A Discrete-Mapping-Based Cross-Component Prediction Paradigm for Screen Content Coding

Bharath Vishwanath, *Member, IEEE*, Kai Zhang, *Senior Member, IEEE*, and Li Zhang, *Senior Member, IEEE*

***Abstract*— Cross-component prediction is an important intra-prediction tool in the modern video coders. Existing prediction methods to exploit cross-component correlation include cross-component linear model and its extension of multi-model linear model. These models are designed for camera captured content. For screen content coding, where videos exhibit different signal characteristics, a cross-component prediction model tailored to their characteristics is desirable. As a pioneering work, we propose a *discrete-mapping based cross-component prediction model* for screen content coding. Our model relies on the core observation that, screen content videos typically comprise of regions with *a few distinct colors* and luma value (almost always) uniquely conveys chroma value. Based on this, the proposed method learns a discrete-mapping function from available reconstructed luma-chroma pairs and uses this function to derive chroma prediction from the co-located luma samples. To achieve higher accuracy, a multi-filter approach is employed to derive co-located luma values. The proposed method achieves 2.61%, 3.51% and 3.92% Y, U and V bit-rate savings respectively over Enhanced Compression Model (ECM) 4.0, with negligible complexity, for text and graphics media under all-intra configuration.**

***Index Terms*— Cross-component prediction, screen content coding, ECM, VVC.**

I. Introduction

SCREEN content media refers to non-camera captured sequences with the content dominated by texts and graphics. Recently, there is a rapid growth in the consumption of screen content media due to increased prevalence of scenarios like online education, virtual meetings and conferences. Since screen content serves as a backbone for many vital applications, there is an obvious need for efficient compression algorithms tailored to their signal characteristics. Realizing this, the ISO/IEC Moving Picture Expert Group and the ITU-T Video Coding Experts Group, also known as "Joint Collaborative Team on Video Coding" (JCTVC) developed a screen content coding (SCC) extension for high efficiency video coding (HEVC) [1], [2]. Further, the Joint Video Experts Team (JVET) developed low-level coding tools specifically for screen content videos and included them in the main profile of versatile video coding (VVC) [3], [4], highlighting the importance of screen content coding. The JVET group is now studying the potential need for standardization of future video coding technology with a compression capability that significantly exceeds that of the current VVC standard under the name Enhanced Compression Model (ECM). This standardization action, which began in 2021, could either take the form of additional extension(s) of VVC or an entirely new standard. Similar to VVC, screen content coding continues to be an important aspect in the development of ECM.

A notable prediction tool in the modern video codecs is the cross-component prediction [5], [6], [7]. To elaborate, for videos with multiple channels, one often observes correlations across different channels. To exploit this, early works in [8] and [9] introduced linear prediction models for RGB 4:4:4 format. Following this, a series of works [10], [11], [12] led to a cross-component linear model (CCLM), which is a part of VVC and ECM. In CCLM, as the names suggests, the correlation between luma and chroma component is modeled by a simple linear model. Realizing that the cross-component relationship could be more complex, multi-model linear model (MMLM) was proposed in [13]. In MMLM, the cross-component correlation is captured by segmented regression or piece-wise linear regression. Considering the significant gains it offers, MMLM was adopted in ECM. It is to be noted that for CCLM and MMLM, the model parameters are learnt by using the reconstructed block boundary samples as the training set of samples, mitigating the need to explicitly send the model parameters to the decoder.

Screen content videos are known to exhibit different signal characteristics as compared to camera captured content. Thus, specialized coding tools have been developed for SCC. For instance, repeated patterns are often observed, which is exploited in intra block copy [14], [15]; regions with few distinct colors are observed, which is exploited in palette coding [2], [16], [17]. Other notable tools include adaptive color transform (ACT) [18] and adaptive motion vector resolution (AMVR) [19], [20]. Since screen content videos exhibit different signal characteristics, predictors and transforms designed for natural sequences might not be optimal for them. For instance, a staircase transform was shown to outperform trigonometric transforms for SCC in [21]. With a similar rationale, we question the optimality of the existing cross-component prediction models for SCC. To this end, we did a thorough analysis, where we observed that for a large number of blocks, luma and chroma are largely

Manuscript received 14 April 2023; revised 3 October 2023; accepted 13 November 2023. Date of publication 29 November 2023; date of current version 4 December 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alessandro Gnutti. *(Corresponding author: Bharath Vishwanath.)*

The authors are with ByteDance Inc., San Diego, CA 92121 USA (e-mail: bharath.vishwanath@bytedance.com; zhangkai.video@bytedance.com; lizhang.idm@bytedance.com).

Digital Object Identifier 10.1109/TIP.2023.3334970

uncorrelated which renders existing linear models sub-optimal. However, an important observation for these blocks is that they typically comprise of a few distinct colors and luma uniquely conveys chroma value. Both CCLM and MMLM fail to capture this observation. This motivates for a new cross-component prediction model for SCC.

In the light of our observation, we realize that an appropriate model for SCC is a *discrete-mapping model* that simply maps the luma values to their corresponding chroma values. This is in sharp contrast to the existing methods that employ continuous models and perform regression to derive model parameters. To derive prediction for a given chroma block, we learn the mapping function from the reconstructed neighbors and obtain prediction from the co-located luma samples and the learnt mapping function. We emphasise that the proposed method gives good prediction despite luma and chroma being (largely) uncorrelated, a scenario where achieving good prediction is difficult and for which the existing methods perform very poorly. For 4:2:0 sequences, the efficacy of the proposed method depends on the down-sampling filter used. To achieve adaptivity, we allow multiple down-sampling filters and convey the best filter to the decoder. Substantial gains in experiments validate the efficacy of the proposed model.

We note that the current journal subsumes our previous work in [22]. In our previous work, the discrete mapping model was presented as an improvement to the palette coding. However, we realize that the model can be an independent cross-component prediction method and not necessarily depend on palette coding. Thus, the model is presented as a new cross-component prediction method, independent of palette coding. The method shows substantial gains despite the presence or absence palette mode, demonstrating our claim. Compared to our previous work, we extend the method to include a larger neighboring area to derive mapping function, which results in substantial improvement in coding performance. More importantly, we present methods to reduce computational complexity, that brings down the increase in encoding and decoding complexity from over 7% to at most 1% increase in encoding and decoding complexity. New simulation results are provided that demonstrate the utility of the model in low bit-rate regime (corresponding to luma PSNR range of 30-40 dB). Additionally, statistical investigations are presented that validate the efficacy of the proposed model and the multiple down-sampling filters. Further, we provide an in-depth discussion of observations in the simulations.

The rest of the paper is organized as follows. Section II provides the relevant background. Section III introduces our observation, the proposed discrete-mapping model and the prediction method as implemented in ECM. Section IV discusses various methods to reduce computational complexity. Experimental results are presented in section V, followed by conclusions in section VI.

II. Relevant Background

In this section, we introduce cross-component prediction and palette coding in ECM.

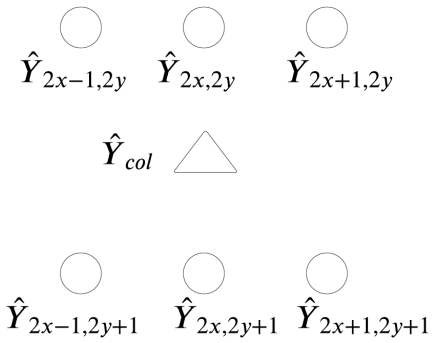


Fig. 1. Luma down-sampling position, marked triangle, in ECM. The samples in original resolution are marked by circles.

A. Cross-Component Prediction in ECM

1) Models: To exploit the correlation between luma and chroma components, ECM allows for the following two prediction models:

- *Cross-Component Linear Model (CCLM)*: In CCLM, chroma samples are predicted from co-located luma samples as,

$$\tilde{C} = \alpha \hat{Y}_{col} + \beta \tag{1}$$

  where $\tilde{C}$ is the chroma prediction, $\hat{Y}_{col}$ is the co-located luma value and $\{\alpha, \beta\}$ are the model parameters.
- *Multi-model Linear Model (CCLM)* In ECM, in addition to CCLM, a multi-model linear model (MMLM) is introduced. In MMLM, luma values are classified to obtain clusters and separate linear models with parameters $\{\alpha_i, \beta_i\}$ are employed for each cluster i. To keep the design simple, number of clusters is chosen to be two. The two clusters are obtained by using the average luma value as the classification boundary. Please refer to [23] for more details.

2) Luma Down-Sampling: For the case of 4:2:0 sequences, for both CCLM and MMLM, to obtain co-located luma sample for chroma prediction, a fixed six-tap down-sampling filter is used to generate co-located luma as,

$$\begin{aligned}\hat{Y}_{col} = \{&2\hat{Y}_{2x,2y} + 2\hat{Y}_{2x,2y+1} + \hat{Y}_{2x-1,2y} \\ &+ \hat{Y}_{2x+1,2y} + \hat{Y}_{2x-1,2y+1} + \hat{Y}_{2x+1,2y+1}\} \gg 3\end{aligned} \tag{2}$$

Fig. 1 shows the six adjacent luma samples (marked as circles) involved in the filter to generate the down-sampled luma sample (marked as a triangle).

3) Model Parameter Estimation: Cross-component correlation is a localized phenomenon, in that, it is observed in certain blocks/regions of a video. Further, different blocks that benefit from cross-component prediction, prefer different model parameters. One option to account for this would be to determine the best parameters for a block at the encoder and convey it to the decoder. However, this would incur significant cost to convey the parameters. To alleviate this, the model parameters are derived at both encoder and decoder by using the already reconstructed boundary samples as the training set of samples, under the assumption that the samples of the current block are highly correlated with its reconstructed neighbors. With the boundary samples as the training samples,

Index	Y	Cb	Cr
0	225	25	30
1	62	128	100
2	210	0	50
3	2	12	180

(a)

0	0	0	1	1	1	1	1
0	0	0	0	2	2	2	1
0	0	2	2	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	1	2	2	2
1	1	1	1	1	2	2	2
1	1	0	0	3	3	3	0
0	0	0	0	0	0	0	5

(b)

Fig. 2. Illustration of palette coding: (a) Palette used for a coding unit; (b) Index map for the coding unit.

the model parameters are derived to minimize the squared-error distortion.

4) Multi-Directional Linear Model: Samples of the current block could at times be more correlated with the samples from top boundary and less correlated with the left boundary or vice-versa. To exploit this observation, ECM allows for multi-directional linear models (MDLM) [23], [24]. In MDLM-T, only top boundary samples are used for model parameter derivation. In MDLM-L, only left samples are used and in MDLM-TL, both top and left samples are used. The idea of MDLM applies for both CCLM and MMLM. Accordingly, ECM has six prediction modes: {CCLM-T, CCLM-L, CCLM-TL, MMLM-T, MMLM-L, MMLM-TL}. Note that, to get more training samples in MDLM-L and MDLM-T, the top and left reconstructed boundaries are extended respectively, when the extended boundary reconstructions are available.

B. Palette Coding

A notable characteristic of screen content videos is that they typically comprise of regions with a few distinct colors. To exploit this observation in terms of compression, a special tool named palette coding is designed [2], [16], [17]. Palette coding is a pure quantization and entropy coding based coding method. First, a table of representative colors named palette is coded. Palette is thus essentially a quantization codebook. Next, each input sample in the coding unit is quantized to the palette entries to derive an index map. The index map is efficiently compressed and conveyed to the decoder. Fig. 2 depicts the idea of palette coding.

In the initial design of palette coding in HEVC, palette consisted of color *triplets* as entries. However, VVC and ECM allow for dual-tree structure, in which luma and chroma components can have independent partitioning for intra slices. The dual-tree structure obviously conflicts with the initial design of palette coding. As a remedy, a simple approach of using separate palette for luma and chroma is currently used in VVC and ECM [17], [25]. However, such a simple design neglects the underlying cross-component correlations. The proposed model bridges this gap leading to considerable coding gains as will be demonstrated in the experiments.

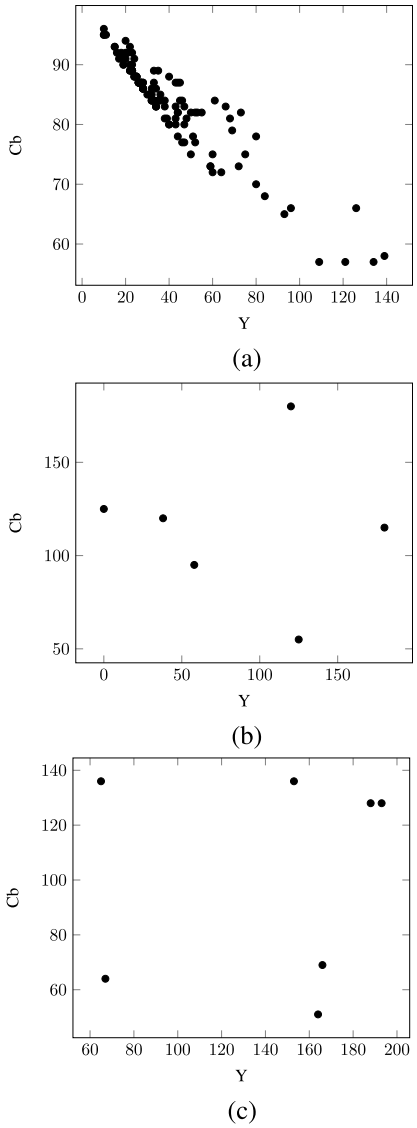


Fig. 3. Plots of Y-Cb samples extracted from (a) a block of camera captured sequence; (b) and (c) blocks of screen content sequences.

III. Proposed Method

In this section, we first introduce our observations in screen content sequences that pave way to the discrete-mapping model. We then introduce the new cross-component prediction method that we employ in ECM.

A. Observation

The main motivation for our work stems from questioning the optimality of existing cross-component prediction methods for screen content coding. In the quest of this, we did a thorough analysis and to illustrate it, we consider some example plots as shown in Fig. 3. In Fig. 3a, we show a plot of Y-Cb samples taken from a block of a camera-captured sequence which exhibits cross-component correlation. (Of

course, a lot of other blocks in camera-captured content don't exhibit correlation. We have picked a block that potentially benefits from cross-component prediction). In contrast to this, we consider example plots of Y-Cb samples taken from blocks of screen content sequences as shown in Fig. 3b and 3c. The kind of plots shown in Fig. 3b and 3c were frequently observed and we have considered these two plots as representative plots for illustration. (Statistics are presented in the experiments that demonstrate that these blocks are frequently encountered). For these plots, we observe that the luma-chroma are (largely) uncorrelated, implying very poor prediction from the existing linear prediction methods. One could try to fit higher order models, but this comes at a cost of increased complexity and still not necessarily provide good prediction. However, a careful observation of these plots reveals that, compared to the luma-chroma plots for natural sequences, screen content sequences have the following peculiar characteristics: *i)* the block comprises of a *few distinct colors* and that *each point corresponds to a distinct color* in the current block. *ii)* the *luma value uniquely determines the chroma value*. Since the blocks comprise of a few distinct colors, instead of learning continuous models, one simply needs to know the mapping between these few luma and chroma values to achieve optimal prediction. This naturally leads to a *discrete-mapping model* that maps luma values to their corresponding chroma values. Chroma prediction can then be obtained from the mapping function and the co-located luma.

An obvious hurdle is to convey this mapping function to the decoder. Conveying this mapping function can prove very costly and can easily nullify the benefits of the model, especially at low-rates. To this end, we leverage an additional observation that the set of colors in the current block are mostly observed in the reconstructed neighbors. Thus, instead of explicitly conveying the mapping function to the decoder, both encoder and decoder learn the mapping function from the reconstructed samples.

We note that for a given set of training samples, discrete-mapping model is an extreme case of over-fitting. It is interesting to note that, an extremely over-fitted model is the preferred model for cross-component prediction in screen content coding. To elaborate, camera captured content typically has noise, including both acquisition noise and the innovation term involved in modeling the spatial evolution of pixels. This limits one to employ simple linear models to avoid the fear of over-fitting. However, screen content has very low level of noise. We could thus benefit from an over-fitted model.

Stitching the above observations, we next propose our cross-component prediction paradigm for SCC.

B. Discrete-Mapping Model Based Cross-Component Prediction

Our observations in the previous section lead to a cross-component prediction method that (a) simply 'remembers' the luma and chroma values of reconstructed neighbors as a discrete-mapping function (b) while coding the current chroma block, derives the chroma prediction from the mapping function based on the co-located luma sample value.

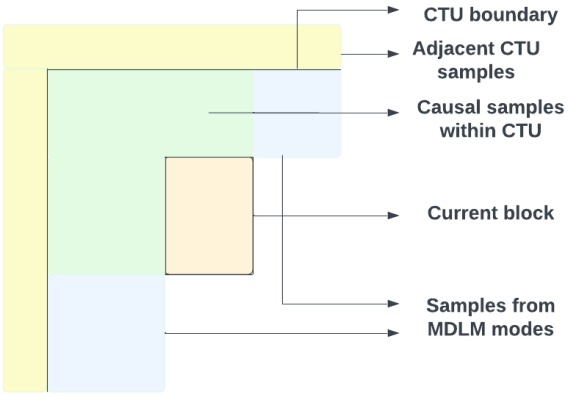


Fig. 4. Samples used to derive discrete-mapping function.

As an algorithm, the proposed method consists of the following three steps to obtain chroma prediction for a block:

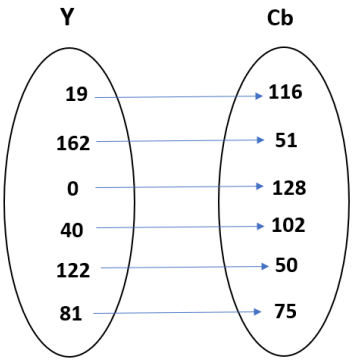
1) Learning Discrete-Mapping Function: We learn the mapping function from the reconstructed neighbors. The derivation simply involves remembering the already reconstructed neighboring luma-chroma sample pairs as a mapping function. An example discrete-mapping function (for one chroma channel) is shown in Fig. 5a. Additionally, we keep average of the chroma values from the reconstructed neighbors handy, whose utility will be clear shortly. Note that, since the mapping function is derived from reconstructed neighbors, there is no need to convey any side-information to the decoder about the mapping function. Also, the cardinality of the domain of the mapping function is essentially the number of distinct colors observed in the neighborhood.

The coding efficiency of the proposed method leans heavily on having a rich set of samples in the mapping function, an observation that we note from our previous work in [22]. Thus, to include a rich set of colors in the mapping function, we derive the function from reconstructed samples in the following regions, which are illustrated in Fig. 4:

- All causal samples within the current coding tree unit (CTU)
- Region spanned by top and left eight chroma reference lines in the neighboring CTUs
- Non-causal samples used by multi-directional linear model (MDLM) modes in ECM and their extensions as shown in Fig. 4. (With the rows and columns indexed by i and j respectively, if $i = M$ and $j = N$ respectively correspond to the bottom and right boundary of a block, then samples with $i > M$ or $j > N$ correspond to non-casual samples.)

The reconstructed samples shown in Fig. 4 are processed in raster scan order. Based on our experiments, the order of processing has negligible impact on the R-D performance. Further, while scanning the region, mapping table entries are overwritten if newer entries are found. Note that each block derives it own table from its reconstructed neighbors. The table is neither propagated to a future block, nor dependent on the table from a previously coded block. Thus, there is no need to borrow or reset a table.

2) Chroma Mapping: For a chroma sample in the current block, we fetch the co-located luma sample value, denoted as



(a) Learnt mapping function from reconstructed neighbors

0	122	122	40	81	40	40	81
0	122	122	40	81	40	40	81
0	122	122	40	81	40	40	81
0	122	122	40	81	80	80	81
0	122	122	40	81	119	119	81
0	122	122	40	81	119	119	81
0	122	122	40	81	119	119	81
0	122	122	40	81	119	119	81

(b) Co-located luma block

128	50	50	102	75	102	102	75
128	50	50	102	75	102	102	75
128	50	50	102	75	102	102	75
128	50	50	102	75	75	75	75
128	50	50	102	75	50	50	75
128	50	50	102	75	50	50	75
128	50	50	102	75	50	50	75
128	50	50	102	75	50	50	75

(c) Mapped chroma values

Fig. 5. Demonstration of the proposed method.

$\hat{Y}_{col}$, and search for it in the domain of the mapping function. If $\hat{Y}_{col}$ cannot be found in the domain, we search in the vicinity, i.e, we search, in order, for values $\hat{Y}_{col} \pm 1$, $\hat{Y}_{col} \pm 2$ and $\hat{Y}_{col} \pm 3$ in the domain of the mapping function. The matched luma value in the domain will be used to derive the chroma mapping. If none of these values are found in the domain, implying a new color in the block not observed before, we set the average of the chroma values in neighborhood mentioned earlier as the mapping chroma sample value. To demonstrate this step, we draw reader's attention to Fig. 5. In the co-located luma block in 5b, for luma value of 40, we find a match in the domain of the mapping function in 5a, for which the chroma mapping is 102. However, for the luma value 119, we don't find a matching value in the domain of the mapping function. We thus search in the vicinity to find a matching value of 122, for which the chroma mapping is 50.

3) Chroma Prediction: For each chroma sample, the mapped chroma sample value is its predicted value. We include this step to note that we don't perform any further processing after chroma mapping.

The chroma prediction residual is transformed, quantized and entropy coded, similar to other cross-component prediction modes in ECM.

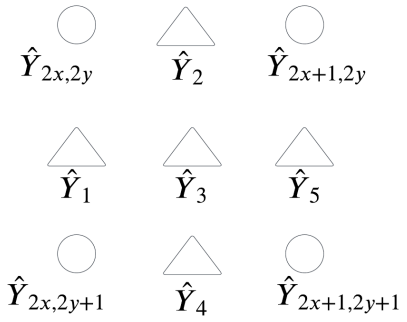


Fig. 6. Luma down-sampling positions.

TABLE I
LUMA DOWN-SAMPLING FILTERS

Luma position	Filter
$\hat{Y}_1$	$(\hat{Y}_{2x,2y} + \hat{Y}_{2x,2y+1}) >> 1$
$\hat{Y}_2$	$(\hat{Y}_{2x,2y} + \hat{Y}_{2x+1,2y}) >> 1$
$\hat{Y}_3$	$(\hat{Y}_{2x,2y} + \hat{Y}_{2x+1,2y} + \hat{Y}_{2x,2y+1} + Y_{2x+1,2y+1}) >> 2$
$\hat{Y}_4$	$(\hat{Y}_{2x,2y+1} + Y_{2x+1,2y+1}) >> 1$
$\hat{Y}_5$	$(\hat{Y}_{2x+1,2y} + \hat{Y}_{2x+1,2y+1}) >> 1$

C. Multi-Filter Discrete-Mapping Method

A critical aspect that dictates the efficacy of the proposed method is to be able to find a matching luma value in the domain of the derived mapping function for a given co-located luma value. For 4:2:0 sequences, this correspondence directly depends on the derivation of the co-located luma-sample value. As mentioned in Section II, ECM employs a fixed down-sampling filter which might not be always optimal. Optimal down-sampling filter for a block depends on the direction of the edges, gradients and other factors. A similar observation was made in [13] and it was demonstrated that, among a set of down-sampling filters, the default down-sampling filter in VVC/ECM was seldom the most dominant filter. Realizing the critical need for having flexibility in the down-sampling process to achieve accurate prediction for the proposed model, we employ a set of down-sampling filters and select the best down-sampling per CU. Specifically, we allow five additional down-sampling filter positions as shown in Fig. 6, whose corresponding down-sampling filters are listed in Table I. Details of the encoder workflow for selection of the best filter will be discussed in the experimental results section.

IV. A Note on Complexity Reduction

In this section, we discuss various optimizations that we performed in implementing our method. The implementation and optimizations for each step is as follows:

A. Discrete-Mapping Derivation

In our previous work [22], we included at most sixteen chroma reference lines for deriving the mapping function.

This was due to a drastic increase in the complexity as we increased the number of chroma reference lines. This obviously motivated us to explore ways to reduce complexity to derive the mapping function.

The discrete-mapping function was initially implemented as an unordered-map, where the chroma values were stored with corresponding luma value as the key. We note that in our preliminary results in [22], we derived the discrete-mapping based on the unordered-map. Based on our analysis, searching the luma value in the unordered map was accounting for increased complexity. To overcome this, we took an array-based approach for discrete-mapping derivation. To elaborate, for luma-value at bit-depth B, we consider a 1 D array of size 2^B to store chroma values (per-chroma channel), such that the corresponding luma values are simply the indices of the array. For luma values that are found in the reconstructed neighborhood, the array value indexed by luma value will be the corresponding chroma value. The rest of the array will be marked invalid. To draw differences between the two approaches, in unordered-map based implementation, the number of entries in the table is the same as the number of distinct luma-chroma pairs observed in the reconstructed neighbors, making its memory requirement to be less than an array-based approach. However, this memory optimization comes at the cost of increased computational complexity. Although in comparison, the array based approach needs more memory, the actual required memory is still negligible. Thus, we chose an array-based implementation to achieve less computational complexity.

The workflow in ECM is to process Cb and Cr blocks sequentially. Thus, while deriving the Y-Cb and Y-Cr mapping functions, in our initial implementation in [22], we had to scan the reconstructed neighbors twice. We remove this redundancy and derive both the mapping functions from a single scan of reconstructed neighbors.

The above optimizations significantly reduce the complexity and opens door to include all the reconstructed samples introduced previously during mapping function derivation, enabling us to achieve high coding gains at a negligible increase in complexity.

B. Chroma Prediction

To derive the chroma prediction for a sample in the current block, we need to look for the nearest match for the co-located luma sample in the domain of the mapping function. Based on our profiling, this search could be time-consuming. As mentioned before, in ECM, the Cb and Cr prediction is done sequentially. We observe that the search process of locating nearest entry for the co-located luma sample in the domain of the mapping function is *identical* for both the chroma channels. Thus, to avoid redundant search, while predicting Cb, we store the matching domain values for each co-located luma value and reuse it while predicting Cr.

Note that all the optimizations presented above are lossless and doesn't impact R-D performance.

TABLE II
BINARIZATION SCHEME FOR THE FILTER INDEX

Filter Index	Binarization
0	00
1	01
2	100
3	101
4	110
5	111

V. EXPERIMENTAL RESULTS

The proposed method is implemented and tested with ECM 4.0 [26]. ECM has multiple cross-component prediction modes and the proposed method is implemented as an additional prediction mode. If the cross-component prediction flag for a coding unit (CU) is equal to one, an additional (context or regular coded [27]) flag is signalled to indicate whether the proposed model is used. Recall that the mapping function is derived from reconstructed neighbors available to both encoder and decoder, thereby mitigating the need to send any information related to the mapping function to the decoder. For 4:2:0 sequences, if the proposed method is chosen, additional flags are signaled to indicate the luma down-sampling filter chosen. The binarization scheme for the filter index is presented in Table II. Filter index 0 corresponds to the default filter in ECM. Filter indices 1-5 correspond to the filters listed in Table I. For the bins of the filter index, the first bin is context (regular) coded and other bins are bypass coded. (We did try other possible options for binarization and coding. The impact was rather minimal.) From the encoder perspective, the encoder first determines the best down-sampling filter based on sum of absolute differences (SAD) criterion, i.e., without any RD cost computation. RD cost is then computed using the best filter and compared with other prediction modes to determine the best coding mode.

As mentioned earlier, we derive the discrete-mapping from the reconstructed neighbors. In ECM, internal bit-depth is set to 10, meaning, the reconstructed luma and chroma neighbors are stored at 10-bit precision. For the luma value in discrete-mapping, based on our simulations, changing luma bit-depth to 8 bits led to a slightly improved coding time with almost no impact on compression performance. Considering this, and also keeping an eye on the memory of the mapping function, the luma in the mapping function is stored at 8-bit precision. Chroma is stored at 10-bit precision. An example mapping function is shown in Fig. 7

Experiments were conducted under the test conditions specified in [28]. Bit-rate reduction is calculated as per [29]. We present results for class F and class G, also known as class TGM (text and graphics with motion) sequences. Class TGM mainly comprises of screen content media whereas a class F sequence is a mixture of screen content and natural content. All sequences are in 4:2:0 format. As noted in the introduction, our method was presented as an improvement to palette coding in [22]. Realizing that the method need not necessarily depend on palette coding, we implemented the method as an independent cross-component prediction mode.

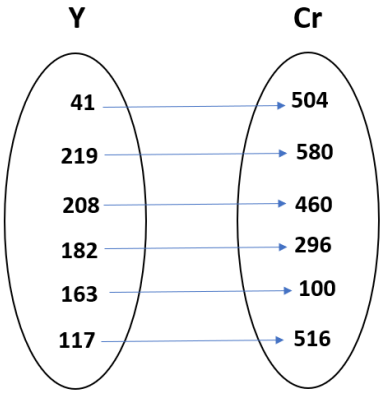


Fig. 7. An example luma-chroma mapping.

TABLE III

BD-RATE GAINS IN % OVER ECM 4.0 FOR AI CONFIGURATION (PALETTE CODING IS DISABLED IN ANCHOR AND THE PROPOSED METHOD)

Class	Sequence	Y	U	V
F	BasketballDrillText	-0.11	-0.23	-0.28
	ArenaofValor	-0.04	-0.13	-0.06
	SlideEditing	-0.51	-1.28	-1.03
	SlideShow	-0.13	0.30	-0.31
	Average	**-0.2**	**-0.34**	**-0.42**
	Δ **EncT**: 1%			
	Δ **DecT**: 1%			
G	FlyingGraphic	-2.24	-3.33	-4.04
	Desktop	-4.45	-5.25	-6.66
	Console	-3.35	-4.86	-4.53
	ChineseEditing	-0.40	-0.61	-0.46
	Average	**-2.61**	**-3.51**	**-3.92**
	Δ **EncT**: 1%			
	Δ **DecT**: 1%			

To demonstrate that the method offers coding gains under both the scenarios, let us consider:

A. Performance Without Palette Coding in Anchor and the Proposed Method

The average bit-rate savings for all-intra (AI) and randomaccess configurations (RA) for this scenario are shown in Tables III and IV respectively. Evidently, the proposed method brings significant coding gains with an average of 2.61% luma bit-rate reduction and up to 4.45% luma bit-rate reduction under AI configuration for TGM sequences. RD curves in Fig. 8 (see next page) illustrates consistent performance across all bit-rates.

B. Performance With Palette Coding in Anchor and the Proposed Method

The average bit-rate savings for AI and RA configurations for this scenario are shown in Tables V and VI respectively. Although there is a dip in the compression performance compared to the previous scenario, the method still brings significant coding gains with an average of 2.34% luma bit-rate reduction and up to 3.69% luma bit-rate reduction under AI configuration for TGM sequences. Note that the motivation for our model bears similarity with the principle of palette coding. The reason that we still have significant gains is as follows: as mentioned earlier, VVC and ECM allow for dual-tree structure, where luma and chroma can have independent partitioning, necessitating to have separate palette for luma and chroma. This results in a gap between luma and chroma coding, that is bridged by the proposed model. Thus, the proposed model supplements palette coding resulting in the observed compression gains.

TABLE IV

BD-RATE GAINS IN % OVER ECM 4.0 FOR RA CONFIGURATION (PALETTE CODING IS DISABLED IN ANCHOR AND THE PROPOSED METHOD)

Class	Sequence	Y	U	V
F	BasketballDrillText	0.04	-0.32	-0.39
	ArenaofValor	-0.02	-0.14	0.03
	SlideEditing	-1.30	-3.00	-2.65
	SlideShow	-0.17	-1.12	-0.93
	Average	**-0.36**	**-1.14**	**-0.99**
	Δ **EncT**: 1%			
	Δ **DecT**: 0%			
G	FlyingGraphic	-0.37	-0.84	-0.99
	Desktop	-3.17	. -3.54	-4.28
	Console	-0.96	-1.13	-1.42
	ChineseEditing	-0.51	-0.92	-0.63
	Average	**-1.25**	**-1.61**	**-1.83**
	Δ **EncT**: 0%			
	Δ **DecT**: 0%			

TABLE V

BD-RATE GAINS IN % OVER ECM 4.0 FOR AI CONFIGURATION (PALETTE CODING IS ENABLED IN ANCHOR AND THE PROPOSED METHOD)

Class	Sequence	Y	U	V
F	BasketballDrillText	-0.07	-0.37	-0.49
	ArenaofValor	-0.02	-0.02	-0.03
	SlideEditing	-0.40	-1.40	-1.16
	SlideShow	-0.04	-0.60	-0.72
	Average	**-0.13**	**-0.60**	**-0.60**
	Δ **EncT**: 1%			
	Δ **DecT**: 0%			
G	FlyingGraphic	-2.22	-3.11	-3.37
	Desktop	-3.69	-2.75	-3.98
	Console	-2.95	-2.27	-3.38
	ChineseEditing	-0.50	-0.52	-0.41
	Average	**-2.34**	**-2.17**	**-2.87**
	Δ **EncT**: 1%			
	Δ **DecT**: 1%			

TABLE VI

BD-RATE GAINS IN % OVER ECM 4.0 FOR RA CONFIGURATION (PALETTE CODING IS ENABLED IN ANCHOR AND THE PROPOSED METHOD)

Class	Sequence	Y	U	V
F	BasketballDrillText	-0.08	-0.20	-0.34
	ArenaofValor	-0.02	-0.03	-0.06
	SlideEditing	-1.05	-3.13	-2.68
	SlideShow	-0.18	-0.05	-0.62
	Average	**-0.33**	**-0.85**	**-0.90**
	Δ **EncT**: 1%			
	Δ **DecT**: 1%			
G	FlyingGraphic	-0.24	-0.67	-0.87
	Desktop	-1.95	-1.77	-2.33
	Console	-0.57	-0.20	-0.84
	ChineseEditing	-0.47	-0.79	-0.68
	Average	**-0.81**	**-0.86**	**-1.18**
	Δ **EncT**: 1%			
	Δ **DecT**: 1%			

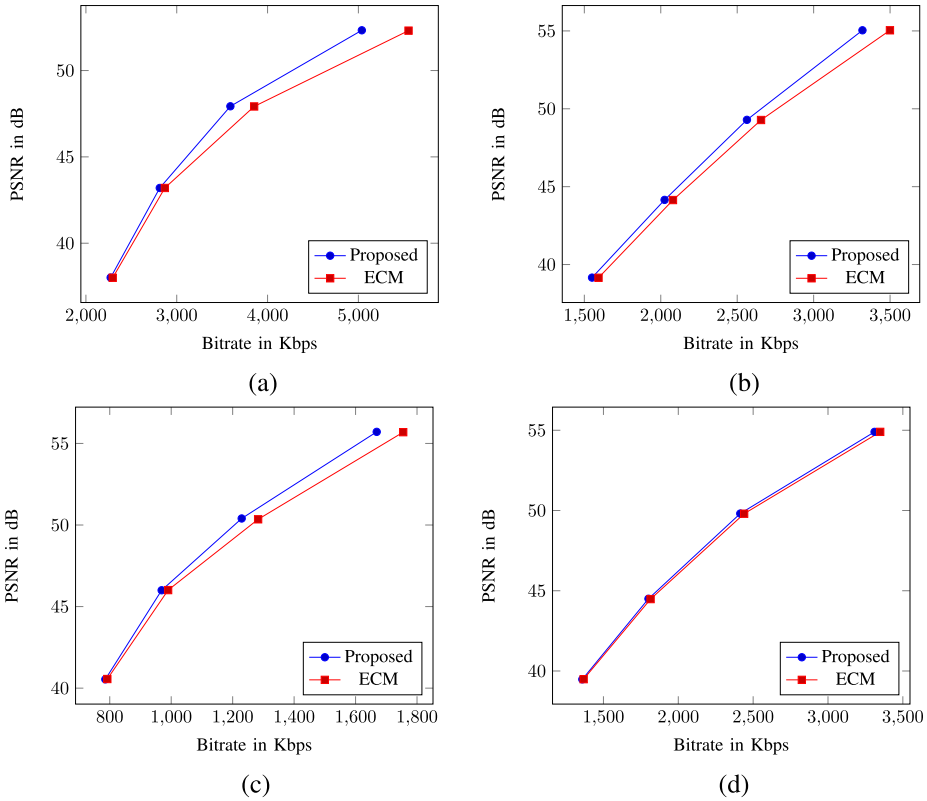


Fig. 8. RD curves for: (a) Desktop, (b) Console sequences for AI configuration and (c) Desktop, (d) Console sequences for RA configuration for the QP set {22, 27, 32, 37} (palette coding is disabled in both anchor and the proposed method).

TABLE VII

PERFORMANCE WITH VARYING NUMBER OF REFERENCE LINES FOR LOOK-UP TABLE DERIVATION FOR CLASS-G UNDER AI CONFIGURATION

Number of chroma ref lines	Y (%)	U (%)	V (%)
1	-0.45	-0.53	-0.71
4	-0.98	-0.88	-1.39
8	-1.49	-1.31	-1.98
16	-1.93	-1.69	-2.47

TABLE VIII

BD-RATE GAINS IN % OVER ECM 4.0 FOR AI CONFIGURATION FOR CLASS TGM WITH SINGLE FILTER AND MULTIPLE FILTERS

Filter(s)	Sequence	Y	U	V
Single	FlyingGraphic	-1.41	-2.28	-2.62
	Desktop	-1.52	-2.16	-3.65
	Console	-1.85	-3.01	-2.76
	ChineseEditing	-0.14	-0.31	-0.23
	Average	**-1.23**	**-2.18**	**-2.32**
Multiple	FlyingGraphic	-2.24	-3.33	-4.04
	Desktop	-4.45	-5.25	-6.66
	Console	-3.35	-4.86	-4.53
	ChineseEditing	-0.40	-0.61	-0.46
	Average	**-2.61**	**-3.51**	**-3.92**

It is evident from the above results that the proposed model brings substantial gains despite enabling or disabling palette coding, demonstrating our claim. For the rest of the discussion that follows, we choose the configuration where our method is an independent cross-component mode and not conditioned on palette coding.

The performance of our method depends on having a rich set of samples in the mapping function which directly relates to the number of reference lines used for deriving the mapping function. To illustrate this dependency, we vary the number of reference lines used for deriving the look-up table and record the compression performance. Coding performance for different number of chroma reference lines is presented in Table VII. It is evident that the performance improves substantially when we include more reference lines for deriving the mapping function.

As mentioned earlier, the accuracy of the prediction depends on the down-sampling filter used. The comparison of gains with single filter and multiple filters is presented in Table VIII. It is evident that using multiple filters brings significant gains. To confirm the efficacy of the filters we introduced, we consider the frequency of the usage of these filters for Desktop and Console sequences under AI configuration and present them in Fig. 9. From the plot, it is clear that different content prefer different down-sampling filters.

As regards the complexity, without the optimizations presented in section IV, for the results in Table III, it would incur over 7% increase in encoding and decoding complexity. With the optimizations, the increase in complexity is clamped at 1%. Specifically, the first optimization in IV-A brings about 4-5% reduction in coding time and the second optimization in IV-B brings about 2-3% reduction in coding time.

The main benefit of the proposed method is that it can provide good prediction despite luma and chroma being largely uncorrelated. To confirm this, we extract the blocks that chose

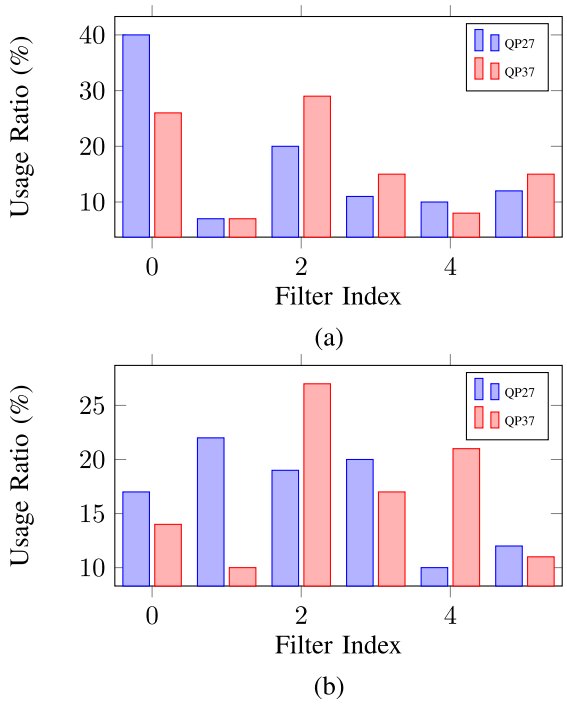


Fig. 9. Histogram of the filter usage statistics for (a) Desktop sequence and (b) Console sequence under AI configuration.

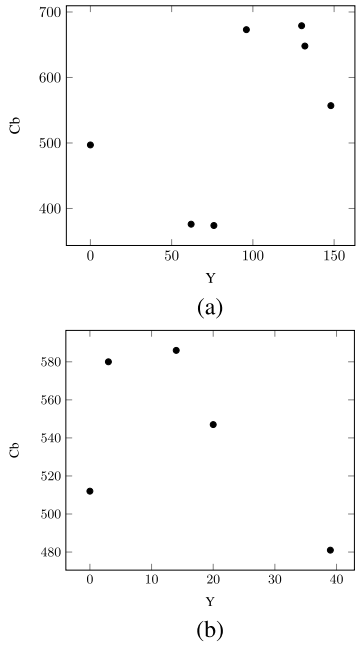


Fig. 10. Plots of Y-Cb samples taken from blocks of screen content sequences that choose the proposed method.

our method and plot their luma-chroma characteristics in Fig. 10. From the plot, it is evident that the blocks that exhibit mostly uncorrelated luma-chroma characteristics chose our prediction method. To further quantify the performance, we consider the following two measures:

1) Frequency of Mode Selection: We first consider statistics of how frequently the presented mode is chosen. We present this in Table IX (for AI configuration), which shows the percentage of the total cross-component predicted blocks that choose the discrete mapping model. From the table, it is evident that the model is frequently chosen with the usage being as high as 70% for the Desktop sequence.

TABLE IX
PERCENTAGE OF THE TOTAL CROSS-COMPONENT PREDICTED BLOCKS THAT USE THE PROPOSED MODEL (AI CONFIGURATION)

Sequence	QP	Percentage
FlyingGraphic	27	48%
FlyingGraphic	37	43%
Desktop	27	72%
Desktop	37	68%
Console	27	49%
Console	37	50%
ChineseEditing	27	20%
ChineseEditing	37	12%

2) Prediction Gain: Having established that the proposed method is frequently used, we next consider its impact in terms prediction gain. To this end, let us consider the prediction gain for an example block that chose our method, where prediction gain, G_P, and its corresponding expression in dB, $\mathrm{SNR}_P[dB]$, are traditionally defined as [30]

$$G_P = \frac{\sigma^2}{\sigma_e^2}, \quad \mathrm{SNR}_P[dB] = 10 \log G_P,$$

where σ^2 denotes the energy of samples in the block before prediction, and σ_e^2 is the energy in the prediction residual. For the block whose characteristics is plotted in Fig. 10a, the prediction gain from the linear model is 9.5dB and the prediction gain from our model is a significant 21.9dB. This clearly illustrates the benefit of our model over linear models. We would like to emphasise that, the more diverse is the set of colors in the block and more uncorrelated is their luma-chroma characteristics, better is the performance of the proposed method compared to the linear models. Further, it is easy to see that the situation in which we have a block with diverse set of colors that exhibits highly uncorrelated cross-component relationship is when it is hard to achieve good compression, which exactly is the scenario in which the proposed method offers significant benefit.

C. Performance in Low Bit-Rate Regime

The rate points considered in the experiments so far correspond to QP set of {22, 27, 32, 37}, as recommended in the common test conditions of ECM. However, a careful observation of Fig. 8 reveals that the PSNR range is 40-50 dB, which is rather high. This makes one question the utility of the model for lower rates (or PSNRs). To address this, we consider the QP set of {37, 40, 43, 46} such that we observe the performance in PSNR range of 30-40 dB. (The lowest rate point in earlier experiments is now the highest rate point. Further, we determined empirically that for QP of 46, the PSNR was around 30 dB, which made us choose the presented new QP set.) The average bit-rate savings for all-intra (AI) and randomaccess configurations (RA) for this scenario for class TGM are shown in Table X. Evidently, the proposed method brings significant coding gains with an average of 3.43% luma bit-rate reduction and up to 5% luma bit-rate reduction under AI configuration. RD curves

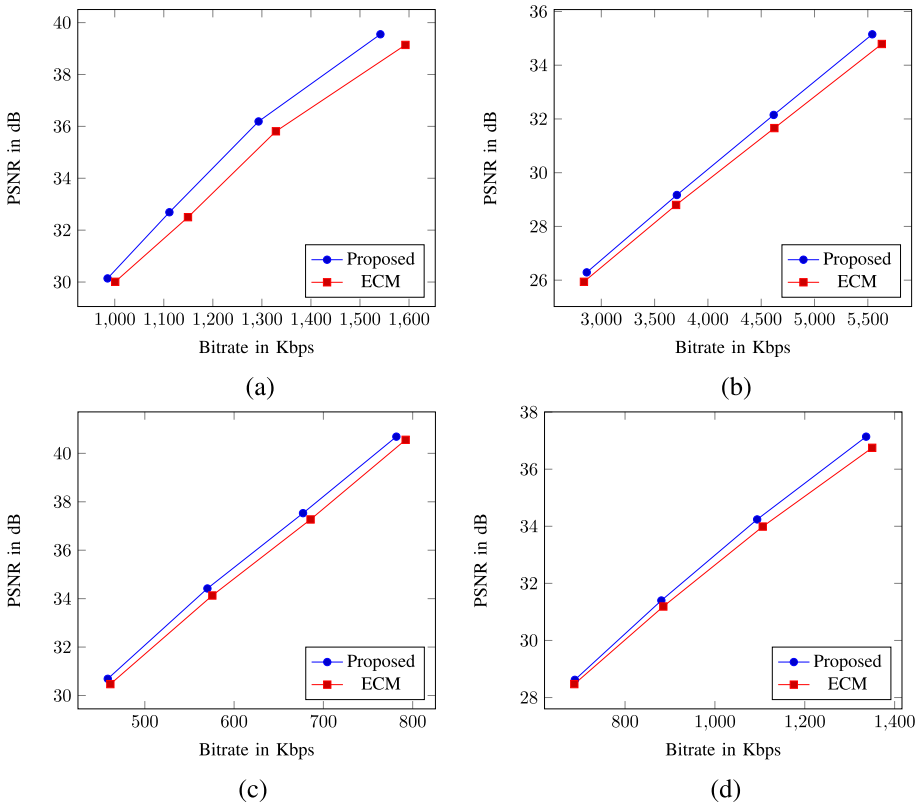


Fig. 11. RD curves for: (a) Console, (b) ChineseEditing sequences for AI configuration and (c) Console, (d) ChineseEditing sequences for RA configuration for QP set {37, 40, 43, 46} (palette coding is disabled in both anchor and the proposed method).

TABLE X

BD-RATE GAINS IN % OVER ECM 4.0 FOR AI AND RA CONFIGURATIONS FOR QP SET {37, 40, 43, 46} FOR CLASS TGM (PALETTE CODING IS DISABLED IN ANCHOR AND THE PROPOSED METHOD)

Configuration	Sequence	Y	U	V
AI	FlyingGraphic	-2.38	-4.15	-5.36
	Desktop	-3.22	-1.24	-3.24
	Console	-4.99	-7.65	-5.33
	ChineseEditing	-3.13	-1.44	-1.67
	Average	**-3.43**	**-3.62**	**-3.90**
	Δ EncT: 1%			
	Δ DecT: 1%			
RA	FlyingGraphic	-0.59	-1.38	-1.30
	Desktop	-2.39	-1.02	-2.16
	Console	-1.43	-1.80	-1.09
	ChineseEditing	-2.55	-1.05	-1.66
	Average	**-1.74**	**-1.31**	**-1.55**
	Δ EncT: 1%			
	Δ DecT: 1%			

in Fig. 11 illustrates consistent performance across all bit-rates. Compared to our earlier results in Tables III and IV, we now have improved performance. Our reasoning for this observation is as follows: at lower rates, the encoder is forced to select larger blocks since block partitioning can be expensive. This in turn increases the likelihood of encountering blocks with diverse set of colors with uncorrelated cross-component characteristics. Thus, employing discrete mapping model yields better gains.

VI. CONCLUSION

This paper proposes a novel cross-component prediction model for screen content coding. The proposed method departs significantly from the conventional continuous models for cross-component prediction and shows the benefit of a discrete-mapping model in obtaining accurate prediction for screen content coding. Substantial gains compared to ECM demonstrate the effectiveness of the proposed model.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 50–62, Jan. 2016.

[3] B. Bross et al., "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[4] T. Nguyen et al., "Overview of the screen content support in VVC: Applications, coding tools, and performance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3801–3817, Oct. 2021.

[5] W.-S. Kim et al., "Cross-component prediction in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1699–1708, Jun. 2020.

[6] J. Pfaff et al., "Intra prediction and mode coding in VVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3834–3847, Oct. 2021.

[7] L. Trudeau, N. Egge, and D. Barr, "Predicting chroma from Luma in AV1," in *Proc. Data Compress. Conf.*, Mar. 2018, pp. 374–382.

[8] L. Goffman-Vinopal and M. Porat, "Color image compression using inter-color correlation," in *Proc. Int. Conf. Image Process.*, 2002, p. II.

[9] B. C. Song, Y. G. Lee, and N. H. Kim, "Block adaptive inter-color compensation algorithm for RGB 4:4:4 video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, pp. 1447–1451, Oct. 2008.

[10] S. H. Lee and N. I. Cho, "Intra prediction method based on the linear relationship between the channels for YUV 4:2:0 intra coding," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 1037–1040.

[11] J. Kim, S.-W. Park, J.-Y. Park, and B.-M. Jeon, *Intra Chroma Prediction Using Inter Channel Correlation*, document JCTVC-B021, 2020.

[12] J. Chen et al., *Chroma Intra Prediction by Scaled Luma Samples Using Integer Operations*, document JCTVC-C206, Oct. 2020.
[13] K. Zhang, J. Chen, L. Zhang, X. Li, and M. Karczewicz, "Enhanced cross-component linear model for chroma intra-prediction in video coding," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3983–3997, Aug. 2018.
[14] M. Budagavi and D-K. Kwon, *(AHG8): Video Coding Using Intra Motion Compensation*, document JCTVC-M0350, Apr. 2013.
[15] X. Xu et al., "Intra block copy in HEVC screen content coding extensions," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 409–419, Dec. 2016.
[16] X. Xiu et al., "Palette-based coding in the screen content coding extension of the HEVC standard," in *Proc. Data Compress. Conf.*, Apr. 2015, pp. 253–262.
[17] Y.-C. Sun, J. Lou, Y.-H. Chao, H. Wang, V. Seregin, and M. Karczewicz, "Analysis of palette mode on versatile video coding," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Mar. 2019, pp. 455–458.
[18] L. Zhang et al., "Adaptive color-space transform in HEVC screen content coding," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 446–459, Dec. 2016.
[19] X. Li, J. Sole, and M. Karczewicz, *Adaptive MV Precision for Screen Content Coding*, document JCTVC-P0283, Jan. 2014.
[20] B. Li, J. Xu, G. J. Sullivan, Y. Zhou, and B. Lin, *Adaptive MV Precision for Screen Content Coding*, document JCTVC-S0085, Oct. 2014.
[21] C. Chen, J. Han, Y. Xu, and J. Bankoski, "A staircase transform coding scheme for screen content video coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2365–2369.
[22] B. Vishwanath, K. Zhang, and L. Zhang, "A cross-component prediction model for screen content coding," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2022, pp. 361–365.
[23] M. Coban, F. L. Léannec, and J. Ström, *Algorithm Description of Enhanced Compression Model 2 (ECM 2)*, document JVET-W2025, 2021.
[24] X. Zhang, C. Gisquet, E. François, F. Zou, and O. C. Au, "Chroma intra prediction based on inter-channel correlation for HEVC," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 274–286, Jan. 2014.
[25] A. Browne, J. Chen, Y. Ye, and S. Kim, *Algorithm Description for Versatile Video Coding and Test model 14 (VTM 14)*, document JVET-W2002, 2021.
[26] (2022). *ECM-4.0*. [Online]. Available: https://vcgit.hhi.fraunhofer.de/ecm/ECM/-/tree/ECM-4.0
[27] H. Schwarz et al., "Quantization and entropy coding in the versatile video coding (VVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3891–3906, Oct. 2021.
[28] M. Karczewicz and Y. Ye, *Common Test Conditions and Evaluation Procedures for Enhanced Compression Tool Testing*, document JVET-X2017, 2021.
[29] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33 ITU-T Q6/16, Austin, TX, USA, Apr. 2001.
[30] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, vol. 159. New York, NY, USA: Kluwer, 2012.

Bharath Vishwanath (Member, IEEE) received the B.E. degree in electronics and communications engineering from the National Institute of Technology Karnataka, India, in 2014, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara (UCSB), Santa Barbara, CA, USA, in 2016 and 2021, respectively. He has interned with Interdigital Communications Inc., San Diego, CA, USA, during the Summer of 2016 and 2017, and Dolby Laboratories during the Summer of 2019. He is currently a Multimedia Research Scientist with ByteDance Inc., San Diego, CA, USA. His research interests include video coding, non-convex optimization, and information theory.

Kai Zhang (Senior Member, IEEE) received the B.S. degree in computer science from Nankai University, Tianjin, China, in 2004, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011. From 2011 to 2012, he was a Researcher with Tencent Inc., Beijing. From 2012 to 2016, he was the Team Manager with Mediatek Inc., Beijing, leading a research team to propose novel technologies for emerging video coding standards. From 2016 to 2018, he was with Qualcomm Inc., San Diego, CA, USA, with a focus on video coding standardization. He is currently leading the Standardization Team, ByteDance Inc. Since 2006, he has contributed more than 500 proposals to JVT, VCEG, JCT-VC, JCT-3V, JVET, JPEG, MPEG, and AVS, covering many important aspects of major standards, such as H.264/AVC, HEVC, 3D-HEVC, VVC, JPEG-AI, MPEG-GPCC, and AVS-1,2,3. He is the inventor of 100 of granted or pending U.S. patent applications. Most of these patents are essential to popular video coding standards. He serves as a coordinator for the reference software known as ECM in JVET, to explore future video coding technologies beyond VVC. He has coauthored more than 70 journal or conference papers. His research interests include video/image compression, coding, processing, and communication, especially video coding standardization. He co-chaired several core experiments and branches of groups during the development of VVC. He serves as an Associate Editor for *IET Image Processing* and a reviewer of several well-known journals and conferences.

Li Zhang (Senior Member, IEEE) received the B.S. degree in computer science from Dalian Maritime University, Dalian, China, in 2003, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2009. She was a Software Coordinator for Audio and Video Coding Standard (AVS) and the 3D extensions of High Efficiency Video Coding (HEVC). From 2009 to 2011, she was a Postdoctoral Researcher with the Institute of Digital Media, Peking University, Beijing. From 2011 to 2018, she was a Senior Staff Engineer with the Multimedia Research and Development and Standards Group, Qualcomm Inc., San Diego, CA, USA. She is currently the Lead of the Multimedia Laboratory, ByteDance Inc., San Diego. She has authored more than 500 standardization contributions, more than 300 granted U.S. patents, and more than 100 technical articles in related book chapters, journals, and proceedings in image/video coding and video processing. She has been an active contributor to the Versatile Video Coding, Advanced AVS, the IEEE 1857, 3D Video (3DV) coding extensions of H.264/AVC and HEVC, and HEVC screen content coding extensions. During the development of those video coding standards, she co-chaired several ad hoc groups and core experiments. Her research interests include 2D/3D image/video coding, video processing, and transmission. She organized/co-chaired multiple special sessions and grand challenges at various conferences/journals. She has been appointed as an Editor of AVS and the Main Editor of the Software Test Model for 3DV Standards. She serves as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the Publicity Subcommittee Chair of the Technical Committee Member of Visual Signal Processing and Communications in IEEE CAS Society (VSPC TC).