%	-33.3%	-34.6%
	720p	-40.3%	-42.9%	-36.3%	-34.9%	-47.3%	-51.7%	-41.8%	-38.0%
	1080p	-47.8%	-50.0%	-43.7%	-42.3%	-54.2%	-58.8%	-48.7%	-42.0%
libvpx vs. x264	360p	-4.1%	-5.8%	-1.1%	-5.5%	-18.5%	-20.4%	-14.7%	-23.6%
	720p	-15.3%	-18.4%	-8.4%	-8.7%	-34.0%	-38.5%	-27.1%	-28.7%
	1080p	-26.5%	-28.6%	-19.1%	-23.8%	-44.9%	-49.2%	-38.3%	-38.4%
x265 vs. libvpx	360p	-26.2%	-26.2%	-26.5%	-23.7%	-21.2%	-22.2%	-21.2%	-11.7%
	720p	-28.1%	-28.4%	-28.8%	-25.0%	-19.6%	-20.6%	-19.6%	-8.8%
	1080p	-27.0%	-27.8%	-28.2%	-18.2%	-16.4%	-17.9%	-15.6%	1.3%

savings compared to x264 range from around 21% to 52%. The largest gains for x265 and libvpx are observed for higher resolutions and in the low bitrate range. Zeroing in on Configuration VQ, which is optimized for visual quality, VMAF-based comparison resulted in 33% to 58% bitrate savings for x265 and 13% to 46% savings for libvpx, compared to x264 at the same VMAF.

x265 outperforms libvpx in almost all the table entries reported in this paper. Configuration VQ with VMAF-based BD-rate gives the most pronounced efficiency gap between libvpx and x265, with x265 requiring 23% less bits for the low bitrate, low resolution scenario. In contrast, for both configurations, encoding 1080p video and calculating BD-rate based on PSNR, SSIM and MS-SSIM, x265 and libvpx compression efficiencies are close, with libvpx requiring a few percent more bits or performing at par. The BD-rate comparisons which split the dataset into animation and non-animation clips indicate that libvpx is suboptimal for animation content and is likely tuned for non-animation video content.

The results presented in this paper are in line with the codec comparison data from the Moscow State University[17] study. In general, x265 finishes ahead among the codecs investigated. Our numbers contradict some earlier results [20,21] that state that libvpx is inferior to H.264/AVC. The data shows that libvpx outperforms x264 for all encoding scenarios evaluated and it is close in efficiency to x265 for high resolutions.

ACKNOWLEDGMENTS

We thank our other team members, Ioannis Katsavounidis and Andrey Norkin, for the fruitful discussions related to encoder parameter settings, and Megha Manohara and Zhi Li for their valuable engineering support. These experiments were run on the Netflix cloud-based encoding pipeline – the scale and robustness is a product of the work of the entire Media Cloud Engineering and Encoding Technologies teams.

REFERENCES

[1] Sandvine, *Global Internet Phenomena Latin America and North America.* Sandvine Intelligent Broadband Networks (Jun 2016). https://www.sandvine.com/trends/global-internet-phenomena/.

[2] ITU-T and ISO/IEC JTC 1, *Advanced Video Coding for Generic Audiovisual Services.* ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC) (2003).

[3] Wiegand, T., Sullivan, G. J., Bjontegaard, G., and Luthra, A., "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology* **13**, 560–576 (Jul 2003).

[4] ITU-T and ISO/IEC JTC 1, *High Efficiency Video Coding.* ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC) (2013).

[5] Sullivan, G. J., Ohm, J.-R., Han, W.-J., and Wiegand, T., "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology* **22**(12), 1649–1668 (2012).

[6] Bossen, F., *JCTVC-L1100: Common HM Test Conditions and Software Reference Configurations.* ITU-T and ISO/IEC JTC 1 Joint Collaborative Team on Video Coding (Jan 2013).

[7] Ohm, J.-R., Sullivan, G. J., Schwarz, H., Tan, T. K., and Wiegand, T., "Comparison of the coding efficiency of video coding standards including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology* **22**(12), 1669–1684 (2012).

[8] Wikipedia, *High Efficiency Video Coding Implementations and Products.* "https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding_implementations_and_products".

[9] Grange, A., de Rivaz, P., and Hunt, J., *VP9 Bitstream and Decoding Process Specification.* Google (Mar 2016). https://www.webmproject.org/vp9/.

[10] Mukherjee, D., Bankoski, J., Grange, A., Han, J., Koleszar, J., Wilkins, P., Xu, Y., and Bultje, R., "The latest open-source video codec VP9 - an overview and preliminary results," in [*Picture Coding Symposium (PCS)*], 390–393, IEEE (2013).

[11] *libvpx Code Repository.* https://chromium.googlesource.com/webm/libvpx.

[12] Robertson, S., "VP9: Faster, better, buffer-free youtube videos," *YouTube Engineering and Developers Blog* (Apr 2015). http://youtube-eng.blogspot.com/2015/04/vp9-faster-better-buffer-free-youtube.html.

[13] *x264 Home Page.* http://www.videolan.org/developers/x264.html.

[14] *x264 Code Repository.* http://git.videolan.org/git/x264.git.

[15] *x265 Home Page.* http://x265.org/.

[16] *x265 Code Repository.* https://bitbucket.org/multicoreware/x265.

[17] Vatolin, D., Kulikov, D., Erofeev, M., and Yanushkovsky, V., *HEVC/H.265 Video Codecs Comparison Free Version.* Moscow State University Image and Graphics Lab (Oct 2015).

[18] *VP9 Home Page.* http://www.webmproject.org/vp9/.

[19] *WebM Repositories.* https://www.webmproject.org/code/.

[20] Hanhart, P., Řeřábek, M., De Simone, F., and Ebrahimi, T., "Subjective quality evaluation of the upcoming HEVC video compression standard," in [*SPIE Optical Engineering + Applications*], International Society for Optics and Photonics (2012).

[21] Grois, D., Marpe, D., Mulayoff, A., Itzhaky, B., and Hadar, O., "Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders," in [*Picture Coding Symposium (PCS)*], 394–397, IEEE (2013).

[22] Řeřábek, M. and Ebrahimi, T., "Comparison of compression efficiency between HEVC/H.265 and VP9 based on subjective assessments," in [*SPIE Optical Engineering + Applications*], International Society for Optics and Photonics (2014).

[23] ITU-T P. 910, *Subjective Video Quality Assessment Methods for Multimedia Applications* (2008).

[24] Bjontegaard, G., *VCEG-M33: Calculation of average PSNR differences between RD-curves*. ITU-T Video Coding Experts Group (April 2001).

[25] Merritt, L. and Vanam, R., "Improved rate control and motion estimation for H.264 encoder," in [*IEEE International Conference on Image Processing (ICIP)*], **5**, 309–312, IEEE (2007).

[26] Wang, Z. and Bovik, A. C., "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine* **26**(1), 98–117 (2009).

[27] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004).

[28] Wang, Z., Simoncelli, E. P., and Bovik, A. C., "Multiscale structural similarity for image quality assessment," in [*Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*], **2**, 1398–1402, IEEE (2003).

[29] Sheikh, H. R. and Bovik, A. C., "Image information and visual quality," *IEEE Transactions on Image Processing* **15**(2), 430–444 (2006).

[30] Lin, J. Y., Wu, C.-H., Katsavounidis, I., Li, Z., Aaron, A., and Kuo, C.-C. J., "EVQA: an ensemble-learning-based video quality assessment index," in [*IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*], 1–6, IEEE (2015).

[31] *VMAF Code Repository*. https://github.com/Netflix/vmaf.git.

[32] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., and Manohara, M., "Toward a practical perceptual video quality metric," *The Netflix Tech Blog* . http://techblog.netflix.com/2016/06/toward-practical-perceptual-video.html.

[33] Pinson, M. H., Choi, L. K., and Bovik, A. C., "Temporal video quality model accounting for variable frame delay distortions," *IEEE Transactions on Broadcasting* **60**(4), 637–649 (2014).

[34] Seshadrinathan, K. and Bovik, A. C., "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Transactions on Image Processing* **19**(2), 335–350 (2010).

[35] Fritsch, F. N. and Carlson, R. E., "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis* **17**(2), 238–246 (1980).

[36] Suehring, K. and Li, X., *JVET-B1010: JVET common test conditions and software reference configurations*. ITU-T and ISO/IEC JTC 1 Joint Video Exploration Team (Feb 2016).

[37] Aaron, A. and Ronca, D., "High quality video encoding at scale," *The Netflix Tech Blog* . http://techblog.netflix.com/2016/06/toward-practical-perceptual-video.html.

[38] *FFmpeg Home Page*. https://ffmpeg.org/.

[39] *FFmpeg Code Repository*. https://git.ffmpeg.org/ffmpeg.git.

[40] Aaron, A., Li, Z., Manohara, M., Lin, J. Y., Wu, E. C.-H., and Kuo, C.-C. J., "Challenges in cloud based ingest and encoding for high quality streaming media," in [*IEEE International Conference on Image Processing (ICIP)*], 1732–1736, IEEE (2015).