Special Paper
방송공학회논문지 제27권 제7호, 2022년 12월 (JBE Vol. 27, No. 7, December 2022)
https://doi.org/10.5909/JBE.2022.27.7.985
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

Intra Coding Tools of Enhanced Compression beyond VVC Capability

Bumyoon Kim[a)] and Byeungwoo Jeon[a)‡]

Abstract

Even after the standardization of VVC completed in 2020, continuing efforts for developing even better video coding technology are still under way. In this context, the Joint Video Experts Group (JVET) is developing various video compression technologies under the name of enhanced compression beyond VVC capability which has reported BDBR gain of -6.75%, -14.05%, and -15.25% on its test model, ECM version 5.0, respectively in Y, Cb, and Cr channels by just having a few new or improved intra coding tools. The current activity has so far adopted 11 intra coding tools beyond VVC which can generate more sophisticated predictors, reduce signaling overhead, or combine various predictors. In this tutorial paper, we will review these techniques.

Keyword : ECM, Video compression, Intra coding, Enhanced compression beyond VVC capability

I. Introduction

As clearly manifested by recent explosive increase in video data traffic, it is needless to say that more powerful video coding techniques are still in a great demand for efficient video data transmission and storage. The Joint Video Experts Team (JVET) formed in 2015 by ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) completed the new video coding standard, Versatile Video Coding (VVC), in 2020 which has shown significantly enhanced coding performance compared to its predecessor, HEVC. Compared to HEVC, VVC has achieved Bjøntegaard delta bitrate (BDBR) gain of -25.06%, -25.37%, and -26.85%, respectively for Y, Cb, and Cr channels under the all intra (AI) configuration[1]. After completing standardization of VVC, JVET decided to continue improving video coding performance under the name of enhanced compression beyond VVC capability and is currently building up a common SW test platform named as Enhanced Compression Model (ECM)[2] to objectively test and evaluate coding performance of various coding tools. Many coding tools have been already evaluated on top of ECM to see how well they function and to decide whether or not to adopt them into ECM. Up to the 26th JVET meeting,

---

a) Department of Electrical and Computer Engineering, Sungkyunkwan University

‡ Corresponding Author : Byeungwoo Jeon
E-mail: bjeon@skku.edu
Tel: +82-31-290-7144
ORCID: https://orcid.org/0000-0002-5650-2881

※This research was supported the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2020R1A2C2007673).

· Manuscript October 4, 2022; Revised December 12, 2022; Accepted December, 13, 2022.

April 2022, the compression performance has been enhanced beyond VVC by BDBR gain of about -6.75%, -14.05%, and -15.25%, respectively in Y, Cb, and Cr channels under AI configuration while encoding and decoding time-complexity increased respectively to 409% and 267%[3]. This performance is based on the ECM version 5.0. In this paper, we review the 11 intra coding tools adopted so far in ECM along with their brief analysis.

Ⅱ. Intra Coding Tools beyond VVC

The newly adopted 11 intra coding tools include extensions of the already existing coding tools in VVC, those proposed to VVC but not adopted, and those newly proposed for ECM[4]. As the first step in the development of intra coding beyond VVC[5], the multi-model linear model (MMLM)[7], gradient position dependent intra prediction combination (PDPC)[9], secondary most probable mode (SMPM)[11], reference sample interpolation and smoothing[12], decoder side intra mode derivation (DIMD)[13]-[16], and intra template matching prediction (TMP)[17][18] were implemented on ECM at first. The template-based intra mode derivation (TIMD)[20]-[22], extended multiple reference line (MRL)[23], slope adjustment for cross-component linear model (CCLM)[24][25], DIMD chroma[26][27], and fusion of chroma intra prediction modes [26][27] have been later incorporated into ECM over several JVET meetings.

In this Section, the intra coding tools aforementioned will be reviewed with their key performance (i.e., BDBR gain) under AI configuration of the common test condition (CTC). Note that the corresponding performance stated in this paper is only that of the tool at the time when each tool was proposed, thus the ECM or VTM version of each tool test in this paper may not be identical. The performance comparison of all tools using the same ECM version with the same common test condition is still not available. CTC is a common experimental environment that must be followed in evaluating the performance of a proposed tool. Its purpose is to more objectively confirm the general performance of the proposed technique. Although CTC has been slightly changed since its first availability, however, there has been no change in the AI configuration except that class TGM (Text and Graphics with Motion) has been added as test sequences[30]-[35]. The CTC specifies several conditions as follows:

- Four quantization parameter values to use (that is, 22, 27, 32, and 37)
- Test sequences to use (see Table 1)
- The number of frames to be encoded and frame rate to use for each sequence

Table 1 shows the latest JVET common test sequences[35].

Table 1. JVET common test sequences[35]

Class	Resolution	Sequence Name	Frames to be encoded	Frame rate	Bit depth
A1	3840x2160	Tango2	294	60	10
		FoodMarket4	300	60	10
		Campfire	300	30	10
A2		CatRobot	300	60	10
		DaylightRoad2	300	60	10
		ParkRunning3	300	50	10
B	1920x1080	MarketPlace	600	60	10
		RitualDance	600	60	10
		Cactus	500	50	8
		BasketballDrive	500	50	8
		BQTerrace	600	60	8
C	832x480	RaceHorses	300	30	8
		BQMall	600	60	8
		PartyScene	500	50	8
		BasketballDrill	500	50	8
D	416x240	RaceHorses	300	30	8
		BQSquare	600	60	8
		BlowingBubbles	500	50	8
		BasketballPass	500	50	8
E	1280x720	FourPeople	600	60	8
		Johnny	600	60	8
		KristenAndSara	600	60	8
F	1920x1080	ArenaOfValor	600	60	8
	832x480	BasketballDrillText	500	50	8
	1280x720	SlideEditing	300	30	8
	1280x720	SlideShow	500	20	8
TGM	1920x1080	FlyingGraphics	300	60	8
		Desktop	600	60	8
		Console	600	60	8
		ChineseEditing	600	60	8

When encoding under the AI configuration, every 8th frame of the input video is encoded as I-frame regardless of frame rate, and every block in the I-frame is coded by using intra coding tools only. Before describing the intra coding tools, the flowcharts in Figure 1 are given first to help understand the overall flow of intra mode coding for enhanced compression beyond VVC capability. It can provide general idea about how the intra coding actually takes place. In case of the intra luma mode coding, as can be seen in Figure 1, each flag indicates whether the current block is coded with the corresponding intra coding tool or not. For MIP, PMPM, SMPM, and MPM remainder cases, additional mode index coding follows to indicate a specific mode to use in decoding. However, in the case of DIMD, intra TMP, and TIMD, additional index coding is not necessary because their intra prediction mode information can be derived at decoder.

In the case of intra chroma mode coding, unless the prediction mode of the current block is the linear model (LM) mode (i.e., planar, DC, horizontal, vertical, and DM), chroma fusion is possible. When the intra prediction mode of the current block is the LM mode, a flag indicating whether slope adjustment is applied or not is coded only when both reference sides are used.

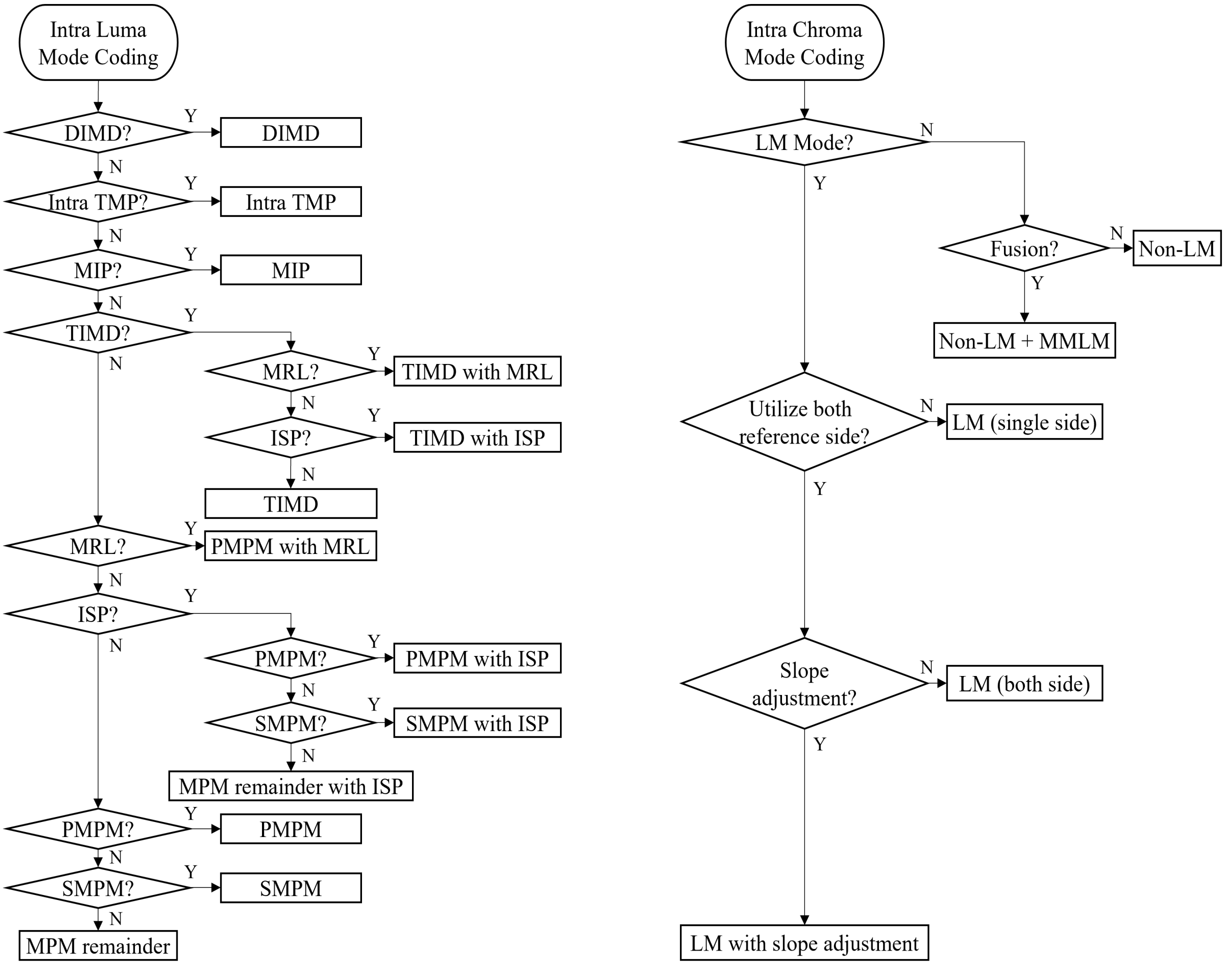


Fig. 1. Flowchart for intra coding of luma and chroma channels

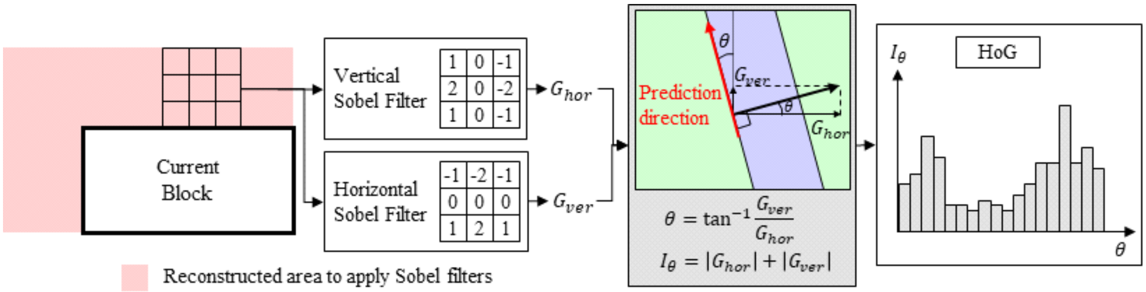


Fig. 2. HoG generation

1. Decoder-side derivation tools

1.1 Decoder-side intra mode derivation

By making it possible for a decoder to generate a predictor by itself without transmitting the intra prediction mode information, DIMD considerably reduces the signaling overhead for the intra prediction mode[13]. To generate an intra predictor in DIMD, the decoder derives two angular intra prediction modes from the reconstructed area around the current block. Two angular intra prediction modes are derived from a histogram of gradient (HoG) built by gathering gradients of every position for the reconstructed area around the current block.

The horizontal and vertical 3x3 Sobel filters are applied to an adjacent reconstructed area with thickness of 3 pixels to compute the gradient as shown in Figure 2[16]. The horizontal gradients are obtained by the vertical Sobel filter, and the vertical gradients are obtained by the horizontal Sobel filter. The prediction direction(θ) and its intensity(I_θ) are derived from the horizontal and vertical gradients. Figure 2 helps to understand the relationship between the gradient and prediction direction.

Based on HoG built with θ and I_θ, a final predictor is formed by combining three predictors generated with planar mode and two angular modes corresponding to two directions having the most prominent intensities. The three predictors are combined as follows[14]:

$$pred = w_{planar} \cdot pred_{planar} + (1 - w_{planar}) \cdot (w_1 \cdot pred_{\theta 1} + w_2 \cdot pred_{\theta 2}) \qquad (1)$$

where $w_{planar} = \frac{21}{64}, w_1 = \frac{I_{\theta 1}}{I_{\theta 1} + I_{\theta 2}}, w_2 = \frac{I_{\theta 2}}{I_{\theta 1} + I_{\theta 2}}, I_{\theta 1}$ and $I_{\theta 2}$ correspond to the two most prominent intensity values in HoG.

DIMD achieved -0.45%, -0.26%, and -0.30% coding gain in Y, Cb, and Cr channels with its encoding and decoding time-complexity increased to 113% and 110% [6]. The anchor was VTM 11.0. The intensity is calculated by absolute sum instead of the square sum to reduce computation. Since the actual intensity is the square root of the sum of squares, it would be useful to see how the coding performance would change when the intensity calculation is modified in the way practically more favorable.

1.2 Intra template matching prediction

Same as DIMD, intra TMP makes the decoder generate the predictor for itself[18]. The decoder finds a reference block inside a given search region whose template is most similar to that of the current block. The sum of absolute differences is computed for similarity evaluation. The template and search region are shown in Figure 3. d and α are respectively set to 4 and 5. It was reported that -0.40%, -0.47%, and -0.49% coding gain was achieved in Y, Cb, and Cr channels respectively with encoding and decoding time-complexity increases respectively of 178% and 168%[18]. The anchor was VTM 10.0.

A coding method that finds a reference block in the current picture and uses it as a predictor, such as intra block copy (IBC), has been proven to be very effective

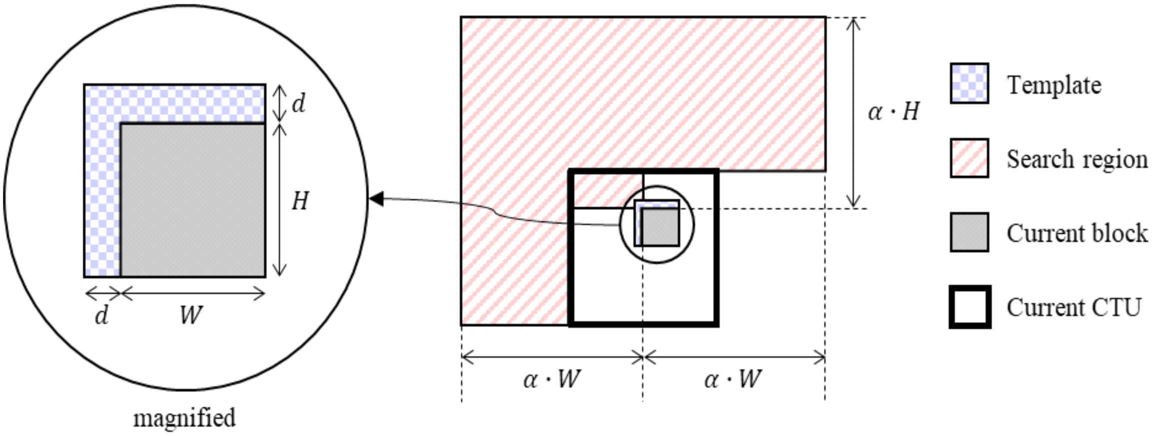


Fig. 3. Template and search range for intra TMP

especially in screen content[19]. Intra TMP also showed remarkable coding performance of -2.16% and -6.47% BDBR gain in Y channel respectively for class F and class TGM sequences both of which are screen content. Intra TMP and IBC exactly generate the same predictors when their reference blocks are the same. The intra TMP obtains significant coding gain like IBC since it does not need to signal position information of the reference block. However, its encoding time-complexity is increased respectively to 132% and 133% for class F and class TGM sequences, and the decoding time-complexity is increased also to 285% and 487% for class F and class TGM sequences. Note that the encoding time-complexity increase is smaller in screen content (that is, class F and TGM) than in camera-generated video content, and the decoding time-complexity increase is much larger in screen content. The lower increase of encoding time-complexity in the case of screen contents can be understood as lesser cases of proceeding to further smaller partitioning due to better prediction. The decoding time-complexity is significantly increased because the intra TMP has to perform a search process at the decoder. Therefore, a larger increased decoding time-complexity in the case of screen contents proves that the number of the blocks coded with the intra TMP is larger in the case of screen contents than that in the case of camera-generated video content.

1.3 Template-based intra mode derivation

TIMD is another decoder-side intra mode derivation method[22]. A decoder identifies the best one or two intra prediction modes among MPMs and uses them for generating predictor. The best modes are determined by comparing the sum of absolute transformed differences (SATD) in the template area.

The template consists of the above and left areas of the current block as shown in Figure 4. Note that, the template area is already reconstructed. The thickness of the template is determined according to the width and height of the current block, that is, if the width (or height) of a block is greater than 8, the thickness of the left (or above) template is 4, otherwise (i.e., less than or equal to 8) the thickness of the template is 2[20]. The decoder performs intra prediction for the template area using its reference samples [21] according to each intra mode under consideration, and SATD is computed using the difference between the already reconstructed values of the template area and its predicted values. Among all SATD values of the intra modes under consideration, the two minimal SATD costs are selected.

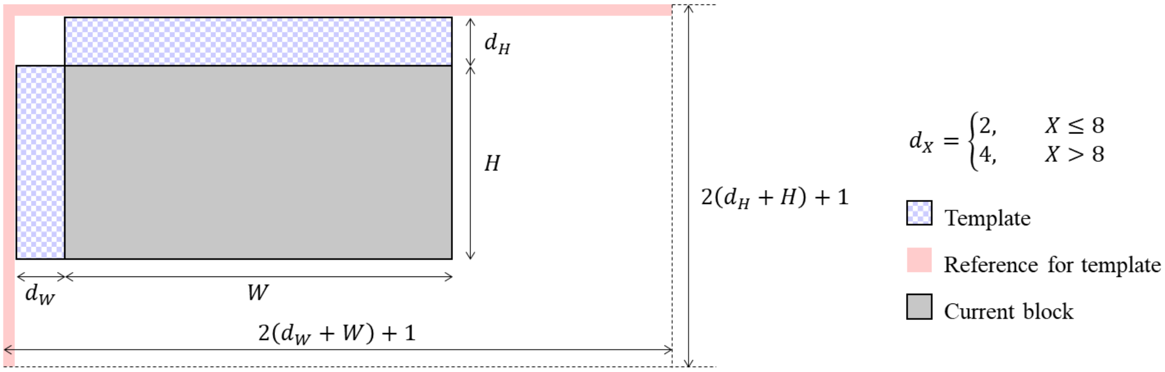


Fig. 4. Template and its reference in TIMD

If the two SATD values are not significantly different each other (that is, the larger SATD is not larger than twice of the smaller SATD), a final predictor is obtained by combining those two corresponding predictors created by the best two modes as follows:

$$pred = \frac{SATD_2}{SATD_1 + SATD_2} \cdot pred_{mode1} + \frac{SATD_1}{SATD_1 + SATD_2} \cdot pred_{mode2} \quad (2)$$

Otherwise, only the mode corresponding to the lowest SATD cost is used for the final prediction. TIMD achieved -0.47%, -0.34%, and -0.37% coding gain in Y, Cb, and Cr channels respectively with its encoding and decoding time-complexity increase to 124% and 111%[22]. The anchor was ECM 1.0.

Similar to the intra TMP, TIMD also gets rid of signaling for intra mode of MPM. Since a decoder has to perform a template-based mode search, the decoding time-complexity increases significantly. If the template thickness can be more precisely set depending on the block size, coding performance is expected to improve at the minimum increase of computation.

2. Intra chroma coding tools

2.1 Multi-model linear model

MMLM is an extension of CCLM in VVC which utilizes a linear relationship between luma and chroma channels for chroma intra prediction[7]. The CCLM in VVC selects four samples for each channel and derives one LM based on the average values of the two smallest and the two largest luma sample values as shown in Figure 5.

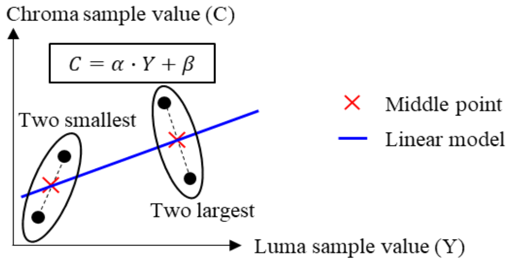


Fig. 5. Linear model derivation for CCLM in VVC

The luma sample corresponding to the chroma sample is obtained by downsampling in case of 4:2:0 chroma sub-sampling format. CCLM is divided into three modes (i.e., CCLM_LT, CCLM_L, and CCLM_T modes) according to the location where the samples are selected. Figure 6 shows how the downsampling is done according to the three CCLM modes in VVC.

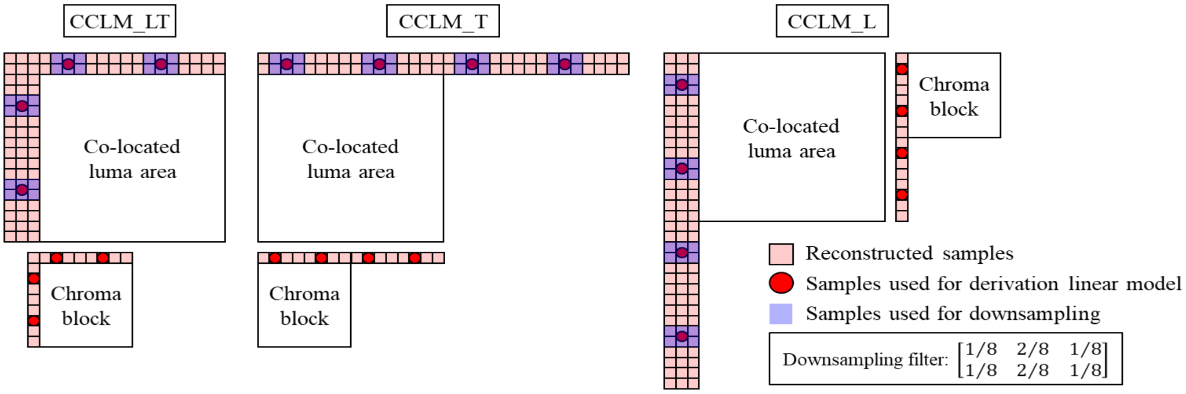


Fig. 6. Sample selection for the linear model derivation of CCLM in VVC

On the other hand, MMLM derives two linear models by dividing all available reference samples into two groups as shown in Figure 7. An average value of luma samples is used as a threshold for dividing the groups. Two linear models are derived by applying the least mean squared method to the two groups separately. In the same way as CCLM, MMLM also has three modes (i.e., MMLM_LT, MMLM_L, and MMLM_T) defined by the location of the samples being utilized for LM derivation.

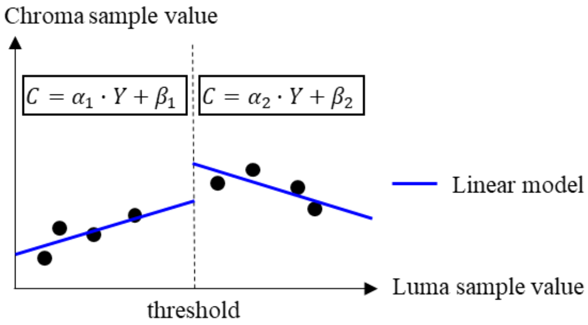


Fig. 7. Linear model derivation of MMLM

MMLM achieved coding gain of -0.21%, -1.57%, and -2.05% respectively in Y, Cb, and Cr channels with encoding time-complexity increased to 101% and decoding time-complexity decreased to 99%[6]. The anchor for comparison was the VVC test model (VTM) version 11.0.

The three MMLM modes are employed as additional intra chroma modes in addition to the already exiting three CCLM modes. Despite the signaling overhead of MMLM modes, its coding gain was found still achievable thus demonstrating that the current single linear model of CCLM was insufficient. Considering the currently early stage of ECM development, the better utilization of relationship between luma and chroma samples is expected to be further developed.

2.2 Slope adjustment for cross-component linear model

For CCLM and MMLM, both of which generate a chroma predictor by using linear model(s) approximating the relationship between luma and chroma, it has become possible to refine the already derived linear models by applying the slope adjustment[24]. The slope adjustment parameter with values from -1/2 to 1/2 (unit is 1/8) adjusts the slope of the linear model as shown in Figure 8. The slope adjustment parameter is transmitted for each chroma channel (i.e., Cb and Cr channels) using an integer values between -4 to 4. For MMLM that derives two linear models, two slope adjustment parameters are signaled to adjust each model. In both CCLM and MMLM, slope adjustment is not used (i.e., no signal is transmitted) for the modes that derive the linear model from only one reference side (i.e., CCLM_L, CCLM_T, MMLM_L, and MMLM_T modes).

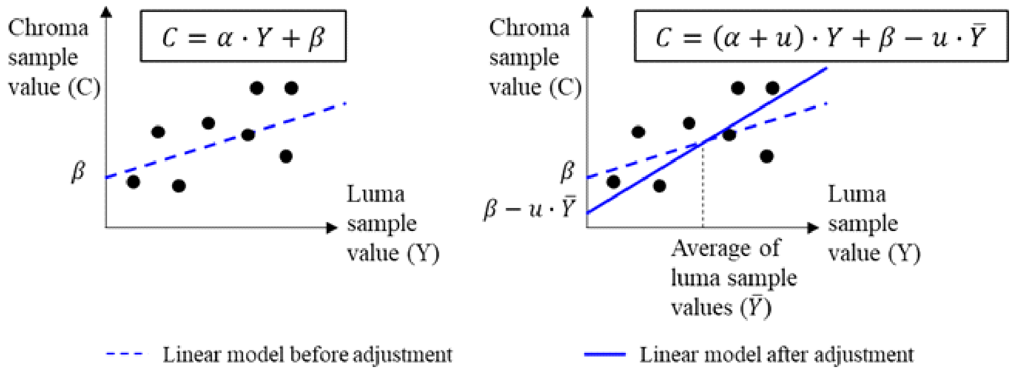


Fig. 8. Slope adjustment for linear model

It achieved -0.04%, -1.16%, and -1.07% BDBR gain respectively in Y, Cb, and Cr channels with 1% encoding time-complexity increase and no time-complexity change in decoding[25]. The anchor was ECM version 4.0.

Adjusting the slope of the linear model derived from the reconstructed pixels shows that there are some samples that are not suitable for linear model derivation. Therefore, if a sample to be used for linear model derivation can be selected adaptively or a non-linear model is derived, coding performance is expected to improve.

2.3 Chroma intra prediction modes fusion

This method combines non-LM mode and MMLM mode [27]. The non-LM mode can be planar, DC, horizontal, vertical, DM, and DIMD chroma mode. The MMLM mode is a mode that derives LM from both reference sides, not from only one side[27]. The two predictors are combined as follows:

$$pred = round(((1+N) \cdot pred_{nonLM} + (3-N) \cdot pred_{MMLM})/4), (N=0,1,2) \quad (3)$$

N indicates the number of blocks coded with non-LM modes among the above block and left block. This combination method showed -0.01%, -0.66%, and -0.61% BDBR gain respectively in Y, Cb, and Cr channels with 2% encoding time-complexity increase when excluding DIMD chroma. The anchor was ECM version 4.0.

Unlike the other intra prediction modes (e.g., planar, DC, and angular intra modes), CCLM and MMLM modes that use the relationship between the luma and the chroma channels do not directly exploit the neighboring reconstruction samples for predictor generation in the same channel. Thus, they may not reflect the relationship between the current block and neighboring samples well or may cause discontinuity on the block boundary. However, this fusion approach appears to address the discontinuity issue effectively.

2.4 Decoder-side intra mode derivation chroma

DIMD chroma derives one intra prediction mode using HoG which is the same as DIMD described before[26]. The area over which gradients are estimated is a neighboring region of co-located luma area of the current chroma block and a neighboring region of the current chroma block as shown in Figure 9. The standalone performance of DIMD chroma has not been reported, but the performance when implemented with another tool has been reported[27]. From the performance of several cases reported in [27], -0.06%, -0.51%, and -0.41% BDBR gain in Y, Cb, and Cr channels can be seen as the standalone performance of DIMD chroma by calculating the difference in performance between with and without it. The anchor was ECM version 4.0.

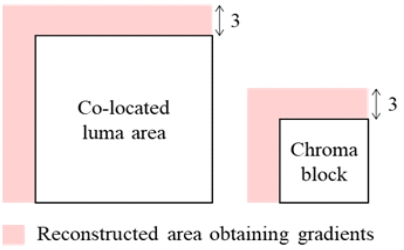


Fig. 9. Reconstructed area for DIMD chroma

The derived mode (DM)[28][29] in HEVC and VVC sets an intra chroma prediction mode same as the co-locating luma channel and it contributes to solving the lack of diversity of prediction mode candidates in chroma intra prediction. In the same spirit, DIMD chroma also increases the number of the intra chroma prediction mode candidates. However, DIMD chroma solves the problem that DM cannot represent the co-located luma area well when the partitioning structure of the luma and the chroma channels is different.

3. Extended coding tools

3.1 Reference sample interpolation and smoothing

A simple modification of increasing the number of samples to be used for interpolation also improves the coding performance[12]. When a predictor is generated using the angular prediction mode, the position of the reference pixel that the current pixel exploits the best may not fall at an integer pixel position because the precision of the prediction direction is 1/32 pel. In this case, interpolation has been performed using a 4-tap filter to increase the prediction accuracy. When generating a predictor with the left-upward prediction mode (i.e., between the horizontal and vertical directions), both top and left reference samples are utilized. Since the L-shaped is less preferred for hardware implementation, a straight reference is used for the prediction by extending the reference as shown in Figure 10.

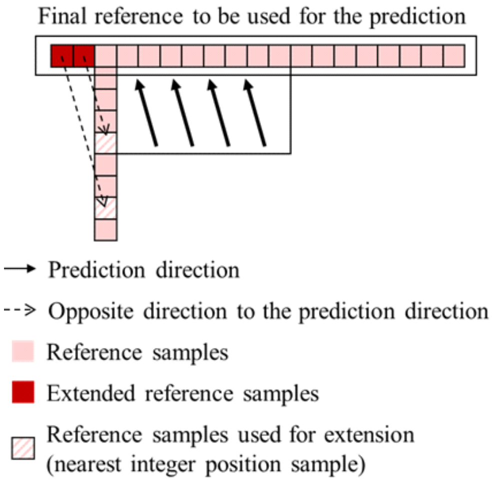


Fig. 10. The reference extension in VVC

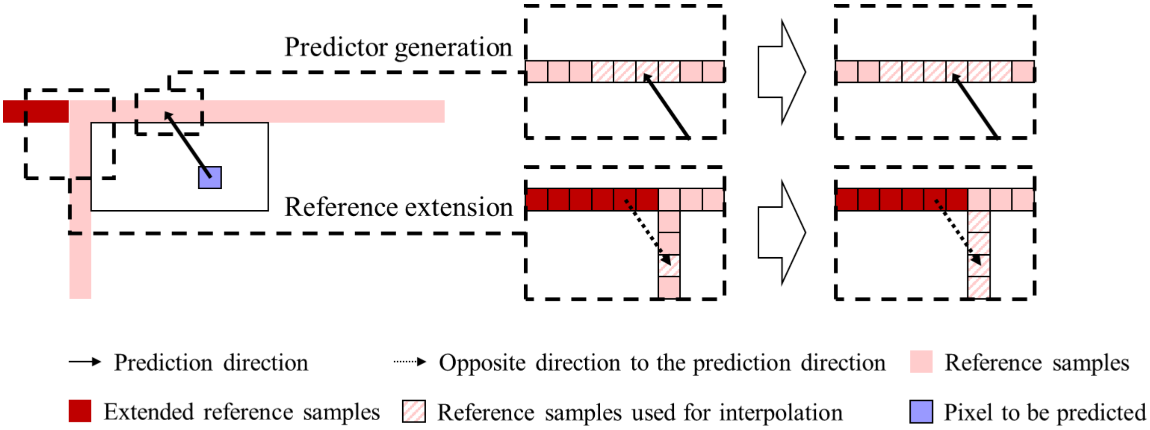


Fig. 11. Change of reference sample interpolation

In the current ECM, as shown in Figure 11, the 4-tap interpolation filters for generating intra predictor have been replaced with 6-tap filters, and the 4-tap interpolation is used for the reference extension instead of using the nearest integer position sample. It was reported that coding gain of -0.14%, -0.17%, and -0.19% has been achieved respectively in Y, Cb, and Cr channels with its encoding and decoding time-complexity increased to 104% and 102%[6]. The anchor was VTM version 11.0.

Although it is a simple way to make the interpolation more precise, but it brings considerable encoding time-complexity increase of 4%. It can be said that the time-complexity increase comes from the increased interpolation filtering operation performed for every pixel in a block and the predictor generation on the same block with various prediction modes in the RDO process.

3.2 Extended multiple reference line

VVC has the MRL scheme employing only three reference lines (their reference line indices are 0,1,2) for intra prediction. In ECM, it has been expanded to include more reference lines, that is, lines having indices 0,1,3,5,7,12[23]. As shown in Figure 12, fairly distant reference samples have become available, and the number of candidates has been increased. It produces -0.09%, -0.05%, and -0.09% BDBR gain in Y, Cb, and Cr channels respectively with no time-complexity increase. The anchor was ECM version 3.1.

Table 2 shows usage percentage of each reference line [23]. Note that the line index of the reference adjoining the current block is 0, not 1. As shown in Table 2, the farther the reference line is, the greater its signaling overhead is. However, the percentage of using the farthest reference line is not the smallest. Moreover, 34.34% of cases used a reference line with a distance greater than 4 (i.e., the minimum block size). It can be said that it is possible to predict well in the case where the current block and the close area around it are heterogeneous by not using the close reference line but the fairly distant reference line.

Table 2. Percentage of selected reference lines

Reference line index	1	3	5	7	12
Bin string	10	110	1110	11110	11111
Percentage	46.37%	19.29%	13.70%	9.04%	11.60%

3.3 Secondary most probable mode

To reduce the signaling overhead of a large number of intra prediction modes, most probable mode (MPM) has been used to exploit the correlation between neighboring blocks[10]. The number of MPMs is increased from 6 to 22 by deriving intra prediction modes more from 5 neighboring blocks (i.e., above-left (AL), above (A), above right (AR), left (L), and bottom-left (BL)) as shown in Figure 13[11]; VVC exploits only above and left neighboring blocks for MPM derivation (i.e., A and L in Figure 13).

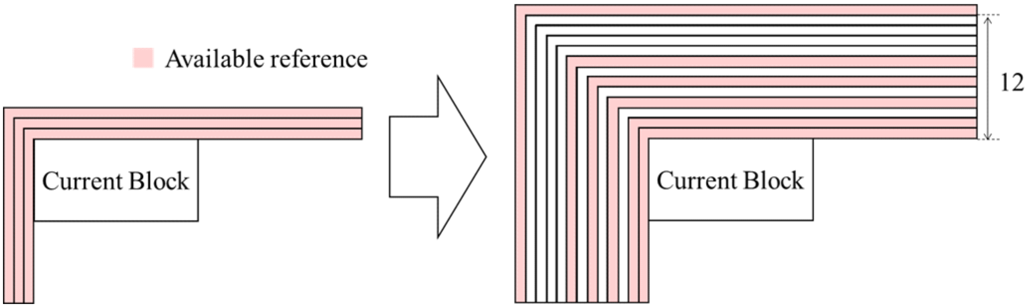


Fig. 12. Extension of MRL

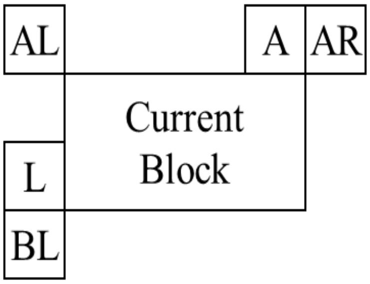


Fig. 13. Neighboring blocks used for MPM derivation

MPM list is filled in the following order: planar, neighboring intra prediction modes, and intra prediction modes that is obtained by adding offsets to neighboring modes. In contrast to the fixed order of filling neighboring intra prediction modes in VVC (i.e., L→A), the filling order in ECM depends on the aspect ratio of the block; for vertically-elongated blocks, it is filled in the order of A→L→BL→AR→AL, and for the other case (that is, horizontally-elongated or square blocks), it follows the order of L→A→BL→AR→AL. If the MPM list is not fully filled with the procedure above, pre-defined modes are additionally inserted into the MPM list. MPM is divided into primary MPM (PMPM) and secondary MPM (SMPM) [11]. In the MPM list, the first 6 candidates are PMPM, and the remaining 16 candidates are SMPM. This MPM improvement method achieved -0.26%, -0.24%, and -0.27% coding gain in Y, Cb, and Cr channels respectively with its encoding and decoding time-complexity increased to 101% [6]. The anchor was VTM version 11.0.

The coding performance was improved not only by using more neighboring intra modes but also by creating the MPM list additionally considering the aspect ratio of a block. Determining the MPM list order according to the aspect ratio of a block means that the aspect ratio of a block is related to the correlation between the current block and the neighboring block. That is, in the case of a horizontally elongated, the current block is more similar to its above block rather than its left block, and in the case of a vertically elongated, the current block is more similar to its left block rather than its above block. It is also noticeable that the above-left block and the current block are related the least.

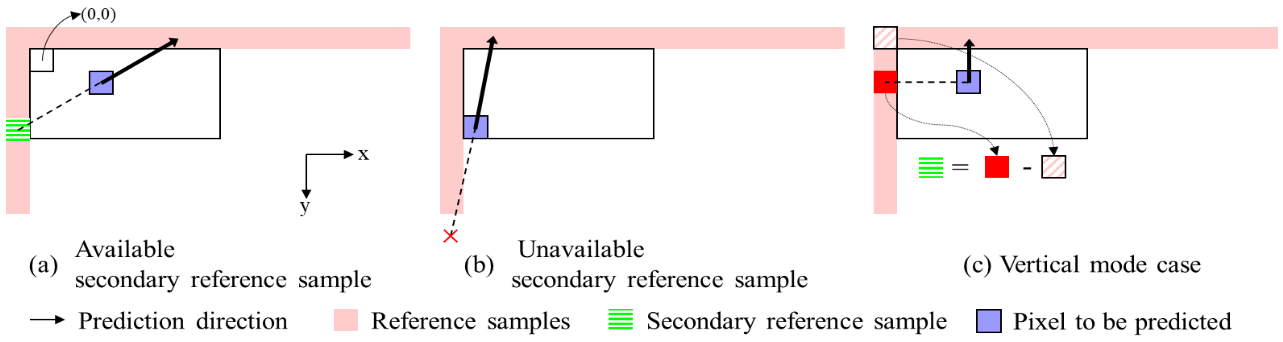
3.4 Gradient position dependent prediction combination

If an intra predictor is generated using only one side of the top or left reference samples, discontinuity is likely to occur on the block boundary of the other side. PDPC alleviates this issue of discontinuity by additionally using secondary reference samples in computing the final predictor through a weighted sum [8]. As shown in Figure 14(a), the secondary reference is located in the opposite direction of the prediction direction. When intra mode is close to the vertical or horizontal mode, its secondary reference sample may not be unavailable as shown in Figure 14(b), in this case, PDPC is not applied in VVC. However as shown in Figure 14(c), in horizontal and vertical modes, the secondary reference sample can be estimated by gradient (i.e., an amount of change along the prediction direction on the reference line) as follows:

$$ref_{secondary}(x,y) = \begin{cases} ref(x,-1) - ref(-1,-1) & \text{for horizontal mode} \\ ref(-1,y) - ref(-1,-1) & \text{for vertical mode} \end{cases} \quad (4)$$

Gradient PDPC [9] makes it possible to apply the PDPC by using the gradient mentioned above when the secondary reference sample is unavailable. As shown in Figure 15, the gradient is obtained as an amount of change along the prediction direction from one reference side to the other reference side as follows:

$$ref_{secondary}(x,y) = \begin{cases} ref(x,-1) - ref(-1,-1+d) & \text{for close to horizontal mode} \\ ref(-1,y) - ref(-1+d,-1) & \text{for close to vertical mode} \end{cases} \quad (5)$$



(a) Available secondary reference sample (b) Unavailable secondary reference sample (c) Vertical mode case

Fig. 14. Secondary reference samples in PDPC

where d denotes an offset that varies depending on the prediction direction and the distance from the reference sample. If d is not an integer, $ref(-1, y+d)$ or $ref(x+d, -1)$ is obtained by interpolation. Gradient PDPC achieved coding gain of -0.08%, -0.10%, and -0.12% respectively in Y, Cb, and Cr channels with its encoding and decoding time-complexity increased respectively to 101% and 102% [6]. Its anchor was VTM 11.0. The gradient PDPC scheme is shown to be able to improve coding performance only with a minor increase in encoding and decoding time-complexity. Due to the introduction of gradient PDPC, discontinuity reduction at block boundary is now applied for all intra prediction modes that generate predictor using only single reference side.

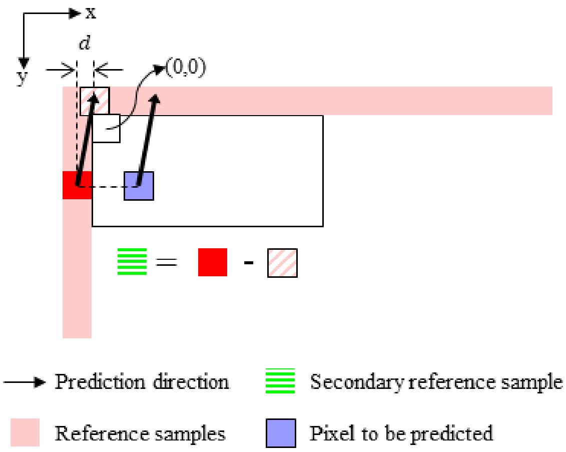


Fig. 15. Calculation of the secondary reference sample

Ⅲ. Conclusion

It can be seen that there are roughly three ways to improve coding performance. The first is to make sophisticated predictors despite of increased coding time-complexity, and the second is to generate more various predictors even if signaling overhead is increased. The last is to reduce signaling overhead by deriving a predictor or a prediction mode by decoder itself. MMLM, gradient PDPC, and reference sample interpolation and smoothing belong to the first category while SMPM, extended MRL, slope adjustment for CCLM, and combination of chroma intra prediction modes belong to the second category. The last category includes DIMD, intra TMP, TIMD, and DIMD chroma.

However, in the case of decoder-side intra mode or predictor derivation tools (i.e., DIMD, intra TMP, TIMD, and DIMD chroma), they make the decoder refer to the large template area, which is particularly unfriendly to the hardware design of intra coding. Even with a slight loss in BDBR gain, hardware-friendly design should be considered by reducing the size of the template as much as possible or by utilizing a virtual pipeline data unit (VPDU). In particular, in the case of intra TMP, which is very similar to IBC, just as IBC limits the search range within VPDU, the performance of intra TMP should also be checked under the condition that VPDU is considered.

In addition, various methods that can reduce the time-complexity of ECM encoder can be considered. For example, similar to the method proposed in [36], the optimal on/off setting of coding tools can be found by considering the trade-off between BDBR and encoder time-complexity. As an another example, similar to the method proposed in [37], the encoder time-complexity can be reduced by removing some RD tests using hardware-friendly designed machine learning techniques.

References

[1] F. Bossen, X. Li, and K. Suehring, *AHG report: Test model software development (AHG3)*, Joint Video Experts Team (JVET), 20th Meeting, teleconference, JVET-T0003, v.1, 2020.

[2] ECM reference software, https://vcgit.hhi.fraunhofer.de/ecm/ECM/-/tree/ECM-5.0 (accessed Oct. 4).

[3] M. Karczewicz et al., *JVET AHG report: Enhanced compression beyond VVC capability (AHG12),* Joint Video Experts Team (JVET), 27th Meeting, teleconference, JVET-AA0012, v.1, 2022.

[4] M. Coban, F. Léannec, K. Naser, and J. Ström, *Algorithm description of Enhanced Compression Model 5 (ECM 5)*, Joint Video Experts Team (JVET), 26th Meeting, teleconference, JVET- Z2025, v.2, 2022.

[5] M. Coban, F. Léanne, J. Ström, and Y. Ye, *Preliminary draft of algorithm description for Enhanced Compression Model 1 Software (ECM 1),* Joint Video Experts Team (JVET), 23rd Meeting, teleconference, JVET-W0102, v.2, 2021.

[6] Y. Chang et al., *EE2: Tests of compression efficiency methods beyond VVC*, Joint Video Experts Team (JVET), 22nd Meeting, teleconference, JVET-V0120, v.2, 2021.

[7] K. Zhang, J. Chen, L. Zhang, and M. Karczewicz, *Enhanced Cross-component Linear Model Intra-prediction*, Joint Video Experts Team (JVET), 4th Meeting, Chengdu, JVET-D0110, v.3, 2016.

[8] G. Auwera, V. Seregin, A. Said, A. Ramasubramonian, and M. Karczewicz, *CE3: Simplified PDPC (Test 2.4.1)*, Joint Video Experts Team (JVET), 11th Meeting, Ljubljana, JVET-K0063, v.2, Jul. 2018.

[9] B. Ray, G. Auwera, and M. Karczewicz, *Unified PDPC for angular intra modes*, Joint Video Experts Team (JVET), 17th Meeting, Brussels, JVET-Q0391, v.3, 2020.

[10] L. Li et al., *CE3-6.2.1: Extended MPM list*, Joint Video Experts Team (JVET), 12th Meeting, Macao, JVET-L0165, v.4, 2018.

[11] V. Seregin, W. Chien, and M. Karczewicz, *Block shape dependent intra mode coding*, Joint Video Experts Team (JVET), 4th Meeting, Chengdu, JVET-D0114, v.1, 2016.

[12] X. Zhao, V. Seregin, and M. Karczewicz, *Six tap intra interpolation filter*, Joint Video Experts Team (JVET), 4th Meeting, Chengdu, JVET-D0119, v.1, 2016.

[13] M. Abdoli et al., *Non-CE3: Decoder-side Intra Mode Derivation with Prediction Fusion Using Planar*, Joint Video Experts Team (JVET), 15th Meeting, Gothenburg, JVET-O0449, v.2, 2019.

[14] M. Abdoli, E. Mora, T. Guionnet, and M. Raulet, *Non-CE3: Decoder-side Intra Mode Derivation with Prediction Fusion*, Joint Video Experts Team (JVET), 14th Meeting, Geneva, JVET-N0342, v.5, 2019.

[15] E. Mora, A. Nasrallah, M. Abdoli, and M. Raulet, *CE3: Decoder-side Intra Mode Derivation (tests 3.1.1, 3.1.2, 3.1.3 and 3.1.4)*, Joint Video Experts Team (JVET), 13th Meeting, Marrakech, JVET- M0094, v.2, 2019.

[16] E. Mora, A. Nasrallah, and M. Raulet, *CE3-related: Decoder-side Intra Mode Derivation*, Joint Video Experts Team (JVET), 12th Meeting, Macao, JVET-L0164, v.2, 2018.

[17] K. Naser, F. Leannec, T. Poirier, and F. Galpin, *Evaluation of Template Matching Prediction for VVC*, Joint Video Experts Team (JVET), 21st Meeting, teleconference, JVET-U0048, v.1, 2021.

[18] K. Naser T. Poirier, F. Léannec, and G. Martin-Cocher, *EE2: Intra Template Matching*, Joint Video Experts Team (JVET), 22nd Meeting, teleconference, JVET-V0130, v.6, 2021.

[19] X. Xu, X. Li, and S. Liu, *CE8: CPR mode with local search ranges (Test CE8.3.1 and CE8.3.2)*, Joint Video Experts Team (JVET), 12th Meeting, Macao, JVET-L0293, v.2, 2018.

[20] X. Xiu, Y. He, and Y. Ye, *Decoder-side intra mode derivation*, Joint Video Experts Team (JVET), 3rd Meeting, Geneva, JVET-C0061, v.1, 2016.

[21] Y. Wang, L. Zhang, K. Zhang, Z. Deng, and N. Zhang, *EE2-related: Template-based intra mode derivation using MPMs*, Joint Video Experts Team (JVET), 22nd Meeting, teleconference, JVET-V0098-v2, 2021.

[22] K. Cao, N. Hu, V. Seregin, and M. Karczewicz, *EE2-related: Fusion for template-based intra mode derivation*, Joint Video Experts Team (JVET), 23rd Meeting, teleconference, JVET-W0123, v.2, 2021.

[23] K. Cao, Y. Chang, B. Ray, V. Seregin, and M. Karczewicz, *EE2-2.1: Extended MRL candidate list*," Joint Video Experts Team (JVET), 25th Meeting, teleconference, JVET-Y0116, v.1, 2022.

[24] J. Lainema, A. Aminlou, P. Astola, and R. Youvalari, *AHG12: Slope adjustment for CCLM*, Joint Video Experts Team (JVET), 25th Meeting, teleconference, JVET-Y0055, v.1, 2022.

[25] J. Lainema, A. Aminlou, P. Astola, and R. Youvalari, *EE2-1.1: Slope adjustment for CCLM*, Joint Video Experts Team (JVET), 26th Meeting, teleconference, JVET-Z0049, v.1, 2022.

[26] X. Li, R. Liao, J. Chen, and Y. Ye, *Non-EE2: On chroma intra prediction mode*, Joint Video Experts Team (JVET), 25th Meeting, teleconference, JVET-Y0092, v.3, 2022.

[27] X. Li, R. Liao, J. Chen, and Y. Ye, *EE2-1.2: On chroma intra prediction*, Joint Video Experts Team (JVET), 26th Meeting, teleconference, JVET-Z0051, v.1, 2022.

[28] L. Zhang, K. Zhang, H. Liu, Y. Wang, P. Zhao, and D. Hong, *CE3-related: Modified chroma derived mode*, Joint Video Experts Team (JVET), 12th Meeting, Macao, JVET-L0272, v.2, 2018.

[29] N. Choi, M. Park, and K. Choi, *CE3-related: Chroma DM*

modification, Joint Video Experts Team (JVET), 12th Meeting, Macao, JVET-L0053, v.3, 2018.

[30] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, *JVET common test conditions and software reference configurations for SDR video*, Joint Video Experts Team (JVET), 14th Meeting, Geneva, JVET-N1010, v.1, 2019.

[31] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, *VTM common test conditions and software reference configurations for SDR video*, Joint Video Experts Team (JVET), 20th Meeting, tele- conference, JVET-T2010, v.1, 2020.

[32] M. Karczewicz and Y. Ye, *Common test conditions and evaluation procedures for enhanced compression tool testing*, Joint Video Experts Team (JVET), 21st Meeting, teleconference, JVET-U2017, v.1, 2021.

[33] M. Karczewicz and Y. Ye, *Common test conditions and evaluation procedures for enhanced compression tool testing*, Joint Video Experts Team (JVET), 22nd Meeting, teleconference, JVET-V2017, v.1, 2021.

[34] M. Karczewicz and Y. Ye, *Common test conditions and evaluation procedures for enhanced compression tool testing*, Joint Video Experts Team (JVET), 24th Meeting, teleconference, JVET-X2017, v.1, 2021.

[35] M. Karczewicz and Y. Ye, *Common test conditions and evaluation procedures for enhanced compression tool testing*, Joint Video Experts Team (JVET), 25th Meeting, teleconference, JVET-Y2017, v.1, 2022.

[36] M. Xu and B. Jeon, "User-Priority Based AV1 Coding Tool Selection," *IEEE Trans. Broadcast.*, Vol.67, No.3, pp. 736-745, Sept. 2021, doi: https://doi.org/10.1109/TBC.2021.3071013.

[37] J. Park, B. Kim, and B. Jeon, "Machine Learning-Based Early Skip Decision for Intra Subpartition Prediction in VVC," *IEEE Access*, Vol.10, pp. 111052-111065, Oct. 2022. doi: https://doi.org/10.1109/ACCESS.2022.3215163.

Introduction Authors

Bumyoon Kim

- Aug. 2020 : Sungkyunkwan University, School of Electronic and Electrical Engineering (Bachelor)
- Sep. 2020 ~ present : Sungkyunkwan University, Dept. Electrical and Computer Engineering (Pursuing Combined master and Ph.D.)
- ORCID : https://orcid.org/0000-0002-9903-0896
- Research interests : image compression, machine learning, image signal processing

Byeungwoo Jeon

- 1985 : Seoul National University, Dept. Electronic Engineering (Bachelor)
- 1987 : Seoul National University, Dept. Electronic Engineering (Master)
- 1992 : Purdue University, School of Electrical Engineering (Ph.D.)
- 1993 ~ 1997 : Samsung Electronics Co., Signal Processing Lab
- 1997 ~ present : Sungkyunkwan University, Dept. Electrical and Computer Engineering (Professor)
- ORCID : https://orcid.org/0000-0002-5650-2881
- Research interests : multi-media, video processing and compression, image recognition, signal processing