MID-DEPTH BASED BLOCK STRUCTURE DETERMINATION FOR AV1

Jiawen Gu, Jiangtao Wen

Department of Computer Science and Technology, Tsinghua University
Email: gujw15@mails.tsinghua.edu.cn, jtwen@tsinghua.edu.cn

ABSTRACT

AV1 is an emerging open-source and royalty-free video compression format as a successor to VP9.The increase in coding efficiency and complexity over VP9 is due to the time required to find the optimal partition structure among the more flexible encoding modes for the coding units (CUs) and prediction units (PUs). Due to differences in the frame structure, existing fast block structure determination algorithm cannot be directly applied to AV1. To tackle this problem, we proposed a novel mid-depth based fast block structure determination algorithm for AV1. It checks the partition from mid-depth to provide information for estimating the posterior probabilistic distribution of the partition decisions as well as fast pruning in the PU prediction. Experimental results show that the proposed method can save up to 29.06% time saving with only 0.95% BD-Rate increase.

***Index Terms*—** Mid-depth, fast determination, AV1.

1. INTRODUCTION

As the demand for royalty-free high efficiency video applications rises and diversifies, Google cofounded the Alliance for Open Media (AOMedia) [1] to work jointly towards a next-generation open video coding format called AV1. As a successor to VP9, AV1 offers a number of new coding tools to achieve around 30% reduction in average bitrate compared with the VP9 encoder with much higher encoding complexity. As a result, fast encoding is very challenging especially in some real time applications for AV1.

Compared with VP9 and standards such as HEVC and H.264, AV1 not only expands the partition-tree to a 10-way structure as shown in Fig. 1, but also increases the largest possible size (referred to as superblock in VP9/AV1 parlance) to 128 × 128. Higher flexibility in partitioning improves motion estimation (ME) performance, while at the same time introduces much higher complexity in determining the rate distortion optimal (RDO) partition methods.

As partition selection is one of the most time-consuming modules in all state-of-the-art video coding standards, many classification-based pruning algorithms, such as [2–9], have been proposed to expedite the process. The AV1 reference software has already incorporated several cost-efficient algorithms into PU pruning, including rate-distortion (RD) cost based PU pruning, which can save around 30% of the encoding time with negligible RD loss. However, these algorithms all use the standard encoding order, which may not be able to provide reliable features for classification as the encoding proceeds. To address this problem, we introduce a novel mid-depth encoding order in this paper. The encoder would check the middle depth first, as opposed to the top-down encoding order. The prediction information of sub-CUs is used to expedite ME, fast PU pruning and early-depth decision. Experiments with the latest AV1 reference software showed an average speedup ratio of 29.06% is achieved with about 0.95% loss in coding efficiency.

This work was supported by the Natural Science Foundation of China (Project Number 61521002).

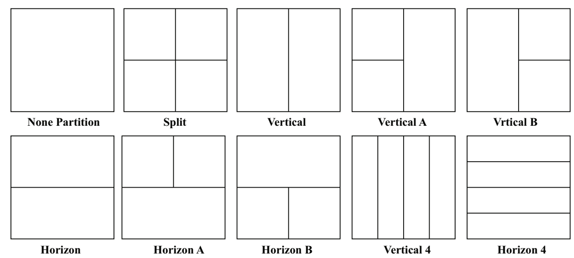


Fig. 1. Ten-way structure of partition-tree in AV1

2. RELATED WORK

Numerous algorithms have been dedicated to reducing encoding complexity of video coding standard such as HEVC or VP9. The most straight-forward solution is to skip some unlikely modes. For example, Lin *et al.* [10] collected rate-distortion costs of skip modes and merge modes to accelerate PU mode decisions for HEVC. Chen *et al.* [11] utilized the information of co-located reference coding units (CU) to decide the partition and depth of the current CU.

Existing pruning algorithms have several limitations: (i) The information is not sufficient for reliable PU/CU pruning. (ii) These algorithms adopted uni-directional top-down search, even though the probability of choosing the largest block size is small in many cases.

To solve the first limitation, Xiong *et al.* [5] found that motion divergence is approximately proportional to the probability for splitting the CU into the next depth. Therefore an estimated optical flow of down sampled frames is provided as additional information, whose calculation is too time-consuming for real time applications. Tang *et al.* [12] proposed an early-split order for HEVC CU-level encoding, where the encoder checks the split mode before the non-square PU partition modes to provide the encoding output of the sub-CUs for PU pruning. However, a similar algorithm has already been incorporated in AV1 as illustrated in Fig. 2.

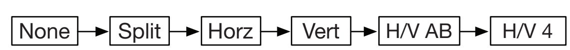


Fig. 2. The default processing order in AV1

To solve the second limitation, Gu *et al.* [13] first proposed bidirectional depth search for intra prediction, where the block with the size of 32 × 32 is first checked and the information will be used to decide whether to check the upper or lower depth subsequently. This encoding order provides more room to skip unnecessary calculations. However, the optimization is only for intra prediction, which is limited in most of the applications.

Inspired by the algorithms in [5] and [13] , we proposed a mid-depth based fast block structure determination scheme targeting AV1 coding structure. The novelties of this paper are as follows: (i) It first integrates the mid-depth encoding structure for inter prediction of AV1, where more reliable information can be utilized in this structure. (ii) It uses fineness instead of optical flow for partition pruning, enabling one-pass encoding as well as real time performance.

3. PROPOSED ALGORITHM

Previous sections introduce several limitations in existing algorithms. Inspired by the work in [13], we found that the mid-depth encoding structure can be utilized for inter-prediction in AV1. More specifically, if we start from the depth in the middle, and then utilized the information to determine whether to check the upper or the lower depth, the pruning process can be bidirectional. The proposed framework is illustrated in Fig. 3. Assuming that the encoding process starts from the middle depth d_m. It is obvious that as the value of d_m increases, more time savings can be achieved at the cost of more loss in video quality. To have a good tradeoff between time and quality, we choose $d_m = 2$ as the initial depth in this paper.

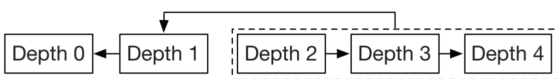


Fig. 3. The proposed Framework

To avoid ambiguity, here we also define the absolute depth of one prediction unit as follows:

$$D = (D_W + D_H)/2 = [\log_2(\bar{W}/W) + \log_2(\bar{H}/H)]/2, \quad (1)$$

where D_W and D_H are two components that describe the depth in width and height respectively. W and H are the width and the height of the current PU, while $\bar{W}$ and $\bar{H}$ are the width and the height of current CU.

3.1. A bayesian inference model for fast structure determination

In [5], it is proved that motion vector (MV) variance is an effective tool for PU partition decision and split determination, where the MVs are derived by optical flow of down sampled frames. MV variance for block X with the size of $n \times n$ is defined as

$$Var_{mv} = \frac{1}{n^4} \sum_{i \in X} ||\bar{m} - m_i||^2, \quad (2)$$

where m_i is the MV of point i in the block X and $\bar{m} = \sum m_i/n^2$.

In our proposed mid-depth encoding framework, we can roughly substitute the block's MVs for the MVs obtained by optical flow. However, the block-based MV is not accurate compared with the MV generated by optical flow and sometimes it cannot be obtained from the mid-depth encoding process due to the existence of intra mode. An alternative choice is to use the degree of partitioning to substitute the MV variance due to the fact that the block will not be split if the points in this block share the same MV. In other words, if a block has a higher partition degree, the probability of larger MV variance is higher.

To better understand the correlation between partition degree and MV variance, we conducted the following analysis. The MV variance for None partition is 0. Once a block is split, the variance becomes positive as split blocks will not share the same MV. More generally, we consider the situation that one sub-block Y in X is split into two part Y_1 and Y_2, where the area of both Y_1 and Y_2 are S and the MV of Y when not split is m_Y. Other partitions such as Horizon A/B, Split mode can be regarded as a further division of this situation. Given two independent two-dimensional normally distributed random variable $\varepsilon_1, \varepsilon_2 \sim N(0, 0, \sigma^2, \sigma^2, 0)$, new MV of each point in X is defined as follows:

$$m'_i = \begin{cases} m_i, & i \in X \cap \bar{Y}, \\ m_Y + \varepsilon_j, & i \in Y_j. \end{cases} \quad (3)$$

Now we have the new variance:

$$Var'_{mv} = \frac{1}{n^4} \sum_{i \in X} ||\bar{m}' - m'_i||^2, \quad (4)$$

where $\bar{m}' = \sum m'_i/n^2 = \bar{m} + \bar{\varepsilon}$, and $\bar{\varepsilon} = (\varepsilon_1 + \varepsilon_2)/n^2$. We now turn to the change of variance:

$$\Delta = Var'_{mv} - Var_{mv}. \quad (5)$$

We substitute (2) and (4) into (5) and we have:

$$\Delta = \frac{n^4 - 2S}{n^4} ||\bar{\varepsilon}||^2 - \frac{2(n^2-2) \cdot S}{n^4} \bar{\varepsilon} \cdot (\bar{m} - m_p)^T + C, \quad (6)$$

$$C = \frac{S}{n^4}(||\bar{\varepsilon} - \varepsilon_1||^2 + ||\bar{\varepsilon} - \varepsilon_2||^2). \quad (7)$$

Another formation of (6) is:

$$\Delta - C = \tau_1 \cdot \bar{\varepsilon} \cdot (\bar{\varepsilon} - \tau_2)^T, \quad (8)$$

where τ_1 is a constant value, and τ_2 is a constant MV.

It should be noticed that $\bar{\varepsilon}$ is also normally distributed and $\bar{\varepsilon} \sim N(0, 0, 2\sigma^2/n^4, 2\sigma^2/n^4, 0)$. Therefore it is obvious that $p(\Delta - C > 0) > 1/2$, and furthermore, $p(\Delta > 0) > 1/2$ since $C > 0$ is always true, or there is higher probability that MV variance increase when partition degree increases.

As it is demonstrated in [5] that higher MV variance represents higher possibility of Split mode, we further assume that higher partition degree is also an evidence of selecting Split mode.

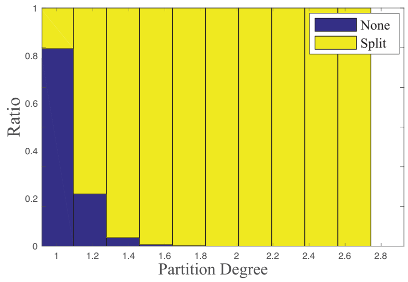
In order to quantify the partition degree of one block, we define the partition degree P_D of one block by the weighted average of the absolute depth of all the PUs in this block:

$$P_D = \sum_i w_i \cdot D_i = \sum_i w_i D_{i,W} + \sum_i w_i D_{i,H} \quad (9)$$

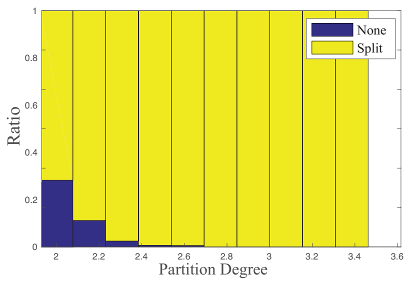
where the w_i is proportional to the area of the PU. For example, the partition degree of the sub-CU with the partition mode None, Vertical, Vertical A and Vertical 4, Split are 1, 1.5, 1.75, 2.0 and 2.0, respectively.

To demonstrate the relationship between partition degree and the probability of selecting Split mode, 100 frames of sequence *Johnny* were tested with quality levels set at 27. Fig. 4 shows the correlation between the ratio of Split and None and the average partition degree of four sub-CUs when the Block is 128×128 and 64×64. Observed that the ratio of choosing split mode increases as the partition degree becomes larger, which is consistent with our expectations.

The analysis proves that partition degree is an effective tool for early depth decision. Within the scope of video coding, making the



(a) Block 128×128



(b) Block 64×64

Fig. 4. The correlation between partition degree and split determination

wrong decision of checking the upper depth or not will result in performance loss. Thus, we define the $L_{c,s}$ as the loss of making a decision as CHECK (check the upper depth) while the correct decision should be SKIP (skip the check process). This type of error is referred as Error I. Likewise, we define $L_{s,c}$ as the loss of making decision SKIP while the correct decision should be CHECK (Error II). It should be noticed that $L_{c,s}$ represents the loss of time reduction due to the unnecessary RDO search, while $L_{s,c}$ represents the loss of RD performance. Noticed that no loss is introduced when making a right decision, we have $L_{i,i} = 0, i \in \{s, c\}$. Then we can obtain the Bayesian risk $R(i|P_D)$ for case i, $i \in \{c, s\}$,

$$R(s|P_D) = L_{s,s}p(s|P_D) + L_{s,c}p(s|P_D) = L_{s,c}p(s|P_D), \quad (10)$$

$$R(c|P_D) = L_{c,c}p(c|P_D) + L_{c,s}p(c|P_D) = L_{c,s}p(c|P_D). \quad (11)$$

The fundamental rule is to make the decision of SKIP, if $R(s|P_D) < \varepsilon R(c|P_D)$, and to choose CHECK, if $R(s|P_D) \geq \varepsilon R(c|P_D)$, where ε is used to balance the loss between time reduction and RD performance. Since we have already obtained the prior probability $p(P_D|s)$ and $p(P_D|c)$, by way of Bayes' theorem:

$$p(i|P_D) = \frac{p(P_D|i) \cdot p(i)}{p(P_D)}, i \in \{c, s\}, \quad (12)$$

where $p(P_D|i), i \in \{c, s\}$ denotes the class-conditional probability density function of non-split mode and split mode in essence. Thus, the decision rule can be rewritten as:

$$\begin{cases} \dfrac{p(P_D|s)}{p(P_D|c)} < \varepsilon \cdot \dfrac{L_{c,s}p(c)}{L_{s,c}p(s)}, & SKIP, \\ else, & CHECK. \end{cases} \quad (13)$$

Since it is difficult to measure the correlation between time reduction and RD performance, it is hard to determine a suitable ε for the tradeoff. Fortunately, the observation in Fig. 4 shows that $p(P_D|s)/p(P_D|c)$ decreases as P_D increases for both depths. Assuming that the right part in Equ. (13) is fixed, we only need to find a suitable threshold η for each depth: SKIP is chosen if P_D is larger than η, otherwise CHECK is selected. In practice, the resulting RD loss needs to be limited, while reducing the time cost as much as possible. Therefore, an error rate threshold α for Error I is set to determine the threshold of P_D. It is obvious that a smaller α reduces the $L_{c,s}$, improving the RD performance, but in turn increases the $L_{s,c}$ and reduces time savings, and vice versa. Therefore, α can be used to control the trade-off between acceleration and RD performance.

Due to the requirements of the prior knowledge of the relationship between the partition degree and the final decision, the first 5 frames of the input after a scene change are encoded without the fast determination algorithm in our implementation. Consider the importance of some special frames that are selected for long-term reference, such as golden-frames and altref-frames, we use these frames to update the model after the first 5 frames of a scene. This ensures that the majority of frames are encoded with acceleration, while keeping the statistical model up to date.

3.2. Improved Pruning method for PU partition

AV1 has already incorporated some cost-efficient pruning methods for PU partition, mainly by using RD cost to determine whether to check some PU modes. In this section, we propose a scheme for improving pruning method for PU partition based on partition degree. We categorize the non-square partition modes into three classes based on its partition degree. Specifically, Vertical and Horizon belong to category I, Vertical A,B and Horizon A,B belong to category II, and Vertical 4 and Horizon 4 are in category III.

For category I and II, the problem is to decide whether we should combine the two sub-CUs into one rectangular PU. Based on earlier analysis, we assume that there is a small chance that we combine the two sub-CUs into one rectangular PU if the partition degree of the two sub-CUs is large. Table 1 records the probability that no further split occurs in the two sub-CUs ($P_D < 2$) located in the rectangular PU, where the average probability exceeds 94.3% in the experiments. This observation allows us to use the partition degree to skip unlikely PU partitions.

Table 1. Ratio of no further split in sub-CU

Size	HorzA	HorzB	VertA	VertB
128x128	83.2%	90.8%	88.4%	91.2%
64x64	96.1%	95.4%	95.7%	96.6%
Size	**Horz top**	**Horz bottom**	**Vert left**	**Vert right**
128x128	96.1%	97.6%	94.1%	93.3%
64x64	97.8%	98.5%	96.5%	96.8%

If the optimal mode belongs to category III, a straightforward assumption is that the sub-CUs should use a compatible partition structure and direction (i.e., not partitioned in the perpendicular direction). Since the defined partition degree cannot reflect the direction of partition, we decomposed the partition degree into two directions, namely P_W and P_H, where $P_W = \sum_i w_i D_{i,W}$ and $P_H = \sum_i w_i D_{i,H}$. In practice, P_W and P_H represent the partition degree of the vertical and horizontal directions. Through experiments, we found that when $P_W > 1.25$, the probability of selecting Horizontal 4 has decreased to 12.7%. Likewise, when $P_H > 1.25$

and the probability of selecting Vertical 4 has decreased to 10.7%. Therefore, P_W and P_H can be used to determine whether to skip the partition mode of Vertical 4 and Horizontal 4. A fine-tuned parameter will improve the tradeoff between RD performance and time reduction. However, the above observations mainly used the geometric feature, therefore we set 1.25 as the threshold for the pruning.

In total, the overall workflow can be illustrated in Fig. 5. The posterior probabilistic distributions of the partition decisions for each depth are obtained from those frames that are checked from mid-depth. And then the thresholds for each depth can be obtained for further decision.

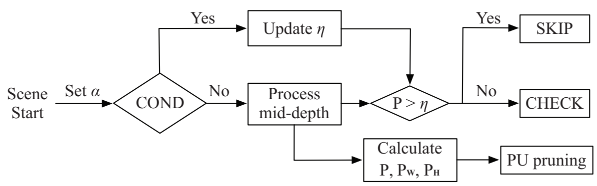


Fig. 5. The proposed workflow. COND represents whether this frame is the first 5 frames of the input after a scene change or the special frames.

4. EXPERIMENTAL RESULTS

4.1. Test condition

We tested the approach in this paper using AV1 codec v1.0.0 running on an Ubuntu 16.04 server with a 2.60 GHz Intel Xeon E5-2640 CPU and 64GB of RAM. The common configuration that will be used throughout this paper is given in Table 2. To demonstrate the effectiveness of our proposed algorithm, the mid-depth based fast block structure determination algorithm is integrated into AV1 encoder to encode eight test sequences from CTC [14] by using a constant of QP of 22, 27, 32, 37. RD performance and encoding time are measured by BD-rate [15, 16] and CPU time.

Table 2. Common configurations of the AV1 encoder

Param	Value	Param	Value	Param	Value
cpu-used	0	kf-min-dist	0	bit-depth	8
end-usage	q	kf-max-dist	9999	auto-altref	1
pass	1	kf-mode	1	drop-frame	0

4.2. Overall performance

Two sets of experiments were conducted, where the error rate threshold α for Error I was set at 0.1 and 0.2 to demonstrate its capability for controlling the trade-off between acceleration and RD performance. Table. 3 shows the results of the overall performance, where ΔT is the time reduction compared to the original AV1 encoder. It is observed that $\alpha = 0.1$ achieves 22.48% average time reduction with a negligible 0.83% BD-rate loss, while $\alpha = 0.2$ achieves a higher average time reduction of 29.02% but was also hit with a higher 0.95% BD-rate loss.

For comparison, the latest PU pruning algorithm in [12] is also evaluated using the same test settings. Although [12] is based on HEVC, it can be migrated to AV1 without difficulty.Table. 5 shows the performance of the algorithm in [12]. [12] reported a time saving of 48%, and a 0.8% BD-rate on average, a significant better tradeoff than our results for AV1. We believe that the difference resulted from the structural difference between AV1 and HEVC. And AV1 has already had its own optimization on PU pruning and partitioning. Comparing to our proposed algorithm where α set at 0.1, they both achieve about 22% average time reduction, but the BD-rate of our proposed algorithm is much slower. This proves the effectiveness of our proposed algorithm.

Table 3. Encoding results for $\alpha = 0.1$

Sequence	BD-Rate	BD-PSNR	ΔT
BQMall (480P)	0.61%	-0.017dB	22.52%
PartyScene (480P)	0.64%	-0.028dB	25.37%
BasketballPass (240P)	0.51%	-0.029dB	27.37%
BlowingBubbles (240P)	0.76%	-0.032dB	21.44%
BQSquare (240P)	0.93%	-0.033dB	25.07%
FourPeople (720P)	0.88%	-0.013dB	23.40%
Johnny (720P)	1.83%	-0.022dB	18.80%
Kimono (1080P)	0.46%	-0.009dB	15.91%
Average	0.83%	-0.023dB	22.48%

Table 4. Encoding results for $\alpha = 0.2$

Sequence	BD-Rate	BD-PSNR	ΔT
BQMall (480P)	0.71%	-0.020dB	29.89%
PartyScene (480P)	0.69%	-0.030dB	26.71%
BasketballPass (240P)	0.65%	-0.036dB	31.59%
BlowingBubbles (240P)	0.90%	-0.038dB	28.55%
BQSquare (240P)	0.98%	-0.034dB	31.11%
FourPeople (720P)	0.91%	-0.016dB	31.57%
Johnny (720P)	1.87%	-0.019dB	20.95%
Kimono (1080P)	0.92%	-0.018dB	32.12%
Average	0.95%	-0.027dB	29.06%

Table 5. Encoding results for algorithm in [12]

Sequence	BD-Rate	BD-PSNR	ΔT
BQMall (480P)	2.74%	-0.077dB	33.14%
PartyScene (480P)	3.51%	-0.156dB	23.91%
BasketballPass (240P)	1.72%	-0.097dB	13.77%
BlowingBubbles (240P)	2.71%	-0.114dB	13.16%
BQSquare (240P)	2.15%	-0.079dB	14.45%
FourPeople (720P)	2.40%	-0.048dB	22.42%
Johnny (720P)	2.40%	-0.048dB	22.42%
Kimono (1080P)	2.05%	-0.040dB	37.70%
Average	2.46%	-0.082dB	22.62%

5. CONCLUSION

In this paper, we consider two limitations in the current fast block determination algorithms. We first present the proposed mid-depth framework, then we analyze the available information, namely partition degree, that can be used for split decision. This information can be further used to accelerate the process of PU partition selection. Experimental results show that our proposed algorithm offers the capability to control the trade-off between RD performance and time reduction, achieving 22.48%-29.06% time reduction while keeping BD-rate loss at 0.83%-0.95%. Many aspects can be incorporated into the proposed framework, including RD cost, reference frame, MV, for further optimizations.

6. REFERENCES

[1] "Alliance for open media," https://aomedia.org/.

[2] Kiho Choi, Sang-Hyo Park, and Euee S Jang, "Coding tree pruning based cu early termination," *JCT-VC document, JCTVC-F092*, 2011.

[3] Yun Zhang, Sam Kwong, Xu Wang, Hui Yuan, Zhaoqing Pan, and Long Xu, "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 24, no. 7, pp. 2225–2238, 2015.

[4] Qiang Hu, Xiaoyun Zhang, Zhiru Shi, and Zhiyong Gao, "Neyman-pearson-based early mode decision for hevc encoding," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 379–391, 2016.

[5] Jian Xiong, Hongliang Li, Qingbo Wu, and Fanman Meng, "A fast hevc inter cu selection method based on pyramid motion divergence," *IEEE transactions on multimedia*, vol. 16, no. 2, pp. 559–564, 2014.

[6] Xiaolin Shen and Lu Yu, "Cu splitting early termination based on weighted svm," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 4, 2013.

[7] Liquan Shen, Zhaoyang Zhang, and Zhi Liu, "Effective cu size decision for hevc intracoding.," *IEEE Trans. Image Processing*, vol. 23, no. 10, pp. 4232–4241, 2014.

[8] Liquan Shen, Zhaoyang Zhang, and Zhi Liu, "Adaptive inter-mode decision for hevc jointly utilizing inter-level and spatiotemporal correlations.," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 24, no. 10, pp. 1709–1722, 2014.

[9] Jian Xiong, Hongliang Li, Fanman Meng, Qingbo Wu, and King Ngi Ngan, "Fast hevc inter cu decision based on latent sad estimation," *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2147–2159, 2015.

[10] Kao-Min Lin, Jie-Ru Lin, Mei-Juan Chen, Chia-Hung Yeh, and Cheng-An Lee, "Fast inter-prediction algorithm based on motion vector information for high efficiency video coding," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 99, 2018.

[11] Mei-Juan Chen, Wu Yu-De, Chia-Hung Yeh, Kao-Min Lin, and Shinfeng D Lin, "Efficient cu and pu decision based on motion information for inter-prediction of hevc," *IEEE Transactions on Industrial Informatics*, 2018.

[12] Minhao Tang, Xinyao Chen, Jiawen Gu, Yuxing Han, Jiangtao Wen, and Shiqiang Yang, "Accelerating hevc encoding using early-split," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 209–213, 2018.

[13] Jiawen Gu, Minhao Tang, Jiangtao Wen, and Yuxing Han, "Adaptive intra candidate selection with early depth decision for fast intra prediction in hevc," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 159–163, 2018.

[14] Frank Bossen et al., "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, 2013.

[15] Gisle Bjontegaard, "Calculation of average psnr differences between rd-curves," *VCEG-M33*, 2001.

[16] Gisle Bjontegaard, "Improvements of the bd-psnr model, vceg-ai11," in *ITU-T Q. 6/SG16, 34th VCEG Meeting, Berlin, Germany (July 2008)*, 2008.