|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extension Development**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  3rd Meeting: Geneva, CH, 17-23 Jan. 2013 | Document: JCT3V-C0054 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Simplifications for adaptive luminance compensation in 3D-AVC** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Jewon Kang Ying Chen Li Zhang Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121 USA | Tel: Email: | 1-858-651-8457 [jewonk@qti.qualcomm.com](mailto:jewonk@qti.qualcomm.com)  1-858-845-6589  [cheny@qti.qualcomm.com](mailto:cheny@qti.qualcomm.com)  +1-858-651-6660 [lizhang@qti.qualcomm.com](mailto:lizhang@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This proposal presents simplification of the adaptive luminance compensation (ALC) in 3D/AVC. In the current ATM design, the above and the left regions of the current block are matched in the reference picture using a motion vector, and the weighted factor derivation process is performed to predict the luminance discrepancy of the sample pixels. However, the current design may suffer from significant complexity in decoding because the 4x4 blocks inside each macroblock (MB) needs to be processed in series even when the MB has only MB partitions larger than 8x8. In this contribution, it is proposed to derive the prediction weights just once for a MB partition if its size is larger than 8x8 or equal to 8x8 without further sub-block partition. Thus, the 4x4 block processing is limited as much as possible while the design can be consistent with the original one and have a minor change in coding efficiency.

# Introduction

In the current 3D-AVC [1], adaptive luminance compensation (ALC) [2] is employed to compensate the luminance discrepancy in inter-view pictures, and achieves a significant coding gain around 2% BD-rate saving in the common test condition [3]. Each luma sample in a macroblock (MB) partition is predicted with a set of prediction weights that is derived using above and left regions outside the partition. The above and left regions for the coded block and the reference block are matched using motion compensation with interpolation, if needed, and, thus, there is no side information for the region matching. The process is performed during the normal motion compensation process. The estimated prediction weights are applied to a block in a similar manner to the weighted prediction in H.264/AVC.

The ALC can be applied to P\_L0\_16x16, P\_L0\_L0\_16x8, P\_L0\_L0\_8x16, P\_8x8, P\_8x8ref0 modes when mb\_alc\_flag signaled in macroblock-wise is equal to 1. The above and the left regions, respectively, have the Mx4 and 4xN block shapes, wherein M and N are decided by the MB partition of a coded block and a transform size as shown in Table 1. For example, when a 4x4 DCT is applied to a MB partition, the partition is divided into 4x4 blocks, and the 4x4 blocks are coded with the ALC in series. However, as noted, the ALC is performed with the motion compensation process and the prediction weights are derived in a 4x4 block-level. Therefore, in the decoder side, all the 4x4 blocks inside the MB partition may go through motion compensation (MC) as well as the inverse DCT/Quantization one by one. One can first perform the MC before the ALC. However, the interpolation of the adjacent blocks needs to be done again. Furthermore, the current block cannot be decoded until the above and the left 4x4 block becomes available for the region matching even though they have the same motion parameters in an MB partition. The 4x4 block processing with inter-block dependency may significantly degrade the decoding throughput. Hence, we propose a simplification of the current design in the ALC so as to reduce the complexity while changing a minimal coding gain in this contribution.

Table 1 – Parameters (M and N) specifying the left and top regions for prediction weights derivation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **mb\_type** | **transform\_size\_8x8\_flag equal to 1** | | **transform\_size\_8x8\_flag equal to 0** | |
| **M** | **N** | **M** | **N** |
| P\_L0\_16x16 | 16 | 16 | 8 | 8 |
| P\_L0\_L0\_16x8 | 16 | 8 | 4 | 4 |
| P\_L0\_L0\_8x16 | 8 | 16 | 4 | 4 |
| P\_8x8, \_8x8ref0 | 8 | 8 | 4 | 4 |
| P\_Skip | Undefined | | 16 | 16 |

# Proposal

To tackle the problems aforementioned, we modify the current ALC design in this proposal. The contributions of this proposal are summarized into two points:

1. The prediction weight should not be derived multiple times when the MB partition is larger than 8x8 or equal to 8x8 with no further sub-MB partitions. Thus, the ALC is performed one time for the whole samples in the partition.
2. The transform size is not further considered in the selection of the matching region shapes. The proposed specification is shown in Table 2 replacing the Table 1. The Table 1, which is complicated, actually can be removed since Table 2 is a simple mapping of MB partition size to the parameters.

Table 2 – Parameters (M and N) specifying the left and top regions for prediction weights derivation: independent to transform size.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **mb\_type** | **M** | **N** | **mb\_type (sub\_mb\_type)** | | **M** | **N** |
| P\_L0\_16x16 | 16 | 16 | P\_8x8, P\_8x8ref0 | P\_L0\_8x8 | 8 | 8 |
| P\_L0\_L0\_16x8 | 16 | 8 | P\_L0\_8x4 | 4 | 4 |
| P\_L0\_L0\_8x16 | 8 | 16 | P\_L0\_4x8 | 4 | 4 |
| P\_Skip | 16 | 16 | P\_L0\_4x4 | 4 | 4 |

It is also noted that the proposed simplification falls back into the original design in transform\_size\_8x8\_flag equal to 1 when the MB partition is larger than 8x8, as can be seen in the left columns of Table 1. As for the 8x8 MB partition, the proposed simplification uses the same size for each sub-block partition regardless of the transform size. Hence, the region selections are simple, but still consistent to the original design.

However, beside to the consistent design, the simplification can take a large advantage over the original one. The difference between the proposed simplification and the ALC in the current ATM design is illustrated in Figure 1 (a) and (b) when a 4x4 DCT is applied to an 8x16 MB partition. The matching regions outside the current partition are colored with orange. As can be seen in Figure 1 (a), in the proposed simplification, the prediction weight is computed using the orange regions pointed by the motion vector, and applied to all the pixels in the partition. In contrast, in the ATM, the derivation can be conducted in each 4x4 block inside though the blocks share the common motion vector in a partition. Furthermore, the 4x4 blocks can be decoded only if the above and the left blocks inside the partition have been decoded (as specified by the blue arrows in Figure 1(b)). In sum, the proposal aims to minimize the inter-block dependency as well as 4x4 block-wise processing in ALC via the MB partition-wise processing.

16

8

4

4

(mv\_x, mv\_y)

4

4

(mv\_x, mv\_y)

(a) Proposed ALC method and (b) the ALC in the current ATM when the transform size is 4.

Figure 1– Example for an 8x16 MB partition

# Experimental results

Simulation results of the proposal are shown in Table 2. The implementation was based on ATM6.0, and simulations are done under common test conditions [3].

As shown in Table 2, the proposed algorithm affects only minor change (i.e. 0.04% BD-rate increase) in the coding efficiency, and can greatly reduce the complexity in decoding and in implementation.

Table 3 – Proposed method VS ATM6.0 rev1.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Texture Coding | | Depth Coding | | Total (Coded PSNR) | | Total (Synthesed PSNR) | | Complexity estimate (ratio to anchor) | | |
|  | dBR, % | dPSNR, dB | dBR, % | dPSNR, dB | dBR, % | dPSNR, dB | dBR, % | dPSNR, dB | Encoder Time, % | Decoder Time, % | Rendering Time, % |
| S01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.08 | 0.00 | 96.32 | 95.69 | 108.30 |
| S02 | 0.02 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 | 99.92 | 98.82 | 106.58 |
| S03 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 97.20 | 100.42 | 99.96 |
| S04 | -0.02 | 0.00 | 0.00 | 0.00 | -0.01 | 0.00 | -0.02 | 0.00 | 100.45 | 98.46 | 100.17 |
| S05 | 0.08 | 0.00 | 0.00 | 0.00 | 0.07 | 0.00 | 0.02 | 0.00 | 100.68 | 94.68 | 91.50 |
| S06 | 0.09 | 0.00 | 0.00 | 0.00 | 0.09 | 0.00 | 0.04 | 0.00 | 96.22 | 96.31 | 103.86 |
| S08 | 0.09 | 0.00 | 0.00 | 0.00 | 0.09 | 0.00 | 0.04 | 0.00 | 97.14 | 98.52 | 96.92 |
| Average | 0.04 | 0.00 | 0.00 | 0.00 | 0.04 | 0.00 | 0.02 | 0.00 | 98.27 | 97.56 | 101.04 |

# Conclusion

The proposal presents simplified prediction weight derivation in the ALC by disabling the 4x4 block-wise processes as possible. The proposed simplification may significantly reduce the decoding complexity while changing only minor coding efficiency.

# References

1. M. M. Hannuksela, Y. Chen, T. Suzuki, J.-R. Ohm, G. J. Sullivan, “3D-AVC draft text 4”, JCT3V-B1002, Oct. 2012.
2. M. Mishurovskiy, A. Fartukov, I. Kovliga, J. Lee, “3D-CE2.a: A results on inter-view coding with adaptive luminance compensation,” JCT3V-B0031, Oct. 2012.
3. D. Rusanovskyy, K. Müller, A. Vetro, “Common Test Conditions of 3DV Core Experiments,” JCT3V-B1100, Oct. 2012.

# Patent rights declaration(s)

**Qualcomm Incorporated 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).**