# Coding Efficiency Comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC Encoders

Dan Grois, *Senior Member, IEEE,* Tung Nguyen, Detlev Marpe, *Fellow*, *IEEE*

Video Coding & Analytics Department, Image & Video Coding Group,
Fraunhofer Heinrich Hertz Institute (HHI), Berlin, Germany
E-mail: grois@ieee.org, {tung.nguyen,detlev.marpe}@hhi.fraunhofer.de

*Abstract* — **This work presents a coding efficiency evaluation of the recently published first release of the video coding scheme of the Alliance for Open Media (AOM), so called AOM/AV1, in comparison to the video coding standards H.264/MPEG-AVC (Advanced Video Coding) and H.265/MPEG-HEVC (High-Efficiency Video Coding). As representatives of the two last-mentioned video coding standards, the corresponding reference software encoders of JM and HM were selected, and for HEVC, in addition, the Fraunhofer HHI HEVC commercial software encoder and the open source software implementation x265 were used. According to the experimental results, which were obtained by using similar configurations for all examined representative encoders, the H.265/MPEG-HEVC reference software implementation provides significant average bit-rate savings of 38.4% and 32.8% compared to AOM/AV1 and H.264/MPEG-AVC, respectively. Particularly, when directly compared to H.264/MPEG-AVC High Profile, the AOM/AV1 encoder produces an average bit-rate overhead of 10.5% at the same objective quality. In addition, it was observed that the AOM/AV1 encoding times are quite similar to those of the full-fledged HM and JM reference software encoders. On the other hand, the typical encoding times of the HM encoder are in the range of 30-300 times higher on average than those measured for the configurable HHI HEVC encoder, depending on its chosen trade-off between encoding speed and coding efficiency.**

***Terms — H.265, HEVC, AV1, VP9, H.264, AVC, HM, x265, JM, video coding, coding efficiency.***

## I. Introduction

The development of the first version of HEVC by the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG was officially finalized in January 2013. After that, the final aligned HEVC specification was approved by ITU-T as Recommendation H.265 [1] and by ISO/IEC as MPEG-H, Part 2. About one year later, the second HEVC version was finalized, incorporating the Range Extensions (RExt) as well as the Scalable and Multiview Extensions (SHVC and MV-HEVC, respectively) [2]. In turn, the third and fourth HEVC edition were issued in 2015 and 2016, further containing the 3D Video Coding Extensions (3D-HEVC) [3] and the Screen Content Coding Extension (HEVC-SCC), respectively.

When developing the H.265/MPEG-HEVC standard, high-resolution video coding was considered as one of its main potential application scenarios, while keeping it applicable to almost all existing use cases that were already targeted by H.264/MPEG-AVC [4]. The development process of H.265/MPEG-HEVC was driven by the most recent scientific and technological achievements in the video coding field. As a result, when compared to the High Profile of its predecessor H.264/MPEG-AVC, H.265/MPEG-HEVC is able to achieve a bit-rate reduction of roughly 50% for substantially the same *visual* quality [5], [6].

While the joint video coding standardization activities of ITU-T and ISO/IEC organizations rely on an open and collaborative process driven by its active members, several companies individually developed their own video coding formats. Technologically, these proprietary video coding formats were often designed around variants of the state-of-the-art coding algorithms supplemented with the companies' own patented or secretly kept technologies. For example, the VP8 codec [7]-[8] that was developed privately by On2 Technologies, a company acquired by Google in 2009, is one of such proprietary video codecs. The development of VP9 [9], as the successor of VP8, started about two years after the acquisition of On2 Technologies and was finalized in 2013. Performance comparisons of VP9 with H.264/MPEG-AVC and H.265/MPEG-HEVC based on objective assessments were presented in detail in [10] and [11]. In addition, subjective assessments of HEVC and VP9 can be found in [12].

In 2015, the so-called Alliance for Open Media (AOM) was formed by Amazon, Google, Microsoft, Cisco, Intel, Mozilla, and Netflix with the objective to work towards next-generation media formats in general and with a particular short-term focus on the development of a royalty-free video format. About half a year later, it was announced that three additional members have joined, particularly, ARM, AMD and Nvidia, and recently, the four additional members Vidyo, Ateme, Adobe and Ittiam joined AOM [13].

In April 2016, AOM released a first version of its proprietary video-coding scheme, so called AOM/AV1, which is evaluated in detail in this work. The final version of AOM/AV1 is currently expected to be available by the end of 2016, and it is claimed to provide a significant coding-efficiency gain over current state-of-the-art video codecs. However, given that there is currently very little known about the coding efficiency of the first version of AOM/AV1 there is a lot of confusion about the ability of future versions of AOM/AV1 to actually compete with or even outperform HEVC-based encoder implementations.

In addition, several open source implementations for encoding video into the H.265/MPEG-HEVC compression format have recently emerged. One of such representatives is x265, which is considered to be the most popular open source encoder implementation of H.265/MPEG-HEVC [14]. Recently, a performance comparison of different video codecs including VP9 and x265 was published with a particular focus on video-on-demand applications [15]. However, the results of this study are difficult to interpret, because the investigated encoders were operated by using individually tuned multi-pass rate-control mechanisms, which somehow obscure the coding-efficiency capabilities provided by the particular syntax of each video coding format. At least for HEVC-based encoding, it was

978-1-5090-5966-9/16/$31.00 ©2016 IEEE

shown that by adjusting the quantization parameter (QP) for modeling inter-frame dependencies in the Lagrangian rate-distortion (R-D) optimized encoder control provides an average gain in bit-rate savings of around 5-6% when compared to the conventional Lagrangian R-D encoder control using fixed QP settings per frame [16]. Similar methods of QP adjustments, although not publicly documented, are probably also used in typical AV1/VP9 encoders.

In order to separate these rate-control related aspects from purely syntax based performance capabilities, this paper presents an experimental study of coding efficiency for five different encoder realizations of the three video coding formats AV1/VP9, AVC, and HEVC using *fixed* QPs only. The detailed settings along with a discussion of the selected software implementations, the choice of coding parameters, and the corresponding evaluation setup are given below.

This paper is organized as follows. In the next section, the selected representative encoders are introduced. *Section III* contains a description of the test methodology and evaluation setup. The detailed experimental results are presented in *Section IV*, and this paper is concluded in *Section V.*

## II. Selected Encoder Implementations

In this section, a brief overview of the selected representative encoders is presented.

### *A. AOM/AV1 Encoder*

As already noted, the first version of AV1 was released by AOM in April 2016 [17]. To the best of the authors' knowledge and based on their analysis of the source code, AOM/AV1 (Version 1) is fully based on VP9. The only difference is that AOM/AV1 contains two additional in-loop filters, which are currently disabled by default. According to [18], the corresponding increase in coding efficiency (in terms of bit-rate savings), when enabling these filters, is expected to be in the region of 2.5%. However, when enabling these filters, this gain could not be verified by the authors of this paper due to a compilation error of the current AOM/AV1 code base.

It should be noted that AOM/AV1 encoder has a two-pass coding option, which results in improved rate-distortion performance. So, this feature was enabled by the authors. Further, it should be noted that parallel processing in AOM/AV1 strictly relies on the usage of tiles, thereby implying a coding-efficiency penalty relative to the case where no tiles are used. As a consequence, multithreading in AOM/AV1 has been disabled in our experiments to not adversely affect its coding efficiency.

### *B. x265 Encoder*

x265 is developed by the VideoLAN organization [14], which is well known for the development of x264 video codec that is based on H.264/MPEG-AVC [19].

x265 provides a quite flexible trade-off between coding efficiency and computational complexity. Similarly to AOM/AV1, x265 has the two-pass coding option, which was also enabled by the authors.

### *C. H.265/MPEG-HEVC Reference Software Encoder*

For H.265/MPEG-HEVC-based encoding, the HM reference software encoder [20] was selected as the main HEVC representative. Particularly, for conducting the corresponding HEVC performance evaluation, the authors selected one of its latest versions, i.e., HM 16.6 [20].

### *D. HHI HEVC Encoder*

Fraunhofer Heinrich Hertz Institute (HHI) developed a commercial software-based encoder solution that is fully compliant with the H.265/MPEG-HEVC video coding standard [21]. This so-called HHI HEVC encoder can be operated in real-time mode (for resolutions up to 4K) as well as in non-real-time mode with different configurations to trade-off between encoding speed and coding efficiency.

The HHI HEVC encoder has been selected as an additional HEVC representative to demonstrate the capabilities of a professional HEVC-based encoding solution.

### *E. H.264/MPEG-AVC Encoder*

For evaluating the H.264/MPEG-AVC standard, the JM reference software [22] was selected due to its stability and consistency in terms of configurability. Particularly, the authors used its latest version, i.e. JM 19.0 [22].

## III. Test Methodology and Evaluation Setup

For performing the detailed performance analysis and in order to be as fair as possible due to the significant difference in the capabilities of the individual encoders, the authors of this paper used very similar settings for all tested encoders, i.e. for the AOM/AV1 coding scheme, JM reference software, HM reference software, x265, and HHI HEVC encoder. The test methodology and the evaluation setup are explained in detail below. Particularly, in *Sub-Section III.A,* the AOM/AV1 configuration is discussed, followed by the discussion of the x265 coding scheme in *Sub-Section III.B*. After that, in *Sub-Section III.C*, the HM reference software configuration is presented. Then, the HHI HEVC and JM configurations are provided in Section *III.D* and *III.E,* respectively. Finally, an overview of the performed Bjøntegaard-Delta Bit-Rate (BD-BR) measurements is presented in *Sub-Section III.F.*

### *A. AOM/AV1 Configuration*

The AOM/AV1 configuration settings are presented in Table I below. It should be noted that there is currently no official AOM/AV1 specification as well as no official AOM/AV1 encoder manual. However, as already noted in *Section II.A*, the first version of AOM/AV1 is fully based on the VP9 coding scheme (except for the mentioned in-loop de-ringing filters), thereby employing substantially the same configuration settings as VP9. Therefore, for configuring AOM/AV1, the authors used the latest recommended best-quality settings for VP9 [7]-[9]. The QP range of AOM/AV1 coding scheme was adjusted to fit the bit-rates produced by the HM reference software encoder.

Table I. Selected Settings for the AOM/AV1 Encoder

| **CODEC** | **AOM/AV1** |
|---|---|
| **Version** | **AOMedia Project AV1 Encoder, Version: b6724815f22876ca88f43b57dba09a555ef4e1b0** |
| **Recommended settings** | --best --psnr --tune=psnr --end-usage=q --passes=2 --tile-columns=0 --arnr-strength=5 --min-q=$QP --max-q=$QP --cq-level=$QP |

The reader is referred to [7]-[9] for obtaining more detailed information regarding the AOM/AV1 settings of Table I.

The IntraPeriod interval as well as the QP values were set to be similar to those used for running other tested encoders in order to be consistent (they are presented as $IntraPeriod and $QP, respectively). By such a way, it is ensured that *I-pictures* are inserted in regular time intervals (which are the “1 sec.” intervals), and at exactly the same time instances [5]. Also, it should be noted that the AOM/AV1 encoder configuration was tuned for best PSNR values.

The authors also conducted various experiments using different settings, such as setting an “arnr-maxframes” value equal to 16 (note that it is set to 7, by default), or setting a “lag-in-frames” value larger than the default value of 25, or setting “end-usage” equal to 3, etc. It finally turned out that the results are either worse or similar to those obtained using the configuration in Table I.

TABLE II. SELECTED SETTINGS FOR THE X265 ENCODER

| **CODEC** | **x265** |
|---|---|
| **Version** | **VideoLAN Project x265 Encoder, Version: 2.0** |
| **Recommended settings** | --profile=main –p=placebo --psnr --tune=psnr --pools none --no-pmode --no-pme --no-allow-non-conformance --rd=6 --rect --amp –qp=$QP --keyint=$IntraPeriod --min-keyint=$IntraPeriod --pass=2 |

### B. *x265 Configuration*

The configuration settings for the x265 encoder are presented in Table II. The reader is referred to [14] for obtaining more detailed information with regard to the x265 settings of Table II. The IntraPeriod interval as well as the QP values in the above x265 configuration were set to be similar to those used for running other tested encoders in order to be consistent (they are presented as $IntraPeriod and $QP, respectively). By such a way, it is ensured that I-pictures are inserted in regular time intervals (which are the “1 sec.” intervals), and at exactly the same time instances [5].

Also, similarly to AOM/AV1, the x265 encoder configuration was tuned for best PSNR values.

### C. *HM Reference Software Configuration*

For the HM reference software encoder [20] as well as for the HHI HEVC and the JM encoder, a Random Access (RA) configuration based on dyadic high-delay hierarchical B pictures was selected. This choice is motivated by the fact that, first, this kind of the RA configuration is typically chosen for all broadcasting and streaming applications, and second, it provides in most cases better results in terms of coding efficiency compared to low delay configurations [5]. The Group of Picture (GOP) size was set to 8 pictures, and the Intra Period was set to 24, 32, 48, and 64 pictures for video content with 24, 30, 50, and 60 fps, respectively. Also, the hierarchical B pictures were used with a fixed QP increase of 1 (i.e., the quantization step size increase of 12% [5]) between each temporal level. It is noted that the above test conditions were selected similarly to the test conditions presented in [5]. Table III below summarizes the above-mentioned HM reference software encoder [20] configuration with the Main Profile being chosen.

TABLE III. SETTINGS FOR THE HM REFERENCE SOFTWARE ENCODER

| **CODING OPTIONS** | **CHOSEN PARAMETERS** |
|---|---|
| Encoder Version | HM 16.6 |
| Profile | Main |
| Reference Frames | 4 |
| R/D Optimization | Enabled |
| Motion Estimation | TZ search |
| Search Range | 64 |
| GOP | 8 |
| Hierarchical Encoding | Enabled |
| Temporal Levels | 4 |
| Intra Period | 1 sec |
| Deblocking Filter | Enabled |
| Coding Unit Size/Depth | 64/4 |
| Transform Unit Size (Min/Max) | 4/32 |
| TransformSkip | Enabled |
| TransformSkipFast | Enabled |
| Hadamard ME | Enabled |
| Asymmetric Motion Partitioning (AMP) | Enabled |
| Fast Encoding | Enabled |
| Fast Merge Decision | Enabled |
| Sample adaptive offset (SAO) | Enabled |
| Rate Control | Disabled |
| Internal Bit Depth | 8 |

### D. *HHI HEVC Encoder Configuration*

For the HHI HEVC encoder [21], the configuration settings are most closely corresponding to the configuration settings of the HM reference software encoder, as presented in Table III above.

The HHI HEVC encoder is operated in non-real-time mode as a function of several predefined speed-ups (as further discussed in details in *Section IV*), while the encoding for each speed-up level is executed with 8 parallel threads.

### E. *JM Reference Software Configuration*

Table IV below summarizes the JM reference software encoder configuration.

TABLE IV. SETTINGS FOR THE JM REFERENCE SOFTWARE ENCODER

| **CODING OPTIONS** | **CHOSEN PARAMETERS** |
|---|---|
| Encoder Version | JM 19.0 |
| Profile | High |
| Reference Frames | 4 |
| R/D Optimization | Enabled |
| Motion Estimation | EPZS |
| Search Range | 64 |
| GOP | 8 |
| Hierarchical Encoding | Enabled |
| Temporal Levels | 4 |
| Intra Period | 1 sec |
| Deblocking Filter | Enabled |
| 8x8 Transform | Enabled |
| Rate Control | Disabled |
| Internal Bit Depth | 8 |

The above JM configuration was selected in line with the corresponding AOM/AV1, HM and HHI HEVC configurations. Note that the High Profile was selected.

### *F. Bjøntegaard-Delta Bit-Rate Measurements*

For the R-D performance assessment, the authors used a Bjøntegaard-Delta bit-rate (BD-BR) measurement method for calculating average bit-rate differences between R-D curves for the same distortion (e.g., for the same $PSNR_{YUV}$ values) [23], where *negative BD-BR values* indicate actual bit-rate savings.

The authors used R-D curves of the combined luma (Y) and chroma (U,V) components, while the combined

978-1-5090-5966-9/16/$31.00 ©2016 IEEE

$PSNR_{YUV}$ value were calculated as a weighted sum of the PSNR values per each picture of each individual component [5], i.e., of $PSNR_Y$, $PSNR_U$, and $PSNR_V$:

$$PSNR_{YUV} = (6 \cdot PSNR_Y + PSNR_U + PSNR_V)/8 \quad (1)$$

As a result, using the combined $PSNR_{YUV}$ and bit-rate values as an input to the BD-BR measurement method enables to determine a single average difference in bit-rate that takes into account the fidelity of both the luma and the two chroma components [5].

## IV. Experimental Results

For obtaining experimental results, the test sequences were selected according to the HEVC common test conditions (CTC) and software reference configurations [24]. Also, for each of these video sequences, four QP values were used: 22, 27, 32, and 37, which are the QP values used for the I-frame coding. Further, the quantization parameters were adjusted for AOM/AV1 and x265, thereby resulting in similar bit-rates as those produced by the HM reference software encoder.

Table V presents a summary of calculated BD-BR savings [23] for HM vs. AOM/AV1 as well as for HM vs. JM per each *Class A* to *D*, in accordance with the common test conditions [24], where *negative BD-BR values* indicate bit-rate savings (in contrast to positive values, which indicate the required overhead in bit-rate to achieve the same $PSNR_{YUV}$ values). It should be noted that *Class E* was not evaluated in accordance with the common test conditions [24] of the RA configuration, as well as *Class F*, which represents screen content that is out of scope of this work. Further, it should be noted that for *Class A*, the experiments were conducted only for the 8-bit sources "Traffic" and "PeopleOnStreet".

Table V. Summary of BD-BR Savings of HM Encoder vs. AOM/AV1 and JM Encoders .

| CTC Classes | HM vs. AOM/AV1 | HM vs. JM |
|---|---|---|
| Class A | **-33.5%** | **-28.8%** |
| Class B | **-40.7%** | **-40.7%** |
| Class C | **-38.4%** | **-30.4%** |
| Class D | **-38.8%** | **-28.6%** |
| **Average for all sequences** | **-38.4%** | **-32.8%** |

As seen from Table V, the average BD-BR savings of the HEVC-based reference encoder relative to AOM/AV1 and H.264/MPEG-4 AVC codecs are 38.4%, and 32.8%, respectively.

As an outcome of another experiment, Figure 1 below illustrates the speedup of the HHI HEVC encoder versus bit-rate overhead for various presets (i.e., $S_0$, $S_1$, $S_2$ and $S_3$) relative to the HM reference encoder (shown as the blue square in the origin of the coordinate axes). As can be observed from Figure 1, the HHI HEVC encoder provides a configurable trade-off between encoding speed and coding efficiency. For instance, for the speed-up level $S_0$, the performance of the HHI HEVC encoder in terms of coding efficiency is very close to that of the HM reference software encoder while its encoding speed is 31 times higher than that of the HM. At the speed-up level $S_3$ the encoding time of the HM encoder is much higher (e.g., 300 times higher) than that measured for the HHI HEVC encoder, albeit at the expense of a 12.5% BD-BR increase for the HHI HEVC encoder.

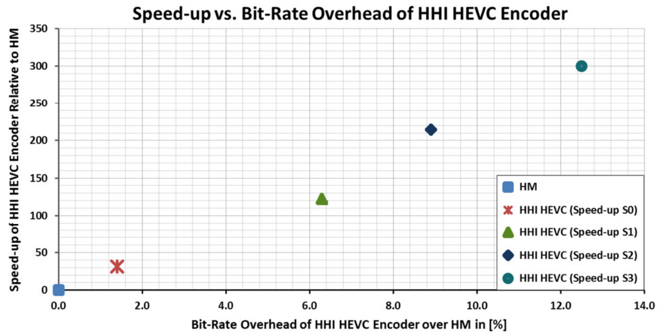


**Figure 1.** Speed-up vs. bit-rate overhead (in terms of BD-BR) for various presets of the HHI HEVC encoder relative the HM reference encoder.

Table VI below presents a summary of the BD-BR savings of HM and HHI HEVC (Speed-up $S_3$, as shown in Figure 1) encoders vs. x265. As can be clearly observed, both HM and HHI HEVC have a significant coding gain over x265. Particularly, HHI HEVC encoder achieves bit-rate savings of 3.4% (speed-up $S_3$) compared to x265, while the encoding is still 101 times faster compared to x265.

Table VI. Summary of the BD-BR Savings of HM and HHI HEVC Encoders vs. x265 Encoder.

| CTC Classes | HM vs. x265 | HHI HEVC (Speed-up $S_3$) vs. x265 |
|---|---|---|
| Class A | **-9.2%** | **0.9%** |
| Class B | **-12.7%** | **-5.1%** |
| Class C | **-15.4%** | **-2.9%** |
| Class D | **-15.8%** | **-3.1%** |
| **Average for all sequences** | **-13.9%** | **-3.4%** |

In addition, Table VII presents a summary of the BD-BR savings of x265 over both AOM/AV1 and JM Encoders. Particularly, x265 encoder has a significant coding gain of 27.4% compared to AOM/AV1, while the encoding is more than 2 times faster compared to AOM/AV1.

Table VII. Summary of the BD-BR Savings of x265 Encoder vs. AOM/AV1 and JM Encoders.

| CTC Classes | x265 vs. AOM/AV1 | x265 vs. JM |
|---|---|---|
| Class A | **-26.1%** | **-21.2%** |
| Class B | **-30.2%** | **-30.1%** |
| Class C | **-26.3%** | **-16.6%** |
| Class D | **-26.9%** | **-14.1%** |
| **Average for all sequences** | **-27.4%** | **-20.6%** |

Further, Table VIII provides a summary of the BD-BR performance evaluation (per each CTC Class) of the JM reference encoder vs. AOM/AV1 and the HHI HEVC encoder operated at speed-up preset $S_3$.

Table VIII. Summary of the BD-BR Performance Evaluation of JM Encoder vs. AOM/AV1 and HHI HEVC Encoders.

| CTC Classes | JM vs. AOM/AV1 | JM vs. HHI HEVC (Speed-up $S_3$) |
|---|---|---|
| Class A | **-6.4%** | **27.7%** |
| Class B | **1.4%** | **55.8%** |
| Class C | **-11.5%** | **25.5%** |
| Class D | **-15.1%** | **22.0%** |
| **Average for all sequences** | **-7.6%** | **34.5%** |

978-1-5090-5966-9/16/$31.00 ©2016 IEEE

In terms of encoding speed, it was found that the typical encoding times of the AOM/AV1 coding scheme implementation are very similar to those of both the HM and the JM reference encoder. Particularly, the AOM/AV1 encoder performs, on average, only 1.3 times faster than the full-fledged HM reference software, and only 1.2 times faster than the JM reference software.

To summarize the BD-BR performance of all tested encoders, Table IX provides an overview of the outcome of the conducted experiments.

TABLE IX. SUMMARIZED BD-BR EXPERIMENTAL RESULTS

| ***CODEC*** | **HEVC** | **HHI HEVC Speed-up $S_0$** | **H.264/ AVC** | **x265** | **AOM/ AV1** |
|---|---|---|---|---|---|
| **HEVC** | | **-1.4%** | **-32.8%** | **-13.9%** | **-38.4%** |
| **HHI HEVC Speed-up $S_0$** | **1.4%** | | **-31.8%** | **-12.7%** | **-37.5%** |
| **H.264/AVC** | **50.7%** | **48.4%** | | **27.3%** | **-7.6%** |
| **x265** | **16.5%** | **14.9%** | **-20.6%** | | **-27.4%** |
| **AOM/AV1** | **65.7%** | **63.1%** | **10.5%** | **39.1%** | |

As shown in Table IX, the H.264/MPEG-AVC High Profile encoder achieves an average gain of 7.6% in terms of BD-BR savings compared to AOM/AV1. Also, in order to achieve the same average $PSNR_{YUV}$ values of HEVC, when using the AOM/AV1 encoder, the BD-BR overhead of 65.7% is required.

When compared with x265, AOM/AV1 has a significant overhead of 39.1%, while the encoding is more than 2 times slower. On the other hand, HM has a coding gain of 13.9% over x265, while the encoding is about 3 times slower. Also, compared to x265, the HHI HEVC encoder at speed-up $S_0$ provides a coding gain of 12.7% with more than 10 times faster encoding. It is noted that since the fitting of R-D curves slightly differs when fitting the R-D curve of one encoder to another and *vice versa*, the product $(100 + b_1)(100 + b_2)$ for each pair $(b_1, b_2)$ of corresponding BD-BR values (e.g., H.264/AVC vs. AOM/AV1 and AOM/AV1 vs. H.264/AVC) is approximately equal to 10.000.

## V. CONCLUSION

A performance evaluation of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC encoders was presented and discussed in detail. According to the experimental results, the coding efficiency of AV1/VP9 was shown to be inferior to all tested encoders, in particular relative to the reference encoders of AVC and HEVC with an average bit-rate overhead at the same objective quality of 10.5% and 65.7%, respectively. Also, it was shown that for these bit-rate overheads, the AV1/VP9 encoding times are quite similar to those of the full-fledged HM and JM reference software encoders. Further, the HM encoding times are larger by a factor of 300 compared to those of the commercial HHI HEVC encoder with a trade-off of 12.5% bit-rate overhead only; on the other hand, for the small bit-rate overhead of just 1.4%, the HHI HEVC encoder is still about 31 times faster, on average, than the HM reference encoder. When compared to x265, the HHI HEVC encoder provides coding gain of 12.7%, while the encoding is more than 10 times faster compared to x265; on the other hand, for smaller bit-rate savings of 3.4% relative to x265, the HHI HEVC encoder is about 100 times faster than x265.

## REFERENCES


[1] ITU-T, Recommendation H.265 (04/13), Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of Moving Video, High Efficiency Video Coding.

[2] ITU-T, Recommendation H.265 (10/14), Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of Moving Video, High Efficiency Video Coding.

[3] ITU-T, Recommendation H.265 (04/15), Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of Moving Video, High Efficiency Video Coding, Online: http://www.itu.int/rec/T-REC-H.265-201504-I.

[4] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol.13, no.7, pp.560-576, Jul. 2003.

[5] J. 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 Transactions on Circuits and Systems for Video Technology,* vol. 22, no.12, pp.1669-1684, Dec. 2012.

[6] B. Li, G. J. Sullivan, and J. Xu, "Comparison of compression performance of HEVC Draft 9 with AVC high profile and performance of HM9.0 with temporal scalability characteristics," JCTVC-L0322, 12th JCT-VC meeting, Geneva, Switzerland, Jan. 2013.

[7] WebM™: an open web media project, VP8 Encode Parameter Guide, 2-Pass Best Quality VBR Encoding, Online: http://www.webmproject.org/docs/encoder-parameters/#2-pass-best-quality-vbr-encoding

[8] J. Bankoski, P. Wilkins, and X. Yaowu, "Technical overview of VP8, an open source video codec for the web," *IEEE International Conference on Multimedia and Expo (ICME) 2011*, pp.1,6, 11-15 July 2011.

[9] Chromium open-source browser project, VP9 source code, Online: http://git.chromium.org/gitweb/?p=webm/libvpx.git;a=tree;f=vp9;hb=aaf61dfbcab414bfacc3171501be17d191ff8506

[10] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders," *Picture Coding Symposium (PCS), 2013*, pp.394-397, 8-11 Dec. 2013.

[11] D. Grois, D. Marpe, T. Nguyen, and O. Hadar, "Comparative Assessment of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders for Low-Delay Video Applications", *Proc. SPIE* Vol. 9217, Sept. 2014.

[12] M. Řeřábek, and T. Ebrahimi, "Comparison of compression efficiency between HEVC/H.265 and VP9 based on subjective assessments", *Proc. SPIE* Vol. 9217, Sept. 2014.

[13] Alliance for Open Media, Press Release Online: http://aomedia.org/press-release/

[14] Projects from VideoLAN, x265 software library and application, Online: https://www.videolan.org/developers/x265.html

[15] J. De Cock, A. Mavlankar, A. Moorthy, and A. Aaron, "A Large-Scale Video Codec Comparison of x264, x265 and libvpx for Practical VOD Applications", *Proc. SPIE* Vol. 9971, Sept. 2016.

[16] S. Li et al., "Lagrangian Multiplier Adaptation for Rate-Distortion Optimization with Inter-frame Dependency", *IEEE Transactions on Circ. and Systems for Video Technology,* vol. 26, no.1, pp. 117-129, 2016.

[17] Alliance for Open Media, git repository online at: https://aomedia.googlesource.com/aom

[18] J.-M. Valin, "Directional Deringing Filter", Internet Draft, Network Working Group, Internet Engineering Task Force, 2016.

[19] Projects from VideoLAN, x264 software library and application, Online: https://www.videolan.org/developers/x264.html

[20] HEVC Reference Software, Online: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[21] Fraunhofer HHI, "HEVC - Software and Hardware Solutions", Online: http://www.hhi.fraunhofer.de/departments/video-coding-analytics/products-technologies/hevc-software-and-hardware-solutions.html.

[22] H.264/AVC Software Coordination, JM Reference Software, Online: http://iphome.hhi.de/suehring/tml

[23] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves", *ITU-T Q.6/SG16 VCEG 13th Meeting*, Document VCEG-M33, Austin, USA, Apr. 2001.

[24] F. Bossen, "Common HM test conditions and software reference configurations," document JCTVC-L1100 of JCT-VC, Geneva, CH, Jan. 2013.