ORIGINAL PAPER

Extended cross-component prediction in HEVC

MISCHA SIEKMANN[1], ALI KHAIRAT[1], TUNG NGUYEN[1], DETLEV MARPE[1], AND THOMAS WIEGAND[1,2]

With Version 2 of the high-efficiency video coding standard, a new compression efficiency tool targeting redundancies among color components is specified for all 4:4:4 profiles, and referred to as cross-component prediction (CCP). This paper describes and analyses two additional extensions to the specified CCP variant. In the first extension, an additional predictor is introduced. Particularly, beside the luma component, also the first chroma component can serve as a reference for prediction of the second chroma component. The second extension proposes a method for predicting the CCP model parameter from the statistics of already reconstructed neighboring blocks. A performance analysis of coding RGB content in different color representations is given in comparison with CCP and both extensions. Experimental results show that the proposed extensions can improve the compression efficiency effectively compared with CCP, when applied in the YC_bC_r domain.

I. INTRODUCTION

Version 1 of the high-efficiency video coding (HEVC) standard [1, 2] mainly focuses on consumer applications and therefore only supports the 4:2:0 chroma sampling format. This is motivated by the well-known fact that the human visual system is much less sensitive to high-frequency components in chroma than in luma, such that sub-sampling of the chroma signal typically results in significant bit-rate savings for those consumer-oriented target applications. The Range Extensions (RExt) of HEVC [3], which are included in Version 2 of the HEVC standard [4], extend the supported formats to 4:4:4 and 4:2:2 chroma sampling formats and bit depths beyond 10 bits per sample. Furthermore, new coding tools are supported by RExt. These tools aim at improving the coding efficiency for specific application scenarios, such as screen content (SC) coding, high-bitrate, and lossless coding. Typical 4:4:4 or 4:2:2 video material shows significant statistical dependencies between its color components. This redundancy is especially high when the components are represented in absolute amplitudes, as in RGB. But even a representation in luma and chroma components, such as YC_bC_r, does not decorrelate the components perfectly. Given that significant and typically local dependencies remain between color components, it is essential to exploit these dependencies.

The cross-component prediction (CCP [5, 6]), as it is specified for all 4:4:4 profiles in HEVC Version 2, follows a linear luma-to-chroma prediction scheme in order to exploit the similarities between color components. Specifically, the residual samples of both, the first and second chroma components, can be predicted by a weighted value of the reconstructed luma residual samples. Here, and in the rest of this paper, the first component is always referred to as the luma component, while the remaining components are denoted as the first and second chroma components, respectively. This holds, regardless of the actual used color representation. CCP is an adaptive forward-driven linear prediction system. On the encoder side, a prediction weight α is determined for each chroma transform block (TB) and si]gnaled within the bit-stream. Hence, no additional decoder complexity is involved for deriving the prediction parameters.

In this paper, two possible extensions to the CCP scheme are introduced and analyzed. In the first extension, a chroma-to-chroma prediction is included. This implies that also the reconstructed residual samples of the first chroma component can act as a reference signal for prediction of the residual samples of the second chroma component. The second extension addresses the reduction of parameter signaling overhead. It introduces a method for predicting the model parameter α from statistics of already reconstructed neighboring blocks, and hence, only an offset needs to be signaled.

This paper is organized as follows. Section II gives an overview of different approaches for exploiting the redundancies among color components in the context of video compression. A detailed description of the CCP approach, as specified in HEVC Version 2, is given in Section III, and two extensions to that scheme are presented in Section IV.

[1]Image and Video Coding Group, Department of Video Coding & Analytics, Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany
[2]Image Communication Chair, Technical University of Berlin, Germany

Corresponding author:
Mischa Siekmann
Email: mischa.siekmann@hhi.fraunhofer.de

Experimental results are given and discussed in Section V, followed by the conclusion.

II. CROSS-COMPONENT DEPENDENCIES IN VIDEO COMPRESSION

Exploiting the redundancy among color components of videos is crucial for an efficient compression. One widely used approach is to find an appropriate color representation for the video to be compressed, in which its color components are less correlated compared with the original video signal space, and to code the video in that representation instead. After decoding, the video signal has to be transformed back to its original representation. When choosing a color representation, also human visual perception characteristics should be taken into account. The human visual system (HVS) is less sensitive to chroma degradation. Hence, a different treatment of luma and chroma components is beneficial for compression. In the following Section II-A, a few examples of such color representations are described and some of their properties are highlighted.

In general, video signals are highly in-stationary in terms of their color characteristics. Hence, it can be valuable to adapt to local color statistics within the coding process. Adaptivity is the focus of Section II-B.

A) Color video representations

The inherent color video representation of today's capture and display devices is mostly based on the R′G′B′ color space. The prime symbol (′) is the widely used notation for gamma corrected representation. Gamma correction is always assumed in this paper, and hence, a specific notation is omitted from now on. As opposed to the RGB representation, the separation between luma and chroma provides the opportunity of chroma subsampling. Chroma subsampling keeps the impact on the subjective quality low, while reducing the overall bit-rate significantly. With the support of 4:4:4 chroma formats, in HEVC Version 2, also the direct coding of RGB content got available. As the luma component in HEVC has a preferential treatment compared with chroma components, it is advisable to code the green component (G) as the luma component. This is motivated by the highest contribution to the intensity perception in the HVS among the RGB components and referred to as GBR coding in the following.

1) YC_bC_r

In video compression, the YC_bC_r color space is widely used for video representation. Here, Y refers to the luma component, while C_b and C_r are the chroma components, respectively. This color representation was originally found by principal component analysis (PCA) performed on a set of video signals represented in YIQ color space (i.e., the color space of the NTSC television standard), while trying to approximate the intensity perception of the HVS by the luma component Y [7]. Several definitions for the YC_bC_r transform can be found, one is shown in equation (1) and its inverse in (2) [8].

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.7152 & 0.0722 & 0.2126 \\ -0.3854 & 0.5 & -0.1146 \\ -0.4542 & -0.0458 & 0.5 \end{pmatrix} \begin{pmatrix} G \\ B \\ R \end{pmatrix}, \quad (1)$$

$$\begin{pmatrix} G \\ B \\ R \end{pmatrix} = \begin{pmatrix} 1 & -0.1873 & -0.4681 \\ 1 & 1.856 & 0 \\ 1 & 0 & 1.575 \end{pmatrix} \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix}. \quad (2)$$

It can be observed, that the luma component consists of a weighted amount of each RGB component, where the green component (G) contributes with the highest scaling factor. The chroma components on the other hand, are represented relatively to the luma component. Transforming video signals from RGB color space to YC_bC_r reduces the statistical dependencies among the components significantly. This is particularly the case for natural scene content.

The floating point arithmetic involved in the forward and backward transformations of YC_bC_r, is a weakness of this representation. This gets especially relevant, when it comes to high bit-rate coding or even lossless coding. In such cases, the rounding error introduced by transforming becomes significant, and hence, YC_bC_r is not feasible for these applications.

2) YC_oC_g

Motivated by the question, whether the YC_bC_r color space is the best color space for compression, considering that it was derived decades ago and modern high-resolution cameras producing sharper images nowadays, the color space YC_oC_g was developed and presented in [7]. This transform was obtained by approximating a Karhunen–Loève–Transform (KLT) estimated on the Kodak image database. Note, also in this transform the green channel contributes most to the luma channel. The forward transform can be described as:

$$\begin{pmatrix} Y \\ C_o \\ C_g \end{pmatrix} = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 0 & -1/2 & 1/2 \\ 1/2 & -1/4 & -1/4 \end{pmatrix} \begin{pmatrix} G \\ B \\ R \end{pmatrix}. \quad (3)$$

One advantage of this transform is its simplicity in terms of implementation considerations. Only additions and shift operations are needed. A lossless version of YC_oC_g can be realized by utilizing the lifting technique. This version (YC_oC_g-R) can be exactly inverted in integer arithmetic as described by the following operations:

$$\begin{aligned} C_o &= R - B, \\ t &= B + (C_o \gg 1), \\ C_g &= G - t, \\ Y &= t + (C_g \gg 1). \end{aligned} \quad (4)$$

When representing a video in YC_oC_g-R, the dynamic range of both chroma components (C_o,C_g) is increased by one bit, compared to the original transformation in (3).

3) $G\,R_b\,R_r$

An even simpler transformation can be obtained by taking the G component as the luma component itself, and represent the remaining components as difference signals to the G component. This transform is also reversible in integer arithmetic and was described in [9], where it is referred $G\,R_b\,R_r$ transform. The forward $G\,R_b\,R_r$ transform can be expressed as:

$$\begin{pmatrix} G \\ R_b \\ R_r \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} G \\ B \\ R \end{pmatrix}. \tag{5}$$

B) Adaptive CCP

Having the extra processing steps of color transformations before encoding, and after decoding, is not feasible for all applications (e.g., coding of SC). In this cases, additional methods for exploiting the cross-component correlation within the compression procedure are indispensable. On the other hand, including the action of decorrelation inside the compression process, can lead to an improved coding-efficiency in general, as it rises the opportunity of adapting to local image statistics.

This adaptivity can be achieved by choosing appropriate color transformations for different color characteristics of the video signal. They can be applied at different levels of the encoding/decoding process. In [9], the color representation of the residual signal is independently chosen for each macro block. The specified alternatives to the input color space are $Y C_o C_g$ and $G\,R_b\,R_r$. Instead of defining a set of available transforms, one can also make use of a prediction model. The transformation defined in equation (5) can be seen as a linear prediction scheme. Here, the component G acts as the predictor for the B and R components, respectively. Adaptivity can be obtained by introducing prediction weights, which can be adjusted as desired. This can be expressed as:

$$\begin{pmatrix} G \\ R_b \\ R_r \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\alpha_1 & 1 & 0 \\ -\alpha_2 & 0 & 1 \end{pmatrix} \begin{pmatrix} G \\ B \\ R \end{pmatrix}. \tag{6}$$

The prediction parameters α_1 and α_2 are needed for reconstruction and may be derived backward-adaptively [10], or signaled within the bitstream.

CCP utilizes a prediction model as described in equation (7), which is forward-driven. A detailed description of CCP is given in Section III.

This model can be extended by also allowing the first chroma component to act as predictor for the second chroma component. One of such models is expressed in equation (7) for the GBR domain. Here, an additional parameter $\rho \in \{0, 1\}$ allows to switch between both predictors. One of the extensions of CCP discussed in this paper, follows this scheme and is described in Section IV.

$$\begin{pmatrix} G \\ R_b \\ R_r \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\alpha_1 & 1 & 0 \\ -(1-\rho)\alpha_2 & -\rho\alpha_2 & 1 \end{pmatrix} \begin{pmatrix} G \\ B \\ R \end{pmatrix}. \tag{7}$$

III. CCP IN HEVC

This section gives a detailed description of CCP as it is specified for 4:4:4 profiles in HEVC Version 2. As described earlier, when coding RGB content, it is advisable to choose the green component (G) as luma and components (R) and (B) as chroma (GBR coding). In the following description, this is always assumed and no further distinction between input color spaces is made.

A) Coding scheme

The CCP scheme operates in the residual domain. Particularly, it is applied to chroma residual samples, obtained, after inter or intra prediction has been performed. Transform coding of prediction residuals in HEVC comprise a partitioning into TBs according to a quadtree structure, which is signaled within the bit stream and denoted as residual quadtree (RQT) [11]. This partitioning allows an adaptation to local statistics of the residual signals. Residual samples belonging to the same TB are jointly transform coded. This efficiently helps exploiting the spatial correlation within the same color component. CCP, on the other hand, addresses the correlation between chroma and luma residual samples at the same spatial location. Following the description of Section II-B, a chroma residual sample is predicted by a weighted amount of the spatially aligned luma sample. CCP operates block adaptive at the TB level, namely, chroma residual samples of the same TB share the prediction weight α. This can be expressed as:

$$\hat{\mathbf{r}}^{(Ci)} = \alpha_i \cdot \tilde{\mathbf{r}}^{(L)}, \tag{8}$$

where $\tilde{\mathbf{r}}^{(L)}$ denotes the reconstructed residual samples of one luma TB and $\hat{\mathbf{r}}^{(Ci)}$ denotes the obtained prediction for one of the chroma TBs, aligned with $\tilde{\mathbf{r}}^{(L)}$. The index $i \in \{1, 2\}$ emphasizes that each of the two chroma components are predicted individually, and $\tilde{\mathbf{r}}$ indicates that the samples might be degraded due to quantization. It should be noted, that $\tilde{\mathbf{r}}^{(L)}$ can be equal to $\mathbf{0}$. This occurs when the luma samples could be perfectly predicted by intra/inter prediction or when signaling of $\tilde{\mathbf{r}}^{(L)}$ is not reasonable in rate-distortion (RD) sense. In such cases, CCP is not meaningful. If the predictor signal is available, on the other hand, the weighting parameters for both corresponding chroma TBs are signaled within the bitstream. CCP can be omitted for each chroma TB by setting its parameter α to 0.

When CCP is applied to a chroma TB, the difference between its residual samples $\mathbf{r}^{(C)}$ and their predicted samples $\hat{\mathbf{r}}^{(C)}$ is transform coded instead of $\mathbf{r}^{(C)}$:

$$\Delta\mathbf{r}^{(C)} = \mathbf{r}^{(C)} - \alpha \cdot \tilde{\mathbf{r}}^{(L)}. \tag{9}$$

The following reconstruction rule follows for the decoder:

$$\tilde{\mathbf{r}}^{(C)} = \Delta\tilde{\mathbf{r}}^{(C)} + \alpha \cdot \tilde{\mathbf{r}}^{(L)}. \tag{10}$$

In order to keep the overhead for signaling α low, the prediction weights are limited to a fixed set:

$$\alpha \in \left\{ 0, \pm\frac{1}{8}, \pm\frac{1}{4}, \pm\frac{1}{2}, \pm 1 \right\}. \tag{11}$$

This set of weights allows a simple implementation, which can be expressed for each sample of a TB as follows:

$$\tilde{r}_c = \Delta\tilde{r}_c + \left\lfloor \frac{\alpha' \cdot \tilde{r}_l}{8} \right\rfloor \tag{12}$$

$$= \Delta\tilde{r}_c + (\alpha' \cdot \tilde{r}_l) \gg 3. \tag{13}$$

Hence, only one addition and one shift operation are needed per sample and the value of $\alpha' \in \{0, \pm 1, \pm 2, \pm 4, \pm 8\}$ is binarized and entropy coded. Particularly, the sign and absolute value of α' are coded separately, where the truncated unary binarization scheme of CABAC [12] is used for the index to the absolute value. The sign of α' needs to be coded only, if $|\alpha'| \neq 0$. As the statistics of both chroma components may differ significantly, a separate set of context models is introduced for each chroma component. Furthermore, having separate context models for signaling the sign and for each of the four unary bins, a total amount of ten context models is defined for the CCP scheme.

B) Encoder considerations

During the encoding process, the prediction weight has to be chosen for each chroma TB. A straight forward and optimal method (in sense of RD costs), is to test each allowed weight in (11) and to take that α, that leads to the lowest RD cost. However, this approach is often not practical in terms of encoder run times. When the residual samples are assumed to be realizations of a stationary random process, with known second-order statistics, an analytical solution can be found by the least-mean-square approach for linear models. The α_M with minimum mean-square prediction error is then given by:

$$\alpha_M = \frac{\text{cov}(\tilde{r}_l, r_c)}{\text{var}(\tilde{r}_l)}. \tag{14}$$

The HEVC reference software encoder (HM 16.2 [13]), for example, utilizes equation (14) by taking sample statistics from aligned chroma and luma TBs, respectively. The obtained α_s is then quantized to α'_s following equation (15) and look-up-table (LUT) (16):

$$\alpha'_s = \text{sign}(\alpha_s) \cdot \text{LUT}_\alpha(|\alpha_s|), \tag{15}$$

$$\text{LUT}_\alpha(x) = \begin{cases} 0 & x < 1/16 \\ 1 & x \in [1/16, 3/16) \\ 2 & x \in [3/16, 3/8) \\ 4 & x \in [3/8, 3/4) \\ 8 & x \geq 3/4. \end{cases} \tag{16}$$

The RD performance of CCP with $\alpha' = \alpha'_s$ is then compared with the performance when omitting CCP for the TB (i.e., $\alpha' = 0$), and α' is set accordingly.

From equation (9) it can be seen that $\Delta\mathbf{r}^{(C)}$ has an increased dynamic range of 1 bit compared with $\mathbf{r}^{(C)}$. Hence, the internal bit-depth of the chroma component representation should be increased by 1 bit, in order to prevent an excessive quantization of the prediction differences $\Delta\mathbf{r}^{(C)}$.

IV. CCP EXTENSION

Two extensions are described in the following: the introduction of an additional predictor and a prediction scheme for model parameter α.

A) Additional predictor

As described in Section II it might be valuable to also exploit the correlation among the first and second chroma components. Because the first chroma component is reconstructed first, it can serve as a predictor for the second chroma component. In this section, an extension to CCP is described in which the predictor for the second chroma component can be chosen adaptively on the TB level. An additional flag $\rho \in \{0, 1\}$ is introduced for each TB of the second chroma component. Here, a value of $\rho = 1$ indicates that the reconstructed samples of the first component, weighted by α_2, are used for prediction. The reconstructed luma residual samples are used otherwise:

$$\hat{\mathbf{r}}^{(C2)} = \alpha_2 \cdot \tilde{\mathbf{r}}^{(L)}, \quad \text{if } \rho = 0, \tag{17}$$

$$\hat{\mathbf{r}}^{(C2)} = \alpha_2 \cdot \tilde{\mathbf{r}}^{(C1)}, \quad \text{if } \rho = 1. \tag{18}$$

These predictions are only meaningful, when the reconstructed TB of the predictor component contains significant residuals. Hence, only when both conditions, $\tilde{\mathbf{r}}^{(L)} \neq \mathbf{0}$ and $\tilde{\mathbf{r}}^{(C1)} \neq \mathbf{0}$ hold, the value of ρ has to be signaled. Its value can be derived otherwise. It should be noted, that the first chroma component can also act as a predictor, when it got predicted by the luma component itself, but no prediction difference was signaled (i.e., $\text{CBF}^{(C1)} = 0$ and $\alpha_1 \neq 0$).

Table 1 summarizes the different predictors $\hat{\mathbf{r}}^{(C2)}$ for the second chroma component residual.

The additional syntax element ρ is signaled using CABAC entropy coding. The prediction weight α_2 is binarized following the same scheme as in CCP. However, a distinct set of context models is used when the first chroma component is used as the predictor (i.e., $\rho = 1$).

Table 1. Condition for ρ and choice of the predictor.

$\text{CBF}^{(L)}$	$\text{CBF}^{(C1)}$	α_1	$\hat{\mathbf{r}}^{(C2)}$
0	0	–	–
0	1	–	$\alpha_2\tilde{\mathbf{r}}^{(C1)}$
1	0	0	$\alpha_2\tilde{\mathbf{r}}^{(L)}$
1	0	$\neq 0$	$\alpha_2[(1-\rho) + \rho\alpha_1] \cdot \tilde{\mathbf{r}}^{(L)}$
1	1	$\forall\alpha$	$\alpha_2[(1-\rho)\tilde{\mathbf{r}}^{(L)} + \rho\tilde{\mathbf{r}}^{(C1)}]$

B) Predicting the model parameter α

As seen in Section III-B, the optimal prediction weight α_M (in sense of mean square error), is described by equation (14) and can be estimated on the encoder side by the residual sample-statistics of the current TB. If a local stationarity can be assumed, an estimator $\hat{\alpha}$ obtained from sample-statistics of already decoded neighboring blocks can also be reasonable and serve as a predictor for the CCP model parameter α of succeeding TBs.

Following this approach, the second extension to CCP presented in this paper defines the following prediction scheme: For each decoded TB one separate weight $\hat{\alpha}_i$ is estimated from its quantized luma $\tilde{\mathbf{r}}^{(L)}$ and chroma $\tilde{\mathbf{r}}^{(C)}$ residual samples, which then serves as a predictor for neighboring TBs. In the case were no residual was signaled, the corresponding $\hat{\alpha}_i$ is set to zero. Now, let $\hat{\alpha}_L$ denote the estimated prediction weight of the left neighboring TB, and $\hat{\alpha}_T$ the corresponding weight of the TB above. The CCP model parameter used for the current TB is then given by the arithmetic mean of both predictors, plus some correction term $\Delta\alpha$:

$$\alpha = \text{mean}(\hat{\alpha}_L, \hat{\alpha}_T) + \Delta\alpha. \tag{19}$$

Here, $\Delta\alpha$ allows a correction of the predicted parameter and is signaled for each TB where CCP is applicable. More precisely, a flag is signaled whether a correction term is present, the correction value itself is coded utilizing a unary binarization. It should be noted, that in this extension the resulting CCP prediction weight α is no longer restricted to the set defined in (11). Both, the prediction and the correction factor, are uniformly quantized with a step size of $1/8$. This not only allows a finer CCP compared with the base variant of CCP, but also results in an increased signaling cost when the prediction $\hat{\alpha}$ from neighboring blocks is poor.

V. PERFORMANCE EVALUATION

The performances of CCP and the two described extensions are evaluated and discussed in this section. The test sequences are given in RGB domain and are coded in both, GBR and YC_bC_r representation. The direct coding in GBR domain without any additional decorrelation methods (i.e., all CCP variants disabled), is taken as the reference test scenario.

A) Experimental setup

All schemes are implemented on top of HEVC RExt reference software HM-10.1 RExt 3 [14]. Whenever CCP or one of its extensions is enabled, the internal chroma bit-depth is increased by 1. In the CCP variants where the model parameters are signaled directly, the encoder is configured to select α by testing all allowed values. In the case of predicting α from its neighborhood, the correction term $\Delta\alpha$ is estimated as described in Section III-B. The performed simulations are based on the common test conditions (CTC) [15], defined during the standardization of HEVC RExt. The set of test sequences include camera captured (CC) sequences and computer rendered sequences, referred to as SC material. The evaluation is focused on random access (RA) configuration. However, in the application of SC coding a low latency is of great interest. Hence, SC results are also presented for low-delay (LD) configuration. Only sequences with 4:4:4 chroma format, and those included in both, RGB and YC_bC_r test sets, are considered. For simplicity reasons, when coded in YC_bC_r domain, the reconstructed video is not converted back to RGB domain. Instead, the distortion is measured in the coding domain and its contribution to the components of RGB is estimated mathematically. In particular, the distortion of different components is assumed to be uncorrelated, then taking the square of each weight of the inverse transform, equation (2), leads to the following estimation:

$$\begin{pmatrix} \text{MSE}_G \\ \text{MSE}_B \\ \text{MSE}_R \end{pmatrix} = \begin{pmatrix} 1 & 0.351 & 0.2191 \\ 1 & 3.443 & 0 \\ 1 & 0 & 2.48 \end{pmatrix} \begin{pmatrix} \text{MSE}_Y \\ \text{MSE}_{Cb} \\ \text{MSE}_{Cr} \end{pmatrix}.$$

It should be noted, that the rounding error due to transformation before compression cannot be measured in this setup. Hence, only main tier configurations (i.e., quantization parameter $\text{QP} \in \{22, 27, 32, 37\}$) are considered. At those RD operation points the rounding errors are assumed to be negligible.

B) BD-rate performance

The bit-rate saving for each test condition is given in terms of Bjøntegaard delta (BD) rate [16]. The obtained results, averaged over sequences of each class, are listed in Tables 2–4. The first three rows show the obtained bit-rate saving when coding in GBR representation, while the results in the bottom rows are obtained by coding the sequences in YC_bC_r representation. Here, the additional predictor extension is denoted as CCP-AP, and the extension for predicting the model parameter α, as CCP-PP. The + sign indicates that the coding tool is applied in addition to the color transformation to YC_bC_r domain. According to the observations of bit-rate distributions among the color components made in [17], the $\overline{\text{GBR}}$-BD rate is calculated based on an weighted average over the PSNRs of each component:

$$\overline{\text{GBR}}_{\text{PSNR}} = \frac{4 \cdot G_{\text{PSNR}} + B_{\text{PSNR}} + R_{\text{PSNR}}}{6}. \tag{20}$$

Table 2. BD-rate of CC sequences (RA).

	G%	B%	R%	$\overline{\text{GBR}}$%
CCP	−55.9	−23.2	−27.3	−48.7
CCPE-AP	−55.9	−23.2	−27.4	−48.6
CCPE-PP	−56.5	2.6	3.8	−46.5
YC_bC_r	−59.4	0.2	−17.1	−50.6
+CCP	−59.6	−5.4	−19.3	−51.4
+CCP-AP	−59.7	−5.2	−19.7	−51.5
+CCP-PP	−60.0	−4.3	−19.6	−51.7

Table 3. BD-rate of SC sequences (RA).

	G%	B%	R%	$\overline{GBR}$
CCP	−40.0	−19.0	−20.5	−34.1
CCP-AP	−40,3	−19,6	−21,2	−34.5
CCP-PP	−38,1	−16.7	−18,2	−32.1
YC_bC_r	−37.3	−18.8	−26.2	−32.8
+CCP	−40.9	−24.5	−31.1	−36.9
+CCP-AP	−42.1	−23.1	−29.3	−37.6
+CCP-PP	−42.6	−26.5	−32,9	−38.7

Table 4. BD-rate of SC sequences (LD).

	G%	B%	R%	$\overline{GBR}$%
CCP	−40.9	−19.6	−21.2	−34.9
CCP-AP	−40,4	−18.6	−20,4	−34.4
CCP-PP	−37,7	−15.8	−17,3	−31.6
YC_bC_r	−38.5	−19.9	−27.0	−33.9
+CCP	−41.6	−24.4	−30.9	−37.4
+CCP-AP	−41.7	−24.6	−31.6	−37.6
+CCP-PP	−43.9	−27.8	−33,9	−40.0

The following can be observed for the set of CC sequences: The average bit-rate saving due to coding after a transformation to YC_bC_r representation amounts to 50% in terms of the $\overline{GBR}$-BD rate. A similar, yet smaller, coding efficiency can be achieved with the CCP scheme when coding in GBR representation directly. Moreover, when coding in YC_bC_r domain the remaining cross-component redundancy can still be reduced by CCP, and hence, an additional coding gain is achieved. Both of the described extensions generate a further minor bit-rate reduction when applied in the YC_bC_r domain, however, no extra coding gain can be achieved in the GBR representation, when compared with the unmodified version of CCP.

For the SC sequences the results are given for RA (Table 3) and LD (Table 4) configuration, respectively. In both cases, a smaller bit-rate saving, of about 33%, is obtained due to the coding in YC_bC_r domain. However, the remaining dependencies can be reduced by CCP and a further BD rate saving of 4% is attained. In the RA case, the additional predictor gives a further saving of almost 1%, while in LD configuration the improvement is only about 0.2%. Predicting α spatially in YC_bC_r domain gives the highest coding-gain among the tested conditions. Here, a total $\overline{GBR}$-BD rate saving of about 39% (RA) and 40% (LD)

is reached. When it comes to direct coding in GBR representation, CCP achieves a higher gain compared with coding in YC_bC_r without any further decorrelations processes. Here, only the additional predictor achieves a slightly higher coding gain when coded in RA configuration. Predicting the model parameter α, and signaling a correction term instead, leads to a significant decrease in coding performance of about 2–3% in the case of coding in GBRrepresentation.

In order to get a more detailed impression of the advantages of each decorrelation approach, the results of individual sequences are shown and discussed for the RA configuration in the following. For each class, the BD-rate of sequences with different color characteristics are shown in Table 5. They have been chosen because of their quite different behavior when coded in color representations other than GBR. It is of interest how well CCP and the extensions deal with those characteristics. The results for YC_oC_g and GR_bR_r are taken from [17] and are also shown in Table 5. It should be pointed out, that the simulations performed in [17] are based on different conditions compared to the results presented in this paper. Hence, these results serve only as an indicator for the sequence characteristics and the coding performance cannot be compared directly.

It can be seen that the impact of color transforming differ immensely between the SC sequences. The sequence *TwistTunnel* is mainly monochromatic, consequently the correlation among the components in RGB representation is close to 1, and hence, transforming to GR_bR_r domain should be almost optimal in sense of decorrelation. Also all CCP-based coding tools show a high bit-rate saving for this sequence. However, it can be seen that a representation in YC_bC_r already decorrelates the luma and chroma components almost completely, hence applying CCP additionally, only results in signaling overhead. Predicting the model parameter α from the spatial neighborhood, as in CCP-PP, compensates this overhead, and the finer CCP even leads to a small coding gain. When a color transform is not applicable, CCP also gives a comparable gain when applied in GBR domain. Here however, the signaling overhead is even increased by CCP-PP. This can be explained by the fact that α is assumed to be zero in the case when no residual signal is available for neighboring blocks. Knowing that in the GBR case the optimal α is close to one for this sequence, the prediction for α is very poor in this cases. When coding in YC_bC_r domain on the other hand, the remaining correlation between color components should be close to zero. Hence, the correction factor $\Delta\alpha$ is also small and

Table 5. $\overline{GBR}$ BD-rate for a subset of sequences (RA).

Sequence	CCP%	CCP-AP%	CCP-PP%	YC_bC_r%	+CCP%	+CCP-AP%	+CCP-PP%	$YC_oC_g - R$%	GR_bR_r%
EBURainFruit	−47, 8	−47, 7	−45, 0	−49, 8	−50, 6	−50, 7	−51, 3	−47.7	−50.1
Kimon1	−49, 2	−49, 3	−47, 8	−51, 0	−51, 4	−51, 4	−51, 8	−37.9	−46.2
TwistTunnel	−52, 0	−51, 9	−43, 3	−55, 7	−51, 3	−55, 4	−55, 8	−52.7	−53.2
Waveform	−22, 9	−23, 4	−22, 4	2, 6	−11, 8	−25, 5	−16, 6	2.4	−8.5
Webbrowsing	−26, 7	−26, 6	−23, 7	−25, 1	−25, 9	−29, 0	−26, 2	−21.3	−28.1

its finer granularity leads to a further improvement in the YC_bC_r domain.

For the *Waveform* sequence even a loss in terms of bit-rate savings can be observed when coding in YC_oC_g-R. This can be explained by the high in-stationarity of color characteristics in this sequence. Thus, an adaptive approach such as CCP is essential. It can be seen that the option of chroma-to-chroma prediction increases the coding-efficiency. For both sequences, *Waveform* and *Webbrowsing*, coding in YC_bC_r with CCP and an additional predictor leads to the best coding gain in our setup.

Even though both selected CC sequences differ in there BD-rate saving when coded in YC_oC_g-R representation, they show quite similar behavior for all test scenarios in our setup. These results are consistent with the averages shown in Table 2.

VI. CONCLUSION

Two extensions to the CCP scheme of the RExt of HEVC Version 2 have been presented and evaluated in this paper. The first extension introduces an optional chroma-to-chroma prediction for the second chroma component. More specifically, for each TB of the second chroma component, it can be chosen adaptively whether the corresponding reconstructed luma residual samples or the reconstructed residual samples of the first chroma component serve as a reference for prediction of the second chroma component. The second extension proposes a method for predicting the model parameter α from the sample statistics of already reconstructed neighboring blocks.

The performances of both extensions were tested along with the original CCP scheme for two different color representations of RGB sequences, namely, direct coding in GBR order and coding after transforming to YC_bC_r domain. CC sequences as well as SC material were considered. It has been shown that in the case of CC sequences, CCP provides a good alternative when a pre-processing transformation to YC_bC_r is not applicable. In addition, remaining redundancies in YC_bC_r representation can still be further reduced by applying CCP in the YC_bC_r domain. In this case, the additional predictor extension gives a further small improvement.

In terms of the SC material, coding in GBR representation with CCP outperforms coding in YCbCr, provided that no further decorrelation step is applied. It turns out that remaining redundancies between color components can be better exploited by CCP when it is applied in the YC_bC_r domain. In this case, both extensions give further improvements in terms of bit-rate savings. In our test configurations, the additional predictor achieves an average gain of 0.7% compared with the original CCP, while predicting the model parameters gives an improvement of 1.8% bit-rate saving. However, directly coding in GBR domain is often favored in SC applications. Here, only the additional predictor leads to an average bit-rate reduction of about 0.5% for a RA coding configuration.

REFERENCES

[1] ISO/IEC and ITU-T: Rec. H.265: High Efficiency Video Coding (HEVC), April 2013.

[2] Sullivan, G.; Ohm, J.R.; Han, W.-J.; Wiegand, T.: Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Trans. Circuits Syst. Video Technol., **22** (2012), 1649–1668.

[3] Flynn, D. *et al.*: Overview of the range extensions for the HEVC standard: tools, profiles, and performance. IEEE Trans. Circuits Syst. Video Technol., **26** (2016), 4–19.

[4] ISO/IEC and ITU-T: Rec. H.265 (V2): High Efficiency Video Coding (HEVC), October 2014.

[5] Khairat, A.; Nguyen, T.; Siekmann, M.; Marpe, D.; Wiegand, T.: Adaptive cross-component prediction for 4:4:4 high efficiency video coding, in *IEEE Int. Conf. Image Processing 2014 (ICIP 2014)*, Paris, France, October 2014.

[6] Kim, W.-S. *et al.*: Cross-component prediction in HEVC. IEEE Trans. Circuits Syst. Video Technol., **PP** (99) (2015), 1–1.

[7] Malvar, H.S.; Sullivan, G.J.; Srinivasan, S.: Lifting-based reversible color transformations for image compression, in *Optical Engineering+Applications*, September 2008, 707307–707307, International Society for Optics and Photonics.

[8] ITU-T: Rec. BT.709-6: parameter values for the HDTV standards for production and international programme exchange, June 2015.

[9] Marpe, D.; Kirchhoffer, H.; George, V.; Kauff, P.; Wiegand, T.: Macroblock-adaptive residual color space transforms for 4:4:4 video coding, in *IEEE Int. Conf. Image Processing*, October 2006, 3157–3160.

[10] Zhang, X.; Gisquet, C.; Francois, E.; Zou, F.; Au, O.: Chroma intra prediction based on inter-channel correlation for HEVC. IEEE Trans. Image Process., **23** (2014), 274–286.

[11] Nguyen, T. *et al.*: Transform coding techniques in HEVC. IEEE J. Selected Top. Signal Process., **7** (2013), 978–989.

[12] Marpe, D.; Schwarz, H.; Wiegand, T.: Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. IEEE Trans. Circuits Syst. Video Technol., **13** (2003), 620–636.

[13] HEVC Reference Software (HM) version 16.2, [Online] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.2.

[14] HEVC Reference Software (HM) Version 10.1 RExt 3, [Online] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-10.1+RExt-3.0.

[15] Flynn, D.; Sharman, K.; Rosewarne, C.: Common Test Conditions and Software Reference Configurations for HEVC Range Extensions, Document JCTVC-O1006 of JCT-VC, October 2013.

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

[17] Nguyen, T.; Khairat, A.; Marpe, D.: Non-RCE1/Non-RCE2/AHG5/AHG8: Adaptive Inter-Plane Prediction for RGB Content, Document JCTVC-M0230 of JCT-VC, April 2013.

Mischa Siekmann received the degree 'Dipl.-Ing. in Electrical Engineering' from Technical University of Braunschweig, Braunschweig, Germany, in 2008. His studies included an exchange with the University of Southampton, U.K., and an internship with British Sky Broadcasting, London, U.K. After university he joined the Image and Video Coding Group of the Video Coding & Analytics Department at Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, Germany. His major

research interests include image and video compression, video quality assessment, and signal processing.

Ali Khairat earned his B.Sc. degree in Digital Media Engineering and Technology from the German University in Cairo, Egypt in 2008. He received his M.Sc. degree in Systems of Information and Multimedia Technology from the Friedrich-Alexander University of Erlangen-Nuremberg, Germany, in 2011. From 2012–2015, he has worked with the Image and Video Coding Group, Video Coding and Analytics Department, Fraunhofer Institute for Telecommunications Heinrich Hertz Institute, Berlin 10587, Germany. His research interests include video coding for range extensions for HEVC. He is with the BMW Team, Managed Services Department, Virtustream Dell Technologies, Munich 85737, Germany.

Tung Nguyen received the Diploma degree in computer science (Dipl.-Inf.) from the Technical University of Berlin (TUB), Berlin, Germany, in 2008. He joined the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute (HHI), Berlin in 2009. Since that time, he has worked as a Research Associate, and his current association within the HHI is the Image and Video Coding Group of the Video Coding and Analytics Department. From 2009 to 2015, he actively participated in the standardization activity of the Joint Collaborative Team on Video Coding (JCT-VC). He successfully contributed as a member of the JCT-VC on the topic of entropy coding for the development of HEVC, Version 1. He is also one of the main contributors of the Cross-Component Prediction scheme for the Range Extensions in Version 2 of HEVC. His current research interests include image and video processing and their efficient implementation.

Detlev Marpe received the Dipl.- Math. degree (Highest Hons.) from the Technical University of Berlin (TUB), Berlin, Germany, and the Dr.-Ing. degree from the University of Rostock, Rostock, Germany. He is Head of the Video Coding & Analytics Department and Head of the Image & Video Coding Group at the Fraunhofer Heinrich Hertz Institute (HHI), Berlin. He is also active as a part-time lecturer at TUB. For more than 15 years, he has successfully contributed to the standardization activities of the ITU-T Visual Coding Experts Group, the ISO/IEC Joint Photographic Experts Group, and the ISO/IEC Moving Picture Experts Group for still image and video coding. During the development of the H.264 | Advanced Video Coding (AVC) standard, he was chief Architect of the CABAC entropy coding scheme, and one of the main technical and editorial contributors to the so-called Fidelity Range Extensions with the addition of the High Profile in H.264/AVC. He was also one of the key people in designing the basic architecture of Scalable Video Coding and Multiview Video Coding as algorithmic and syntactical extensions of H.264/AVC. He also made successful contributions to the recent development of the H.265 | MPEG-H High Efficiency Video Coding (HEVC) standard, including its Range Extensions and 3D extensions. He is author or co-author of more than 200 publications in the area of video coding and signal processing, and he holds numerous internationally issued patents and patent applications in this field. For his substantial contributions to the field of video coding, he received numerous awards, including a Nomination for the 2012 German Future Prize, the Karl Heinz Beckurts Award 2011, two Emmy Engineering Awards in 2008 and 2009, the 2009 Best Paper Award of the IEEE Circuits and Systems Society, the Joseph von Fraunhofer Prize 2004, and the Best Paper Award of the German Information Technology Society in 2004. He also received the IEEE Best Paper Award at the 2013 IEEE International Conference on Consumer Electronics-Berlin in 2013 and the SMPTE Journal Certificate of Merit in 2014. He is an IEEE Fellow and Member of the ITG (German Information Technology Society). Since 2014, he has served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. His research interests include still image and video coding, signal processing for communications, and computer vision and information theory.

Thomas Wiegand (M'05, SM'08) is a professor in the department of Electrical Engineering and Computer Science at the Technical University of Berlin and is jointly heading the Fraunhofer Heinrich Hertz Institute, Berlin, Germany. He received the Dipl.- Ing. degree in Electrical Engineering from the Technical University of Hamburg-Harburg, Germany, in 1995 and the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Germany, in 2000. As a student, he was a Visiting Researcher at Kobe University, Japan, the University of California at Santa Barbara and Stanford University, USA, where he also returned as a visiting professor. He served as a consultant to several start-up ventures and is currently a consultant to Vidyo, Inc., Hackensack, NJ, USA. He has been an active participant in standardization for video coding multimedia with many successful submissions to ITU-T and ISO/IEC. He is the Associated Rapporteur of ITU-T VCEG. The projects that he co-chaired for the development of the H.264/MPEGAVC standard have been recognized by an ATAS Primetime Emmy Engineering Award and a pair of NATAS Technology & Engineering Emmy Awards. For his research in video coding and transmission, he received numerous awards including the Vodafone Innovations Award, the EURASIP Group Technical Achievement Award, the Eduard Rhein Technology Award, the Karl Heinz Beckurts Award, the IEEE Masaru Ibuka Technical Field Award, and the IMTC Leadership Award. He received multiple best paper awards for his publications. Since 2014, Thomson Reuters named him in their list of "The World's Most Influential Scientific Minds" as one of the most cited researchers in his field. He is a recipient of the ITU150 Award.