



# Overview of Cross-Component In-Loop Filters in Video Coding Standards

LI Zhaoyu<sup>1</sup>, MENG Xuewei<sup>1</sup>, ZHANG Jiaqi<sup>1</sup>,  
HUANG Cheng<sup>2,3</sup>, JIA Chuanmin<sup>1</sup>, MA Siwei<sup>1</sup>, JIANG Yun<sup>1</sup>

(1. National Engineer Research Center of Visual Technology, Peking University, Beijing 100871, China;

2. ZTE Corporation, Shenzhen 518057, China;

3. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China)

DOI: 10.12142/ZTECOM.202502009

<https://kns.cnki.net/kcms/detail/34.1294.TN.20250512.1043.002.html>,  
published online May 13, 2025

Manuscript received: 2024-07-09

**Abstract:** In-loop filters have been comprehensively explored during the development of video coding standards due to their remarkable noise-reduction capabilities. In the early stage of video coding, in-loop filters, such as the deblocking filter, sample adaptive offset, and adaptive loop filter, were performed separately for each component. Recently, cross-component filters have been studied to improve chroma fidelity by exploiting correlations between the luma and chroma channels. This paper introduces the cross-component filters used in the state-of-the-art video coding standards, including the cross-component adaptive loop filter and cross-component sample adaptive offset. Cross-component filters aim to reduce compression artifacts based on the correlation between different components and provide more accurate pixel reconstruction values. We present their origin, development, and status in the current video coding standards. Finally, we conduct discussions on the further evolution of cross-component filters.

**Keywords:** cross-component in-loop filter; adaptive loop filter; sample adaptive offset; video coding

**Citation** (Format 1): LI Z Y, MENG X W, ZHANG J Q, et al. Overview of cross-component in-loop filters in video coding standards [J]. *ZTE Communications*, 2025, 23(2): 85 – 95. DOI: 10.12142/ZTECOM.202502009

**Citation** (Format 2): Z. Y. Li, X. W. Meng, J. Q. Zhang, et al., “Overview of cross-component in-loop filters in video coding standards,” *ZTE Communications*, vol. 23, no. 2, pp. 85 – 95, Jun. 2025. doi: 10.12142/ZTECOM.202502009.

## 1 Introduction

With the development of video capture, storage, compression, and display technologies, numerous video applications continue to emerge, such as video communications, online conferences, cloud gaming, and immersive video experiences. The advancement brings forth new challenges to video coding technologies. To meet the increasing demand for video compression, various video coding tools and technologies have been proposed, leading to continuous evolution in video coding standards. A significant milestone in this progression was the finalization of the high efficiency video coding (HEVC)<sup>[1]</sup> standard in 2013, which achieved approximately 50% bitrate savings compared with its predecessor, the advanced video coding (AVC) standard<sup>[2]</sup>. The latest video coding standard, versatile video coding (VVC)<sup>[3]</sup>, has further improved upon HEVC by achieving

roughly 50% bitrate reduction. While H.266/VVC demonstrates excellent video compression capabilities, there remains significant potential in further enhancing video coding efficiency. In the pursuit of exploring advanced video encoding tools, a software model named the enhanced compression model (ECM) has been introduced to explore the potential of video compression further<sup>[4]</sup>.

As a result of the prevalent utilization of block-based operations and coarse quantization within contemporary video coding standards, artifacts such as blocking and ringing have become inherent in compressed frames, thereby markedly diminishing both objective and subjective qualities. To mitigate these compression artifacts, extensive exploration has been conducted on in-loop filter algorithms during the evolution of video coding standards. These filters enhance the quality of reconstructed frames while furnishing high-fidelity reference frames for subsequent images, thereby facilitating more accurate motion compensation.

There are four kinds of in-loop filters in VVC<sup>[5]</sup>, i.e., the deblocking filter (DBF)<sup>[6]</sup>, the sample adaptive offset (SAO)<sup>[7]</sup>, the adaptive loop filter (ALF)<sup>[8]</sup>, and luma mapping with chroma

This work was supported in part by National Science Foundation of China under Grant No. 62031013, PCL-CMCC Foundation for Science and Innovation under Grant No. 2024ZY1C0040, New Cornerstone Science Foundation for the Explorer Prize, and High performance Computing Platform of Peking University.

scaling (LMCS)<sup>[9]</sup>. The bilateral filter (BIF)<sup>[10]</sup> has been newly adopted in ECM. These filters are depicted in Fig. 1. DBF aims to remove the blocking artifact by applying low-pass filters to the boundaries of the coding unit, the prediction unit, and the transform unit. SAO is conducted by conditionally adding an offset to the reconstructed samples after DBF, which shows promising performance in reducing the mean sample distortion and the ringing artifacts. ALF is a Wiener-based spatial filter. It enhances reconstructed video fidelity by taking the weighted average of reference samples as the filtered samples. The weighting coefficients are derived by minimizing the mean square error between the original and decoded samples in the encoder, and then they are transmitted to the decoder. LMCS does not particularly focus on artifact reduction but aims to boost coding efficiency by better utilizing the dynamic range. BIF is a nonlinear, edge-preserving, and noise-reducing filter that has been newly introduced to ECM. Similar to the ALF, it also replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. While the difference lies in that the weights of BIF depend on the Euclidean distance of pixels and the radiometric differences, which preserves sharp edges. These weights can be calculated both in the encoder and decoder.

In addition to the above-mentioned local filters adopted in the ECM, some other in-loop filters based on the image non-local similarity have been studied, such as a structure-driven adaptive non-local filter (SANF)<sup>[11]</sup>, a non-local structure-based loop filter (NLSF)<sup>[12-14]</sup>, a novel adaptive loop filter uti-

lizing image non-local prior knowledge<sup>[15]</sup>, a parametric non-local loop filter (PNLF)<sup>[16]</sup>, and a deformable Wiener filter (DWF)<sup>[17]</sup>. Some of these methods were also discussed in the joint video experts team (JVET) meetings<sup>[18-21]</sup>.

Though the aforementioned in-loop filters effectively reduce compression artifacts, these conventional methodologies, characterized by hand-crafted designs, exhibit constraints in addressing more intricate artifacts. In response to this constraint, in-loop filters leveraging convolutional neural networks (CNNs) have been developed, demonstrating superior performance over conventional filtering methods<sup>[22-25]</sup>. Various neural network-based loop filtering tools have been proposed and adopted by ECM, achieving significant performance improvement<sup>[26-29]</sup>.

While the coding techniques mentioned above only focus on single-component in-loop filtering, ignoring the correlation between different components. Extensive research has demonstrated a high correlation between luma and chroma components in the YUV format<sup>[30-33]</sup>. Based on this correlation, some prediction techniques were proposed, such as cross-component prediction (CCP)<sup>[34]</sup> supported in the HEVC range extensions and cross-component linear model (CCLM)<sup>[35]</sup>. Besides, cross-component techniques are also applied in end-to-end image compression<sup>[36]</sup>, which effectively improves compression performance. Recently, the correlation among different components was also considered in in-loop filters.

Several cross-component in-loop techniques were proposed and adopted in H.266/VVC and the audio video coding standard (AVS3), an independently developed Chinese audio-video coding standard. Continuous studies have been carried out on these methods during the development of ECM. In the ECM-12.0, there are two cross-component filters, namely the cross-component adaptive loop filter (CCALF) and the cross-component sample adaptive offset (CCSAO). CCALF was initially proposed and adopted during the development of H.266/VVC and was optimized and improved in ECM. Similar to ALF, CCALF is also a Wiener filter. The difference is that it only applies to chroma samples, and it utilizes luma samples as the reference samples and corrects the target chroma pixel by applying a linear filter to these selected luma samples. The filter parameters are trained following the principle of minimizing the mean square error (MSE) in the encoder and transmitted to the decoder. CCSAO is adopted by AVS3 and ECM. Specifically, it uses the correlation between luma and

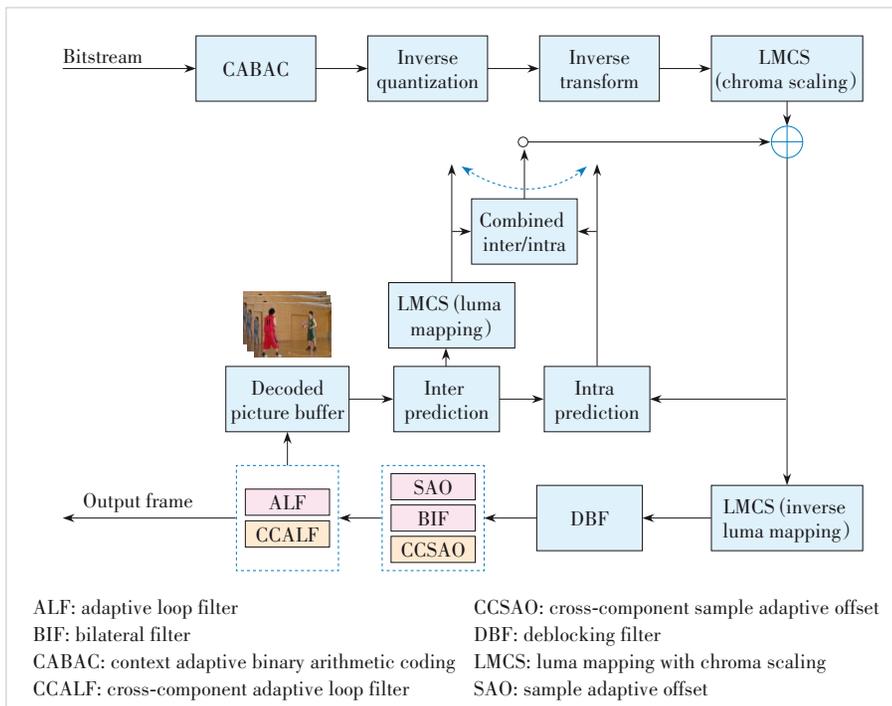


Figure 1. Illustration of the ECM video decoder diagram, with golden boxes corresponding to cross-component filters

chroma components to classify the reconstructed samples into different categories and assigns each category an offset value for sample adjustment.

Compared with ECM-12.0 without CCALF<sup>[4]</sup>, ECM-12.0 with CCALF achieves 2.49% and 2.90% coding gains for the Cb and Cr components under All Intra (AI) configuration, and 1.48% and 2.12% coding gains for Cb and Cr components under random access (RA) configuration. While in VTM-10.0, CCALF can achieve 13.88% and 13.73% coding gains under AI configuration, and 9.69% and 8.55% coding gains under RA configuration for Cb and Cr components respectively<sup>[37]</sup>. The decrease in the coding gain may be caused by the new cross-component techniques introduced in the prediction process of ECM. For CCSAO, 1.28% and 1.08% coding gains can be achieved for Cb and Cr components under AI configuration, and 3.02% and 2.79% coding gains for Cb and Cr components can be achieved under RA configuration, respectively.

The remainder of this paper is organized as follows. Section 2 introduces the theory of CCALF and summarizes its development. Section 3 introduces the fundamental principles and the proposals about CCSAO. Experimental results and discussions are shown in Section 4. Section 5 concludes this paper.

## 2 CCALF

CCALF is fundamentally a Wiener filter<sup>[38]</sup>. Specifically, CCALF derives a correction signal for chroma samples based on the weighted average of luma reference samples. These reference samples are the neighboring samples of the collocated luma sample. The coordinate of the collocated luma sample is derived based on the chroma format of the video. Both the ALF and CCALF use the reconstructed sample of SAO as input, while CCALF only calculates the offsets for chroma components as shown in Fig. 2. The filtering operation can be represented using the conditions below, and we assume the fol-

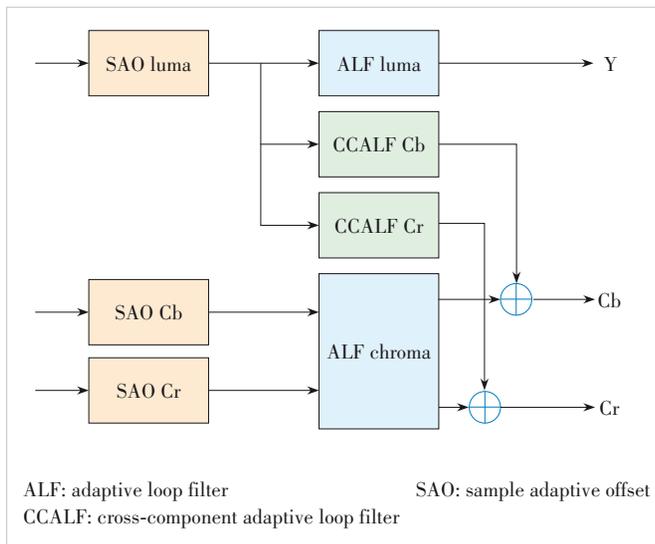


Figure 2. Illustration of CCALF

lowing for 2D images.

$$I'(r) = I(r) + \sum_{i=0}^{N-1} c_i p_i \quad (1),$$

$$p_i = L(r' + \mathbf{d}_i) - I(r) \quad (2),$$

where sample location  $r = (x, y)$  belongs to the to-be-filtered region  $R$ , and  $r' = (x', y')$  means the collocated luma sample position of the to-be-filtered chroma sample;  $s[r]$  is the original sample,  $I[r]$  is the to-be-filtered sample, and  $L(r')$  is the collocated luma samples of  $[r]$ ;  $\mathbf{c} = [c_0, c_1, c_2, \dots, c_{N-1}]$  means  $N$ -tap filter coefficients;  $\{\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{N-1}\}$  is the filter tap position offset, where  $\mathbf{d}_i$  denotes the sample location offset to  $L(r')$  of the  $i$ -th filter tap;  $\mathbf{p} = [p_0, p_1, p_2, \dots, p_{N-1}]$  shows the difference values between neighboring reference luma samples and the to-be-filtered chroma sample;  $I'(r)$  is the filtered chroma sample.

The coefficients of CCALF are derived by minimizing the mean square error between the reconstructed chroma component after SAO and the original chroma sample, similar to the parameter derivation process of chroma-ALF. Specifically, a correlation matrix is derived, and the coefficients are calculated using the Cholesky decomposition solver to minimize the mean square error.

The coefficient values at different positions are obtained from the bitstream. The filter coefficients are derived by solving the optimization problem shown in Eq. (3).

$$\min_c \sum_{r \in R} (\mathbf{c} \odot \mathbf{p} - s[r])^2 \quad (3),$$

$$\mathbf{c} = \mathbf{R}_{r,r}^{-1} \mathbf{R}_{r,s} \quad (4),$$

where  $\odot$  is the inner product. By solving the Wiener-Hopf equation as in Eq. (4), the filter coefficients can be calculated.  $\mathbf{R}_{r,r}^{-1}$  denotes the auto-correlation matrix of the to-be-filtered samples, and  $\mathbf{R}_{r,s}$  is the cross-correlation matrix of the to-be-filtered and the original samples.

### 2.1 Filter Shape

The filter shape of CCALF was a  $5 \times 6$  diamond-shaped filter with 14 filter coefficients and 18 taps when it was initially proposed<sup>[39]</sup>. Considering the trade-off among performance, line buffer, and computational complexity, several reduced filter shapes were proposed<sup>[40-43]</sup>. Finally, the  $3 \times 4$  diamond filter shape was adopted in H.266/VVC. Fig. 3 illustrates the relative location of the chroma sample being filtered and its support region in the luma sample when CCALF is adopted in H.266/VVC. Consequently, each CCALF filter has only 8 filter coefficients, and the filtering operation is shown in Eq. (1), where  $N = 8$ .

To improve the performance of CCALF, numerous propos-

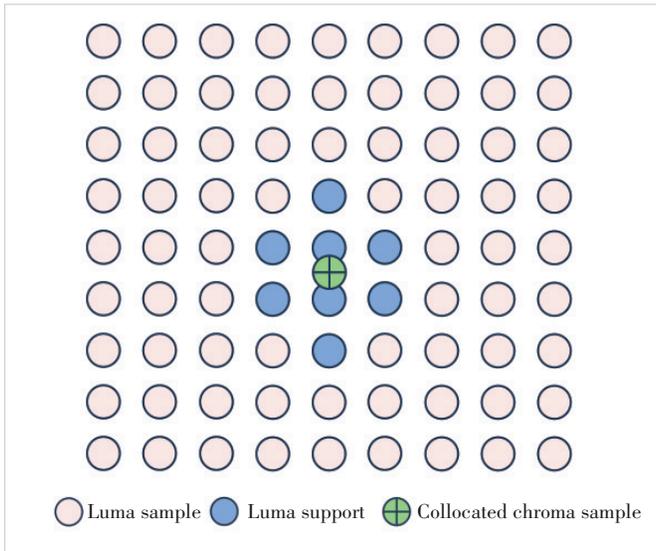


Figure 3. Illustration of the relative location of filtered chroma sample of CCALF and its support in the luma channel for 4 : 2 : 0 chroma format in H.266/VVC

als have been put forward, considering the trade-off between performance and running time, complexity, and other factors. However, some of these proposals were not adopted.

In Ref. [44], an extension to CCALF was proposed. This contribution suggests extensions to CCALF in both the number and size of filters. While this extension can enhance chroma components, it may lead to some loss in the luma component. Additionally, CCALF could introduce artifacts in chroma components, which is why certain constraints are set in high quantization parameter (QP) regions. Therefore, proposals regarding CCALF must avoid reintroducing these artifacts.

Considering that the correlation between neighboring pixels may depend on the characteristics of the video content, a single filter shape may not be optimal for different video content. A coding tree block (CTB) level filter shape selection scheme was proposed to optimize the CCALF framework<sup>[45]</sup>. This contribution introduces two filter shapes shown in Fig. 4. Within each adaptation parameter set (APS), multiple filters and their corresponding shapes with coefficients are signaled.

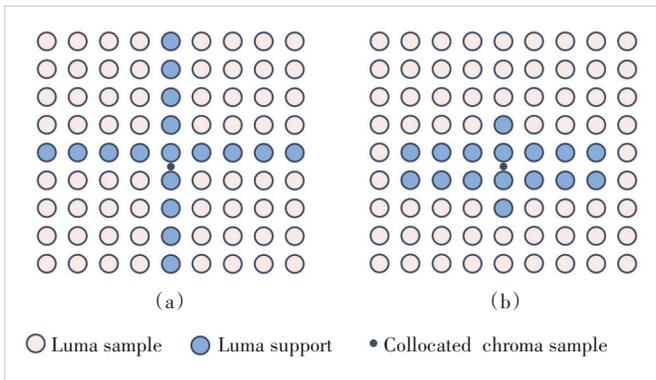


Figure 4. Illustration of the two filter shapes of CCALF in Ref. [45]

For each CTB, the decoder specifies which filter shapes or coefficients are used based on the signaled index.

This contribution demonstrates significant gains in chroma components. However, the necessity of adaptively selecting CCALF shapes is questioned. In a subsequent exploration experience<sup>[46]</sup>, in addition to the adaptive selection of the two filter shapes<sup>[45]</sup>, another scheme involving larger-size filters was proposed. Specifically, a 25-tap long-tap CCALF was introduced. This long-tap filter was considered a simpler scheme to achieve better gain. After joint tests of the modified CCALF and other in-loop filters<sup>[47]</sup>, the long-tap CCALF scheme was eventually adopted. The new shape of CCALF in ECM is illustrated in Fig. 5, and the filtering operation is shown in Eq. (1) where  $N = 25$ .

Because residual values have been stored and used in luma ALF, the concept of residual-based taps in chroma ALF and CCALF was proposed<sup>[48]</sup>. Before this contribution, CCALF only had one online-trained CCALF filter with a cross-like filter shape mentioned above, as depicted in Fig. 5. Since the residual values are utilized in the unfixed luma filter of ALF, there is no need to store luma residual values additionally. In this contribution, only one luma-residual-based tap was added. Furthermore, chroma residual values were incorporated into the chroma online-trained filter of ALF, while luma residual values were employed in CCALF. However, considering that chroma residual values were not stored previously and the additional memory required, the resulting gain was comparatively low. Therefore, this proposal is recommended for further study.

At the 31st JVET meeting, luma residual taps in chroma ALF and CCALF were introduced<sup>[49]</sup>. Five luma residual taps in a cross 3×3 shape were added. These extended taps took the collocated and neighboring luma residual values as input. The inclusion of the luma residual taps in CCALF was adopted due to its relatively higher standalone gain<sup>[50]</sup>. The filter shape of CCALF in ECM-12.0 is illustrated in Fig. 6.

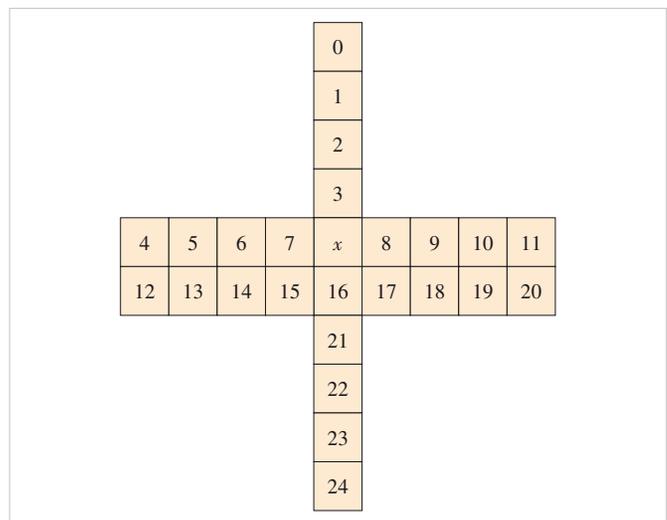


Figure 5. Illustration of the filter shape of CCALF with 25 taps

The coefficients that need to be calculated are divided into two parts: spatial luma sample-based taps and luma residual-based taps. The linear filtering operation can be represented using Eq. (5).

$$I'(x,y) = I(x,y) + \sum_{i=0,12} c_i(f_{i,0} + f_{i,1}) + \sum_{i=1}^{21} c_i f_i + \sum_{i=22}^{26} c_i g_i \quad (5)$$

$$f_{i,j} = L(x' + x_i, y' + y_j) - I(x,y) \quad (6)$$

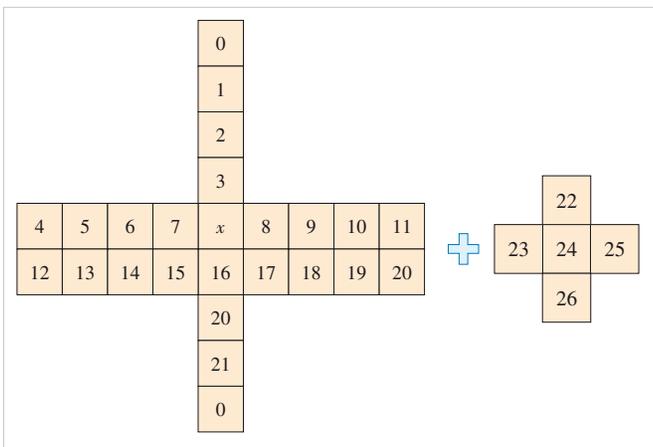
$$g_i = \text{Clip}(R(x' + x_i, y' + y_j)) \quad (7)$$

where  $(x, y)$  is the coordinate of the center sample, and  $(x', y')$  is the coordinate of collocated luma sample;  $(x' + x_i, y' + y_j)$  and  $(x' + x_{ij}, y' + y_{ij})$  are the coordinates of the reconstructed samples corresponding to coefficient  $c_i$ ;  $f_{i,j}$  and  $f_i$  show the difference between neighboring luma samples  $L(x', y')$  and current sample  $I(x, y)$ ;  $g_i$  is the clipped value of luma residual samples  $R(x', y')$ , which is the residual between prediction samples and reconstructed samples; Clip is the function that limits the values within a certain range to reduce the impact of significant differences in sample values, and the value of the clipping operation depends on the clipIdx of APS and bitDepth.

## 2.2 Filter Coefficient Calculation and Representation

Except for the filter shape of the CCALF, the optimization of coefficient calculation and signaling<sup>[39, 51-54]</sup> is important to improve the performance of CCALF.

When CCALF was proposed, each filter had 14 filter coefficients and 18 taps, and every coefficient had an 8-bit dynamic



**Figure 6.** Illustration of CCALF's shape at ECM-12.0 (the left cross-like filter uses the reconstructed spatial sample of luma sample adaptive offset as input with 23 taps, and the right one uses luma residual samples as input)

range and was signaled with a third-order exponential-Golomb code<sup>[39]</sup>. However, it would increase complexity with additional multiplications per chroma pixels. To simplify the computation overhead, a bit shifting scheme was proposed to replace the multiplications<sup>[51]</sup>. The results show that this scheme can reduce the complexity of the CCALF filter with an accepted loss, so it was adopted. Besides, a contribution was proposed to reduce memory access, encoding latency, and power consumption<sup>[54]</sup>. It proposes a method to estimate CCALF filtering distortion without conducting real filter operations. With this proposal, the number of encoding passes can be reduced from 152 to 1 without affecting the coding performance. As a desirable simplification, this proposal was adopted.

At the 32nd JVET meeting, coefficient precision adjustment for ALF was proposed, demonstrating promising coding performance with negligible increases in encoding and decoding time<sup>[55]</sup>. Similarly, at the 33rd JVET meeting, adaptive coefficient precision for CCALF was introduced<sup>[56-57]</sup>. Since CCALF involves different coefficient derivations compared with ALF, removing the power of 2 constraints was also proposed in this context. This adjustment can enhance the accuracy of coefficients, though a 2-bit syntax element needs to be signaled for per luma filter set to indicate the number of bits. These two contributions have been further investigated.

## 2.3 Syntax Design

Compared with H. 266/VVC, ECM-12.0 utilizes luma residual samples additionally, as shown in Fig. 6. The residual correction is generated for chroma samples according to Eq. (5). For each picture, two types of information need to be coded for CCALF, i.e., filter coefficient parameters and filter control on/off flags. The filter coefficient parameters include the number of cross-component filters and the coefficients of the corresponding filter. CCALF can transmit up to 8 CCALF filters, with the resulting filters being indicated for each of the two chroma channels on a CTU basis. Each slice only has one APS, and the Cb component and Cr component can have different APSs, which are signaled separately at the slice header. Similar to luma ALF, to reduce bit overhead, filter coefficients of different classifications can be merged. The filter control on/off flags enable better local adaptation, with hierarchical control at the sequence-level, picture-level, slice-level and CTU-level. When the value of sequence-level and picture-level control flags is not present, it is inferred to be equal to 0. When the slice-level on/off control flag is not present, it is inferred to be equal to picture-level on/off control flags. If the slice-level on/off control flag indicates ALF-on, CTU-level filter on/off control flags are interleaved in slice data and coded with CTUs; otherwise, no additional CTU-level filter on/off control flags are coded and all CTUs of the slice are inferred as ALF-off.

Due to the abundant texture features of the luma component, CCALF may introduce artifacts with overly abundant chroma texture, thereby reducing the subjective quality of the

image, especially at high QP. Therefore, the H.266/VVC reference encoder can achieve subjective tuning through configuration file adjustments. Specifically, it can attenuate the application of CCALF in high QP encoding and areas with high-frequency luminance. Algorithmically, CCALF is deactivated on CTUs when any of the following conditions is true:

- 1) The slice QP value minus 1 is less than or equal to the base QP value;
- 2) The number of chroma samples exhibiting local contrast exceeding  $(1 \llcorner (\text{bitDepth} - 2)) - 1$  surpasses the CTU height, where the local contrast is the difference between the maximum and minimum luma sample values within the filter support region;
- 3) More than a quarter of chroma samples are in the range between  $(1 \llcorner (\text{bitDepth} - 1)) - 16$  and  $(1 \llcorner (\text{bitDepth} - 1)) + 16$ .

### 3 CCSAO

CCSAO is conceptually similar to SAO, as it initially classifies the samples to be filtered into different categories, then derives an offset value for each category, and finally corrects the pixels in that category with the corresponding offset value. It uses the reconstructed sample of DBF, which is the same as SAO, and the offsets are derived for three channels respectively. The reconstruction operation of CCSAO can be represented by the equation below.

$$C'_{\text{rec}} = \text{Clip}(C_{\text{rec}} + \text{offset}_i) \tag{8}$$

where  $C_{\text{rec}}$  and  $C'_{\text{rec}}$  are the reconstructed samples after DBF and CCSAO, respectively,  $i$  represents the class index of the corresponding sample, and  $\text{offset}_i$  is the corresponding offset value.

The difference between SAO and CCSAO lies in CCSAO's utilization of the strong correlation between the luma and chroma components in the classification process. It optimizes the reconstruction of one component of the sample by leveraging the information contained in the other component of the sample<sup>[58]</sup>.

#### 3.1 Classifier Extension

The original CCSAO includes only a classification based on band information to avoid a significant increase in complexity. Corresponding band offsets are obtained by minimizing the sum of squared error (SSE) between the original sample and the corrected reconstruction sample. This approach keeps computational complexity low while enabling CCSAO to handle certain encoded artifacts. It should be noted that the offsets need to be signaled in the bitstream.

CCSAO is applied to the output of DBF reconstructed samples, and the offset calculated for each category is added to the output sample from the SAO process. Therefore, CCSAO can be parallelized with SAO, as shown in Fig. 7.

The band information-based classification of CCSAO utilizes the reconstructed sample of three components to process the classification for each component. Specifically, the collocated samples for each component are first selected. Then, an index representing a category is calculated based on the band number of the three components and their collocated samples. The offset value of a sample depends on its category. Regarding the collocated samples for each component, the collocated luma sample can be chosen from 9 candidates, while the collocated chroma samples have fixed positions, as shown in Fig. 8.

CCSAO was first proposed and adopted<sup>[59]</sup> in the AVS3 video coding standard, in which collocated luma component samples are classified by equally dividing the range of the sample values. For each category, an offset value is derived and used for the chroma samples whose collocated luma sample belongs to the category.

Although cross-component tools in in-loop filters always act on chroma components, regarding cross-component proposals, attention should not only be given to the gain of chroma components but also to the effects on the luma component. Furthermore, subjective quality improvement needs to be considered as well. Considering these reasons, CCSAO was introduced to ECM. This proposal showed great performance im-

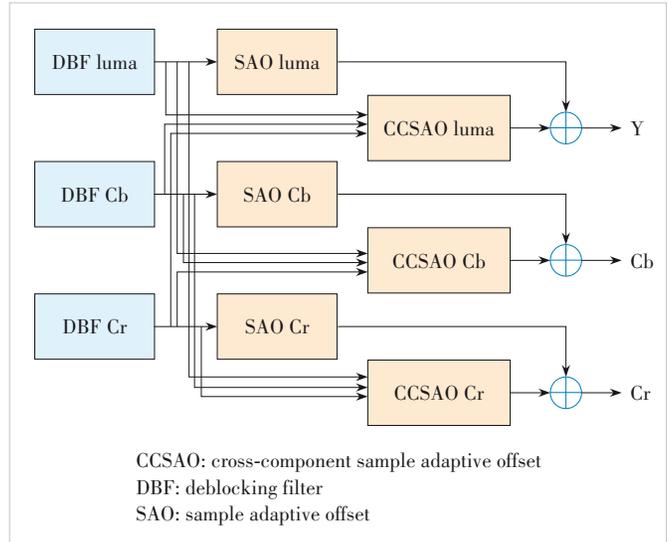


Figure 7. Illustration of SAO process when CCSAO is applied

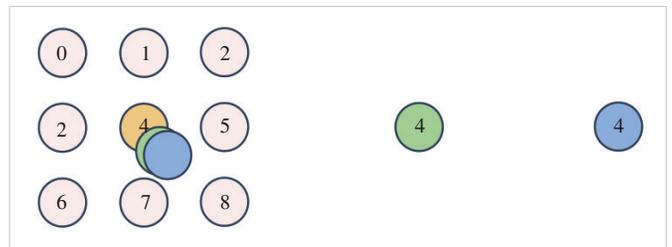


Figure 8. Illustration of the collocated sample used for the CCSAO classification. The left graph shows the 9 locations of the luma component (one of the 9 samples will be chosen based on rate-distortion optimization) and the green and blue samples show the two collocated chroma samples

provement in chroma components while introducing minimal loss in the luma component. Initially, CCSAO only used the band classifier when it was adopted in ECM<sup>[60]</sup>, and the category index is calculated using the equations below.

$$\text{classIndex} = \text{BandNum} \times (N_{\text{cb}} \times N_{\text{cr}}) + \text{band}_{\text{cb}} \times N_{\text{cr}} + \text{band}_{\text{cr}} \quad (9),$$

$$\text{band}_L = P(x_Y, y_Y) \times N_Y \gg \text{BitDepth} \quad (10),$$

$$\text{band}_{\text{cb}} = P(x_{\text{cb}}, y_{\text{cr}}) \times N_{\text{cb}} \gg \text{BitDepth} \quad (11),$$

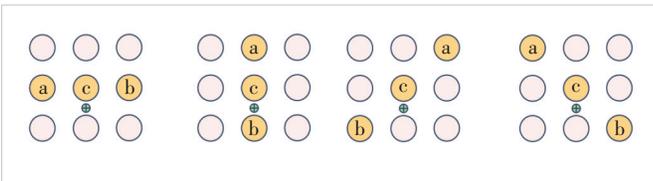
$$\text{band}_{\text{cr}} = P(x_{\text{cr}}, y_{\text{cr}}) \times N_{\text{cr}} \gg \text{BitDepth} \quad (12),$$

where  $P(i, j)$  is the sample value of different components at position  $(i, j)$ ,  $N_i$  is the number of band for each component,  $(x_{\text{cb}}, y_{\text{cb}})$  and  $(x_{\text{cr}}, y_{\text{cr}})$  are the current chroma sample positions, and  $(x_Y, y_Y)$  is the collocated luma sample position.

As a new in-loop filter tool, several schemes have been proposed to optimize the original CCSAO. An extension of CCSAO was proposed at the 24th JVET meeting, where the proponents extended the design of CCSAO by adding the edge-based classifier<sup>[61-62]</sup>. Similar to the edge-based classification method in SAO, the edge-based classification of CCSAO also uses four 1-D directional patterns, including horizontal, vertical, 45°, and 135°, as shown in Fig. 9. The best direction mode is determined at the encoder through rate-distortion optimization (RDO). Edge information used for classification is derived by calculating the difference between the center pixel and its two adjacent pixels, and then comparing the difference with a predefined threshold value to derive the final class index. The best threshold values are also selected from an array of predefined threshold values based on RDO. If the edge-based classifier is selected, the category index will be calculated as follows, given the chroma sample and the collocated luma samples.

$$\text{classIndex} = \text{BandNum} \times 16 + q_a \times 4 + q_b \quad (13),$$

$$q_i = \begin{cases} 0 & d_i < -Th \\ 1 & -Th < d_i < 0 \\ 2 & 0 < d_i < Th \\ 3 & Th < d_i \end{cases} \quad (14),$$



**Figure 9. Illustration of the edge-based classification of CCSAO. Four graphs show four different directions, where the yellow samples are the locations used for calculating the class index at different directional patterns**

$$\text{BandNum} = \text{cur}_i \times N_i \gg \text{BitDepth} \quad (15),$$

where  $i$  can be chosen from the two co-located samples based on RDO,  $d_i$  is the delta value between the center sample  $c$  and the neighboring sample  $a$  or  $b$ .  $q_i$  is the quantized value of  $d_i$ . The position of neighboring sample  $a$  or  $b$  depends on the best 1-D directional pattern selected from the four 1-D directional patterns. Besides, the offset value is constrained to the range of  $[-15, 15]$  and these offsets need to be transmitted to the decoder.

Unlike SAO, the edge-based classifier in CCSAO combines the luma edge and the band index of the sample at the corresponding collocated position to determine the final classification of a given sample. Additionally, CCSAO uses collocated luma samples to derive edge information for chroma samples, while SAO uses neighboring samples of the same component to derive edge information.

A similar contribution was introduced to AVS<sup>[63]</sup>, where the enhanced cross-component sample adaptive offset (ECCSAO) method further improves encoding performance, which extends the edge-based classification by using the edge information of collocated luma samples to classify chroma samples. Moreover, a four-layer quad-tree structure was proposed. The former method has been adopted by AVS.

In the ECM, the edge classifier was further optimized with more edge/band combinations, and the component used for edge classification can be selected from any of the three components<sup>[64-65]</sup>. The new edge-based classification scheme, a subset of the original one with fewer edge range divisions, was added. This allows for more flexible edge/band combinations to adapt to the local characteristics of video sequences. This contribution was adopted at the 31st JVET meeting. The second edge-based classifier is formulated as follows.

$$\text{classIndex} = \text{BandNum} \times 4 + q_a \times 2 + q_b \quad (16),$$

$$q_i = \begin{cases} 0 & d_i < Th \\ 1 & d_i \geq T \end{cases} \quad (17).$$

### 3.2 Signaling Overhead Reduction

Similar to the APS design in H.266/VVC, the inheritance scheme of CCSAO was also proposed<sup>[64-65]</sup>. There is a strong correlation between the CCSAO offsets and classifier parameters of different pictures. To reduce signaling overhead, the offsets/parameters of some coded pictures can be stored at both the encoder and decoder, allowing them to be used by future pictures. This contribution has also been adopted.

## 4 Performance Evaluation

To improve the coding performance, both CCALF and CCSAO are integrated into ECM-12.0 seamlessly. A comparative analysis is conducted to evaluate the efficiency and effec-

tiveness of the cross-component in-loop filter tools. With continuous development, both CCSAO and CCALF have achieved remarkable performance gains. To evaluate the coding performance of CCALF and CCSAO, ECM-12.0 without CCALF and CCSAO are regarded as the anchor respectively<sup>[66]</sup>.

As shown in Table 1, CCALF can achieve 2.49% and 2.90% coding gains for Cb and Cr components under AI configuration. For RA configuration, 1.48% and 2.12% coding gains for Cb and Cr components can be achieved. In VTM-10.0, CCALF can achieve 13.88% and 13.73% coding gains for Cb and Cr components under AI configuration, and 9.69% and 8.55% coding gains for Cb and Cr components under RA configuration. The decrease in gain may be caused by the newly proposed and optimized cross-component techniques in the prediction process. For CCSAO, as shown in Table 2, 1.28% and 1.08% coding gains can be achieved for Cb and Cr components under AI configuration. For RA configuration,

**Table 1. Experimental results of ECM-12.0 (anchor: ECM-12.0 without CCALF)**

Class	AI			RA		
	Y	Cb	Cr	Y	Cb	Cr
A1	0.09%	-1.21%	-3.32%	0.07%	-1.00%	-3.96%
A2	0.11%	-2.78%	-3.23%	0.13%	-2.62%	-4.94%
B	0.12%	-3.35%	-3.22%	0.15%	-4.31%	-3.41%
C	0.10%	-1.67%	-1.91%	0.03%	-1.48%	-2.12%
E	0.15%	-3.12%	-2.96%	-	-	-
Average	0.11%	-2.49%	-2.90%	0.10%	-2.56%	-3.48%
D	0.02%	-0.42%	-0.18%	-0.01%	-0.94%	-0.53%
F	0.10%	-1.77%	-1.07%	0.15%	-1.08%	-0.32%
TGM	0.12%	-1.19%	-0.72%	0.16%	-1.26%	-1.03%

AI: All Intra

CCALF: cross-component adaptive loop filter

ECM: enhanced compression model

RA: random access

TGM: text and graphics with motion

**Table 2. Experimental results of ECM-12.0 (anchor: ECM-12.0 without CCSAO)**

Class	AI			RA		
	Y	Cb	Cr	Y	Cb	Cr
A1	-0.28%	-0.83%	-1.36%	-0.42%	-1.89%	-2.42%
A2	0.01%	-0.99%	-1.15%	-0.06%	-1.88%	-2.01%
B	0.08%	-1.94%	-1.63%	-0.16%	-3.76%	-4.07%
C	0.11%	-0.83%	-0.41%	0.00%	-2.10%	-1.20%
E	0.02%	-1.55%	-0.68%	-	-	-
Average	0.01%	-1.28%	-1.08%	-0.15%	-2.57%	-2.56%
D	0.03%	-0.02%	-0.31%	0.10%	-1.56%	-1.05%
F	-0.23%	-1.99%	-1.74%	-0.15%	-2.99%	-1.54%
TGM	-0.73%	-1.64%	-1.81%	-1.01%	-2.72%	-3.38%

AI: All Intra

CCSAO: cross-component sample adaptive offset

ECM: enhanced compression model

RA: random access

TGM: text and graphics with motion

3.02% and 2.79% coding gains for Cb and Cr components can be achieved. It can be noted that the Y component coding performance of CCSAO on screen content sequences is significantly greater than that of natural sequences. This may be caused by the more obvious relationship between the texture and directional features of luma and chroma components in screen content videos.

Furthermore, we compare the subjective performance under different configurations. The subjective testing materials consist of the sequences mentioned in the common test conditions (CTC), with each sequence encoded using four QPs (QP = 22, 27, 32, and 37) under the RA configuration. Partial visual quality comparison results of reconstructed sequences are shown in Fig. 10, where the first column displays decoded images with both CCALF and CCSAO applied, the second column shows decoded images without CCALF, and the last column presents reconstructed images without CCSAO. Red boxes highlight regions with significant subjective improvement. The lines on the clothes are clearer in Fig. 10a, whereas the color and lines in Figs. 10b and 10c appear slightly blurry. Compared with Fig. 10e, the boundaries of the clothes in Fig. 10d are more distinct. The lines in Fig. 10d are cleaner than those in Fig. 10f. Additionally, the wires in Fig. 10g are more coherent and clearer compared with Figs. 10h and 10j.

Building upon the demonstrated performance gains of CCALF and CCSAO, it's important to consider the broader context of loop filter development. Loop filters are designed to correct artifacts introduced prior to loop filtering. Different types of loop filters address various artifacts such as blocking, ringing, blurring, and mosquito noise. In VVC, there are three primary loop filters: DBF, SAO, and ALF. Moreover, the CCALF is integrated with ALF to fully utilize the relationship between luma and chroma components. To further exploit the cross-component relationship, an additional cross-component loop filter, CCSAO, has been proposed during the ECM exploration. CCSAO operates in parallel with SAO. With the advancement of ECM, the classifiers of CCSAO have become more refined and diverse<sup>[61-65]</sup>. Concurrently, the structure of CCALF has evolved to be more complex and comprehensive, incorporating a wider variety of samples into its filters<sup>[48-49]</sup>. Moreover, the shape and calculation methods of the filters are continuously optimized<sup>[40-43, 51, 56]</sup>. In addition, other in-loop filters based on image non-local similarity have been studied<sup>[11-12, 15-16]</sup>. Traditional loop filters in existing video coding standards primarily focus on local correlations. While non-local loop filters can offer performance gains, their high computational demands and hardware limitations make it challenging to implement in video coding standards. Therefore, methods to optimize non-local filters are proposed<sup>[13-14]</sup>. Overall, many new filtering tools are currently being explored. However, further investigation into the relationship among different components remains a crucial direction for video coding.



**Figure 10. Illustration of subjective quality comparison.** (a) – (c): BasketballDrill, RA configuration and QP22; (d) – (f): BQMall, RA configuration and QP22; (g) – (i): MarketPlace, random access configuration and QP32

## 5 Conclusions

Cross-component filters play a crucial role in future video coding standards. By leveraging the correlation between luma and chroma components, cross-component filters can achieve substantial coding performance improvement, leading to the adoption of various video coding standards such as VVC and AVS3. Compression distortion can be effectively mitigated, thereby improving the accuracy of the reconstructed pixel. Nevertheless, the philosophy of current cross-component filters primarily emphasizes utilizing luma information to refine chroma pixels, which neglects the potential impact of chroma information on luma pixels and the correlation between two chroma components. In some scenarios, the chroma texture information and edge details can also contribute to correcting luma inaccuracies. Therefore, cross-component filters still have the potential to achieve substantial performance improvement by delving into the filtering manner and relationship among different channels.

## References

- [1] SULLIVAN G J, OHM J R, HAN W J, et al. Overview of the high efficiency video coding (HEVC) standard [J]. *IEEE transactions on circuits and systems for video technology*, 2012, 22(12): 1649 – 1668. DOI: 10.1109/TCSVT.2012.2221191
- [2] WIEGAND T, SULLIVAN G J, BJONTEGAARD G, et al. Overview of the H.264/AVC video coding standard [J]. *IEEE transactions on circuits and systems for video technology*, 2003, 13(7): 560 – 576. DOI: 10.1109/TCSVT.2003.815165
- [3] BROSS B, WANG Y K, YE Y, et al. Overview of the versatile video coding (VVC) standard and its applications [J]. *IEEE transactions on circuits and systems for video technology*, 2021, 31(10): 3736 – 3764
- [4] COBAN M, LIAO R L, NASER K, et al. Algorithm description of enhanced compression model 12 (ECM 12) [R]. Joint Video Experts Team (JVET), document JVET-AG2025, 2024
- [5] KARCZEWICZ M, HU N, TAQUET J, et al. VVC in-loop filters [J]. *IEEE transactions on circuits and systems for video technology*, 2021, 31(10): 3907 – 3925. DOI: 10.1109/tcsvt.2021.3072297
- [6] ANDERSSON K, MISRA K, IKEDA M, et al. Deblocking filtering in VVC [C]//Picture Coding Symposium (PCS). IEEE, 2021: 1 – 5. DOI: 10.1109/pcs50896.2021.9477477
- [7] FU C M, ALSHINA E, ALSHIN A, et al. Sample adaptive offset in the HEVC standard [J]. *IEEE transactions on circuits and systems for video technology*, 2012, 22(12): 1755 – 1764. DOI: 10.1109/

- TCSVT.2012.2221529
- [8] TSAI C Y, CHEN C Y, YAMAKAGE T, et al. Adaptive loop filtering for video coding [J]. *IEEE journal of selected topics in signal processing*, 2013, 7(6): 934 - 945. DOI: 10.1109/JSTSP.2013.2271974
- [9] LU T R, PU F J, YIN P, et al. Luma mapping with chroma scaling in versatile video coding [C]//Data Compression Conference (DCC). IEEE, 2020: 193 - 202. DOI: 10.1109/DCC47342.2020.00027
- [10] STRÖM J, WENNERSTEN P, ENHORN J, et al. Bilateral loop filter in combination with SAO [C]//Picture Coding Symposium (PCS). IEEE, 2019: 1 - 5. DOI: 10.1109/PCS48520.2019.8954554
- [11] ZHANG J, JIA C M, ZHANG N, et al. Structure-driven adaptive non-local filter for high efficiency video coding (HEVC) [C]//Data Compression Conference (DCC). IEEE, 2016: 91 - 100. DOI: 10.1109/DCC.2016.105
- [12] MA S W, ZHANG X F, ZHANG J, et al. Nonlocal in-loop filter: the way toward next-generation video coding? [J]. *IEEE multimedia*, 2016, 23(2): 16 - 26. DOI: 10.1109/MMUL.2016.16
- [13] MENG X W, JIA C M, WANG S S, et al. Optimized non-local in-loop filter for video coding [C]//Picture Coding Symposium (PCS). IEEE, 2018: 233 - 237. DOI: 10.1109/PCS.2018.8456299
- [14] JIA C M, LUO F L, ZHANG X F, et al. Fast non-local adaptive in-loop filter optimization on GPU [J]. *IEEE transactions on multimedia*, 2020, 23: 39 - 51. DOI: 10.1109/TMM.2020.2981185
- [15] ZHANG X F, XIONG R Q, LIN W S, et al. Low-rank-based nonlocal adaptive loop filter for high-efficiency video compression [J]. *IEEE transactions on circuits and systems for video technology*, 2017, 27(10): 2177 - 2188. DOI: 10.1109/TCSVT.2016.2581618
- [16] MENG X W, JIA C M, ZHANG X F, et al. Parametric non-local in-loop filter for future video coding [C]//Data Compression Conference (DCC). IEEE, 2022: 474. DOI: 10.1109/DCC52660.2022.00085
- [17] MENG X W, JIA C M, ZHANG X F, et al. Deformable Wiener filter for future video coding [J]. *IEEE transactions on image processing*, 2022, 31: 7222 - 7236. DOI: 10.1109/TIP.2022.3221278
- [18] WANG Z, MENG X, JIA C, et al. Description of SDR video coding technology proposal by DJI and Peking University [R]. Joint Video Experts Team (JVET), document JVET-J0011, 2018
- [19] MENG X, JIA C, WANG Z, et al. CE2: non-local structure-based filter [R]. Joint Video Experts Team (JVET), document JVET-K0160, 2018
- [20] CHERNYAK R, Victor S, Ikonin S, et al. CE2: noise suppression filter [R]. Joint Video Experts Team (JVET), document JVET-K0053, 2018
- [21] LAI C Y, CHEN C Y, HUANG Y W, et al. CE2.5.2: nonlocal mean in-loop filter [R]. Joint Video Experts Team (JVET), document JVET-K0236, 2018
- [22] ZHANG K, ZUO W M, CHEN Y J, et al. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising [J]. *IEEE transactions on image processing*, 2017, 26(7): 3142 - 3155. DOI: 10.1109/TIP.2017.2662206
- [23] WANG M Z, WAN S, GONG H, et al. Attention-based dual-scale CNN in-loop filter for versatile video coding [J]. *IEEE access*, 2019, 7: 145214 - 145226
- [24] CHEN S J, CHEN Z Z, WANG Y B, et al. In-loop filter with dense residual convolutional neural network for VVC [C]//IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). IEEE, 2020: 149-152. DOI: 10.1109/mipr49039.2020.00038
- [25] HUANG Z J, GUO X P, SHANG M Y, et al. An efficient QP variable convolutional neural network based in-loop filter for intra coding [C]//Data Compression Conference (DCC). IEEE, 2021: 33 - 42. DOI: 10.1109/DCC50243.2021.00011
- [26] LI Y, ZHANG K, LI J, et al. EE1-1.6: deep in-loop filter with fixed point implementation [R]. Joint Video Experts Team (JVET), document JVET-AA0111, 2022
- [27] LI J, LI Y, ZHANG K, et al. EE1-1.6: RDO considering deep in-loop filtering [R]. Joint Video Experts Team (JVET), document JVET-AB0068, 2022
- [28] LIU D, STRÖM J, DAMGHANIAN M, P. et al. EE1-1.5: combined intra and inter models for luma and chroma [R]. Joint Video Experts Team (JVET), document JVET-AC0089, 2023
- [29] LI Y, ZHANG K, ZHANG L. EE1-1.7: deep in-loop filter with additional input information [R]. Joint Video Experts Team (JVET), document JVET-AC0177, 2023
- [30] GOFFMAN-VINOPAL L, PORAT M. Color image compression using inter-color correlation [C]//International Conference on Image Processing. IEEE, 2002: II. DOI: 10.1109/ICIP.2002.1039960
- [31] SONG B C, LEE Y G, KIM N H. Block adaptive inter-color compensation algorithm for RGB 4 : 4 : 4 video coding [J]. *IEEE transactions on circuits and systems for video technology*, 2008, 18(10): 1447 - 1451. DOI: 10.1109/TCSVT.2008.2002827
- [32] LEE S H, CHO N I. Intra prediction method based on the linear relationship between the channels for YUV 4 : 2 : 0 intra coding [C]//The 16th IEEE International Conference on Image Processing (ICIP). IEEE, 2009: 1037 - 1040. DOI: 10.1109/icip.2009.5413727
- [33] LI J R, WANG M, ZHANG L, et al. Sub-sampled cross-component prediction for emerging video coding standards [J]. *IEEE transactions on image processing*, 2021, 30: 7305 - 7316. DOI: 10.1109/TIP.2021.3104191
- [34] FLYNN D, MARPE D, NACCARI M, et al. Overview of the range extensions for the HEVC standard: tools, profiles, and performance [J]. *IEEE transactions on circuits and systems for video technology*, 2016, 26(1): 4 - 19
- [35] PFAFF J, FILIPPOV A, LIU S, et al. Intra prediction and mode coding in VVC [J]. *IEEE transactions on circuits and systems for video technology*, 2021, 31(10): 3834 - 3847
- [36] DUAN W H, CHANG Z, JIA C M, et al. Learned image compression using cross-component attention mechanism [J]. *IEEE transactions on image processing*, 2023, 32: 5478 - 5493. DOI: 10.1109/TIP.2023.3319275
- [37] CHIEN W J, BOYCE J, CHEN W, et al. JVET AHG report: tool reporting procedure (AHG13) [R]. Joint Video Experts Team (JVET), document JVET-T0013, 2020
- [38] MISRA K, BOSSEN F, SEGALL A. On cross component adaptive loop filter for video compression [C]//Picture Coding Symposium (PCS). IEEE, 2019: 1 - 5. DOI: 10.1109/PCS48520.2019.8954547
- [39] MISRA K, BOSSEN F, SEGALL A. Cross-component adaptive loop filter for chroma [R]. Joint Video Experts Team (JVET), document JVET-00636, 2019
- [40] HU N, DONG J, SEREGIN V, et al. CE5-related: reduced filter shape for cross component adaptive loop filter [R]. Joint Video Experts Team (JVET), document JVET-P0558, 2019
- [41] KOTRA A M, ESENLİK S, WANG B, et al. AHG16/CE5-Related: simplifications for cross component adaptive loop filter [R]. Joint Video Experts Team (JVET), document JVET-P0106, 2019
- [42] LI J Y, LIM C S. AHG16/Non-CE5: cross component ALF simplification [R]. Joint Video Experts Team (JVET), document JVET-P0173, 2019
- [43] ZHAO Y, YANG H T. CE5-related: simplified CCALF with 6 filter coefficients [R]. Joint Video Experts Team (JVET), document JVET-P0251, 2019
- [44] KEATING S, BROWNE A, SHARMAN K. AHG12: extensions to CCALF [R]. Joint Video Experts Team (JVET), document JVET-V0080, 2021
- [45] SARWE M G, LIAO R L, CHEN J, et al. AHG12: CTB level filter shape selection of CCALF [R]. Joint Video Experts Team (JVET), document JVET-W0079, 2021
- [46] SARWER M G, LIAO R L, CHEN J, et al. EE2-4.2, EE2-4.3: on CCALF filter [R]. Joint Video Experts Team (JVET), document JVET-X0045, 2021
- [47] HU N, SEREGIN V, KARCZEWICZ M, et al. EE2-4.8: joint tests of chroma BIF, ALF and CCALF [R]. Joint Video Experts Team (JVET), document JVET-X0071, 2021

- [48] YIN W B, ZHANG K, ZHANG L. Non-EE2: extensions of residual-based taps in ALF and CCALF [R]. Joint Video Experts Team (JVET), document JVET-AD0234, 2023
- [49] YIN W B, ZHANG K, DENG Z P, et al. Non-EE2: luma residual taps in chroma-ALF and CCALF [R]. Joint Video Experts Team (JVET), document JVET-AE0121, 2023
- [50] OHM J R. Meeting report of the 32nd Meeting of the Joint Video Experts Team (JVET) [R]. Joint Video Experts Team (JVET), document JVET-AF1000, 2023
- [51] HU N, DONG J, SEREGIN V, et al. CE5-related: multiplication removal for cross component adaptive loop filter [R]. Joint Video Experts Team (JVET), document JVET-P0557, 2019
- [52] STRÖM J, ANDERSSON K. CE5-related: on the CCALF filtering process [R]. Joint Video Experts Team (JVET), document JVET-Q0165, 2020
- [53] MENG X W, ZHENG X Z, WANG S S, et al. CCALF virtual boundary issue for 4:4:4 and 4:2:2 format [R]. Joint Video Experts Team (JVET), document JVET-R0322, Oct. 2019
- [54] MENG X W, ZHENG X Z, WANG S S, et al. AHG 10: One-pass CCALF [R]. Joint Video Experts Team (JVET), document JVET-R0327, Oct. 2019
- [55] YIN W B, ZHANG K, ZHANG L. Non-EE2: coefficient precision adjustment for ALF [R]. Joint Video Experts Team (JVET) document JVET-AF0198, Oct. 2023
- [56] HU N, KARCZEWICZ M, SEREGIN V, et al. EE2-related: adaptive precision for CCALF coefficients [R]. Joint Video Experts Team (JVET), document JVET-AG0233, 2024
- [57] SONG N, YU Y, YU H P, et al. Non-EE2: adaptive coefficient precision for CCALF [R]. Joint Video Experts Team (JVET), document JVET-AG0065, 2024
- [58] KUO C W, XIU X Y, CHEN Y W, et al. Cross-component sample adaptive offset [C]//Data Compression Conference (DCC). IEEE, 2022: 359 – 368. DOI: 10.1109/DCC52660.2022.00044
- [59] KUO C W, XIU X Y, CHEN W, et al. Cross-component sample adaptive offset [R]. Joint Video Experts Team (JVET), document AVS-M5595, 2020
- [60] KUO C W, XIU X Y, CHEN Y W, et al. EE2-5.1: cross-component sample adaptive offset [R]. Joint Video Experts Team (JVET), document JVET-W0066, 2021
- [61] KOTRA A M, HU N, SEREGIN V, et al. AHG12: edge classifier for cross-component sample adaptive offset (CCSAO) [R]. Joint Video Experts Team (JVET), document JVET-X0105, 2021
- [62] KUO C W, XIU X, CHEN W Y, et al. AHG12: CCSAO classification with edge information [R]. Joint Video Experts Team (JVET), document JVET-X0152, 2021
- [63] JIAN Y R, ZHANG J Q, LI J R, et al. Enhanced cross component sample adaptive offset for AVS3 [C]//International Conference on Visual Communications and Image Processing (VCIP). IEEE, 2021: 1 – 5. DOI: 10.1109/VCIP53242.2021.9675321
- [64] KUO C W, XIU X Y, CHEN W, et al. AHG12: CCSAO with extended edge classifiers and history offsets [R]. Joint Video Experts Team (JVET), document JVET-AD0218, 2023
- [65] KUO C W, XIU X Y, CHEN W, et al. EE2-5.1: CCSAO with extended edge classifiers and history offsets [R]. Joint Video Experts Team (JVET), document JVET-AE0151, 2023
- [66] KARCZEWICZ M, YE Y. Common test conditions and evaluation procedures for enhanced compression tool testing [R]. Joint Video Experts Team (JVET), document JVET-AF2017, 2023

## Biographies

**LI Zhaoyu** received her BE degree in computer science and technology from Chang'an University, China in 2023. She is currently pursuing her MS degree in computer application technology with Peking University, China. Her research interests include video coding and video coding standard.

**MENG Xuewei** received her BE degree in communication engineering from Beijing University of Posts and Telecommunications, China in 2017 and PhD degree in computer application technology from Peking University, China in 2022. She is currently a senior software engineer with Core Media Technology, Disney Streaming. Her research interests include video processing, video compression, and video coding standard. She is actively participating in the research of versatile video coding (VVC) standard.

**ZHANG Jiaqi** received his BS degree from the School of Software Technology, Dalian University of Technology, China in 2017 and PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences in 2023. He is currently a postdoctoral researcher with the School of Computer Science, Peking University, China. His research interests include data compression and image/video coding.

**HUANG Cheng** (huang.cheng5@zte.com.cn) received his MS degree from the School of Computer Science and Engineering, Southeast University, China. He is currently a senior system architect and the project manager of video technology research at ZTE Corporation. His research interests include visual coding, storage, transport, and multimedia systems.

**JIA Chuanmin** received his BE degree in computer science from Beijing University of Posts and Telecommunications, China in 2015 and PhD degree in computer application technology from Peking University, China in 2020. He was a visiting student in New York University, USA in 2018. He is currently an assistant professor with the Wangxuan Institute of Computer Technology, Peking University. His research interests include video compression and multimedia signal processing. He received the Best Paper Award of PCM 2017, Best Paper Award of IEEE MM 2018, and Best Student Paper Award of IEEE MIPR 2019.

**MA Siwei** received his BS degree from Shandong Normal University, China in 1999, and PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, China in 2005. He held a postdoctoral position with the University of Southern California, USA from 2005 to 2007. He joined the School of Electronics Engineering and Computer Science, Institute of Digital Media, Peking University, China where he is currently a professor. He has authored more than 300 technical articles in refereed journals and proceedings in image and video coding, video processing, video streaming, and transmission. He served/serves as an associate editor for the *IEEE Transactions on Circuits and Systems for Video Technology* and *Journal of Visual Communication and Image Representation*.

**JIANG Yun** received his BE degree in 2000 and ME degree in 2003 in information science and technology from Peking University, China. He is the director of Zhejiang Province Advanced Institute of Information Technology, Peking University. He serves as a senior engineer at the School of Computer Science, Peking University, where he is a master's supervisor. His primary research interests focus on image processing and video coding standards.