|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  19th Meeting: Strasbourg, FR, 17–24 Oct. 2014 | Document: JCTVC-S0177\_r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE9 Test A.3 Modifying cross-component prediction to compensate for intra boundary filtering** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | **Robert Cohen Xingyu Zhang** Mitsubishi Electric Research Laboratories 201 Broadway, Cambridge, MA 02139 U.S.A | Tel: Email: | +1-617-621-7500 cohen@merl.com eexyzhang@ust.hk |
| *Source:* | Mitsubishi Electric Research Laboratories (MERL) | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

SCM2.0 and related platforms include a cross-component prediction (CCP) process, which uses a residual for a first component that may have undergone a boundary filtering process to predict a second and third component to which boundary filtering is not applied. This document proposes modifying the CCP process by adding an offset block to the reconstructed luma (or first component) residual block. This offset block is the difference between the luma prediction block after intra boundary filtering is applied and the same block before intra boundary filtering is applied. For blocks that use the adaptive color transform, the offset is halved, and the cross-component prediction process for the Cg component is additionally modified by subtracting the halved offset. The intra boundary filtering process is not modified. Simulations results for making this modification to SCM2.0 are provided. Additional results for when the adaptive color transform is disabled in both the anchor and tested configurations are also provided.

# Introduction

In SCM2.0, a cross-component prediction (CCP) process is used to predict chroma (or 2nd and 3rd components) prediction residuals from the luma (or 1st component) prediction residual. For luma (1st component) prediction blocks, a boundary filtering process is applied for blocks that use the DC, horizontal or vertical intra prediction mode. This boundary filtering process is not applied to chroma components. Therefore, CCP uses a luma residual block generated using a boundary filtering process to predict a chroma residual block generated without using a boundary-filtering process. This document proposes modifying the CCP process to compensate for the changes made to the luma residual block by the intra boundary filtering process. The intra boundary filtering is not modified.

## Intra boundary filtering

In SCM2.0, when a luma block is predicted using DC, horizontal or vertical prediction mode, the boundary pixels of the prediction block may be modified according to an intra boundary filtering process. For an NxN luma block, the DC, horizontal and vertical prediction process consists of two steps:

* Step 1: Form the initial prediction value  for the current block, with x, y = 0..N−1. For DC mode,  is equal to the average of the reference pixels. For horizontal mode, , where are the reference pixels to the left of the current block. For vertical mode, , where are the reference pixels above the current block.
* Step 2: Form the final prediction value  by applying the intra boundary filtering process to when the filtering process is enabled, or by setting when the filtering process is disabled.

## Cross component prediction

In SCM2.0, when a chroma block is intra-coded using DM mode, i.e. intra\_chroma\_pred\_mode[ xCb ][ yCb ]==4, a cross component prediction process may be applied. At the decoder, the chroma residual block is reconstructed as follows:

, (1)

where

denotes the chroma CCP residual block;

denotes the reconstructed luma residual block;

denotes the reconstructed chroma residual block, and

α is a scaling parameter that calculated by the encoder and is signaled in the bit-stream.

## Adaptive color transform

When the method of this document was first proposed in JCTVC-R0219 [1], the adaptive color transform was not part of SCM 1.0. SCM2.0, however, sometimes applies a color transform across all three components, so the proposed method must be modified accordingly. For lossy coding conditions, the forward transform, as described in JCTVC-R0147, is as follows:



For lossless coding conditions, the forward transform is:



# Proposed method

## For blocks when the adaptive color transform is not applied

# This document proposes modifying the cross-component prediction process by adding an offset block to the reconstructed luma (or first component) residual block as follows:

, (2)

where

. (3)

is set to 0 when *x* ≠ 0 and *y* ≠ 0, i.e. only the left and top boundaries can be nonzero. When intra boundary filtering is not applied, all elements of are set to 0. No changes are made to the intra boundary filtering process.

## For blocks when the adaptive color transform is applied

After the intra prediction process but before the cross-component prediction process, the encoder may apply the adaptive color transform. Suppose the first component is G, and the second and third components are B and R. The intra prediction process may apply boundary filtering to the G component. The adaptive color transform is applied, and then cross-component prediction is used to predict between the Y and Co (or Cg) residuals. If there were no color transform, then offset would be added to the G component. Because the encoder applies the cross-component prediction process after the adaptive color transform, we need to determine how to apply this offset to the Y, Co and Cg components. For the forward process in the encoder, adding the offset to the G component and applying the color transform results in:



Substituting the above components into the cross-component prediction equations yields:

The process in the decoder becomes:

Thus, when the adaptive color transform is applied to a block, this proposal modifies the existing decoder’s cross-component prediction process by adding to the Y component inside the CCP process, and by also subtracting after the Cg CCP process.

# Simulation results

The test conditions specified in the CE9 description [2] were used for these experiments.

## Lossy coding results

Decode time ratios may vary due to differences in computing platforms.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **All Intra** | | |
|  | G/Y | B/U | R/V |
| RGB, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% |
| RGB, text & graphics with motion,720p | -0.1% | 0.0% | 0.0% |
| RGB, mixed content, 1440p | 0.0% | -0.1% | -0.1% |
| RGB, mixed content, 1080p | -0.1% | -0.1% | 0.0% |
| RGB, Animation, 720p | -0.1% | 0.0% | -0.1% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | 0.0% | 0.0% | 0.0% |
| YUV, mixed content, 1440p | -0.1% | -0.1% | -0.1% |
| YUV, mixed content, 1080p | -0.1% | -0.1% | 0.0% |
| YUV, Animation, 720p | -0.1% | -0.2% | -0.1% |
| YUV, camera captured, 1080p | 0.0% | -0.1% | -0.3% |
| Enc Time[%] | 101% | | |
| Dec Time[%] | 107% | | |
|  |  |  |  |
|  | **Random Access** | | |
|  | G/Y | B/U | R/V |
| RGB, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% |
| RGB, text & graphics with motion,720p | -0.1% | -0.1% | -0.1% |
| RGB, mixed content, 1440p | -0.1% | -0.1% | -0.1% |
| RGB, mixed content, 1080p | 0.0% | -0.1% | -0.2% |
| RGB, Animation, 720p | -0.1% | -0.1% | -0.1% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | 0.0% | -0.2% | 0.0% |
| YUV, mixed content, 1440p | 0.0% | -0.1% | 0.0% |
| YUV, mixed content, 1080p | 0.0% | -0.4% | -0.2% |
| YUV, Animation, 720p | 0.0% | -0.2% | 0.0% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | -0.2% |
| Enc Time[%] | 101% | | |
| Dec Time[%] | 103% | | |
|  |  |  |  |
|  | **Low delay B** | | |
|  | G/Y | B/U | R/V |
| RGB, text & graphics with motion, 1080p | -0.1% | -0.1% | 0.0% |
| RGB, text & graphics with motion,720p | 0.0% | 0.1% | 0.1% |
| RGB, mixed content, 1440p | -0.2% | -0.1% | -0.1% |
| RGB, mixed content, 1080p | -0.2% | 0.1% | 0.5% |
| RGB, Animation, 720p | 0.0% | 0.0% | 0.1% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | -0.2% | -0.1% | -0.3% |
| YUV, mixed content, 1440p | -0.1% | -0.4% | 0.0% |
| YUV, mixed content, 1080p | -0.1% | -0.1% | -0.8% |
| YUV, Animation, 720p | 0.1% | -0.6% | -0.2% |
| YUV, camera captured, 1080p | 0.0% | -0.1% | 0.1% |
| Enc Time[%] | 101% | | |
| Dec Time[%] | 105% | | |

## Lossless coding results

Decode time ratios may vary due to differences in computing platforms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **All Intra** | | | |
|  | Bit-rate saving (Total) | Bit-rate saving (Average) | Bit-rate saving (Min) | Bit-rate saving (Max) |
|  |
| RGB, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, text & graphics with motion,720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, mixed content, 1440p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, mixed content, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, mixed content, 1440p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, mixed content, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 102% | | | |
| Dec Time[%] | 110% | | | |
|  |  |  |  |  |
|  | **Random Access** | | | |
|  | Bit-rate saving (Total) | Bit-rate saving (Average) | Bit-rate saving (Min) | Bit-rate saving (Max) |
|  |
| RGB, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, text & graphics with motion,720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, mixed content, 1440p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, mixed content, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, mixed content, 1440p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, mixed content, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 100% | | | |
| Dec Time[%] | 103% | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  | **Low Delay B** | | | |
|  | Bit-rate saving (Total) | Bit-rate saving (Average) | Bit-rate saving (Min) | Bit-rate saving (Max) |
|  |
| RGB, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, text & graphics with motion,720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, mixed content, 1440p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, mixed content, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| RGB, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, mixed content, 1440p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, mixed content, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, Animation, 720p | 0.0% | 0.0% | 0.0% | 0.0% |
| YUV, camera captured, 1080p | 0.0% | 0.0% | 0.0% | 0.0% |
| Enc Time[%] | 100% | | | |
| Dec Time[%] | 104% | | | |

# Additional results for when the adaptive color transform is disabled

When the adaptive color transform is disabled in the configurations for the both anchor and tested conditions, the modification of CCP only adds the offset to the luminance component. Simulation results for lossy coding conditions, for when the adaptive color transform is disabled for both the anchor and tested conditions, are as follows:

Decode time ratios may vary due to differences in computing platforms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  |  |
|  | **All Intra** | | | |
|  | G/Y | | B/U | R/V |
| RGB, text & graphics with motion, 1080p | -0.1% | | -0.1% | -0.1% |
| RGB, text & graphics with motion,720p | -0.6% | | -0.6% | -0.6% |
| RGB, mixed content, 1440p | -1.2% | | -1.1% | -1.1% |
| RGB, mixed content, 1080p | -1.3% | | -1.3% | -1.2% |
| RGB, Animation, 720p | -0.7% | | -0.7% | -0.6% |
| RGB, camera captured, 1080p | -1.1% | | -0.9% | -1.2% |
| YUV, text & graphics with motion, 1080p | 0.0% | | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | -0.1% | | 0.0% | -0.1% |
| YUV, mixed content, 1440p | -0.1% | | -0.1% | 0.0% |
| YUV, mixed content, 1080p | -0.1% | | -0.1% | -0.1% |
| YUV, Animation, 720p | -0.1% | | -0.2% | -0.1% |
| YUV, camera captured, 1080p | 0.0% | | -0.1% | -0.3% |
| Enc Time[%] | 101% | | | |
| Dec Time[%] | 109% | | | |
|  |  | |  |  |
|  | **Random Access** | | | |
|  | G/Y | | B/U | R/V |
| RGB, text & graphics with motion, 1080p | -0.1% | | 0.0% | -0.1% |
| RGB, text & graphics with motion,720p | -0.6% | | -0.5% | -0.5% |
| RGB, mixed content, 1440p | -0.5% | | -0.5% | -0.5% |
| RGB, mixed content, 1080p | -0.8% | | -0.8% | -0.9% |
| RGB, Animation, 720p | -0.3% | | -0.3% | -0.3% |
| RGB, camera captured, 1080p | -0.5% | | -0.3% | -0.5% |
| YUV, text & graphics with motion, 1080p | 0.0% | | 0.0% | 0.0% |
| YUV, text & graphics with motion,720p | 0.1% | | 0.0% | 0.0% |
| YUV, mixed content, 1440p | 0.0% | | -0.2% | -0.2% |
| YUV, mixed content, 1080p | 0.0% | | -0.2% | 0.0% |
| YUV, Animation, 720p | 0.0% | | -0.1% | -0.2% |
| YUV, camera captured, 1080p | 0.0% | | 0.0% | -0.1% |
| Enc Time[%] | 101% | | | |
| Dec Time[%] | 102% | | | |
|  |  | |  |  |
|  | **Low delay B** | | | |
|  | G/Y | | B/U | R/V |
| RGB, text & graphics with motion, 1080p | 0.0% | | 0.0% | -0.1% |
| RGB, text & graphics with motion,720p | -0.3% | | -0.3% | -0.2% |
| RGB, mixed content, 1440p | -0.4% | | -0.2% | -0.1% |
| RGB, mixed content, 1080p | 0.4% | | 0.0% | 0.1% |
| RGB, Animation, 720p | -0.1% | | -0.1% | 0.0% |
| RGB, camera captured, 1080p | -0.1% | | 0.0% | -0.1% |
| YUV, text & graphics with motion, 1080p | 0.0% | | 0.0% | 0.1% |
| YUV, text & graphics with motion,720p | 0.0% | | -0.1% | -0.1% |
| YUV, mixed content, 1440p | 0.0% | | -0.2% | 0.2% |
| YUV, mixed content, 1080p | -0.1% | | 0.4% | 0.8% |
| YUV, Animation, 720p | 0.0% | | -0.2% | -0.2% |
| YUV, camera captured, 1080p | 0.0% | | 0.0% | 0.0% |
| Enc Time[%] | 102% | | | |
| Dec Time[%] | 102% | | | |

# Conclusions

This document proposes modifying the CCP process by adding an offset block to the reconstructed luma (or first component) residual block. This offset block is the difference between the luma prediction block after intra boundary filtering is applied and the same block before intra boundary filtering is applied. For blocks that use the adaptive color transform, the offset is halved, and the cross-component prediction process for the Cg component is additionally modified by subtracting the halved offset. The intra boundary filtering is not modified. Simulations results for making this modification to SCM2.0 are provided. Additional results for when the adaptive color transform is disabled in both the anchor and tested configurations are also provided.

We would like to thank participants from Qualcomm for their cross-check of this test.

# Patent rights declaration(s)

**Mitsubishi Electric Research Laboratories may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

# References

1. X. Zhang, R. Cohen, “Improvement of cross-component prediction,” Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-R0219, 18th Meeting: Sapporo, JP, 30 June – 9 July, 2014.
2. R. Cohen, S. Liu, J. Xu, L. Zhang, “Description of Core Experiment 9 (CE9): IBF/CCP interdependency,” Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-Q1109, 18th Meeting: Sapporo, JP, 30 June – 9 July, 2014.