SIP (2021), vol. 10, e11, page 1 of 17 © The Author(s), 2021. Published by Cambridge University Press in association with Asia Pacific Signal and Information Processing Association. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution, and reproduction in any medium, provided the original work is properly cited.
doi:10.1017/ATSIP.2021.10

ORIGINAL PAPER

Compression efficiency analysis of AV1, VVC, and HEVC for random access applications

TUNG NGUYEN AND DETLEV MARPE

AOM Video 1 (AV1) and Versatile Video Coding (VVC) are the outcome of two recent independent video coding technology developments. Although VVC is the successor of High Efficiency Video Coding (HEVC) in the lineage of international video coding standards jointly developed by ITU-T and ISO/IEC within an open and public standardization process, AV1 is a video coding scheme that was developed by the industry consortium Alliance for Open Media (AOM) and that has its technological roots in Google's proprietary VP9 codec. This paper presents a compression efficiency evaluation for the AV1, VVC, and HEVC video coding schemes in a typical video compression application requiring random access. The latter is an important property, without which essential functionalities in digital video broadcasting or streaming could not be provided. For the evaluation, we employed a controlled experimental environment that basically follows the guidelines specified in the Common Test Conditions of the Joint Video Experts Team. As representatives of the corresponding video coding schemes, we selected their freely available reference software implementations. Depending on the application-specific frequency of random access points, the experimental results show averaged bit-rate savings of about 10–15% for AV1 and 36–37% for the VVC reference encoder implementation (VTM), both relative to the HEVC reference encoder implementation (HM) and by using a test set of video sequences with different characteristics regarding content and resolution. A direct comparison between VTM and AV1 reveals averaged bit-rate savings of about 25–29% for VTM, while the averaged encoding and decoding run times of VTM relative to those of AV1 are around 300% and 270%, respectively.

Keywords: AV1, HEVC, VVC

Received 22 July 2019; Revised 4 June 2021

I. INTRODUCTION

More than eight years have passed since the publication of the first version of the High Efficiency Video Coding (HEVC) [1] standard. Research on its successor, the Versatile Video Coding (VVC) [2] standard developed by the Joint Video Experts Team (JVET) [3] of ITU-T and ISO/IEC has been finished recently with the official approval of Recommendation H.266 by ITU-T in August 2020. An alternative video coding scheme resulting from a different line of development that was driven by the industry consortium Alliance for Open Media (AOM) is AOM Video 1 (AV1) [4].

Recently, experts have spent lots of efforts in assessing the compression efficiency of the three video coding schemes mentioned above, as documented in [5–10]. Outcomes and conclusions of those studies, however, vary to a quite large extent due to their varying foci on different application scenarios and their often diverging choices of software encoders and corresponding settings. In contrast to the existing literature, this paper relies on a rigorous evaluation framework that is based on strict requirements for the *random access* functionality, which need to be fulfilled for a quite large application space of real-world video compression usage such as digital video broadcasting or low-latency live streaming. Important key features such as fast channel start-up or channel switching, seeking during playback, or fast error recovery cannot be offered without requiring an appropriate level of random access capability in the first place.

The experiments of this study have been conducted accordingly by using the most recently available reference software implementations. These are HM-16.21 for HEVC, VTM-8.0 for VVC, and a snapshot from the AV1 Codec Library git repository [11]. We conducted the simulations using the three software packages within the controlled experimental environment of JVET, referred to as Common Test Conditions (CTC) [12]. CTC specify a test scenario representing random access applications and a test set covering a broad spectrum of content and resolution characteristics.

As a result of our evaluation, it can be concluded that both AV1 and VTM clearly achieve higher compression efficiency than HM with averaged bit-rate savings of about

Department of Video Communication and Applications, Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany

Corresponding author:
Tung Nguyen
Email: tung.nguyen@hhi.fraunhofer.de

10% for AV1 and about 36% for VTM, respectively. Both software encoders require significantly higher run times than the HM encoder, whereas their decoding run times indicate a manageable increase in computational complexity relative to the HM decoder. In addition to the results generated under the CTC guideline, we also provide additional data for a different random-access configuration that is assumed to be also of practical interest. For reproducing the experiments conducted in this paper, the interested reader is invited to reconstruct all operation points using the provided configuration files for HM/VTM and the command line parameters for AV1. The authors further provide a simple script to run the encoder software implementations. The whole package is available at https://bit.ly/3vU3VCK.

The organization of this paper is as follows. Section II reviews the relevant existing literature and discusses the commonalities and differences relative to the evaluation in this paper. The section also describes the random access property, works out the differences in the representation of the temporal prediction structure between HM/VTM and AV1, and briefly discusses the complexity aspect. Section III describes the experimental setup of the three reference software implementations and Section IV presents and discusses the outcome of the evaluation. Finally, Section V concludes this paper.

II. PROBLEM STATEMENT

This section deals with relevant aspects that need consideration for the proper assessment of the experiments and results presented in this paper. It starts with a brief overview of existing performance analyses and discusses some recent publications in more detail. The following two subsections discuss and analyze the random access property and the representation of inter-predicted pictures between successive random access points. Both aspects play an essential role in the different outcomes reported by existing publications. This section closes with a brief discussion of some aspects related to the assessment of computational complexity.

A) Previous work

A vast number of publications exist that evaluate the compression efficiency of different video coding technologies or video encoder implementations. For the three video coding schemes tested in this paper, the authors of [5–9, 13–16] reported the evaluation outcome in terms of objective quality. In addition, the authors of [17–19] reported evaluation results for both objective and subjective quality. In addition to the evaluation results, the authors of [7–10, 20] give a brief overview of the coding tools in AV1 and VVC. However, not all cited publications focused on compression performance evaluation, and the outcomes and conclusions are not always consistent. One aspect that contributes to the diverging results is the choice of employed software implementation as a representative for a particular video coding scheme. For instance, some papers used a development snapshot of AV1, while others employed the Joint Exploration Model (JEM) software as the representative of VVC, yet others used both. A further aspect is the usage of a configuration that does not support random access, which is necessary for the applications of interest in this paper. This paper's evaluation addresses the two mentioned aspects by using a recent AV1 software implementation, the VVC reference software implementation (VTM), and a configuration supporting random access. Note that JEM is an explorative software implementation created by the JVET before the official VVC standardization activity started, and the JEM software does not represent the software foundation for the VTM implementation.

Even a previous evaluation by the authors of this paper in [6] does not adequately fulfilll the aspects mentioned above because the experiments employed an AV1 development snapshot and the JEM software. Notably, the usage of the final tagged AV1 version or later is crucial because intermediate AV1 software implementations do not show the full potential of the finalized AV1 coding technology. Further descriptions of the main AV1 coding tools are given in [10, 20] together with a performance evaluation relative to other encoders. The investigated temporal prediction structure in those studies consists of an intra-only predicted picture at the beginning of the video sequence and no further random access points. Although this kind of configuration seems to be favorable for AV1 in terms of coding efficiency, we will explain in the subsection below how and why the random access property is of utmost importance for a wide range of applications, and therefore cannot be ignored completely.

Further recent publications are [8, 9, 19, 21], and they reported objective or subjective quality fully, or partially, or both, in different scenarios. The first among the recent publications [19] includes a subjective evaluation of the video coding schemes in an adaptive streaming application. Their outcome indicated that AV1 performs slightly better than HEVC objectively, while the subjective quality is similar to HEVC's delivered quality. The second [8] has a similar outcome to that of [19], although they used JEM as representative of VVC. In that study, the authors reported an objective compression efficiency improvement of about 10% for AV1 relative to HM, but their subjective evaluation showed that AV1 and HM performed very similarly. The authors of [9] observed a relatively close overall performance of AV1 and VTM. Specifically, they reported an overall bit-rate overhead of AV1 relative to VTM in the range of about 5% only. However, they remarked that the bit-rate overhead of AV1 relative to VTM for UHD content is significantly higher with 20%, and they generated their reported results by using a non-random-access capable temporal prediction structure. Finally, the most recent publication [21] reported that for the same bit rate, AV1 and HM show an insignificant difference in perceptual quality, while VTM performs significantly better than AV1 and

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press

Table 1. Distance in number of frames between two IRAP depending on the frame rate of the input sequence following the JVET CTC using GOP 32

Frame rate	≤30 Hz	50 Hz	60 Hz	100 Hz
∼1 s IRAP period	32	64	64	96
∼2 s IRAP period	64	96	128	192

HM. They also used a random-access configuration for their experiments, and their findings matched the observations of [8, 19], although the chosen test sets and the selected encoder configurations for AV1 were different.

B) Random access

A wide range of typical video coding applications requires the so-called random access (RA) property. RA enables the possibility to tune into the bitstream, which, for example, is necessary for broadcasting when users want to switch between different channels. RA also allows for features such as seeking in video content for a specific temporal position. Another functionality provided by RA may be given by a certain degree of error resilience. More precisely, an application that supports RA can guarantee an error-free reconstruction of pictures from the bitstream after a specific amount of time, even if the transmission got temporarily interrupted. An internet-based use case relying on RA is live streaming, where a user can tune into the live stream only at random access points.

HEVC and VVC specify so-called intra random access point (IRAP) pictures as pictures that a conforming decoder can reconstruct without using the content of any other pictures [22]. IRAP pictures are typically uniformly distributed over time, and consequentially, the constant distance between two random access points is a parameter balancing the trade-off between compression efficiency and usability. By increasing the distance between two successive IRAPs, the compression efficiency may be improved depending on the content. On the other hand, the usability suffers from a large temporal IRAP distance due to the delay experienced when tuning into the bitstream. An encoder can achieve the random access property by encoding pictures using intra-only prediction regularly at a distance of a specific time interval. A typical trade-off between compression efficiency and usability, in terms of delay and user experience, is a constant time interval of about 1 s. Since the frame rate dictates the number of displayed pictures within a time frame, the period of random access points depends on the individual frame rate of each test sequence in the test set. Table 1 summarizes the distances in number of frames between two successive random access points for different frame rates, according to the JVET CTC.

In addition to being restricted to intra prediction for an IRAP picture, the decoder also has to reset the reference picture buffer for inter prediction at random access points. Pictures temporally later than the random access point should not use reconstructed pictures earlier than that of the random access point as reference pictures for inter prediction. Furthermore, by providing additional high-level syntax, the functionality of RA can be tailored to the support of specific use cases. For example, the HEVC standard specifies three different IRAP picture types: Instantaneous Decoder Refresh (IDR), Clean Random Access (CRA), and Broken Link Access (BLA) [22].

C) Temporal prediction structure

The representation of the inter-predicted pictures between two successive random access points is a fundamental difference between HM/VTM and AV1. Both the HEVC and the VVC standard allow the display order to be different from the transmission order, resulting in the possibility to reorder pictures within a time period. In combination with bi-prediction and hierarchical prediction and quantization structure, the reordering and organization in so-called Group-of-Pictures (GOP) enables the possibility to achieve higher compression efficiency [23]. Figure 1 illustrates an example of the GOP concept with a GOP size equal to eight, i.e. a GOP spans over eight successive pictures. Each box shape in Fig. 1 labeled with **b** or **B** stands for a picture of the video sequence. Every GOP consists of a keyframe, depicted by the most right box shape in Fig. 1, and the keyframe is the first picture of the GOP transmitted in the bitstream for the associated GOP. The first number in the angle brackets denotes the display order, which is sequential from left to right according to the picture's time stamp. The second number denotes the coding or transmission order. The hierarchical quantization structure, achieved via the picture-level quantization parameter (QP), usually assigns higher QP values for pictures within the GOP that are less referenced by other pictures. Let x be the QP value for the keyframe, then the QP value for the picture at $\langle 4|2 \rangle$ is $x+1$, $x+2$ for the pictures $\langle 2|3 \rangle$ and $\langle 6|6 \rangle$, and $x+3$ for the remaining pictures of the GOP. Note that the mentioned configuration is an example only and does not reflect the actual configuration for HM or VTM.

A video sequence coded with HEVC/VVC syntax in random access is a sequence of GOPs with the reordering limited to pictures within a GOP and an intra-only predicted keyframe for random access points. Fig. 2 shows an example configuration for a 60 Hz video where the IRAP period is equal to 64, and the GOP size is equal to 16. Let the first number of the angle brackets be t, the time stamp of the picture. Let further assume that a conforming decoder wants to tune into the stream at the picture with the time stamp $t=64$, which is also the first picture in the fifth GOP transmission order. However, the decoder does not have to parse and reconstruct the fifth GOP's remaining pictures to tune into the stream. The fifth GOP pictures except the keyframe are so-called Network Abstraction Layer (NAL) units of Random Access Skipped Leading (RASL) picture type. A decoder that tunes into the stream at that point can ignore the RASL pictures completely.

In the AV1 syntax, the display order is the same as the coding order, and a reordering process necessary for GOP structures is consequently not required. A temporal representation similar to GOP structures is still possible thanks

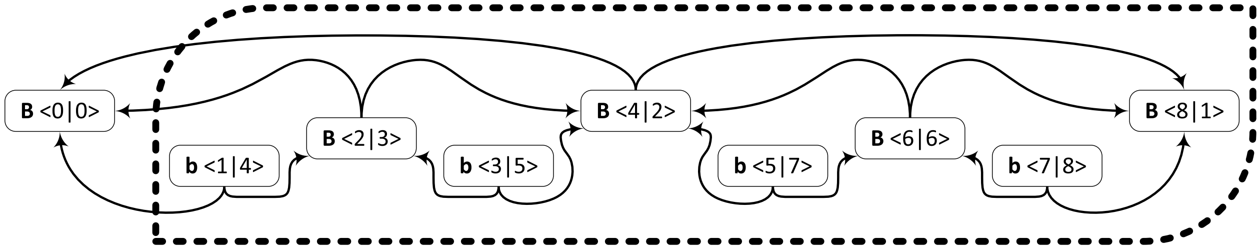


Fig. 1. Diagram shows a Group-of-Pictures (GOP) structure of a size equal to eight. Each box shape denotes a picture, and the pictures within the dotted outline are forming the GOP. The first number in the angle brackets denotes the actual display order, while the second number denotes the coding or transmission order. **B** denotes a reference picture, whereas a non-capitalized **b** stands for a non-reference picture. Finally, the arrows denote the reference pictures for each picture. Note that the vertical arrangement of the boxes reflects the corresponding hierarchical temporal layering of the pictures.

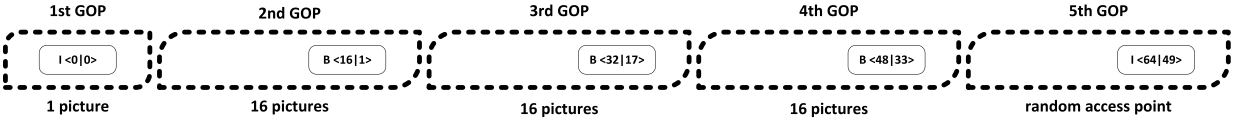


Fig. 2. Diagram shows the GOP sequence for a 60 Hz video with an IRAP period configuration equal to 64 and a GOP size configuration equal to 16. When the keyframe of the GOP (marked as a box) is an IRAP picture, the corresponding GOP provides the random access point, which is the fifth GOP in this illustrated example. At the beginning of the transmission, the first GOP has the size equal to one, consisting of the keyframe only.

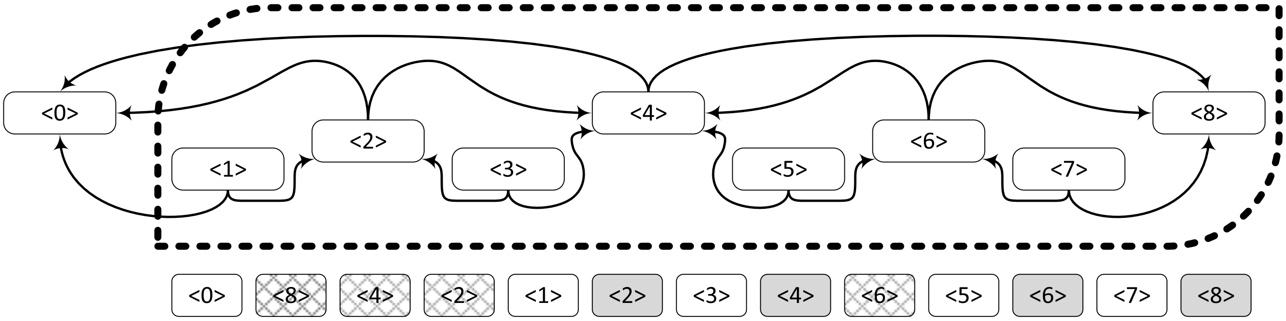


Fig. 3. Diagram shows a simplified GFG structure of a size equal to eight. Each box shape denotes a picture, and the number in the brackets denotes the time stamp of the corresponding picture. In the simplified illustration, the prediction structure is the same as the GOP structure of HM/VTM. However, instead of a reordering process and with the support of the AV1 syntax specification, the AV1 encoder transmits pictures before their designated display time and marks them as non-displayable. Below the GFG structure, the list of pictures denotes the transmission order for the same GFG, and cross-shaded box shapes denote pictures that are not displayed immediately. The white-shaded box shapes denote pictures that are reconstructed and displayed immediately, whereas gray-shaded box shapes denote pictures using the previously transmitted non-displayable pictures.

to a reference buffer for up to seven pictures and the possibility to mark pictures as non-displayable. Typically, the AV1 encoder transmits pictures having a time stamp that lies in the future relative to the current display time as non-displayable. For the display time at the same position as the earlier transmitted non-displayable picture, a dedicated syntax specifies the usage of the already transmitted non-displayable picture. Therefore, pictures consisting of the dedicated syntax consist of a corresponding header syntax only. Figure 3 illustrates a simplified example of such a so-called Golden-Frame Group (GFG). After transmitting the first picture, the three following pictures are non-displayable, followed by a regular picture at display time equal to one. The picture at display time equal to two consists of an "empty" picture with header syntax only that specifies the usage of the fourth transmitted picture. Overall, the whole scheme is a generalized approach of the Alternate Reference Frame (ARF) concept of VP9 [24], without encapsulating two pictures into the same transmission unit, referred to as Open Bitstream Unit (OBU) in AV1. Note that by using the concept of non-displayable pictures together with "empty" pictures that only reference their associated non-displayable counterparts within a GFG, an implicit picture reordering can be achieved in AV1.

Applications requiring the low delay (LD) property, such as video conferencing, typically use an alternative temporal configuration with a single intra-only predicted picture at the beginning of the video sequence. For LD applications, the display order should be the same as the transmission order to minimize the delay, which is also specified by a test scenario in the JVET CTC. An application that may be suitable for a single intra-only predicted picture at the beginning of the video sequence is adaptive streaming supporting the spatial resolution change. In that application scenario, the service provider typically generates and makes available multiple bitstreams of the same video sequence with different time offsets relative to its starting point. It is worth mentioning that for VVC such an adaptive resolution change is possible without using the multiple bitstreams concept above, thanks to the Reference Picture Resampling (RPR) concept [25].

D) Complexity assessment

Many factors contribute to the complexity of a video encoding or decoding algorithm, both in terms of computational complexity and memory bandwidth. Therefore, it is noteworthy that a single empirical metric cannot capture all aspects of practical interest to estimate the complexity of a given algorithm. Nevertheless, for providing some figures that allow at least for a *relative* comparison of our chosen reference software implementations, we measured the execution or run time of encoding or decoding on a single platform. By doing so, we are fully aware of the fact that run times may only give a reasonable indication of the change in complexity if the investigated implementations rely on a similar software architecture and have a similar degree of optimizations. To a certain extent, that remains true for the comparison between VTM and HM, whereas a direct complexity comparison between HM/VTM and AV1 on pure run-time measurements is rather challenging, at least when using the corresponding reference software implementations only. Therefore, the reported run times are only meaningful to a limited extent for a complexity assessment of the investigated video coding schemes, and one should interpret the corresponding numbers with care. Note that only recently an open-source VVC encoder [26] was published that achieves the same rate-distortion performance of VTM at half of its encoding time and can be alternatively configured such that it offers a slightly lower compression-efficiency performance improvement over HM at only 67% of HM encoding time [27].

III. EXPERIMENTAL CONFIGURATION

This section describes the selected encoder configurations according to the CTC in the context of the aspects discussed in the previous section. It starts with a description of the test set, followed by a description of the used software implementations. Finally, the last subsection describes and discusses the actual encoder parameters and configuration values and their relation to the previous section.

A) Test sequences

The employed CTC test set covers a broad spectrum of typical camera captured content with a classification into sequence classes. Table 2 summarizes the test set by listing the test sequence names and their respective class label. Each class corresponds to a different spatial resolution or content characteristic, and the classification provides a better overview. The test set consists of 19 test sequences in total, split into six UHD, five HD, four WVGA, and four QWVGA test sequences. All UHD test sequences and two of the HD test sequences, i.e. **MarketPlace** and **RitualDance**, are 10-bit content, whereas the remaining test sequences are 8-bit content. The composition with a relatively high number of high-resolution and 10-bit content emphasizes the primary focus of the VVC development. Note that the two classes labeled as E and F, not listed in Table 2, are not part of the experiments for this evaluation. In class E, the corresponding test sequences represent typical content that is suitable for LD applications, and thus, it is not part of the random access test scenario. Class F consists of screen content test sequences, and they are optional and out of the scope for the evaluation in this paper. Another peculiarity of the JVET CTC is the exclusion of the class D results for the averaged bit-rate savings due to the focus on high-resolution content. Therefore, we apply the same to the overall averaged results presented in the evaluation outcome, i.e. excluding the class D results for the averaged results.

Table 2. Test sequences of the CTC test set that were used for the experiments in this paper

Class	Sequences	Resolution
A1	Tango2, FoodMarket4, CampfireParty2	UHD (3840×2160)
A2	CatRobot1, DaylightRoad2, ParkRunning3	UHD (3840×2160)
B	MarketPlace, RitualDance, Cactus, BasketballDrive, BQTerrace	HD (1920×1080)
C	BasketballDrill, BQMall, PartyScene, RaceHorsesC	WVGA (832×480)
D	BasketballPass, BQSquare, BlowingBubbles, RaceHorses	QWVGA (416×240)

One concern that often arises is the possibility of overfitting to the test set, i.e. a video coding scheme may perform only well on the test set used for the development. Such an overfitting typically occurs in traditional learning algorithms when the number of model parameters exceeds the number of input or training samples. In the video compression context, the results reported in [19, 28, 29] using different test sequences show that VVC performs similarly for content not included in the JVET CTC test set. Moreover, the authors of [9] report very similar results for the JVET CTC test set and an alternative test set employed for the AOM development. Specifically, the reported difference in averaged bit-rate savings is only 1% between the AOM and the JVET CTC test sets when benchmarking AV1 relative to VTM.

B) Software implementations

The reference software implementation of HEVC is HM, and for the experiments of this evaluation, version 16.21 was used. For VVC, the reference software implementation is VTM, and the experiments used version 8.0, which was released in February 2020 by the JVET. For AV1, the git repository [11] consists of two final tagged versions labeled as **v1.0.0** and **v1.0.0-errata1**. Derived from the time stamps, we regard the two tag versions as finalized software implementations conforming to the AV1 specification. The absence of additional public tags indicates a different software development philosophy relative to the HEVC/VVC activity with incremental software versions after each standardization meeting. Nevertheless, the

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press

AV1 repository indicates regular activity, i.e. changes to the master branch occur regularly and frequently. Therefore, we conducted limited experiments to reveal the progress between the tag version **v1.0.0-errata1** and recent commit states. Several recent commit states show a compression efficiency improvement of about 3% relative to the tag version for the two low-resolution sequence classes C and D **v1.0.0-errata1**. More significant is the reduction in encoding run time, which goes down from 60 to 90 times to about three times relative to the HM encoding run time. Another observation is the consistency of the AV1 reference software implementation newer than the tag version **v1.0.0-errata1** in terms of performance and run times. For all tested commit states, the operation points and the encoding run times are very similar. As a sanity check, we decoded the bitstreams generated by the used recent commit states with the tag version **v1.0.0-errata1** decoder and vice versa. Since both decoders processed the bitstreams without problems, one may assume that the bitstream syntaxes are compatible. The final results generated for this paper used an intermediate commit state with the following commit id:

85a9314

C) Encoder configurations

Recommended settings

According to the JVET CTC, the default IRAP configuration in the random access test scenario is such that the temporal distance between two successive IRAPs is about 1 s. As described in Section II, this chosen setting is a balanced trade-off between usability and compression performance for the given target applications. According to the guidelines of Digital Video Broadcasting (DVB), it is recommended that an IRAP picture occur on average at least every 2 s. Also, it is explicitly mentioned that for applications with rapid channel hopping, it may be appropriate to further reduce the IRAP period to 1 s [30]. For adaptive streaming, the recommended IRAP period typically ranges from 2 to 4 s, depending on whether the underlying service wants to provide better user experience and more stability, one the one hand, or higher compression efficiency on the other hand [31]. Note, however, that there is an ongoing activity in the DASH industry forum [32] to provide support for low-latency streaming of Live Services with an encoder-display latency comparable to the latency when distributing the same content over cable or satellite broadcast, where an IRAP distance of 0.5–1.5 s is recommended. We generated experimental results for an IRAP period configuration equal to 1 s (IRAP1) and equal to 2 s (IRAP2) to satisfy the needs of the two different application scenarios of digital video broadcasting and adaptive streaming.

The coding tool configurations for VTM according to CTC represent a balanced trade-off between compression efficiency and encoder/decoder complexity from the JVET point-of-view. A trade-off configuration example is the Multiple Transform Set (MTS [33]) configuration in VTM. MTS is a forward-adaptive driven scheme specifying the usage of a transform different from the default DCT-II integer approximation. There exists a configuration parameter for MTS in the VTM that limits the number of maximum transform candidates tested during the encoder rate-distortion optimization. To achieve a different trade-off other than the recommended one of the CTC, one can vary the parameter. The results in [34] show that by decreasing the default configuration value by one, the encoding run time can be reduced by 10% at a cost of 0.15% bit-rate overhead only. Although MTS is a transform-level coding tool, the default MTS setting specifies its usage for intra-predicted blocks only due to an imbalanced trade-off for the test scenarios using inter prediction, such as the random access scenario. However, a typical hardware decoder implementation passes the reconstructed transform coefficient levels into the scaling process regardless of the prediction type. Consequently, MTS usage for inter-predicted blocks is an encoder-side limitation in VTM rather than a decoder-side syntax-based limitation. It is up to commercial vendors to use MTS for inter-predicted blocks in their VVC encoder product and to implement a fast encoder search for MTS. Note that the conducted experiments did not use the abovementioned alternative configurations for MTS, i.e. the default MTS configuration according to the provided configuration file was used.

Different recommendations for the AV1 reference encoder settings exist, and one of the crucial parameters is the mode controlling the picture-level quantization. The AV1 reference encoder implementation supports four different modes controlling the picture-level quantization: `cbr`, `vbr`, `cq`, and `q`. Limited experiments conducted for our previous evaluation in [6] showed that the `vbr` mode performs best at that time, whereas the recommendation in [5] is the Constant Quantization Parameter mode `cq`. Other studies by AV1 experts used the `q`mode that we also used in the experiments conducted for the evaluation in this paper to achieve the best compression efficiency for AV1. Note that we conducted further limited experiments for the used AV1 commit state, and the results indicate that the difference in BD-rate between `vbr` and `q` is less than 1% on the used test set. Note that a disadvantage of the selected `q`mode is the non-deterministic behavior of the encoder control, as the additional results provided will confirm, making the evaluation process challenging.

HM-16.21 and VTM-8.0

The HEVC and VVC reference software implementations provide pre-defined configuration files for the test scenarios specified in the CTC. For the tested random-access scenario, the configuration file consists of recommended configuration values for the coding tools and the GOP structure. Specifically, the GOP size is equal to 32 [35], and the IRAP period is according to the values listed in Table 1, i.e. it depends on the frame rate of the video sequence. For reference, the configuration parameters and their respective values passed to the HM-16.21, and VTM-8.0 encoders are as follows:

- -c encoder_random_access_main10.cfg
  This setting specifies the usage of the recommended configuration file for the random access test scenario. In the case of HM-16.21, the configuration file name is encoder_random_access_main10_gop32.cfg, whereas the file name is encoder_random_access_gop32.cfg for VTM-8.0 taken from [35].
- -c [sequence].cfg
  The second configuration file specifies video sequence-specific parameters such as the spatial resolution and frame rate.
- -q [value]
  This setting specifies the QP that controls the quantization step size. Since the Bjøntegaard Delta rate (BD-rate) [36] calculation requires four operation points, the CTC specifies the base QP values as follows.

  $$\mathrm{QP} \in \{22, 27, 32, 37\}$$

AV1 (85a9314)

The configuration parameters and their respective values passed to the AV1 encoder are as follows:

- tune=psnr
  This setting specifies the distortion metric, which is mean squared error (MSE) for measurements in peak-signal-to-noise-ratio (PSNR), and it is essential when comparing in terms of objective evaluation.
- -w [width]
  This parameter specifies the sequence-specific spatial width of each picture within the video sequence.
- -h [height]
  This parameter specifies the sequence specific spatial height of each picture within the video sequence.
- -fps [rate]
  The sequence-specific frame rate of the video sequence is specified by this parameter.
- -b 10
  Since the CTC specifies the usage of 10-bit internal processing for both HM and VTM, we apply the same to AV1, which also enables a slightly higher compression efficiency for 8-bit content.
- cpu-used=0
  This setting turned out to be a speed mode configuration with a configuration value equal to 0 stands for the encoding mode delivering the best compression efficiency. Note that our previous evaluation in [20] used a configuration value equal to 1 due to a better trade-off between encoding run time and compression efficiency at that time. For this evaluation, we used the configuration value equal to 0 thanks to the advance in run time reduction mentioned above to achieve the best performance for the AV1 reference implementation.

Quantization control

- end-usage=q
  The picture-level quantization mode that we have discussed above.
- cq-level=[value]
  This setting specifies the base QP when operating in the picture-level quantization mode **q**. When using the same QP values as specified for HM and VTM, the AV1 operation points significantly differ from those of HM/VTM. Specifically, the operation points have higher bit rates in the first place and lead to higher encoding run times in the second place. Therefore, we used the following base QP values to generate the four operation points for the BD-rate calculation:

  $$\mathrm{QP} \in \{28, 39, 50, 61\}$$

Random access

- kf-min-dist=[value]
- kf-max-dist=[value]
  The AV1 reference encoder can adaptively select the distance between two successive intra pictures, a feature that is non-existent in HM/VTM but typically available for non-reference implementations. These two settings control the minimum and the maximum distance between two successive intra pictures and let the encoder selects the distance depending on the content. By using the same value for both parameters, the AV1 encoder generates bitstreams that have intra pictures in a regular interval. Please be aware that the value depends on the frame rate of the video sequence, and we selected it according to Table 1, which is the same for HM/VTM.
- enable-fwd-kf=1
  From the bits per picture plot in [20], one can observe two peaks close together for AV1 in the center area of the x-axis, which is a random access point. That is because the previous evaluation in [20] did not set a value equal to one for this parameter that enables an open GOP structure similar to that of HM/VTM. In a limited side experiment that we conducted, we found out that the difference in compression efficiency between a configuration with and without the option enabled is about 7–10%.

Temporal prediction structure

- passes=2
  The usage of the two-pass encoding mode seems to be necessary for the correct operation of the ARF technique mentioned in Section II-C). Another setting related to the two-pass encoding mode is the lag-in-frames parameter, which is per default equal to 19 in the used AV1 commit state version. It specifies the number of future frames that the AV1 encoder can use to make decisions on the temporal structure.
- min-gf-interval=[value]
- max-gf-interval=[value]
  Similar to the IRAP period configuration, the AV1 encoder implements an adaptive mechanism to place keyframes depending on the content. This setting controls the minimum and the maximum keyframe interval, or equivalently the minimum and the maximum distance between two successive ARF. Note that AV1 supports multiple ARF buffers, and by the term ARF we denote the

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press

keyframe as described in Section II-C). Both the min-gf-interval and the max-gf-interval have been set to the same value of 16, representing a similar temporal prediction structure as used in HM/VTM. Analysis results presented in [20] indicate that fixing both parameters to the same value results in a fixed GFG size, albeit with the choice of a value of 16 at half the dimension of the GOP size 32 as used in HM/VTM. Note that the version of the AV1 encoder, we used for our experiments, does not allow for the choice of larger values than 16 for the keyframe interval size. When not setting the two parameters, the AV1 encoder uses the so-called automatic reference frame mode, i.e. the encoder decides to select the distance between two successive keyframes based on the statistics of the individual content collected during the first pass. Such a concept is similar to that of adaptive, content-dependent GOP structures, and the results presented in [37, 38] show that adaptive GOP structures may also result in compression efficiency improvements for HM/VTM.

It should be noted, however, that, in general, the two-pass mode of operation of the AV1 encoder causes an encoding delay (in addition to the implicit picture reordering within a GFG), which may be undesirable in real-time applications such as live broadcasting or live streaming.

IV. RESULTS AND DISCUSSION

We generated the BD-rate values by running the reference software implementations and by decoding the generated bitstreams. Instead of merely taking the bit rate and the distortion values, such as PSNR, from the encoder output, the simulation environment calculated the bit rate from the bitstream size given the frame rate of the test sequence. For the distortion metrics, a standalone software performs the averaged calculation by using the reconstructed sequence and the original input sequence. In the case of PSNR, the calculation uses the same formula specified by the JVET CTC. The employed BD-rate calculation requires four operation points with a negative BD-rate value representing bit-rate savings, whereas a positive value representing a bit-rate overhead. Note that the BD-rate value is the result of an area calculation for the area located between two fitted rate-distortion curves. Specifically, a bit-rate saving of $-x$% for A relative to B can be reformulated to a bit-rate overhead y% for B relative to A as follows:

$$y = \frac{x}{100 - x} \times 100$$

and vice versa

$$x = \frac{y}{100 + y} \times 100$$

Although the PSNR metric relies on MSE only and does not necessarily match the subjective perception, it is still the only reasonable choice for assessing objective coding performance. On the other hand, a proper subjective evaluation based on human mean opinion score ratings usually involves relatively high setup costs and would go beyond this paper's scope. Different metrics claiming to have a higher correlation to subjective perception than PSNR, such as multiscale structural similarity (MS-SSIM) [39] or video multimethod assessment fusion (VMAF) [40], often suffer from the disadvantage that their direct use in any practical encoder control is rather challenging and remains to be an open problem. Furthermore, it is remarkable that the BD-rate numbers of PSNR and VMAF are relatively close when considering the average over the whole test set, such as in [21], where the reported difference in the BD-rate numbers accounted for 2% only. Finally, it is worth mentioning that there is a recently published study [41] that unveiled an undesired loophole for tuning the VMAF metric by contrast or color enhancements of a video signal, while the corresponding SSIM scores keep largely unaffected.

Note that the verification tests for HEVC, as reported in [42], showed subjective bit-rate savings of more than 50%, meaning that the subjective gain for HEVC relative to H.264/AVC was even more significant than the corresponding objective result of 40–45% bit-rate savings using PSNR. For the sake of completeness, we include three additional full-reference video quality assessment metrics besides PSNR: XPSNR [43], VMAF, and MS-SSIM. According to the findings in [43], the XPSNR metric has the same or higher correlation (depending on the correlation measure) than VMAF and MS-SSIM. A further remarkable property of XPSNR due to its ease of per-block computability is the possibility to perform an encoder optimization by using it directly in the rate-quality optimization process.

Note, however, that for our experiments all tested encoders were configured to perform an MSE-based optimization rather than a perceptual tuning, which, as already mentioned above, was considered to be out of scope of our present investigation.

A) Experimental results

Table 3 lists luma BD-rate results of AV1 relative to HM in detail for each test sequence, averaged for each class, and averaged over the whole test set. Here, AV1 relative to HM means that the AV1 operation points form the test data, and the HM operation points form the anchor data for the BD-rate calculation. Table 4 provides the same kind of data for VTM, i.e. the BD-rate values are also relative to HM. In both cases, the HM encoder generates bitstreams with inferior rate-distortion operation points, i.e. there is either a bit-rate overhead for the same PSNR quality or the PSNR quality is lower for the same bit rate. On average, i.e. without taking the class D sequences into the overall calculation according to the JVET CTC, the bit-rate saving of AV1 is about 10% relative to HM, while the averaged bit-rate saving of VTM relative to HM is about 36%. Compared to the previous evaluation in [6], the averaged BD-rate result we obtained for AV1 indicates an improved performance, with a significant amount of improvements contributed by the

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press

Table 3. Bit-rate savings of AV1 version **85a9314** relative to HM 16.21

Sequence	Class	PSNR(%)	XPSNR(%)	VMAF(%)	MS-SSIM(%)	EncT(%)	DecT(%)
Tango2	*A1*	−10.80	−10.39	−13.56	−5.58	207	60
FoodMarket4	*A1*	−6.07	−6.40	−9.63	−1.73	154	58
CampfireParty2	*A1*	−25.75	−25.42	−35.11	−24.47	450	59
Overall A1		**−14.20**	**−14.07**	**−19.43**	**−10.59**	**243**	**59**
CatRobot1	*A2*	−11.01	−10.61	−20.70	−5.61	334	63
DaylightRoad2	*A2*	−11.51	−11.97	−24.59	−8.24	364	61
ParkRunning3	*A2*	−17.35	−13.64	−28.46	−11.86	332	68
Overall A2		**−13.29**	**−12.07**	**−24.58**	**−8.57**	**343**	**64**
MarketPlace	*B*	−9.98	−9.28	−14.65	−2.85	325	67
RitualDance	*B*	−6.71	−6.09	−8.03	−3.40	207	70
Cactus	*B*	−5.62	−3.11	−10.66	3.39	516	70
BasketballDrive	*B*	−11.34	−8.86	−15.30	−1.61	302	68
BQTerrace	*B*	−7.31	−5.44	−12.24	12.45	543	67
Overall B		**−8.19**	**−6.56**	**−12.17**	**1.60**	**356**	**69**
BasketballDrill	*C*	−11.12	−12.58	−11.08	−11.93	268	66
BQMall	*C*	−0.63	0.82	−2.97	5.79	281	66
PartyScene	*C*	−8.31	−7.40	−13.74	−3.67	452	71
RaceHorsesC	*C*	−1.55	−0.73	−2.91	8.83	298	70
Overall C		−5.36	−4.97	−7.68	−0.25	**317**	**68**
BasketballPass	*D*	−7.04	−6.51	−8.16	−5.08	335	71
BQSquare	*D*	−4.00	−1.87	−25.11	10.54	434	66
BlowingBubbles	*D*	3.01	4.79	−1.77	9.80	507	69
RaceHorses	*D*	4.75	6.38	1.23	16.04	353	73
Overall D		**−0.82**	**0.70**	**−8.45**	**7.85**	**402**	**70**
Overall		**−9.66**	**−8.74**	**−14.91**	**−3.37**	**317**	**66**

Table 4. Bit-rate savings of VTM 8.0 relative to HM 16.21

Sequence	Class	PSNR(%)	XPSNR(%)	VMAF(%)	MS-SSIM(%)	EncT(%)	DecT(%)
Tango2	*A1*	−38.63	−38.98	−42.34	−38.82	740	176
FoodMarket4	*A1*	−38.34	−38.31	−42.12	−39.37	624	172
CampfireParty2	*A1*	−40.75	−42.51	−42.86	−43.25	1338	183
Overall A1		**−39.24**	**−39.94**	**−42.44**	**−40.48**	**852**	**177**
CatRobot1	*A2*	−46.02	−45.67	−52.82	−43.77	816	178
DaylightRoad2	*A2*	−44.86	−46.02	−53.04	−44.92	884	187
ParkRunning3	*A2*	−40.93	−42.74	−45.06	−45.48	1208	213
Overall A2		**−43.94**	**−44.81**	**−50.30**	**−44.73**	**955**	**192**
MarketPlace	*B*	−35.53	−34.99	−42.28	−37.40	870	178
RitualDance	*B*	−31.39	−32.70	−32.00	−33.19	826	166
Cactus	*B*	−38.64	−38.29	−41.75	−36.28	940	175
BasketballDrive	*B*	−36.46	−37.37	−37.93	−35.45	1071	177
BQTerrace	*B*	−35.02	−36.24	−47.57	−32.93	802	166
Overall B		**−35.41**	**−35.92**	**−40.31**	**−35.05**	**897**	**172**
BasketballDrill	*C*	−32.61	−33.56	−29.26	−32.13	1177	184
BQMall	*C*	−30.54	−31.16	−33.89	−30.91	907	188
PartyScene	*C*	−29.12	−29.43	−32.42	−27.45	1153	199
RaceHorsesC	*C*	−27.74	−28.98	−29.27	−28.89	1383	194
Overall C		**−30.00**	**−30.78**	**−31.21**	**−29.84**	**1142**	**191**
BasketballPass	*D*	−26.93	−27.89	−27.70	−28.00	1326	196
BQSquare	*D*	−34.32	−32.79	−41.80	−26.05	977	194
BlowingBubbles	*D*	−25.45	−26.15	−28.23	−25.47	1256	196
RaceHorses	*D*	−25.16	−25.69	−25.60	−24.41	1479	201
Overall D		**−27.97**	**−28.18**	**−30.83**	**−26.23**	**1245**	**197**
Overall		**−36.44**	**−37.13**	**−40.31**	**−36.68**	**959**	**182**

activation of the enable-fwd-keyframe configuration. On the other hand, the averaged VTM bit-rate saving is about 4% higher than that obtained for the JEM software relative to HM. The bit-rate savings are even higher than the averaged value over the whole test set when only taking into account the high-resolution content of the test set for both AV1 and VTM. Class A2 shows less bit-rate savings than class A1 for AV1, whereas it is the other way round for VTM. That behavior is due to the sequence **CampfireParty2** that is an outlier for the AV1 case with a bit-rate saving of about 26%. This BD-rate number is significantly larger than those for the other two sequences within the same class

Table 5. Bit-rate savings of VTM 8.0 relative to AV1

Sequence	Class	PSNR(%)	XPSNR(%)	VMAF(%)	MS-SSIM(%)	EncT(%)	DecT(%)
Tango2	*A1*	−30.16	−30.77	−30.32	−34.23	357	292
FoodMarket4	*A1*	−34.34	−33.97	−36.39	−38.38	405	298
CampfireParty2	*A1*	−22.82	−24.98	−12.90	−26.90	297	312
Overall A1		**−29.11**	**−29.91**	**−26.54**	**−32.90**	**350**	**301**
CatRobot1	*A2*	−38.81	−38.52	−37.64	−39.95	244	280
DaylightRoad2	*A2*	−36.57	−37.32	−31.77	−38.69	243	307
ParkRunning3	*A2*	−29.34	−34.63	−23.80	−40.20	364	310
Overall A2		**−34.91**	**−36.82**	**−31.07**	**−39.61**	**278**	**298**
MarketPlace	*B*	−27.71	−27.70	−30.62	−35.06	267	265
RitualDance	*B*	−26.39	−28.29	−25.77	−30.97	400	236
Cactus	*B*	−34.92	−36.18	−32.83	−38.35	182	249
BasketballDrive	*B*	−28.10	−31.08	−25.73	−34.45	355	260
BQTerrace	*B*	−27.65	−29.84	−33.49	−38.68	148	248
Overall B		**−28.95**	**−30.62**	**−29.69**	**−35.50**	**252**	**251**
BasketballDrill	*C*	−23.97	−23.79	−20.30	−22.80	440	278
BQMall	*C*	−29.86	−31.50	−30.95	−34.30	323	287
PartyScene	*C*	−22.68	−23.55	−20.77	−24.28	255	278
RaceHorsesC	*C*	−26.47	−28.35	−26.79	−34.13	465	276
Overall C		**−25.75**	**−26.80**	**−24.70**	**−28.88**	**360**	**280**
BasketballPass	*D*	−21.24	−22.76	−20.46	−23.89	395	275
BQSquare	*D*	−31.49	−31.41	−21.03	−33.14	225	293
BlowingBubbles	*D*	−27.41	−29.29	−25.59	−31.61	248	282
RaceHorses	*D*	−28.77	−30.39	−26.14	−35.76	419	277
Overall D		**−27.23**	**−28.46**	**−23.30**	**−31.10**	**310**	**281**
Overall		**−29.32**	**−30.70**	**−28.00**	**−34.04**	**302**	**277**

with bit-rate savings of about 11 and 6%. The highest bit-rate savings for VTM are achieved for the two sequences **CatRobot1** and **DaylightRoad2** of class A2. When excluding the sequence **CampfireParty2**, one discovers a similar behavior for AV1 with the highest bit-rate savings for the class A2 sequences. The bit-rate savings for lower spatial resolutions are smaller for both AV1 and VTM relative to HM. Actually, for AV1 versus HM, the averaged bit-rate saving in class D turns out to be lower than 1%. The averaged bit-rate saving for VTM relative to HM in class D is still around 28%, while the averaged bit-rate savings for all HD and UHD sequences are above 35%.

When using the XPSNR metric, the BD-rate values are close to that of the PSNR-based BD-rate values in all cases. However, the same does not hold for VMAF, where both AV1 and VTM show higher VMAF-based BD-rate values than the PSNR-based BD-rate values, with a significant shift for AV1. On the other hand, for MS-SSIM, the BD-rate values are significantly lower than the PSNR-based BD-rate values for AV1, whereas the MS-SSIM-based values are similar to the PSNR-based values in the case of VTM. This imbalance for XPSNR, VMAF, and MS-SSIM is significant for the sequences **ParkRunning3** and **BQTerrace**, where the latter shows a bit-rate saving for AV1 relative to HM when using VMAF, whereas a bit-rate overhead for AV1 relative to HM has been measured when using MS-SSIM.

The averaged decoding run times of AV1 indicate (with the limitations mentioned above) that the AV1 decoder requires 34% less computational resources than the HM decoder, whereas the VTM decoder requires 82% more than the HM decoder. At the encoder side, the averaged encoding run-time increase relative to the HM encoder is 317% for AV1, i.e. the AV1 reference encoder requires more than thrice of the encoding time consumed by the HM reference encoder. In contrast to that, the averaged encoding run-time increase for VTM relative to HM is 959%, meaning that the VTM reference encoder runs almost ten times longer than the HM encoder to finish a test sequence.

Table 5 shows the BD-rate values for a direct comparison between VTM and AV1, with AV1 being the anchor. Given the results in Tables 3 and 4, the BD-rate results as given in Table 5 are as expected. **CatRobot1** shows the highest bit-rate savings with a BD-rate value equals to 39%, and the averaged overall BD-rate gain for VTM relative to AV1 is about 29%. One can observe the same outlier for the sequence **CampfireParty2** as in the comparison of AV1 relative to HM with the lowest bit-rate savings of about 22%.

Note that the BD-rate calculation uses the total bit rate and the PSNR of the luma component as input. Although the PSNR values of the two chroma components have not been taken into account by this kind of BD-rate calculation, it has been found that the improvement in chroma gain for VTM is even higher than that for luma [44], when compared to HM. Therefore, it was proposed to change the chroma QP offsets to a value of 1 for the VTM reference encoder, which may increase its averaged BD-rate gain of around 1% for luma in the random-access configuration.

Table 6 summarizes the results for the IRAP2 configuration similar to the presentation in Tables 3–5 for the IRAP1 configuration, but without BD-rate values for each sequence. In summary, the results show that the AV1 performance improves relative to the IRAP1 configuration, whereas the performance of VTM relative to HM remains

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press

Table 6. Summarized results for an IRAP period configuration equals to approximately 2 s

Class	PSNR(%)	XPSNR(%)	VMAF(%)	MS-SSIM(%)	EncT(%)	DecT(%)
The bit-rate savings of AV1 version ***85a9314*** *relative to HM 16.21*						
A1	**−19.92**	**−20.41**	**−23.67**	**−19.43**	**248**	**61**
A2	**−16.70**	**−15.38**	**−27.38**	**−11.76**	**359**	**66**
B	**−13.70**	**−12.87**	**−18.91**	**−6.36**	**351**	**68**
C	**−11.74**	**−12.28**	**−13.29**	**−10.40**	**320**	**70**
D	**−9.65**	**−9.36**	**−17.38**	**−6.04**	**407**	**68**
Overall	**−15.02**	**−14.72**	**−20.06**	**−11.13**	**321**	**67**
The bit-rate savings of VTM 8.0 relative to HM 16.21						
A1	**−39.34**	**−40.01**	**−42.55**	**−40.78**	**838**	**177**
A2	**−44.24**	**−45.11**	**−50.41**	**−45.25**	**949**	**194**
B	**−34.89**	**−35.43**	**−39.77**	**−34.51**	**890**	**172**
C	**−30.79**	**−31.64**	**−31.81**	**−30.75**	**1093**	**180**
D	**−29.15**	**−29.38**	**−31.89**	**−27.75**	**1205**	**179**
Overall	**−36.55**	**−37.27**	**−40.33**	**−36.91**	**941**	**179**
The bit-rate savings of VTM 8.0 relative to AV1						
A1	**−22.81**	**−22.95**	**−22.44**	**−23.94**	**338**	**289**
A2	**−32.97**	**−34.98**	**−29.86**	**−38.01**	**264**	**295**
B	**−23.63**	**−24.76**	**−24.47**	**−28.91**	**254**	**255**
C	**−21.16**	**−21.60**	**−20.98**	**−22.42**	**341**	**257**
D	**−21.34**	**−21.72**	**−15.78**	**−22.76**	**296**	**262**
Overall	**−24.67**	**−25.60**	**−24.21**	**−28.01**	**293**	**270**

similar to the IRAP1 configuration. Specifically, the BD-rate numbers for AV1 change by about 5% depending on the comparison, i.e. the bit-rate savings relative to HM increase from 10 to 15%. Similar trends as for the IRAP1 configuration can be observed in the IRAP2 case, such as the opposing trends of BD-rate numbers for some sequences when using VMAF and MS-SSIM.

B) Selected rate-distortion plots

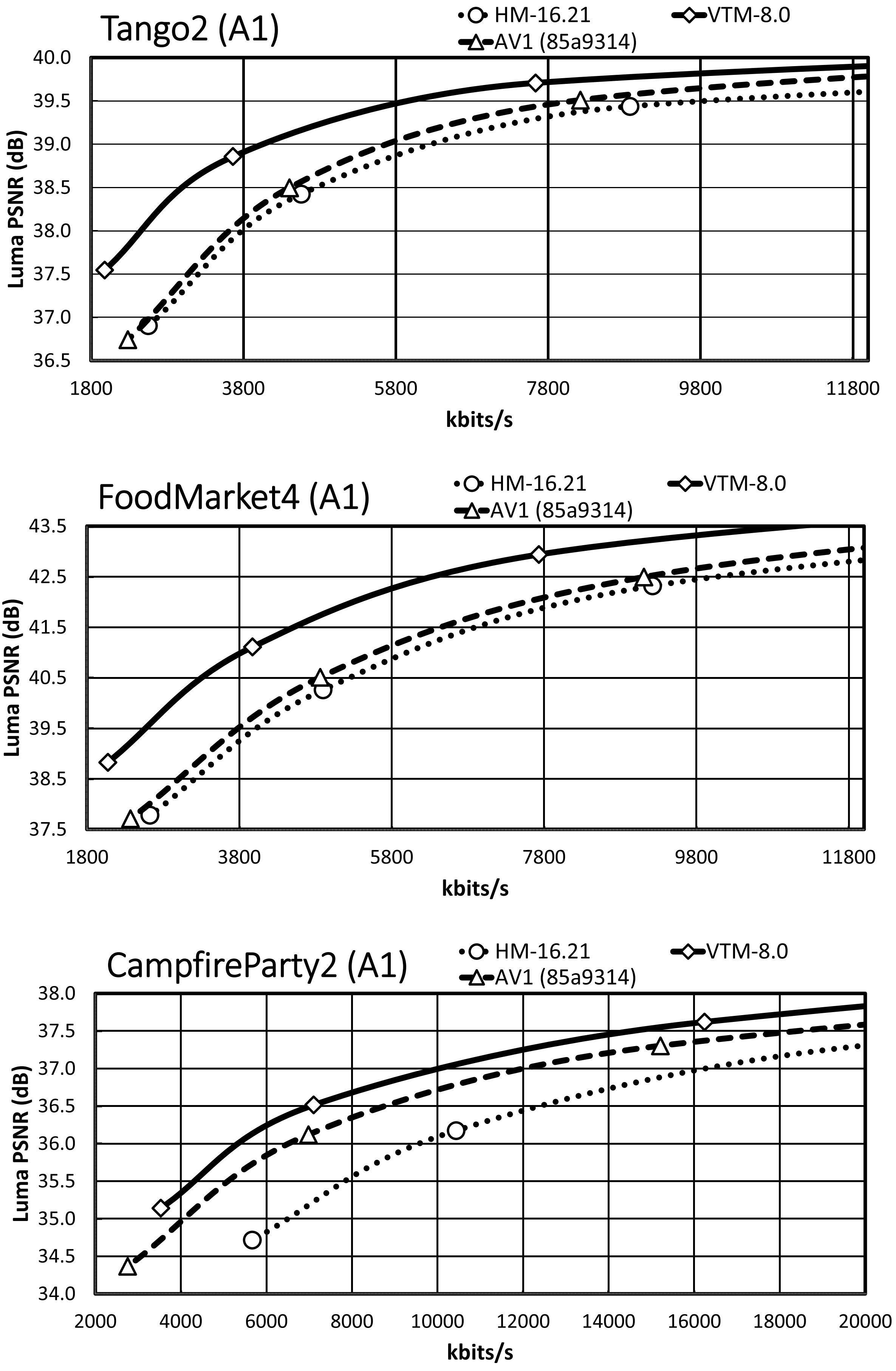
Two aspects are crucial when comparing rate-distortion (RD) curves: first, the bit-rate range, and second, the knee of the RD curves. For the first aspect, it is necessary to look at the bit-rate range that is typically used by the target application, which usually also covers the second aspect. Most of the content operating in the typical bit-rate range inherits the so-called knee of the RD curve, i.e. the left-hand side of the curve that has a steep positive slope.

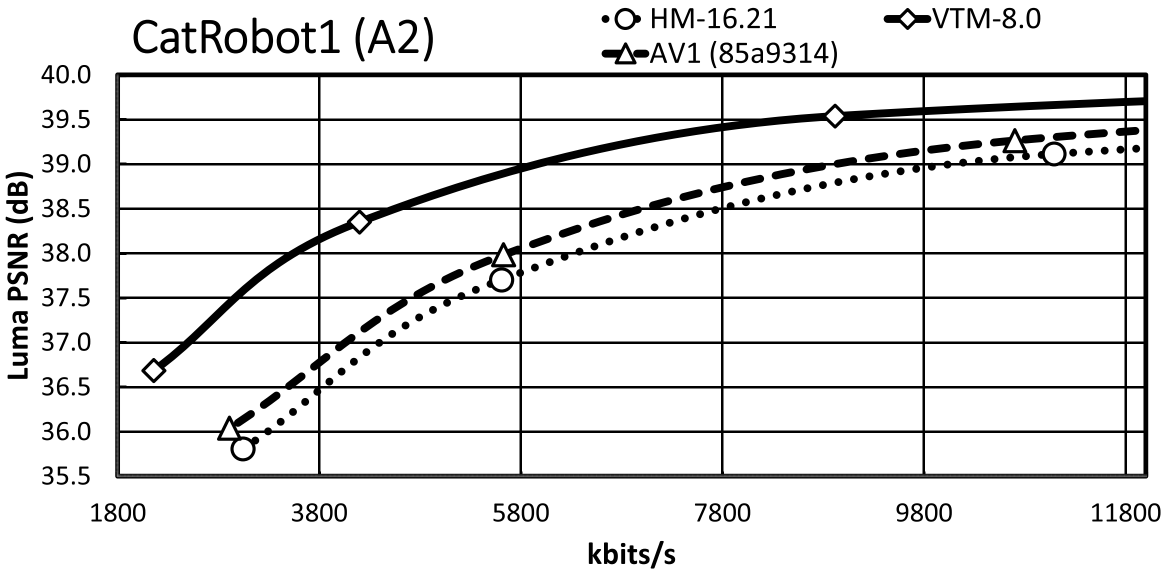
It is essential to consider the bit-rate range, especially the range that is relevant for the target applications. Since an important requirement for JVET was the focus on the development of a standard targeting higher spatial resolutions, we present the RD plots for the high-resolution content, i.e. classes A1, A2, and B. The maximum bit rates of the x-axis are around 7 MBit/s for the HD and around 14 Mbit/s for the UHD content, which are typical bit-rate ranges for random-access applications given the content type. An exception is **ParkRunning3**, where the x-axis of the plot is up to 100 Mbit/s due to the RD curve characteristic. Note that all RD curves show a steep slope at the lower bit-rate range while curves become flattened for higher bit rates.

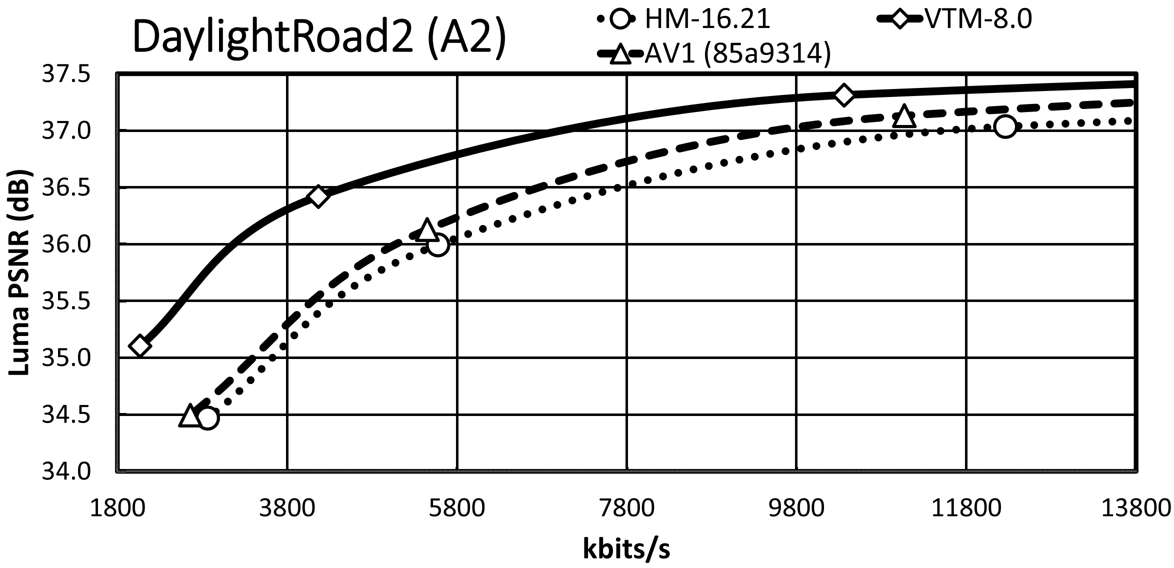
Almost all VTM bit-rate points are slightly lower than the corresponding points of AV1/HM, whereas they all have a significantly higher PSNR value than that of AV1/HM. A further observation is that the AV1 rate points are close to those of HM/VTM, reflecting that the selected **cq-level** values for AV1 are reasonable. In the general case, the VTM RD curves have a significant gap relative to the AV1/HM RD curves, with the AV1 RD curves usually having a smaller gap relative to those of HM.

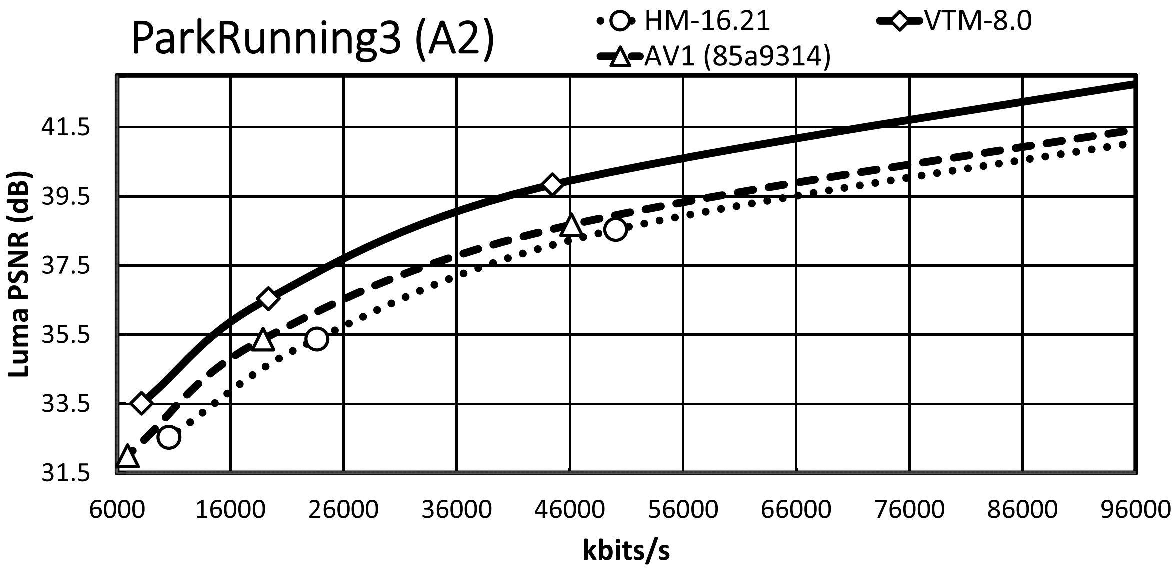
For the **Tango2** sequence, the PSNR difference between the VTM and AV1/HM RD curves is about 1 dB at around 2 MBit/s, whereas the difference decreases to around 0.5 dB at around 5.8 MBit/s. For the lowest rate point, the RD curves of HM and AV1 are very close, whereas the AV1 RD curve departs from the HM RD curve for higher bit rates. At very high bit rates, not fully visible in the plot, the AV1 RD curve indicates an operation point that is between that of HM and VTM. The gap in **FoodMarket4** between the VTM RD curve and those of AV1/HM is over 1 dB at 2 MBit/s, but the gap does not close significantly for the higher bit rates. That is in contrast to the **Tango2** sequence and also reflects the difference in BD-rate values between **Tango2** and **FoodMarket4**. **CampfireParty2** represents the best-performing video sequence for AV1 relative to HM/VTM and the respective plot clearly shows that property. The gap between the VTM and AV1 RD curves is around 0.3 dB at 4 Mbit/s only, and it remains stable up to the third bit-rate points. At 6 Mbit/s, the gap between the HM and VTM RD curves is about 1.5 dB, and the gap between the AV1 and HM RD curve is about 1 dB at the same bit-rate. For **CatRobot1**, the gap between the VTM and the AV1/HM RD curves is more than 1 dB at 4 Mbit/s and decreases for the higher bit-rate range. Specifically, the gap between the VTM and AV1/HM RD curves becomes around 0.5 dB in the range between 8 and 10 MBit/s. The plot shows that the RD curves for **CatRobot1** have a very similar trend as for the **FoodMarket4** sequence. For **DaylightRoad2**, the RD curves show similar behavior as for the **Tango2** sequence. The gap between the VTM and AV1/HM RD curves is higher than for the **Tango2** sequence with more than 1 dB

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press

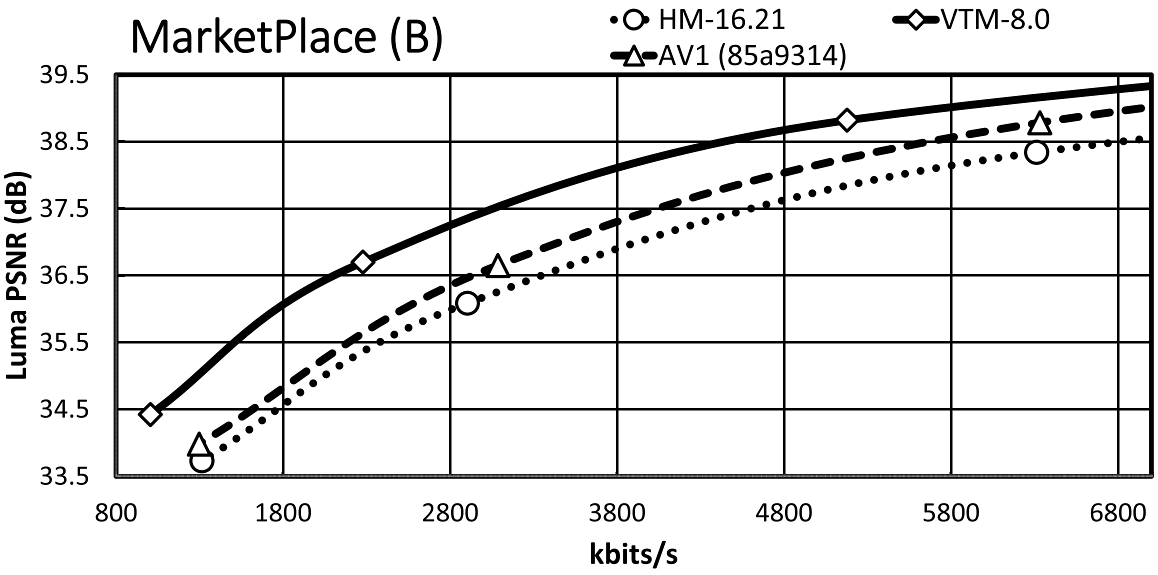

Tango2 (A1)
HM-16.21
VTM-8.0
AV1 (85a9314)
40.0
39.5
39.0
38.5
38.0
37.5
37.0
36.5
Luma PSNR (dB)
1800
3800
5800
7800
9800
11800
kbits/s
FoodMarket4 (A1)
HM-16.21
VTM-8.0
AV1 (85a9314)
43.5
42.5
41.5
40.5
39.5
38.5
37.5
Luma PSNR (dB)
1800
3800
5800
7800
9800
11800
kbits/s
CampfireParty2 (A1)
HM-16.21
VTM-8.0
AV1 (85a9314)
38.0
37.5
37.0
36.5
36.0
35.5
35.0
34.5
34.0
Luma PSNR (dB)
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
kbits/s

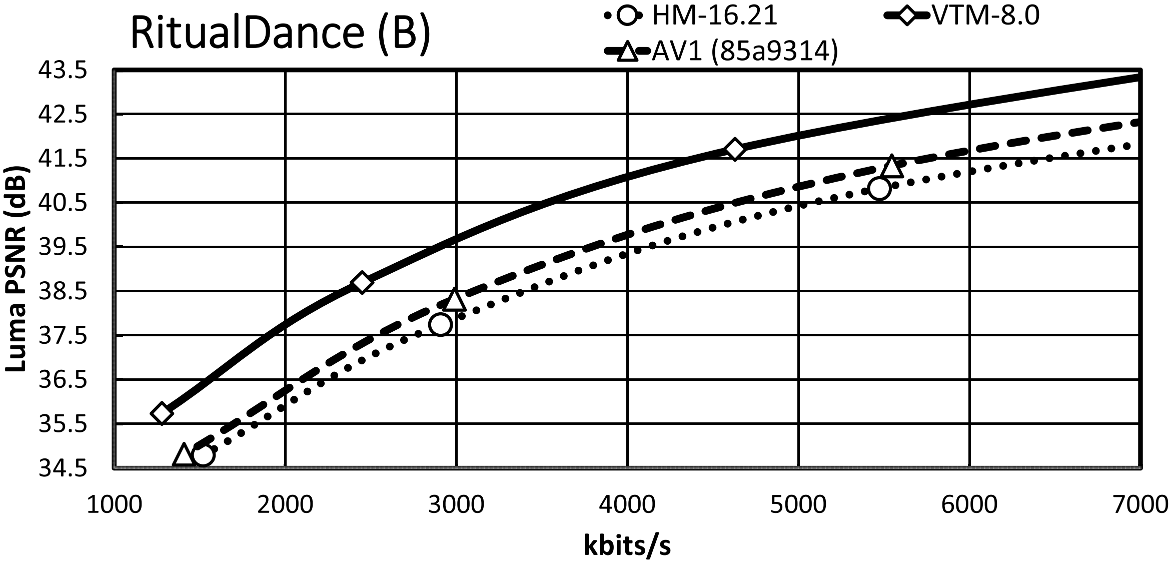


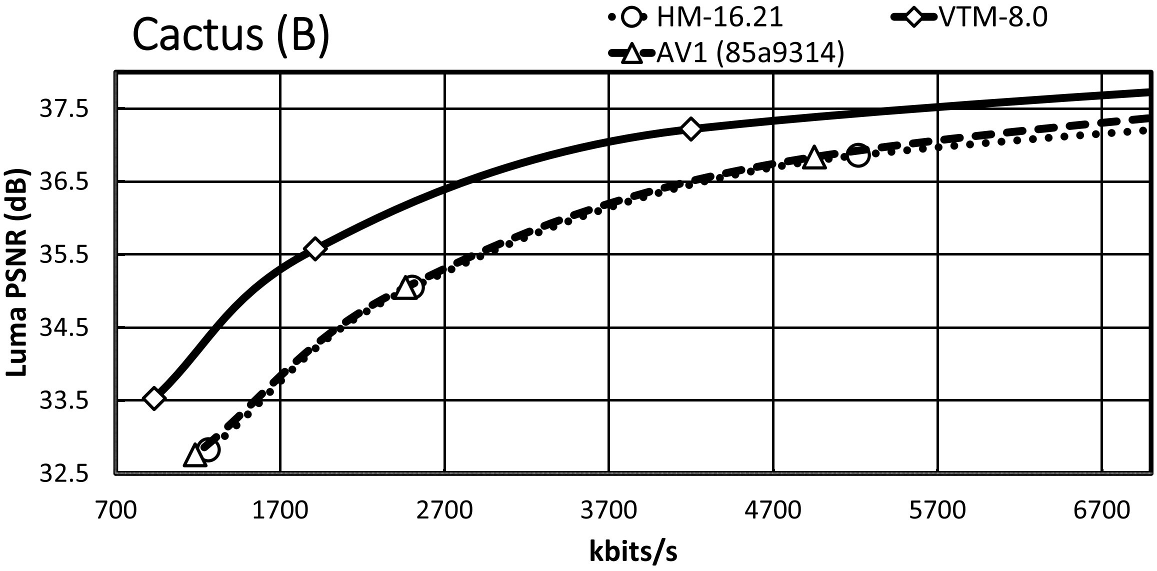


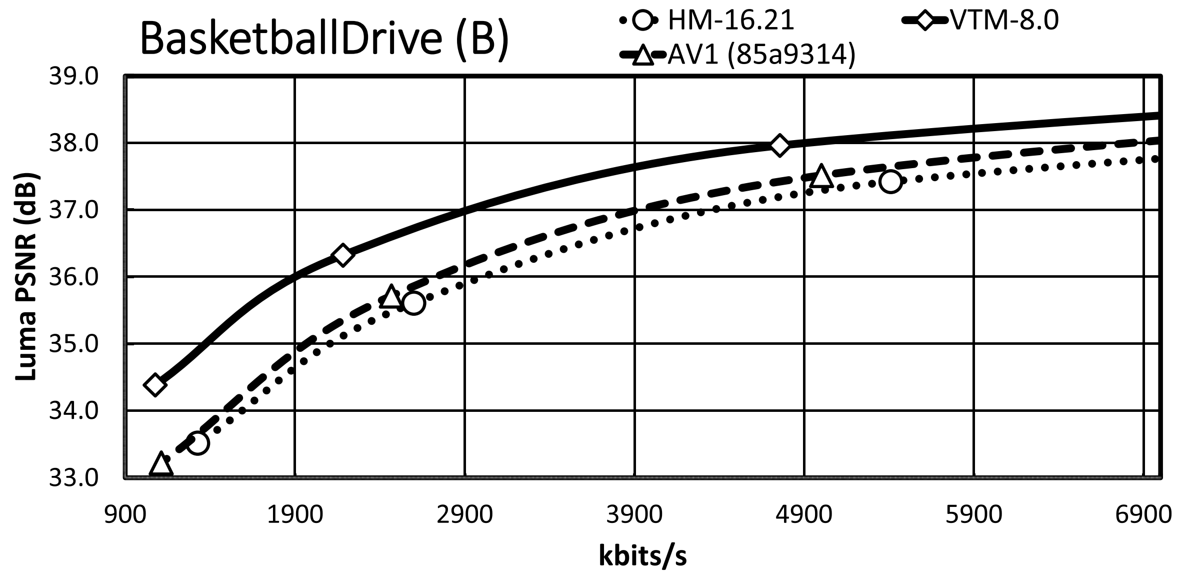


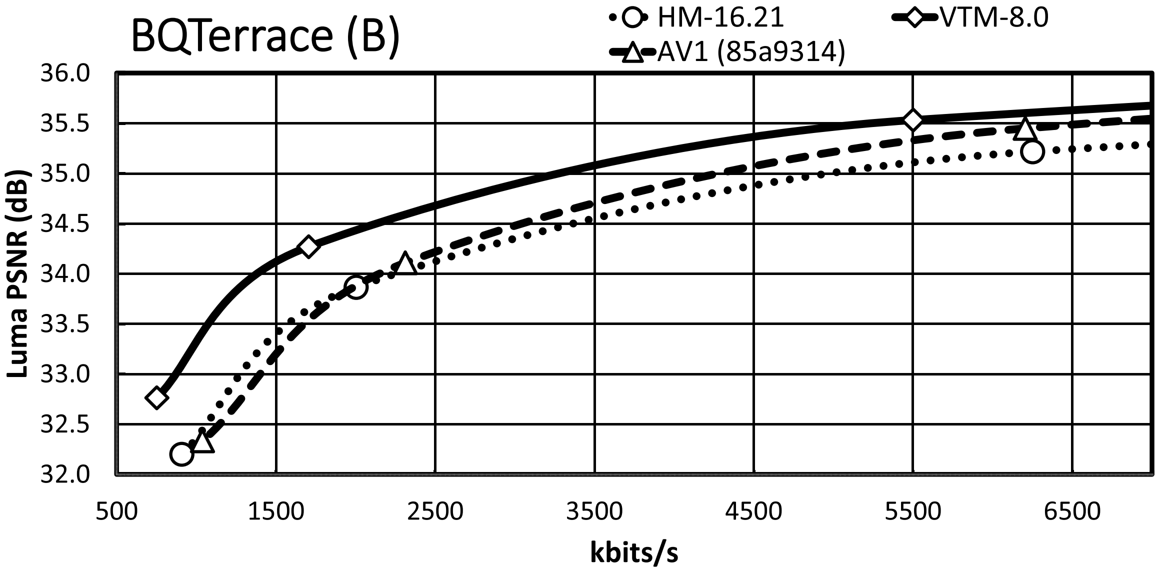
https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press


MarketPlace (B)
HM-16.21
VTM-8.0
AV1 (85a9314)
Luma PSNR (dB)
39.5
38.5
37.5
36.5
35.5
34.5
33.5
800
1800
2800
3800
4800
5800
6800
kbits/s


RitualDance (B)
HM-16.21
VTM-8.0
AV1 (85a9314)
Luma PSNR (dB)
43.5
42.5
41.5
40.5
39.5
38.5
37.5
36.5
35.5
34.5
1000
2000
3000
4000
5000
6000
7000
kbits/s


Cactus (B)
HM-16.21
VTM-8.0
AV1 (85a9314)
Luma PSNR (dB)
37.5
36.5
35.5
34.5
33.5
32.5
700
1700
2700
3700
4700
5700
6700
kbits/s





at around 2 Mbit/s. Compared to the **Tango2** sequence, the gap between the AV1 and the HM RD curves maintains up to around 5 MBit/s, whereas the same behavior occurs at around 8 to 9 Mbit/s for the **Tango2** sequence. **ParkRunning3** represents an exception since the two lowest operation points already generate very high bit rates for all three tested video coding schemes. The AV1 RD curve starts in-between those of VTM and HM but then becomes closer to the HM RD curve at higher bit rates. Also unusual is that the gap becomes more significant between the VTM the AV1/HM RD curves at higher bit rates. For **MarketPlace**, a 10-bit sequence of class B, the gap between the VTM and the AV1 RD curves becomes smaller while approaching the higher bit-rate range. Relative to HM, the gap to the VTM RD curve is about 1.5 dB at 1.8 Mbit/s and decreases down to about 1 dB at 5.8 MBit/s. Similar RD curve characteristics are observed for the **RitualDance** and the **BasketballDrive** video sequences. In contrast to that, the AV1 and the HM RD curves are remarkably similar for the **Cactus** video sequence, with the higher bit-rate range indicating a small divergence. Finally, the AV1 and the HM RD curves cross each other in the plot for the **BQTerrace** sequence, which is a rather unique case.

V. CONCLUSION

This paper presented the results of an objective performance evaluation of the three video coding schemes AV1, VVC, and HEVC for random-access applications. The employed controlled experimental environment was motivated by a typical application space that requires the random access property. The reference software encoder implementations of AV1, VVC (VTM), and HEVC (HM) were used to generate the rate-distortion operation points for the performance evaluation. Both AV1 and VVC showed a further step forward in terms of compression efficiency relative to what previous evaluations reported when using their corresponding work in progress. AV1 achieved averaged bit-rate savings of about 10–15% relative to HM, while VTM achieved 36–37% on average relative to the same anchor. A direct comparison between AV1 and VTM resulted in averaged bit-rate savings of about 25–29% for VTM, depending on the chosen IRAP period. For UHD content and an IRAP period of 1 s, which represents one of the primary foci of the VVC development, the bit-rate savings were found to be larger for both AV1 and VTM. Specifically, UHD-related bit-rate savings were measured as 14% for AV1 and 42% for VTM, both relative to HM, and 32% when comparing VTM to AV1. The AV1 and VVC reference encoders required averaged encoding run times relative to that of the HM encoder by a factor of more than three for AV1 and more than nine for VTM. On the other hand, the decoding run times indicated that the decoder complexity is manageable for both AV1 and VTM with an averaged decoding run time relative to that of the HM decoder of about 66% for AV1 and 182% for VTM, respectively. All the necessary parameters to reproduce the bitstreams for the experiments conducted in this paper can be found at https://bit.ly/3vU3VCK.

REFERENCES

[1] Rec. ITU-T H.265 and ISO/IEC 23008-2, High Efficiency Video Coding, Version 1, https://www.itu.int/rec/T-REC-H.265, April 2013.

[2] Rec. ITU-T H.266 and ISO/IEC 23090-3, Versatile Video Coding, Version 1, https://www.itu.int/rec/T-REC-H.266, August 2020.

[3] Joint Video Experts Team, Document Repository of the Joint Video Experts Team, http://phenix.int-evry.fr/jvet/, 2019.

[4] Alliance for Open Media. AV1 Bitstream & Decoding Process Specification, https://aomediacodec.github.io/av1-spec/av1-spec.pdf, 2019.

[5] L., Guo; J., De Cock; A., Aaron: Compression Performance Comparison of x264, x265, libvpx and aomenc for On-Demand Adaptive Streaming Applications, in *2018 Picture Coding Symposium (PCS)*, June 2018, 26–30.

[6] T., Nguyen; D., Marpe: Future Video Coding Technologies: A Performance Evaluation of AV1, JEM, VP9, and HM, in *2018 Picture Coding Symposium (PCS)*, June 2018, 31–35.

[7] T., Laude; Y.G., Adhisantoso; J., Voges; M., Munderloh; J., Ostermann: A Comparison of JEM and AV1 with HEVC: Coding Tools, Coding Efficiency and Complexity, in *2018 Picture Coding Symposium (PCS)*, June 2018, 36–40.

[8] J.L., Tanou; M., Blestel: Analysis of Emerging Video Codecs: Coding Tools, Compression Efficiency. *SMPTE Motion Imaging J.*, **128** (10) (2019), 14–24.

[9] T., Laude; Y., Adhisantoso; J., Voges; M., Munderloh; J., Ostermann: A Comprehensive Video Codec Comparison. *APSIPA Trans. Signal Inf. Process.*, **8** (2019), 1–16.

[10] Yue, Chen; *et al.*: An Overview of Coding Tools in AV1: the First Video Codec from the Alliance for Open Media. *APSIPA Trans. Signal Inf. Process.*, **9** (2020), e6.

[11] Alliance for Open Media, AV1 Codec Library, https://aomedia.googlesource.com/aom, 2019.

[12] F., Bossen; J., Boyce; X., Li; V., Seregin; K., Sühring: JVET Common Test Conditions and Software Reference Configurations for SDR Video, JVET-M1010, http://phenix.int-evry.fr/jvet/doc_end_user/current_document.php?id=5759, January 2019.

[13] D., Grois; D., Marpe; A., Mulayoff; B., Itzhaky; O., Hadar: Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders, *Picture Coding Symposium (PCS)*, December 2013, 394–397.

[14] D., Grois; D., Marpe; T., Nguyen; O., Hadar: Comparative Assessment of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders for Low-Delay Video Applications, *SPIE*, vol. 9217, September 2014.

[15] D., Grois; T., Nguyen; D., Marpe: Coding Efficiency Comparison of AV1/VP9, H.265/MPEG-HEVC, and H.264/MPEG-AVC Encoders, *Picture Coding Symposium (PCS)*, December 2016.

[16] D., Grois; T., Nguyen; D., Marpe: Performance Comparison of AV1, JEM, VP9, and HEVC Encoders, *SPIE*, vol. 10396, September 2017.

[17] M., Rerabek; T., Ebrahimi: Comparison of Compression Efficiency between HEVC/H.265 and VP9 based on Subjective Assessments, *SPIE*, vol. 9217, September 2014.

[18] P., Akyazi; T., Ebrahimi: Comparison of Compression Efficiency between HEVC/H.265, VP9 and AV1 based on Subjective Quality Assessments, in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2018, 1–6.

[19] A.V., Katsenou; F., Zhang; M., Afonso; D.R., Bull: A Subjective Comparison of AV1 and HEVC for Adaptive Video Streaming, in *2019 IEEE International Conference on Image Processing (ICIP)*, September 2019, 4145–4149.

[20] Y., Chen; *et al.*: An Overview of Core Coding Tools in the AV1 Video Codec, in *2018 Picture Coding Symposium (PCS)*, June 2018, 41–45.

[21] Fan, Zhang; Angeliki V., Katsenou; Mariana, Afonso; Goce, Dimitrov; David R., Bull: Comparing VVC, HEVC and AV1 using Objective and Subjective Assessments, arXiv: 2003.10282, 2020.

[22] R., Sjöberg; *et al.*: Overview of HEVC High-Level Syntax and Reference Picture Management. *IEEE Trans. Circuits. Syst. Video. Technol.*, **22** (12) (2012), 1858–1870.

[23] H., Schwarz; D., Marpe; T., Wiegand: Analysis of Hierarchical B Pictures and MCTF, in *2006 IEEE International Conference on Multimedia and Expo*, July 2006, 1929–1932.

[24] D., Mukherjee; *et al.*: The Latest Open-Source Video Codec VP9 - An Overview and Preliminary Results, in *2013 Picture Coding Symposium (PCS)*, December 2013, 390–393.

[25] R., Skupin; *et al.*: Open GOP Resolution Switching in HTTP Adaptive Streaming with VVC, *Picture Coding Symposium (PCS)*, June 2021.

[26] VVenC Software, Online: https://github.com/fraunhoferhhi/vvenc, 2021.

[27] T., Nguyen; A., Wieckowski; B., Bross; D., Marpe: Objective Evaluation of the Practical Video Encoders VVenC, x265, and aomenc AV1, *Picture Coding Symposium (PCS)*, June 2021.

[28] J., Ström; D., Saffar; P., Wennersten; R., Sjöberg: AHG13: Tool-off Tests on Open Images Dataset, JVET-O0551, http://phenix.int-evry.fr/jvet/doc_end_user/current_document.php?id=7164, July 2019.

[29] S., Nemoto; S., Iwamura; A., Ichigaya and K. Kazui: AHG13: Compression Performance Analysis for 8 K HLG Sequences, JVET-P0616, http://phenix.int-evry.fr/jvet/doc_end_user/current_document.php?id=8410, October 2019.

[30] Digital Video Broadcasting (DVB). *Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream*, 2014.

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press

[31] S., Lederer: Optimal Adaptive Streaming Formats MPEG-DASH & HLS Segment Length, Bitmovin Blog, https://bitmovin.com/mpeg-dash-hls-segment-length/, April 2015.

[32] DASH Industry Forum Interoperability Documents. Low-Latency Modes for DASH, https://dashif.org/docs/CR-Low-Latency-Live-r8.pdf, March 2021.

[33] X., Zhao; J., Chen; M., Karczewicz; L., Zhang; X., Li; W., Chien: Enhanced Multiple Transform for Video Coding, in *2016 Data Compression Conference (DCC)*, March 2016, 73–82.

[34] T., Nguyen; B., Bross; P., Keydel; H., Schwarz; D., Marpe; T., Wiegand: Extended Transform Skip Mode and Fast Multiple Transform Set Selection in VVC, in *2019 Picture Coding Symposium (PCS)*, November 2019, 1–5.

[35] K., Andersson; J., Enhorn; R., Sjöberg; J., Ström; L., Litwic: Addition of a GOP Hierarchy of 32 for Random Access Configuration for VTM, JVET-S0180, June 2020.

[36] G., Bjøntegaard: Calculation of Average PSNR Differences between RD Curves, VCEG-M33, April 2001.

[37] J., Xu; B., Zhou; C., Zhang; N., Ke; W., Jin; S., Hao: The Impact of Bitrate and GOP Pattern on the Video Quality of H.265/HEVC Compression Standard, in *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, September 2018, 1–5.

[38] Y., Sakamoto; R., Yokoyama; M., Takeuchi; Y., Matsuo; J., Katto: Improvement of H.265/HEVC Encoding for 8 K UHDTV by GOP Size and Prediction Mode Selection, in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, January 2019, 1–2.

[39] Z., Wang; E.P., Simoncelli; A.C., Bovik: Multiscale Structural Similarity for Image Quality Assessment, in *The Thirty-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, 2003, vol. 2, pp. 1398–1402.

[40] Z., Li; A., Aaron; I., Katsavounidis; A., Moorthy; M., Manohara: Toward A Practical Perceptual Video Quality Metric, The Netflix Tech Blog, https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652, June 2016.

[41] A., Zvezdakova; S., Zvezdakov; D., Kulikov; D., Vatolin: Hacking VMAF with Video Color and Contrast Distortion, *Proceedings of the 29th International Conference on Computer Graphics and Vision*, September 2019.

[42] T.K., Tan; *et al.*: Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance. *IEEE Trans. Circuits. Syst. Video. Technol.*, **26** (1) (2016), 76–90.

[43] C., Helmrich; S., Bosse; H., Schwarz; D., Marpe; T., Wiegand: A Study of the Extended Perceptually weighted Peak Signal-to-Noise Ratio (XPSNR) for Video Compression with Different Resolutions and Bit Depths. *ITU Journal: ICT Discoveries*, **3** (2020), 1–8.

[44] C., Helmrich; H., Schwarz; D., Marpe; T., Wiegand: On the Use of Chroma QP Offsets in the VVC Common Test Conditions, JVET-M0090, http://phenix.int-evry.fr/jvet/doc_end_user/current_document.php?id=4893, January 2019.

https://doi.org/10.1017/ATSIP.2021.10 Published online by Cambridge University Press