ADAPTIVE CROSS-COMPONENT PREDICTION FOR 4:4:4 HIGH EFFICIENCY VIDEO CODING

Ali Khairat, Tung Nguyen, Mischa Siekmann, Detlev Marpe, and Thomas Wiegand

Fraunhofer Institute for Telecommunications - Heinrich Hertz Institute
Image Processing Department
Einsteinufer 37, D-10587 Berlin, Germany

ABSTRACT

The work on support for higher bit depths and 4:2:2 as well as 4:4:4 chroma sampling formats for the High Efficiency Video Coding (HEVC) standard is currently being conducted under the term Range Extensions (RExt). A technique that exploits the correlation between residual color components in 4:4:4 chroma sampling format, also referred to as Cross-Component Prediction (CCP), has been adopted as part of the current RExt draft. In this paper, this relatively simple but yet effective CCP scheme is presented. Conceptually, CCP relies on the idea that an adaptively switched predictor based on a linear model is invoked for coding of the residuals of the second and third color component by using the residual of the first color component. Experimental results show that, depending on the underlying color space, average bit-rate savings in the range of 2–18% or 3–26% can be achieved by CCP for test sets of natural and screen content, respectively.

Index Terms— HEVC, H.265, Range Extensions, Cross-Component Prediction, Inter-Component Decorrelation

1. INTRODUCTION

The new High Efficiency Video Coding (HEVC) standard [1, 2] with its first version published as ITU-T H.265 and ISO/IEC 23008-2 in 2013 outperforms prior video coding standards by approximately 50% in terms of bit-rate savings for equivalent perceptual video quality [3]. H.265/HEVC Version 1 mainly focuses on consumer entertainment and interactive applications, and hence, 4:2:0 is its only supported chroma sampling format. That is, the chroma components have half the spatial resolution of the luma component in both horizontal and vertical dimension. This type of chroma subsampling is extensively used in video compression and helps reducing the overall bit-rate, albeit at a reduced chroma fidelity. However, for certain professional or high-quality applications there is a demand for higher chroma and luma fidelity.

Range Extensions (RExt) [4] for H.265/HEVC which are currently under development and planned to be finalized in July 2014 [5] address these needs. Besides the support of full chroma resolution in both dimensions, also denoted as 4:4:4 chroma format, as well as support of the 4:2:2 chroma format with full-resolution chroma in vertical but half-resolution chroma in horizontal direction, also coding with bit depths beyond 10 bits per sample will be supported by the upcoming HEVC RExt standard. Beyond these extended input formats, several additional coding methods are considered in the current RExt working draft. Among the prospective target applications of RExt are the coding of screen content and the direct coding of RGB video signals. Note that usually, prior to coding or chroma subsampling, the RGB video signal is transformed to $\mathrm{YC_bC_r}$ [6] with the resulting decorrelated signal representation consisting of one luma (Y) and two chroma ($\mathrm{C_b}$, $\mathrm{C_r}$) components.

For RGB content as well as on top of the $\mathrm{YC_bC_r}$ representation, inter-component decorrelation methods can be applied at different levels of the encoding/decoding processes. In [7], a high-level syntax approach is chosen, where a conversion from RGB to a $\mathrm{YC_bC_r}$-like integer color space with a subsequent chroma subsampling can be enabled at a sequence level, such that the H.265/HEVC Version 1 4:2:0 codec can be reused for 4:4:4 formatted input material. Another approach, referred to as linear model (LM) mode [8], uses a linear model to predict the chroma signal from the reconstructed luma signal on a block-by-block basis. It was originally designed to address the remaining correlation in $\mathrm{YC_bC_r}$ 4:2:0 video signals. In [7] and [9], a decorrelation process is also applied in an adaptively switched block-by-block fashion, but with application to the residual signal components.

In this paper, the so-called Cross-Component Prediction (CCP) method is presented. It combines the advantages of both the previously mentioned LM mode [8] and the macroblock-adaptive residual color transform (MB-ARCT [9]). CCP has been adopted as part of the current working draft of HEVC RExt [5].

The paper is organized as follows. After a brief description of two related approaches in Section 2, a detailed description of CCP is given in Section 3. Section 4 contains the experimental results together with a discussion of their implications. A conclusion is given in Section 5.

2. RELATED WORK

As the presented CCP method is related to MB-ARCT [9] and LM chroma [8], their features are described in more detail.

2.1. Blockwise-Adaptive Residual Color Transform

MB-ARCT [9] was proposed for 4:4:4 RGB content in H.264/AVC. In this approach, the color representation of the prediction residual is chosen adaptively on a macroblock level. This permits an adaptation to non-stationary statistics in terms of color characteristics. A set of three different color representations is defined. Namely, RGB, $\mathrm{YC_oC_g}$ and the newly introduced $\mathrm{Gr_Br_R}$. The $\mathrm{Gr_Br_R}$ color domain is represented by one green color component and two color difference components relative to green as follows

$$\begin{aligned} G &= \tilde{G} \\ r_B &= B - \tilde{G} \\ r_R &= R - \tilde{G} \end{aligned} \tag{1}$$

where $\tilde{G}$ is the quantized green component that is used as the predictor for the second and third components, resulting in r_B and r_R, respectively. Green is taken as the main component since it is a close approximation of the luma channel.

2.2. Chroma Intra Prediction Using Linear Model (LM)

In the LM mode proposal for 4:2:0 sequences [8], an additional intra prediction mode is introduced for chroma blocks, where the samples are predicted from the reconstructed down-sampled luma samples by using a linear model as follows

$$c = \alpha \cdot l + \beta \tag{2}$$

where c is the chroma sample to be predicted, l is the corresponding down-sampled reconstructed luma sample, α and β are the model parameters. The values of α and β are derived from the reconstructed neighboring samples of the current block as follows. The normalized covariance between luma and chroma reference samples is calculated and taken as α. The offset β is obtained by fitting the linear model to the reference samples, utilizing (2) under the knowledge of α.

3. CROSS-COMPONENT PREDICTION

The Cross-Component Prediction (CCP) was initially proposed during the development of H.265/HEVC RExt in [10] and later refined in [11]. Although CCP can also be applied to 4:2:2 and 4:2:0 chroma formats, the current RExt Draft [5] includes its use for the 4:4:4 chroma format only. The scheme itself combines the advantages of both MB-ARCT and LM chroma by operating adaptively on the spatial or temporal prediction residuals using a linear model. An important fact is that CCP itself is to a large extent agnostic of the input color space at hand. This is due to the fact that the reconstructed residual of the main color component (CM) acts as the predictor for each of the second and third remaining color components' (CR) residuals. Instead of performing CCP directly on the given prediction block, it is applied separately for the residual signals that are covered by transform blocks in H.265/HEVC. Generally speaking, in H.265/HEVC the block partitioning for transform coding is to a large extent decoupled from the given prediction block size by using a nested tree structure referred to as residual quadtree (RQT [12]). At this end, i.e., for each transform block, a syntax element, referred to as the coding block flag (CBF), is transmitted specifying the existence of non-zero valued transform coefficient levels. Given that there are residuals in the area covered by the transform block (residual block) of the main component, i.e., $\mathrm{CBF}_{CM} = 1$, the CCP scheme can be applied for the two residual blocks in the two remaining components at the same spatial location.

3.1. Residual-Based Prediction

From the encoder perspective, a weighted version of the main component's residual ($\hat{r}_{CM}$) is subtracted from the residual of the remaining component ($\hat{r}_{CR}$) at the same spatial location, with α being the weighting factor and r_{CR} the final residual, as summarized in the following equation

$$r_{CR} = \hat{r}_{CR} - \alpha \cdot \hat{r}_{CM}. \tag{3}$$

This linear operation is performed before the transform and quantization of the remaining components' residual blocks, and clearly, the inverse operation is applied after scaling and inverse transform at the decoder. In contrast to the prediction of LM chroma in (2), the offset β is dropped. The reason for that is mainly its application to the residual signal. A constant offset is only affecting the DC value after the transform, which can be efficiently coded by the subsequent entropy coding stage. Since the value of α can be different for each of the second and third component, α is transmitted twice, i.e., separately for each residual block of the remaining components with an associated $\mathrm{CBF}_{CM} = 1$. Moreover, the value of α is limited to the range between -1 and 1, inclusively, and its integer representation has three bits of precision. The relation between the accuracy of the integer representation and its signaling in the bit stream is the topic of the following subsection.

3.2. Parameter Signaling

The prediction is forward-driven, i.e., no further parameter derivation is required at the decoder but the encoder specifies the related parameters in the bit stream. In contrast to H.264/AVC, Context-Based Adaptive Binary Arithmetic Coding (CABAC [12, 13]) is the only method for entropy coding in H.265/HEVC. For the signaling of the integer-valued weighting parameter α with CABAC, as the first step,

α is separated into two syntax elements specifying its absolute value and its sign. In order to transmit the absolute value, the so-called truncated unary binarization is used to generate the sequence of binary decisions (bins). However, the maximum number of bins for α is limited to four, and hence, five values can be represented. That is the reason why the calculated absolute value of α is non-uniformly quantized to $\{0, 1, 2, 4, 8\}$ at the encoder side. Both the binarization and the quantization of α enables its reconstruction by a single left shift as shown in (4) at the decoder side, where the parsed value from the bit stream is denoted as TU_{val}:

$$|\alpha| = \begin{cases} 0 & \text{TU}_{val} = 0 \\ 2^{\text{TU}_{val}-1} & \text{TU}_{val} \neq 0 \end{cases} \tag{4}$$

After the transmission of $|\alpha|$, the sign syntax element is coded only if $|\alpha|$ is not equal to zero. Regarding the context modeling, each bin of the bin string resulting from the binarization of $|\alpha|$ uses a separate context model. In addition to that, the sign syntax element uses a separate context model. Furthermore, different context model sets are used for the two CRs resulting in 10 context models in total. Note that the value of $|\alpha|$ is inferred to be equal to zero in the case of $\text{CBF}_{CM} = 0$.

3.3. Calculation of the Weighting Parameter

The calculation of α can be done as implemented in the reference software implementation for RExt (HM+RExt [14]). In particular, α is calculated once for each remaining component's residual block with $\text{CBF}_{CM} = 1$ for the corresponding main component. Due to the linear model for CCP, the calculation is simply the correlation between the two residual blocks and it is summarized in the following formula, i.e.,

$$\alpha = \frac{\mathbf{cov}(r_{CM}, \hat{r}_{CR})}{\mathbf{var}(\hat{r}_{CM}, \hat{r}_{CM})} \tag{5}$$

where $\mathbf{cov}$ and $\mathbf{var}$ denote the estimator for the covariance and the variance, respectively. The implemented calculation uses floating point precision and the resulting α is transformed into an integer representation with four bits of precision. Finally, it is quantized non-uniformly in order to match the syntax of the final binary representation. In the reference implementation, the unquantized residuals for the main component is used for the calculation of α. Another possibility is to use the reconstructed residual of the main component instead of the unquantized residual. It was shown, however, that the calculation using the unquantized residuals has negligible impact on the coding performance, but it provides a possibility for simpler hardware encoder implementations [15].

The estimated α parameter does not guarantee optimal results in rate-distortion (RD) performance because the linear model itself may be suboptimal. Therefore, in the reference encoder, the RD performance for both the case without using CCP ($\alpha = 0$) and the case using CCP based on the estimated α as given in (5) is compared to each other.

4. EXPERIMENTAL RESULTS

The results of our experimental evaluation were generated using the reference software implementation in version 12.1 (HM+RExt [14]). This version mainly corresponds to the description of CCP in the current RExt draft [5]. In the following, the test conditions are described first, followed by a presentation and discussion of the experimental results.

4.1. Experimental Setup

For the conducted experiments, video sequences having 4:4:4 chroma format, as described in the RExt common test conditions (CTC [16]), are used as input. They are mainly divided into two classes with different characteristics, i.e., natural and screen content. Independent from the content classes, the used sequences are varying in spatial resolution, sample bit depth, and color space, i.e., RGB or $\text{YC}_\text{b}\text{C}_\text{r}$. Depending on the color space of the input, the color component order is adjusted while the remaining encoder parameters are chosen as described in the CTC. Particularly, the RGB sequences are encoded in GBR order where the green channel acts as the first component, B and R are the second and third components, respectively. For $\text{YC}_\text{b}\text{C}_\text{r}$ sequences, the luma Y is the first component, C_b is the second component and C_r is the third component. Results are generated for all three test configurations as described in the CTC, i.e., All Intra (AI), Random Access (RA), and Low Delay B (LB). Furthermore, also according to the CTC, results are generated for the main-tier and the high-tier configuration.

In the following, bit-rate savings results are generated using the so-called Bjøntegaard delta rate (BD rate [17]), which is considered to be a useful measure for evaluating RD performance. CCP is enabled in HM+RExt and compared to the default configuration of HM+RExt which serves as the reference for the BD-rate calculation. For the CCP experiments the α parameter was calculated for each block using the original residual signal of the three components.

Results in Tables 1 and 2 show the percentage of BD rates for the three components of RGB or $\text{YC}_\text{b}\text{C}_\text{r}$. Note that the given BD rates are the mean values for the defined set of sequences in the CTC [16]. Negative numbers indicate that there is an increase in RD performance, i.e., a BD rate saving. An average BD rate figure for each color space is presented together with a BD rate figure that is based on an average peak signal-to-noise ratio (PSNR) of all three components, calculated as the mean of four times the PSNR of the first component and the PSNR of each of the two other components. For example, the average PSNR for RGB is given as

$$\overline{\text{GBR}} = \frac{4 \cdot \text{G} + \text{B} + \text{R}}{6}. \tag{6}$$

The weights for the PSNR of the components in Equation (6) were calculated according to the bit-rate ratio of each component relative to the total bit rate [10].

4.2. Results

It can be noticed that the RD performance is increased for all configurations by applying the CCP technique for the screen content sequences as shown in Table 1. In terms of RGB color space in Table 1a, an increase in RD performance of up to 26% for the G component as well as for the averaged RGB is achieved in the AI configuration. A large increase in RD performance of 20.3–24.9% BD rate savings is observed for the other configurations as well. For the YC_bC_r color space, as shown in Table 1b, the gains are considerably smaller than for the RGB color space since the correlation between the components was already reduced to a large extent during the pre-processing color transform stage.

Table 1: Averaged BD rate (%) for 4:4:4 video sequences with screen content

(a) RGB

	Main Tier				High Tier			
	G	B	R	$\overline{GBR}$	G	B	R	$\overline{GBR}$
AI	**-26.4**	-26.1	-25.7	**-26.2**	**-25.2**	-24.6	-24.2	**-24.9**
RA	**-22.1**	-21.9	-21.4	**-21.9**	**-22.0**	-21.5	-21.1	**-21.7**
LB	**-20.8**	-19.5	-19.1	**-20.3**	**-20.9**	-19.5	-19.2	**-20.3**

(b) YC_bC_r

	Main Tier				High Tier			
	Y	C_b	C_r	$\overline{YC_bC_r}$	Y	C_b	C_r	$\overline{YC_bC_r}$
AI	**-3.6**	-10	-7.6	**-8.0**	**-3.9**	-8.0	-7.0	**-5.1**
RA	**-2.3**	-9.7	-7.2	**-4.3**	**-2.6**	-8.5	-7.0	**-4.3**
LB	**-1.5**	-8.2	-5.6	**-3.3**	**-2.1**	-7.4	-5.6	**-3.5**

Table 2 shows the results for the video sequences with the natural content. Similar BD rate savings for the RGB color space, as given in Table 2a, were observed as for the screen content input. Also, for the YC_bC_r color space, as shown in Table 2b, there is always a small but consistent BD rate saving for all components as well as for the averaged cross-component case.

It can also be observed that the BD rate savings for screen content are usually much larger than those for natural content sequences. This is due to the fact that natural sequences typically contain more high-frequency textures leading to less correlation between the components as compared to the screen content case.

CCP also proved to provide better coding efficiency than other suggested related work. A variant of CCP was compared to a number of fixed color transforms for RGB color sequences [10]. This variant is a subset of the presented CCP implementation, so that the comparisons are valid. The results in [10] showed consistently better RD performance for CCP compared to YC_bC_r, YC_oC_g and Gr_Br_R color transforms for all tested configurations.

Table 2: Averaged BD rate (%) for 4:4:4 video sequences with natural content

(a) RGB

	Main Tier				High Tier			
	G	B	R	$\overline{GBR}$	G	B	R	$\overline{GBR}$
AI	**-18.2**	-16.6	-17.3	**-17.7**	**-14.2**	-13.6	-13.8	**-14.0**
RA	**-14.4**	-11.9	-13.8	**-13.8**	**-11.3**	-8.8	-10.7	**-10.7**
LB	**-13.3**	-9.3	-10.8	**-12.2**	**-10.3**	-7.0	-8.4	**-9.4**

(b) YC_bC_r

	Main Tier				High Tier			
	Y	C_b	C_r	$\overline{YC_bC_r}$	Y	C_b	C_r	$\overline{YC_bC_r}$
AI	**-1.4**	-6.5	-6.7	**-3.1**	**-1.7**	-4.4	-6.2	**-2.9**
RA	**-0.4**	-8.4	-7.2	**-2.8**	**-0.8**	-6.2	-7.5	**-2.8**
LB	**-0.2**	-6.0	-5.6	**-2.0**	**-0.6**	-4.3	-6.3	**-2.1**

The CCP variant was also tested against the LM chroma approach and the corresponding results in [18] showed larger BD rate savings in RA and LB configurations as well as nearly the same BD rate savings for the AI configuration. This is due to the fact that LM chroma was designed for intra coding only.

From the experimental results, it can be deduced that by using CCP a high RD performance is achieved for all types of 4:4:4 video sequences independent from the underlying input color space. The adaptivity of the proposed technique guarantees optimal RD performance by selecting the best performing mode for each transform block.

5. CONCLUSION

In this work we have presented the cross-component prediction (CCP) technique that is part of the current draft of the H.265/HEVC Range Extensions. CCP is based on an adaptive block-based linear predictor for the residual signals of the second and third color components using the reconstructed residual signal of the first color component. CCP is inherently agnostic to the given color space representation of the video signal and has proven to provide gains in terms of BD rate savings for both RGB and YC_bC_r content in 4:4:4 chroma format.

6. REFERENCES

[1] "High Efficiency Video Coding," Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan 2013.

[2] G. Sullivan, J. R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, Dec 2012.

[3] J. R. Ohm, G. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards–Including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, Dec 2012.

[4] G. Sullivan, J. Boyce, Y. Chen, J. R. Ohm, A. Segall, and A. Vetro, "Standardized Extensions of High Efficiency Video Coding (HEVC)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 22, pp. 1669–1684, Dec 2012.

[5] D. Flynn, M. Naccari, K. Sharman, C. Rosewarne, J. Sole, G. Sullivan, and T. Suzuki, "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 6," document JCTVC-P1005 of JCT-VC, Jan 2014.

[6] C. Poynton, *Digital Video and HD: Algorithms and Interfaces*, Morgan Kaufmann Publishers Inc., 2003.

[7] W. Dai, M. Krishnan, and P. Topiwala, "AHG7: 4:4:4 Lite: A High-Level Syntax Approach to a Consumer 4:4:4 Codec," document JCTVC-L0175 of JCT-VC, Jan 2013.

[8] X. Zhang, C. Gisquet, E. François, F. Zou, and O. Au, "Chroma Intra Prediction Based on Inter-Channel Correlation for HEVC," *IEEE Transactions on Image Processing*, vol. 23, pp. 274–286, Nov 2013.

[9] D. Marpe, H. Kirchhoffer, V. George, P. Kauff, and T. Wiegand, "Macroblock-Adaptive Residual Color Space Transforms for 4:4:4 Video Coding," in *IEEE International Conference on Image Processing*, Oct 2006, pp. 3157–3160.

[10] T. Nguyen, A. Khairat, and D. Marpe, "Non-RCE1/Non-RCE2/AHG5/AHG8: Adaptive Inter-Plane Prediction for RGB Content," document JCTVC-M0230 of JCT-VC, Apr 2013.

[11] W. Pu, W-S. Kim, J. Chen, K. Rapaka, L. Guo, J. Sole, and M. Karczewicz, "Non-RCE1: Inter Color Component Residual Prediction," document JCTVC-N0266 of JCT-VC, Jul 2013.

[12] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Transform Coding Techniques in HEVC," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, pp. 978–989, Dec 2013.

[13] D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 620–636, Jul 2003.

[14] "HEVC Reference Software 12.1 + Range Extensions 5.1," [Online] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-12.1+RExt-5.1/.

[15] W.-S. Kim and T. Nguyen, "RCE1: Summary Report of HEVC Range Extensions Core Experiment 1 on Inter-Component Decorrelation Methods," document JCTVC-O0035 of JCT-VC, Oct 2013.

[16] D. Flynn, K. Sharman, and C. Rosewarne, "Common Test Conditions and Software Reference Configurations for HEVC Range Extensions," document JCTVC-O1006 of JCT-VC, Oct 2013.

[17] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD Curves," document VCEG-M33 of ITU-T Q6/16, Apr 2001.

[18] T. Nguyen, J. Sole, and J. Kim, "RCE1: Summary Report of HEVC Range Extensions Core Experiment 1 on Inter-Component Decorrelation Methods," document JCTVC-N0034 of JCT-VC, Jul 2013.