2018 Data Compression Conference

Efficient AV1 Video Coding Using A Multi-Layer Framework

Wei-Ting Lin[†], Zoe Liu*, Debargha Mukherjee*, Jingning Han*, Paul Wilkins*,
Yaowu Xu*, and Kenneth Rose[†]

[†]Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA, 93106
{weiting,rose}@ece.ucsb.edu

*WebM Codec Team, Google Inc.
1600 Amphitheatre Parkway, Mountain View, CA, 94043
{zoeliu,debargha,jingning,paulwilkins,yaowu}@google.com

Abstract

This paper proposes a multi-layer multi-reference prediction framework for effective video compression. Current AOM/AV1 baseline uses three reference frames for the inter prediction of each video frame. This paper first presents a new coding tool that extends the total number of reference frames in both forward and backward prediction directions. A multi-layer framework is then described, which suggests the encoder design and places different reference frames within one Golden Frame (GF) group to different layers. The multi-layer framework leverages the existing coding tools in the AV1 baseline, including the tool of "show_existing_frame" and the reference frame buffer update module of a wide flexibility. The use of extended `ALTREF_FRAME`s is proposed, and multiple `ALTREF_FRAME` candidates are selected and widely spaced within one GF group. `ALTREF_FRAME` is a constructed, no-show reference obtained through temporal filtering of a look-ahead frame. In the multi-layer structure, one reference frame may serve different roles for the encoding of different frames through the virtual index manipulation. The experimental results have been collected over several video test sets of various resolutions and characteristics both texture- and motion-wise, which demonstrate that the proposed approach achieves a consistent coding gain compared to the AV1 baseline. For instance, using PSNR as the distortion metric, an average bitrate saving of 5.57+% in BDRate is obtained for the CIF-level resolution set, some of which has a gain of up to 13+%, and 4.47% on average for the VGA-level resolution set, some of which up to 18+%.

1 Introduction

Google embarked on the open-source project entitled WebM [1] in 2010 to develop open-source, royalty unencumbered video codecs for the Web. WebM released two editions, first VP8 [2] and then VP9 [3], where VP9 achieves a coding efficiency similar to the latest video codec from MPEG entitled HEVC [4]. VP9 has delivered a significant improvement to YouTube in terms of quality of experience metrics over the primary format H.264/AVC. Google then joined the Alliance for Open Media (AOM) [5] effort for a Joint Development Foundation project formed with a few other industrial leaders, to define and develop media codecs, media formats, and related technologies [6][7], still under the open standard. In this paper, we focus on

2375-0359/18/$31.00 ©2018 IEEE
DOI 10.1109/DCC.2018.00045

the multiple reference inter prediction aspect for the to-be first edition of the AOM video codec, namely AV1.

The use of multiple reference frames facilitates a better inter prediction for videos with a variety of motion characteristics, such as the presence of occlusion and uncovered objects, lighting changes, fade-in and fade-out effects, static background, etc. The state-of-the-art techniques proposed the use of both short-term references and long-term references (LTR) [8] to adapt to the specific content and motion features presented in the coded frame. The Rate-Distortion (RD) performance optimization requests a trade-off between identifying the best reference for one coded frame and the overhead bits spent in signaling the multi-reference candidates [9–11]. Further, the encoder-side computational complexity should be considered [12]. Leveraging the multiple reference resources, one video frame may be forward predicted or backward predicted or both, referred to as bidirectionally predicted [13]. Special modes have been designed to effectively encode these bi-predictive frames, i.e. B frames, including the use of DIRECT mode [14, 15] and the design of hierarchical B frames [16].

In this paper, we first propose a new coding tool that extends the number of reference frames in AV1 from three to six to increase the flexibility and adaptability for the multi-reference prediction. Furthermore, we describe the encoder design through the exploit of extended `ALTREF_FRAME`s, and form a multi-layer framework facilitated by the two coding tools provided in AV1, namely the "show_existing_frame" and the virtual index manipulation. The experimental results validate the efficiency of the multi-layer structure with a consistent coding gain compared to the AV1 baseline over a variety of video test sets in various resolutions.

2 A New Coding Tool

2.1 AV1 Baseline Reference Frame Design

Current AOM/AV1 baseline uses three reference frames for the coding of each inter-coded frame: `LAST_FRAME`, `GOLDEN_FRAME`, and `ALTREF_FRAME`. The three references used by one specific coded frame are selected from a reference frame buffer that can store up to eight frames. In general, an AV1 encoder may select `LAST_FRAME` from a near past frame, and `GOLDEN_FRAME` from a distant past. `ALTREF_FRAME` is a no-show frame usually constructed from a distant future frame through temporal filtering. An AV1 encoder may apply different temporal filtering strength to construct an `ALTREF_FRAME`, adapting to various motion smoothness levels across frames. A so-called Golden Frame (GF) group can be established, and all the frames within one GF group may share the same `GOLDEN_FRAME` and the same `ALTREF_FRAME`. `LAST_FRAME` may be updated constantly. When the distant future frame that provides `ALTREF_FRAME` is actually being coded, it is referred to as an `OVERLAY` frame but treated as a regular inter frame. `OVERLAY` frames usually cost fairly small amounts of bits as `ALTREF_FRAME` may serve as an ideal prediction.

AV1 baseline designs two types of inter prediction: A block predicted from one reference frame with a corresponding motion vector is said to be in a *single prediction* mode, while a block predicted using two different reference frames and two corre-

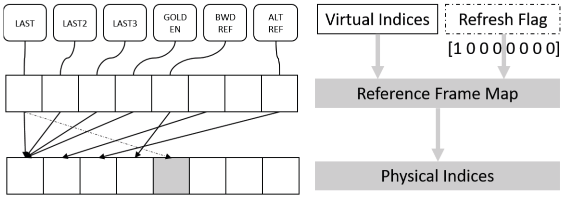


Figure 1: AV1 reference frame buffer update.

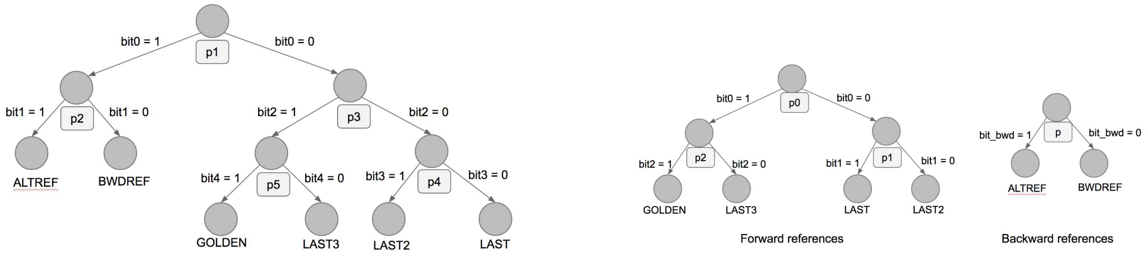
sponding motion vectors is said to be in a *compound* mode. *Compound* prediction always chooses the two predictions from two different directions, and generates a new predictor by simply averaging the two *single* predictors.

The reference frame buffer update in AV1 is realized through two syntaxes in the frame level: First is an eight-bit reference Refresh Flag, with each bit signaling whether the corresponding frame in the reference buffer needs to be refreshed or not by the newly coded frame; The second syntax is a mechanism referred to as "Virtual Index Mapping", as shown in Fig. 1. Each of the three references is labeled by a unique virtual index, and both the encoder and the decoder maintains a Reference Frame Map to associate a virtual index with the corresponding physical index that points to its location within the reference buffer. Both the Refresh Flag and the virtual indices are written into the bitstream. The advantage of using such mapping mechanism is to largely avoid memory copying whenever reference frames are being updated.

2.2 Extended Reference Frame - A New Coding Tool

To make full use of the reference frame buffer designed to store a maximum of eight frames, we propose a new coding tool that extends the number of reference frames for each coded frame from three to six. Specifically, we add `LAST2_FRAME`, `LAST3_FRAME`, and `BWDREF_FRAME`, where the former two references are usually selected from past for forward prediction and the later selected through look-ahead for backward prediction. Moreover, different from `ALTREF_FRAME`, `BWDREF_FRAME` leverages the existing coding tool provided by the AV1 baseline, namely the "show_existing_frame" feature, to encode a look-ahead frame without applying temporal filtering, thus no corresponding `OVERLAY` frame is needed. The use of `BWDREF_FRAME` is more applicable as a backward reference at a relatively shorter future distance. The extended reference frames allow a total of six candidates for the *single prediction* mode, and a total of 8 candidates for the `compound` mode as a combination of a forward predictor and a backward predictor are considered. Consequently each video frame is offered an extensively larger set of multi-reference prediction modes, thus leading to a great potential for the rate-distortion (RD) performance improvement.

To efficiently encode the extended number of references, context-based, bit-level binary tree structures are adopted, as shown in Fig. 2a and Fig. 2b. Depending on the availability and the final coding modes of the two neighboring blocks within the causal window - on the top and at the left, five contexts are designed for the coding of every bit in either *single reference* or *compound* prediction.



(a) *single reference* prediction (b) *compound* prediction

Figure 2: Binary tree structure design for context-based, bit-level entropy coding of the extended reference frames.

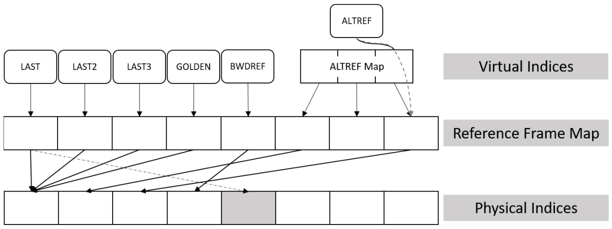
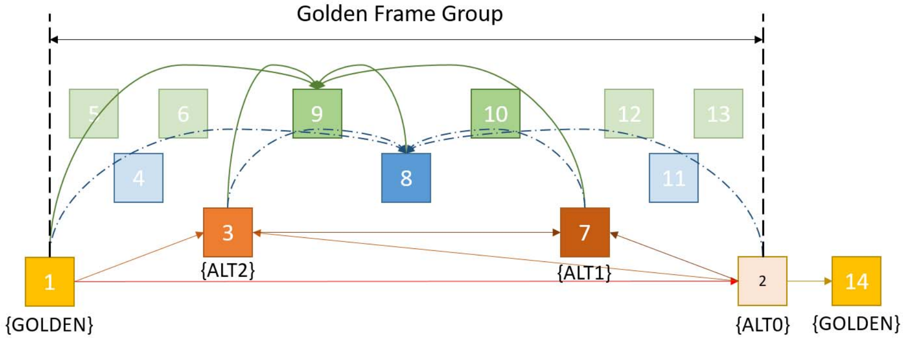


Figure 3: Encoder design using extended `ALTREF_FRAME`s.

Moreover, through the use of `BWDREF_FRAME`, a symmetric framework of multi-reference prediction is established for the *compound* mode: (1) A `BWDREF_FRAME` may be selected from a nearer future frame, paired with the nearer past `LAST_FRAME`; (2) A `BWDREF_FRAME` may be selected from a father future frame, paired with the father past `LAST2_FRAME`; and (3) `ALTREF_FRAME` may be selected from a distant future frame, paired with the `GOLDEN_FRAME` in the distant past. The use of extended reference frames that are spread out widely thus allows an adaptation to the dynamic motion characteristics within one video sequence.

3 Encoder Design - A Multi-Layer Framework

Aligned with the new coding tool introduced in Session 2, we address the encoder design in this session. An extended `ALTREF_FRAME` scheme is proposed, which adopts more than one `ALTREF_FRAME` candidates within one GF group. Still complied with the syntax that allows one `ALTREF_FRAME` at maximum for the coding of each frame, several frames may be buffered to act as `ALTREF_FRAME` serving for different frames. These candidates may be selected from various locations within the GF group and have various temporal filtering strengths applied. A multi-layer framework is then constructed with the aid of the extended `ALTREF_FRAME`s. Such encoder design is targeted to make full use of the eight-frame spots in the reference buffer and best leverage the new coding tool of extended reference frames.



(a) Symmetric multi-reference prediction in display order

(b) Symmetric multi-reference prediction in encoding order (SE for non-filtered ALTREF_FRAMEs and O for filtered ones

Figure 4: An example of the symmetric multi-layer multi-reference framework.

3.1 Extended ALTREF_FRAMEs

As illustrated in Fig. 1, the "Virtual Index Mapping" mechanism specifies how the reference frame buffer is updated. Both the encoder and the decoder use identical virtual indices associate with the same reference frame, and maintain a respective Reference Frame Map to track the corresponding physical location in the reference frame buffer. Within one GF group the encoder may buffer multiple frames to serve as the ALTREF_FRAME candidates, which is referred to as the extended ALTREF_FRAME scheme. To facilitate such an encoder design, an ALTREF Map is exploited only at the encoder side, as shown in Fig. 3. The ALTREF Map in essence is used to track the encoder's choice on the current selected ALTREF_FRAME. It stores the virtual indices of all the ALTREF_FRAME candidates, and the virtual index associated with the current selected ALTREF_FRAME is written to the bitstream.

3.2 Multi-Layer-Multi-Reference Framework

A multi-layer framework may be constructed using the extended ALTREF_FRAMEs, and an example is given in Figure 4a. This framework constructs a multi-layer structure where the top layer frames are coded through the prediction from the lower layers. As discussed in Sec. 2.1, one GF group starts with the coding of either a KEY_FRAME or an OVERLAY frame, serving as the GOLDEN_FRAME, followed by the coding of a distant future ALTREF_FRAME candidate, denoted as ALT0 in the figure. These two frames together form the bottom layer of the multi-layer structure. Given a GF group, we propose to use the new coding tools to construct multi-layer structure with the following steps.

Step 1. Insert k extended `ALTREF_FRAME`s and space them equally in the GF group. Since the extended `ALTREF_FRAME` along with the original `ALTREF_FRAME` lay out the bottom layer of the hierarchy structure, they will all serve as a distant future reference. We ensure there is enough space between each frame in the bottom layer by letting

$$k = \min\left(\left\lfloor \frac{\text{length(GF)}}{4} \right\rfloor - 1, 2\right).$$

Note that due to the size constraint of the reference buffer, the maximum number of `ALTREF_FRAME` allowed is two.

The extended `ALTREF_FRAME`'s divide the GF group into several subgroups. Compared to the original `ALTREF_FRAME`, the extended `ALTREF_FRAME`'s are always located closer to the current coded frame, hence, a predictor of higher quality may be obtained without the use of temporal filtering. When an `ALTREF_FRAME` is not filtered, the "show_exsisting_frame" flag is turned on and no `OVERLAY` frame is added. The coding of both `ALT2` and `ALT1` may choose `ALT0` to serve as their `ALTREF_FRAME`.

Step 2. Following coding order, the `BWDREF_FRAME` in each subgroup is constructed and formed the second layer from the top of the multi-layer structure. Through the virtual index manipulation, coding of the `BWDREF_FRAME` will use the near `ALTREF_FRAME` (e.g. `ALT2` or `ALT1`) to serve as its `BWDREF_FRAME` and the distant `ALTREF_FRAME` (`ALT0`) to serve as its `ALTREF_FRAME`.

Step 3. The remaining frames in the GF group form the top layer of the multi-layer structure. These frames use the near future reference frame as their `BWDREF_FRAME`, and the next future reference frame as their `ALTREF_FRAME`, if available. For instance, in Figure 4a, all the first frames in the top layer of each subgroup have their own `BWDREF_FRAME` and `ALTREF_FRAME` explicitly coded. For those second frames in the top layer of each subgroup, through virtual index manipulation, the two available `ALTREF_FRAME` candidates may serve as `BWDREF_FRAME` and `ALTREF_FRAME` respectively. For instance, for Frame 6, `ALT2` may serve as `BWDREF_FRAME` and `ALT0` may serve as `ALTREF_FRAME`. For the last frame in the last subgroup of the GF group, i.e. Frame 13 in the figure, `ALT0` is the only available backward reference, which may simply act as `ALTREF_FRAME` and no `BWDREF_FRAME` may be used.

Such coding structure is designed to minimize the decoding delay while to maintain a diversifying reference frame list to achieve a larger coding gain for the GF group. It is noted that the virtual index manipulation is only conducted at the encoder side, as the decoder simply identifies the virtual index associated with a specific reference frame from the bitstream. The encoder determines whether one buffered reference frame should act as `BWDREF_FRAME` or act as `ALTREF_FRAME`. We still maintain the size of reference frame buffer in the new coding tool the same as that specified in the AV1 baseline, considering the overall encoder complexity as well as the hardware design for the AV1 codec.

4 Experiment Results

In this section the experimental results of using extended reference frames are presented. The encoder adopts the proposed multi-layer framework and the results are compared against the AV1 baseline. We have tested the new approach over four different data sets, namely *low-res*, *derflr*, *medium-res*, and *hd-res*, where the first two sets contain video clips of the CIF/SIF-level resolution, the third set contains VGA-level resolution, and the last set contains HD-level resolution (e.g. 720p). The overall results are summarized in Table 1. The example results of individual video clips for the *low-res* and *medium-res* are given in Table 3. In all cases, we simply use a VBR bitrate-controlled test condition, where videos are run at a range of target bitrates with a standard rate-control mechanism to obtain RD curves. The BDRate [17] is computed using the global PSNR as the distortion metric.

Compared against AV1 baseline, the new coding tool of the extended reference frames and the corresponding multi-layer encoder design increase the computational complexity at both the encoder and the decoder, but have a nearly negligible impact on the decoder side, as described in Table 2.

Table 1: Coding gains of the multi-layer framework using extended reference frames compared against AV1 baseline in terms of BDRate reduction over datasets of various resolutions.

Data Set	low-res	derflr	medium-res	hd-res
Ext-Refs	-5.573%	-4.465%	-4.471%	-3.192%

Table 2: Computational complexity increment of the proposed approach compared against AV1 baseline.

	Encoder Side	Decoder Side
Ext-Refs	+74.16%	+2.12%

5 Conclusion and Future Work

In this paper, we first introduce a new coding tool that extends the total number of reference frames in the AV1 baseline. We then propose a multi-layer framework for the encoder design, which leverages the new coding tool through the use of extended `ALTREF_FRAME`s and the virtual index manipulation. The multi-layer, multi-reference prediction framework substantially increases the overall coding efficiency over an abundant set of video clips of various content and motion characteristics with a wide range of resolutions, providing evidence for the effectiveness of the proposed framework. The computational complexity at the decoder side is negligible. For the next step we will focus on the encoder-side complexity reduction. For instance, through the use of a much smaller set of block partition/prediction candidates for some of the references (e.g. `LAST2_FRAME` and `LAST3_FRAME`) complexity may be reduced at a sacrifice of the coding gain. We will also investigate the more optimized encoder design specifically applied to the higher resolution videos so that the coding effectiveness on the higher resolution videos may be on par with that on the lower resolution scenarios. Also, it is possible for both the encoder and the decoder to keep track of the update of all the reference frames, and check whether either `LAST2_FRAME` or `LAST3_FRAME` belong to the previous GF group. As the current GF group always start with an updated `GOLDEN_FRAME` it is possible to remove the use of `LAST2_FRAME` or `LAST3_FRAME` if they are not in the current GF group, which may greatly help on the encoder speedup whereas incur negligible coding performance degradation.

Table 3: Coding gains of the multi-layer framework using extended reference frames compared against AV1 baseline in terms of BDRate reduction on the low and mid resolution datasets (50 video clips).

Video	Resolution	BDRate Saving (%)	Video	Resolution	BDRate Saving (%)
akiyo	CIF	-5.789	BQMall	832×480	-6.117
bowing	CIF	-3.885	BasketballDrillText	832×480	-3.937
bridge_close	CIF	-5.908	BasketballDrill	832×480	-2.970
bridge_far	CIF	-6.777	Flowervase	832×480	-4.109
bus	CIF	-4.528	Keiba	832×480	-1.274
city	CIF	-5.041	Mobisode2	832×480	-2.671
coastguard	CIF	-9.797	PartyScene	832×480	-5.837
container	CIF	-12.683	RaceHorses	832×480	-1.340
crew	CIF	-3.642	aspen	480p	-2.751
flower	CIF	-13.176	crowd_run	480p	-11.267
foreman	CIF	-4.433	old_town_cross	480p	-4.323
harbour	CIF	-8.018	red_kayak	480p	1.840
highway	CIF	-2.426	rush_field_cuts	480p	-9.318
husky	CIF	-4.256	sintel_trailer_2k	480p	-4.825
ice	CIF	-4.308	snow_mnt	480p	0.496
mobile	CIF	-12.347	speed_bag	480p	-7.850
motherdaughter	CIF	-4.794	station2	480p	-2.548
news	CIF	-3.214	tears_of_steel1	480p	-4.122
pamphlet	CIF	-1.446	tears_of_steel2	480p	-6.668
paris	CIF	-3.305	touchdown_pass	480p	-2.321
signirene	CIF	-5.419	west_wind_easy	480p	-1.235
silent	CIF	-3.380	controlled_burn	480p	-1.340
students	CIF	-6.415	crew	4CIF	-2.476
tempete	CIF	-9.465	harbour	4CIF	-8.387
waterfall	CIF	-7.412	ice	4CIF	-2.876

References

[1] "WebM," http://www.webmproject.org/.

[2] J. Bankoski, P. Wilkins, and Y. Xu, "Technical overview of VP8, an open source video codec for the web," in *Multimedia and Expo (ICME)*. IEEE, 2011, pp. 1–6.

[3] D. Mukherjee, J. Han, J. Bankoski, R. Bultje, A. Grange, J. Koleszar, P. Wilkins, and Y. Xu, "A technical overview of VP9-the latest open-source video codec," *SMPTE Motion Imaging Journal*, vol. 124, no. 1, pp. 44–54, 2015.

[4] 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," in *SPIE Optical Engineering + Applications*, vol. 9971. International Society for Optics and Photonics, 2016, p. 997116.

[5] "AOM - Alliance for Open Media," http://aomedia.org/.

[6] U. Joshi, D. Mukherjee, J. Han, Y. Chen, S. Parker, H. Su, A. Chiang, Y. Xu, Z. Liu, Y. Wang *et al.*, "Novel inter and intra prediction tools under consideration for the emerging av1 video codec," in *SPIE Optical Engineering + Applications*, vol. 10396. International Society for Optics and Photonics, 2017, p. 103960F.

[7] S. Parker, Y. Chen, J. Han, Z. Liu, D. Mukherjee, H. Su, Y. Wang, J. Bankoski, and S. Li, "On transform coding tools under development for vp10," in *SPIE Optical Engineering + Applications*, vol. 9971. International Society for Optics and Photonics, 2016, p. 997119.

[8] T. Wiegand, X. Zhang, and B. Girod, "Motion-compensating long-term memory prediction," *IEEE International Conference on Image Processing*, vol. 2, pp. 53–56, 1997.

[9] T.-Y. Kuo and H.-J. Lu, "Efficient reference frame selector for H.264," *IEEE Trans. on Circuits and System for Video Technology*, vol. 18, no. 3, pp. 400–405, 2008.

[10] V. Chellappa, P. C. Cosman, and G. M. Voelker, "Dual frame motion compensation with uneven quality assignment," *IEEE Trans. on Circuits and System for Video Technology*, vol. 18, no. 2, pp. 249–256, 2008.

[11] D. Liu, D. Zhao, X. Ji, and W. Gao, "Dual frame motion compensation with optimal long-term reference frame selection and bit allocation," *IEEE Trans. on Circuits and System for Video Technology*, vol. 20, no. 3, pp. 325–339, 2010.

[12] Y.-W. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. on Circuits and System for Video Technology*, vol. 16, no. 4, pp. 507–522, 2006.

[13] M. Flierl and B. Girod, "Generalized B pictures and the draft H.264/AVC video-compression standard," *IEEE Trans. on Circuits and System for Video Technology*, vol. 13, no. 7, pp. 587–597, 2003.

[14] A. M. Tourapis, F. Wu, and S. Li, "Direct mode coding for bipredictive slices in the H.264 standard," *IEEE Trans. on Circuits and System for Video Technology*, vol. 15, no. 1, pp. 119–126, 2005.

[15] Y.-N. Fang, Y. Lin, and H.-J. Hsieh, "Novel direct mode decision for H.264/AVC inter B frame video coding," *International Conference on Computing, Management and Telecommunications (ComManTel)*, pp. 198–202, 2013.

[16] M. Paul, W. Lin, C.-T. Lau, and B. S. Lee, "A long-term reference frame for hierarchical B-picture-based video coding," *IEEE Trans. on Circuits and System for Video Technology*, vol. 24, no. 10, pp. 1729–1742, 2014.

[17] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," *ITU-T Q.6/16 13th VCEG meeting VCEG-M33*, March 2001.