Sub-sampled Cross-Component Prediction For Chroma Component Coding

Junru Li[†], Meng Wang[*], Li Zhang[⊛], Kai Zhang[⊛]
Shiqi Wang[*], Shanshe Wang[†], Siwei Ma[†] and Wen Gao[†]

[†]Institute of Digital Media, Peking University, Beijing, China.
[*]City University of Hong Kong, Hong Kong, China.
[⊛]Bytedance Inc., San Diego CA. 92122 USA.

Abstract

Cross-component prediction, which takes advantage of inter-channel correlations, predicts the chroma block with the luma reconstructed block according to associated linear model. Instead of involving all available reference samples in building the linear model, in this paper, we propose a sub-sampled approach that utilizes at most four neighboring chroma samples and their corresponding down-sampled luma samples, leading to significantly reduced operations in the derivation of model parameters at both encoder and decoder. The proposed scheme is hardware friendly in terms of the overheads of memory access and clock cycles, and greatly benefits the practical implementations of the emerging video coding standard in real applications. Extensive experiments reveal that the proposed sub-sampled method provides simple operations and robust coding performance, leading to the adoption by Versatile Video Coding (VVC) Standard and the third generation Audio Video Coding Standard (AVS3).

1. Introduction

Cross-component linear prediction mode (CCLM) removes inter-channel redundancies to further improve coding performance. It was originally proposed and studied by Kim *et al.* [1] and Chen *et al.* [2] for the development of High Efficiency Video Coding (HEVC) [3] standard. Furthermore, numerous approaches have been investigated to improve CCLM in the literature. In [4], a multi-model based CCLM was studied, which applies two linear models to more precisely represent the inter-channel linear correlations. Adaptive Cr component prediction was proposed in [5], such that Cr can be predicted not only by Y component, but also by Cb component or the adaptive combination of Y and Cb. A hybrid neural network based prediction strategy was investigated in [6] that exploits both spatial and cross-channel correlations. Zhang *et al.* [7] propose two additional CCLM modes that use only left or above reference samples to generate the linear model. The enhanced CCLM was investigated in [8] wherein multi-filters for luma component are involved. An LM-angular method was also proposed in [8] on the basis of multi-hypothesis theory, which combines angular intra prediction and CCLM intra prediction to further enhance the prediction efficiency.

However, regarding the derivations of the linear model, traditional CCLM based approaches adopt classical least square regression (LSR), along with additional surrounding reference samples to obtain a fine-grained prediction model. Assuming a

2375-0359/20/$31.00 ©2020 IEEE
DOI 10.1109/DCC47342.2020.00028

luma coding block (CB) is of size $2N \times 2N$, the corresponding chroma CB is $N \times N$ for the 4:2:0 color format. If a chroma CB is predicted with CCLM, the prediction result associated to position (i, j) can be obtained as follows [1],

$$Pred_C(i,j) = \alpha \cdot \hat{Rec_L}(i,j) + \beta, \tag{1}$$

where $Pred_C$ denotes the prediction of chroma, and $\hat{Rec_L}$ is the down-sampled reconstructed samples of collocated luma coding block. α and β are two linear model parameters that can be derived according to the neighboring luma and chroma reconstructed samples via LSR as

$$\alpha = \frac{M \cdot \sum_{m=1}^{M} Rec_C^{(m)} \cdot \hat{Rec_L}^{(m)} - \sum_{m=1}^{M} Rec_C^{(m)} \sum_{m=1}^{M} \hat{Rec_L}^{(m)}}{M \cdot \sum_{m=1}^{M} (\hat{Rec_L}^{(m)})^2 - (\sum_{m=1}^{M} \hat{Rec_L}^{(m)})^2}, \tag{2}$$

$$\beta = \frac{\sum_{m=1}^{M} Rec_C^{(m)} - \alpha \sum_{m=1}^{M} \hat{Rec_L}^{(m)}}{M}, \tag{3}$$

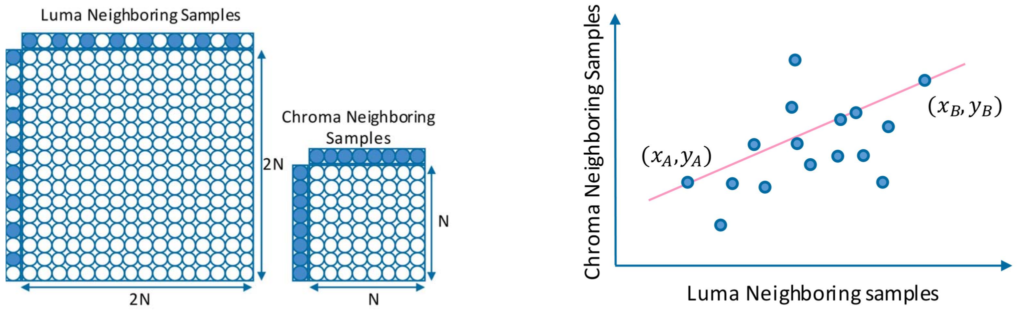
where M denotes the total number of neighboring available samples. Rec_C and $\hat{Rec_L}$ represent the neighboring chroma and down-sampled luma reference samples. The explicit sample positions are illustrated in Fig. 1(a). It should be noted that the division operation is usually replaced by multiplication to reduce complexity. As such, in total the number of multiplication is $2M + 4$, adding operation is $7M + 3$, and the substituted division is 2 in the LSR method.

It is generally acknowledged that multiplication tends to consume more clock cycles than adding and comparison operations. To reduce the number of multiplications, a max-min method [9] was adopted in VTM-4.0 [10] to replace the LSR for generating α and β. In particular, a straight line is directly obtained by two points (couple of luma and chorma) that possess the maximum and minimum luma intensities of all the neighboring samples, as shown in Fig. 1(b). As such, α and β can be feasibly obtained as follows,

$$\alpha = \frac{y_A - y_B}{x_A - x_B} \tag{4}$$

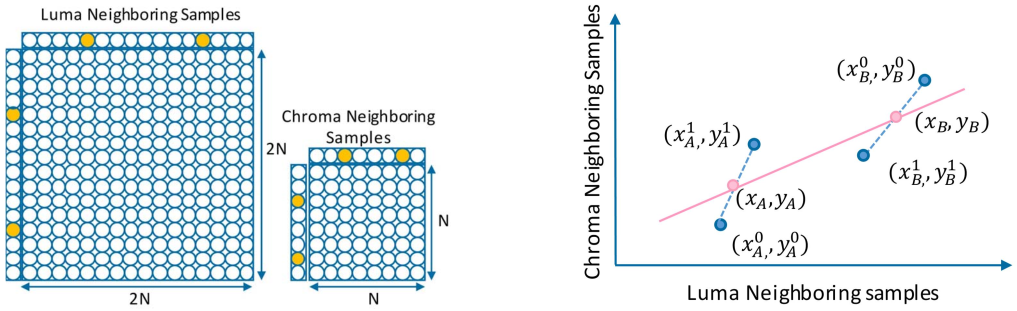
$$\beta = y_A - \alpha x_A \tag{5}$$

where x_A and x_B denote the minimum and maximum down-sampled neighboring luma samples. y_A and y_B are their corresponding chroma sample intensities. In this manner, the multiplication operations are shifted to the searching of the maximum and minimum luma samples.



(a) Locations of the samples (b) Max-min method

Figure 1: Illustration of the locations regarding the neighboring luma and chroma samples, and the derivations of linear model with the Max-min method.



(a) Locations of the samples (b) Proposed sub-sampled method

Figure 2: Illustration of the neighboring luma and chroma samples with fixed position, and the derivation of the linear model with the proposed method.

2. Sub-sampled Cross-Component Prediction and Implementation

2.1. Sub-sampled Cross-Component Prediction

The derivation of CCLM parameters involves quite considerable number of comparison operations, which are not practical in hardware or software design, especially at the decoder side. Herein, we propose a sub-sampled method that derives the model parameters with four neighboring chroma samples and the associated down-sampled luma samples, as illustrated in Fig. 2(a). In this way, luma component down-sampling only applys to the selected positions, which largely enhances the memory access efficiency. On top of that, the number of comparison is reduced to 4, with an effort to obtain the two samples with smaller intensity: x_A^0 and x_A^1, and two larger intensity: x_B^0 and x_B^1. The corresponding chroma sample are denoted as y_A^0, y_A^1, y_B^0 and y_B^1. Consequently, x_A, x_B, y_A and y_B can be derived as,

$$x_A = (x_A^0 + x_A^1 + 1) >> 1, \tag{6}$$

$$x_B = (x_B^0 + x_B^1 + 1) >> 1, \tag{7}$$

$$y_A = (y_A^0 + y_A^1 + 1) >> 1, \tag{8}$$

$$y_B = (y_B^0 + y_B^1 + 1) >> 1. \tag{9}$$

Given these values, the parameters α and β are obtained according to Eqn. (4) and Eqn. (5). We illustrate the derivation of α and β in Fig. 2(b). Since it is not necessary to further compare x_A^0 and x_A^1, or x_B^0 and x_B^1, four comparisons are sufficient. If neither of the left and above neighboring samples are available, a default prediction is used with α equalling to 0 and β equalling to $1 << (bitDepth - 1)$ where $bitDepth$ represents the bit-depth of chroma samples. In addition, for avoiding the division, a look-up table is introduced to achieve the division operation.

The proposed sub-sampled cross-component prediction has two main advantages. Firstly, instead of traversing all the neighboring reconstructed samples to derive the linear model, we elaborately select at most four samples with fixed positions. As such, luma down-sampling is only applied to the chosen four samples. Second, rather than deriving linear model by LSR or max-min method, the linear model is robustly generated with the average of two larger and two smaller values corresponding to luma intensities. Thus, the unnecessary multiplications, adding and comparison operations can be eliminated.

2.2. Implementation Details in VVC

In addition to the conventional CCLM mode, VVC adopts LM-Above and LM-Left modes to better predict the inter-channel mapping correlations. In particular, LM-Above only utilizes the above samples and LM-Left only adopts the left samples to derive the model parameters. As such, CCLM mode is able to access all the available reference samples to establish the linear model.

We propose to employ the sub-sampled method to all LM modes, with the aim of simplifying the linear model derivation [11] [12]. More specifically, the four sample positions are determined according to the block size, the specific LM mode and the availability in terms of the neighboring reference samples. Assuming the current chroma CB dimensions are $W \times H$, we set W' and H' according to the explicit LM modes as follows,

$$W' = \begin{cases} W & \text{CCLM} \\ W + H & \text{LM-Above} \end{cases} \text{and } H' = \begin{cases} H & \text{CCLM} \\ H + W & \text{LM-Left} \end{cases}. \tag{10}$$

As such, the range of above neighbouring positions can be expressed from $S[0, -1]$ to $S[W' - 1, -1]$ and the range of left neighbouring positions are from $S[-1, 0]$ to $S[-1, H' - 1]$. Subsequently, we select the four samples as

- $S[\frac{W'}{4}, -1]$, $S[\frac{3W'}{4}, -1]$, $S[-1, \frac{H'}{4}]$, $S[-1, \frac{3H'}{4}]$ when CCLM mode is applied and both above and left neighbouring samples are accessible;
- $S[\frac{W'}{8}, -1]$, $S[\frac{3W'}{8}, -1]$, $S[\frac{5W'}{8}, -1]$, $S[\frac{7W'}{8}, -1]$ when LM-Above mode is selected or only the above neighbouring samples are available;

Table 1: Complexity comparison for CCLM parameter derivation on a 32 × 32 chroma coding block

	Luma down-sampling	Comparison between luma samples
VTM-4.0	64	128
Proposed	4	4
Proposed/VTM-4.0	6%	3%

- $S[-1, \frac{H'}{8}]$, $S[-1, \frac{3H'}{8}]$, $S[-1, \frac{5H'}{8}]$, $S[-1, \frac{7H'}{8}]$ when LM-Left mode is applied or only the left neighbouring samples are available.

For a typical 32×32 chroma block, when both left and above neighboring samples are accessible, the complexity reduction of the proposed method is illustrated in Table 1 when compared to VTM-4.0. The luma down-sampling for the neighboring reference samples can be reduced from 64 to 4. Meanwhile, the number of comparison between luma samples can be decreased to 3% of the VTM-4.0. Moreover, the proposed sub-sampled method is robust to the codec, and superior coding performance compared with the max-min method in VTM-4.0 is achieved.

2.3. Implementation Details in AVS3

There are five chroma prediction modes in AVS3 intra coding, including derived mode (DM), horizontal mode, vertical mode, DC mode and bi-linear mode. Based on the sub-sampled cross-component prediction, we propose the Two-Step Cross-component Prediction Mode (TSCPM) [13] [14] [15] for chroma intra coding wherein the linear model is derived with sub-sampled pairs. Though the principle behind TSCPM is similar to CCLM, there are some differences from the technical perspective. More specifically, the linear model can be directly applied to the reconstructed luma samples, which generates a temporal chroma prediction block of the same size. Subsequently, down-sampling is performed on the internal block, yielding the final chroma prediction block.

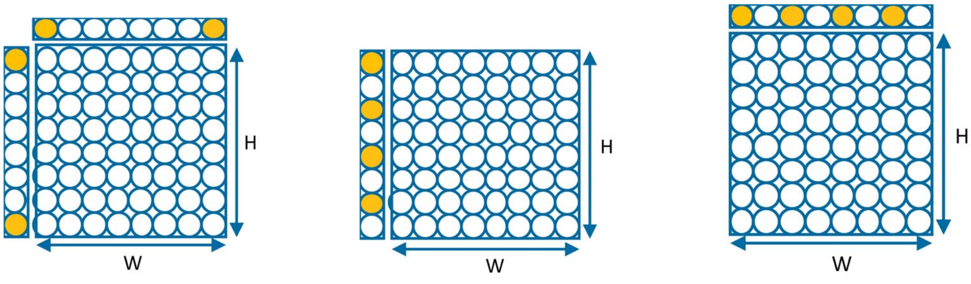
To derive the linear model, the ratio r of block width and height is calculated as follows,

$$r = \begin{cases} W/H & W \geq H \\ H/W & H > W \end{cases}, \tag{11}$$

where W and H denote the width and height of chroma CB, respectively. The number of accessible neighboring samples can be expressed with W and H as well. As such, we need to consider the availability of the left and above neighboring samples. The four samples can be selected as,

- $S[0, -1]$, $S[W - max(1, r), -1]$, $S[-1, 0]$ and $S[-1, H - max(1, r)]$, when both of the left and above reference samples are available.

- $S[0, -1]$, $S[\frac{W}{4}, -1]$, $S[\frac{2W}{4}, -1]$ and $S[\frac{3W}{4}, -1]$, when only above reference samples are available.



(a) (b) (c)

Figure 3: Illustration of the proposed sample selection method; (a) Left and above are both available; (b) Only left is available; (c) Only above is available.

- $S[-1,0]$, $S[-1,\frac{H}{4}]$, $S[-1,\frac{2H}{4}]$ and $S[-1,\frac{3H}{4}]$, when only left reference samples are available.

The sample selection strategy is illustrated in Fig. 3. It should be mentioned that for the case that the chroma block size is 2 and only one side neighboring samples are available, only two samples are used to derive the linear model.

After obtaining the linear model parameters, an internal chroma prediction block is generated as,

$$\hat{Pred}_C(i,j) = \alpha \cdot Rec_L(i,j) + \beta, \tag{12}$$

where $Rec_L(i,j)$ is the actual reconstructed luma sample at position (i,j). Then, a six-tap filter (*i.e.* [1 2 1; 1 2 1]) is introduced for down-sampling the internal chroma prediction block,

$$\begin{aligned} Pred_C(i,j) = (&2 \cdot \hat{Pred}_C(2i,2j) + 2 \cdot \hat{Pred}_C(2i,2j+1) \\ &+ \hat{Pred}_C(2i-1,2j) + \hat{Pred}_C(2i-1,2j+1) \\ &+ \hat{Pred}_C(2i+1,2j) + \hat{Pred}_C(2i+1,2j+1) + 4) >> 3. \end{aligned} \tag{13}$$

However, for chroma samples locating at the left most column, the two-tap average filter is applied instead.

Regarding the signaling of TSCPM, we use one flag to signal whether the mode is TSCPM or not. This flag is coded right after the DM mode. The detailed bin strings for each chroma mode are tabulated in Table 2.

3. Experimental Results

The effectiveness of the proposed sub-sampled cross-component prediction is first validated on VTM-4.0 [10]. Then, we verify the performance of TSCPM on the latest AVS3 test model HPM-5.0 [16].

3.1. Evaluations on VTM-4.0

Experiments are first conducted on VTM-4.0 conforming to JVET common test conditions (CTC) [17]. All-intra (AI) and random access (RA) configurations are both

Table 2: Coding bins signaling with TSCPM in AVS3.

Mode Index	**Mode**	**Bin String**
0	DM	1
5	TSCPM	01
1	DC	001
2	Horizontal	0001
3	Vertical	00001
4	Bi-linear	00000

Table 3: Experimental Results on VTM-4.0 under AI and RA configurations

Class	AI			RA		
	Y	U	V	Y	U	V
A1	-0.17%	-0.16%	-0.46%	-0.16%	-0.06%	-0.40%
A2	-0.03%	-0.04%	-0.08%	-0.04%	0.08%	-0.06%
B	-0.02%	-0.59%	-0.70%	0.00%	-0.82%	-1.13%
C	-0.02%	-0.47%	-0.33%	-0.03%	-0.32%	-0.59%
E	-0.02%	-0.01%	-0.15%			
Overall	**-0.05%**	**-0.30%**	**-0.38%**	**-0.05%**	**-0.36%**	**-0.63%**
D	-0.01%	-0.21%	-0.34%	-0.02%	-0.18%	-0.32%
F	0.01%	-0.04%	-0.21%	0.00%	-0.09%	-0.21%
Enc Time	99%			100%		
Dec Time	100%			100%		

adopted. BD-Rate [18] is employed for evaluating the coding performance, where negative BD-Rate indicates the performance gains. The results are summarized in Table 3. It can be observed that the proposed sub-sampled cross-component prediction can slightly improve the coding performance for VTM-4.0, where 0.05%, 0.30% and 0.38% BD-Rate savings for Y, U and V components can be achieved, respectively under AI configuration. In addition, the coding performance under RA configuration is 0.05%, 0.36% and 0.63% for Y, U and V components. The largest gain comes from the 4K sequences in class A1. As such, the proposed sub-sampled cross-component prediction brings coding performance improvement, meanwhile largely simplify the derivation process of the linear model, leading to the adoption by VVC.

3.2. Evaluations on HPM-5.0

The performance of TSCPM is evaluated on latest AVS3 test model HPM-5.0 [16]. Experiments are conducted confirming to the AVS3 common test conditions (CTC) [19]. The AVS3 recommend test sequences are all involved in our simulation that cover different bit-depths (8bit, 10bit) and resolutions (4K, 1080p, 720p). AI and RA con-

Table 4: Performance of proposed TSCPM on AVS3 test model HPM-5.0

Seq		AI			RA		
		Y	U	V	Y	U	V
720p	City	0.01%	-0.43%	-0.28%	0.09%	-1.20%	-0.13%
	Crew	-0.15%	-5.26%	-4.87%	0.11%	-2.15%	-4.80%
	Vidyo1	-0.05%	-4.08%	-2.41%	0.11%	-3.78%	-2.63%
	Vidyo3	0.03%	-2.66%	-7.72%	-0.24%	-6.27%	-10.26%
1080p	BasketballDrive	-0.34%	-6.00%	-6.29%	-0.27%	-5.68%	-5.16%
	Cactus	-0.40%	-7.08%	-6.47%	-0.27%	-9.23%	-6.26%
	MarketPlace	-0.54%	-11.03%	-11.11%	-0.03%	-12.55%	-9.59%
	RitualDance	-0.30%	-7.90%	-13.55%	-0.18%	-4.41%	-10.17%
4k	Tango2	-1.56%	-22.48%	-24.06%	-0.67%	-20.75%	-20.65%
	Campfire	-7.74%	-32.29%	-32.34%	-4.43%	-16.80%	-21.59%
	ParkRunning3	-0.61%	-1.69%	-1.06%	-0.22%	-0.87%	-0.38%
	DaylightRoad2	-0.18%	-9.21%	-2.74%	-0.04%	-7.85%	-1.95%
720p		-0.04%	-3.11%	-3.82%	0.02%	-3.35%	-4.46%
1080p		-0.40%	-8.00%	-9.36%	-0.19%	-7.97%	-7.80%
4k		-2.52%	-16.42%	-15.05%	-1.34%	-11.57%	-11.14%
Average		**-0.99%**	**-9.18%**	**-9.41%**	**-0.50%**	**-7.63%**	**-7.80%**
Enc Time		100%			99%		
Dec Time		100%			99%		

figurations are adopted.

TSCPM significantly promotes the chroma prediction accuracy, leading to fewer encoding bits, which in turn contributes to the improvement of luma performance. Simulation results are tabulated in Table 4. We notice that TSCPM performs well for all classes, especially for 4K sequences. More specifically, TSCPM achieves on average 0.99%, 9.81% and 9.41% BD-Rate gain on Y, U and V components, respectively, under AI configuration. Meanwhile, on average 0.50%, 7.63% and 7.80% BD-Rate savings are achieved under RA configuration. Significant performance improvement is achieved on sequence “Campfire”, which contains drastic changes in lights and colors, such that traditional chroma intra prediction modes cannot well handle such scenarios.

In addition, we investigate the chroma CB sizes when cooperating with TSCPM prediction. Larger CBs tend to consume less block-level coding bins, leading to better compression performance. The proportions of different CB areas with and without TSCPM are counted at the decoder side. The results are presented in Fig. 4, where p is derived according to

$$p = \log_2(W \times H). \tag{14}$$

According to the statistical results, TSCPM significantly decreases the percentages of smaller CBs and increases the proportions of larger CBs.

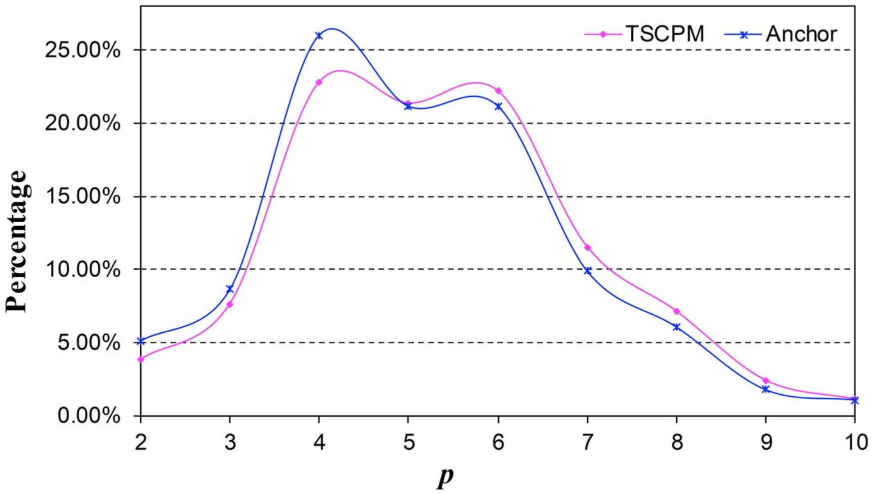


Figure 4: Statistical analysis of chroma CB size with and without TSCPM.

5. Conclusions

In this paper, we have presented an efficient sub-sampled cross component prediction method, with which the linear model parameters can be derived effectively with at most four neighboring sample pairs. The proposed method strikes an excellent tradeoff between the operation complexity and coding performance. Moreover, it has been adopted in VVC and AVS3 owing to the great benefit regarding the hardware and software implementations. Simulations on VTM-4.0 reportedly show that on average 0.05%, 0.30%, 0.38% BD-Rate savings for Y, U and V components can be achieved under AI configuration. Meanwhile 0.05%, 0.36% and 0.63% BD-Rate savings can be observed for Y, U and V components under RA configuration. Moreover, the TSCPM chroma coding method in AVS3, which is designed based on the sub-sampled cross component prediction strategy, brings 0.99%, 9.18% and 9.41% BD-Rate gains under AI configuration, and 0.50%, 7.63% and 7.80% for Y, U and V components under RA configuration.

Acknowledgment

This work was supported by High-performance Computing Platform of Peking University, National Key Research and Development Project (2019YFF0302703), MoE-China Mobile Research Fund Project (MCM20180702), which are gratefully acknowledged. This work was done while the author was a research intern in ByteDance Inc.

References

[1] J. Kim, S.-W. Park, J.-Y. Park, and B.-M. Jeon, "Intra chroma prediction using inter channel correlation," *JCTVC-B021*, Jul. 2010.

[2] J. Chen and V. Seregin, "Chroma intra prediction by scaled luma samples using integer operations," *JCTVC-C206*, Oct. 2010.

[3] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[4] K. Zhang, J. Chen, L. Zhang, X. Li, and M. Karczewicz, "Multi-model based cross-component linear model chroma intra-prediction for video coding," in *Visual Communications and Image Processing (VCIP), 2017 IEEE*. IEEE, 2017, pp. 1–4.

[5] T. Zhang, X. Fan, D. Zhao, and W. Gao, "Improving chroma intra prediction for hevc," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–6.

[6] Y. Li, L. Li, Z. Li, J. Yang, N. Xu, D. Liu, and H. Li, "A hybrid neural network for chroma intra prediction," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1797–1801.

[7] X. Zhang, C. Gisquet, E. Francois, F. Zou, and O. C Au, "Chroma intra prediction based on inter-channel correlation for HEVC," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 274–286, 2014.

[8] K. Zhang, J. Chen, L. Zhang, X. Li, and M. Karczewicz, "Enhanced cross-component linear model for chroma intra-prediction in video coding," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3983–3997, 2018.

[9] G. Larche, J. Taquet, C. Gisquet, and P. Onno, "CE3-5.1: On cross-component linear model simplification," *JVET-L0191*, Oct. 2018.

[10] "VVC software VTM-4.0," `https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-4.0`.

[11] M. Wang, K. Zhang, L. Zhang, H. Liu, J. Xu, S. Wang, J. Li, S. Wang, and W. Gao, "CE3-related: Modified linear model derivation for cclm modes," *JVET-M0274*, Jan. 2019.

[12] J. Huo, Y. Ma, S. Wan, Y. Yu, M. Wang, K. Zhang, L. Zhang, H. Liu, J. Xu, S. Wang, J. Li, S. Wang, and W. Gao, "CE3-1.5: CCLM derived with four neighbouring samples," *JVET-N0271*, Mar. 2019.

[13] J. Li, M. Wang, L. Zhang, K. Zhang, H. Liu, J. Xu, Y. Wang, P. Zhao, D. Hong, and S. Wang, "Chroma coding with two-step cross-component prediction," *AVS-M4632*, Dec. 2018.

[14] J. Li, M. Wang, L. Zhang, K. Zhang, H. Liu, J. Xu, Y. Wang, P. Zhao, D. Hong, and S. Wang, "A simplified LUT for TSCPM," *AVS-M4658*, Jan. 2019.

[15] J. Li, M. Wang, L. Zhang, K. Zhang, H. Liu, J. Xu, Y. Wang, and S. Wang, "Modified comparison logic for TSCPM," *AVS-M4726*, Mar. 2019.

[16] "AVS3 software repository," `/Public/codec/video_codec/HPM/HPM-5.0`.

[17] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Suhring, "JVET common test conditions and software reference configurations for SDR video," *Joint Video Exploration Team (JVET), doc. JVET-M1010*, 2019.

[18] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T SG 16 Q.6 VCEG-M33*, 2001.

[19] J. Chen and K. Fan, "AVS3-P2 common test conditions v8.0," *AVS-N2727*, Sep. 2019.