|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  15th Meeting: Geneva, CH, 23 Oct. – 1 Nov. 2013 | Document: JCTVC-O0237\_r2 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG5: Disable scaling lists within certain spatial regions in a picture** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Rajan Joshi, Joel Sole, Marta Karczewicz 5775 Morehouse Drive San Diego, CA 92121-1714 USA | Tel: Email: | +1-858-658-4511 [rajanj@qti.qualcomm.com](mailto:rajanj@qti.qualcomm.com) |
| *Source:* | Qualcomm Inc. | | |

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

# Abstract

At the 14th JCTVC meeting transform-skip was enabled for all transform block sizes. In the current text specification, when scaling lists are being used, they have to be used for the entire picture, both for transform and transform-skip blocks. There is no flexibility to disable their use for different spatial regions in the picture. We propose a signaling scheme to disable the use of scaling lists for each CU or group of CUs. A PPS level flag is provided to enable the use of this feature.

# Introduction

At the 14th JCTVC meeting transform-skip was extended to all transform block sizes. In the current text specification, scaling lists have to be used for the entire picture. When used, they are applied to both transform and transform-skip blocks. This is counterintuitive since the coefficients in a transform are arranged according to its frequency content and the coefficients in a transform-skipped block are arranged according to the spatial position.

It is desirable to have the flexibility to turn off the use of scaling matrices for transform-skip blocks. However, this was not included in HEVC because it was considered that switching scaling list at each transform block would be too complex.

Video content that contains a mix of natural video and non-camera generated content is becoming increasingly common. Furthermore the segmentation of the picture into natural video and non-camera generated content may be known. The non-camera generated content may be coded with a mixture of transform and transform-skip blocks. In this scenario, it may be desirable to turn off the use of scaling lists for certain spatial areas instead of depending purely on whether a block uses transform or transform-skip.

A CU level flag is proposed to enable this functionality.

# Technical description

A CU level flag *cu\_scaling\_list\_flag* is proposed to signal that the active scaling list should be used for the current CU or a group of CUs. A PPS level flag *cu\_scaling\_list\_enabled\_flag* is used to enable this functionality. To avoid excessive signaling costs, the minimum size at which the *cu\_scaling\_list\_flag* flag is present is signaled in the PPS using a syntax element *diff\_cu\_scaling\_list\_depth* whose semantics are similar to the *diff\_cu\_qp\_delta\_depth* syntax element. Then, instead of signaling the flag for each CU, it is signaled only when the CU size is greater than or equal to a minimum size.

Consider that the minimum size is 16×16. If a 16×16 CU is split further into 4 8×8 CUs, only one flag may be signalled for the 16×16 block. If there is a 32×32 CU that is not split further, only one flag may be sent for the entire 32×32 block. This is similar to the way quantization groups work for sending delta QP. The flag is not signaled if the CU is coded in transquant-bypass mode.

During discussions with JCT-VC experts, it was noted that the use of the flag could be generalized to signal a choice between two sets of scaling lists. The functionality of disabling the scaling list for certain spatial regions is still retained. To achieve this, one set of scaling lists can have all the values set to 16.

# Conclusion

We propose the use of a CU level flag *cu\_scaling\_list\_flag* to signal that the active scaling list should be used for the current CU or a group of CUs. This provides the functionality to turn off the use of scaling lists for certain spatial areas. A PPS level flag *cu\_scaling\_list\_enabled\_flag* is proposed to enable this functionality. Furthermore the minimum size at which the CU level flag may be signaled is indicated by a new syntax element *diff\_cu\_scaling\_list\_depth*.

# Working draft text

The working draft text is on top of JCTVC-N1005\_v3. The changes are marked with yellow.

**7.3.2.2 Picture parameter set RBSP syntax**

|  |  |
| --- | --- |
| pic\_parameter\_set\_rbsp( ) { | Descriptor |
| **…** |  |
| **pps\_scaling\_list\_data\_present\_flag** | u(1) |
| if( pps\_scaling\_list\_data\_present\_flag ) |  |
| scaling\_list\_data( ) |  |
| if( scaling\_list\_enabled\_flag ) { |  |
| **cu\_scaling\_list\_enabled\_flag** | u(1) |
| if( cu\_scaling\_list\_enabled\_flag ) |  |
| **diff\_cu\_scaling\_list\_depth** | ue(v) |
| } |  |
| **lists\_modification\_present\_flag** | u(1) |
| **…** |  |

#### 7.3.8.4 Coding quadtree syntax

|  |  |
| --- | --- |
| coding\_quadtree( x0, y0, log2CbSize, cqtDepth ) { | Descriptor |
| … |  |
| if( cu\_qp\_delta\_enabled\_flag && log2CbSize >= Log2MinCuQpDeltaSize ) { |  |
| IsCuQpDeltaCoded = 0 |  |
| CuQpDeltaVal = 0 |  |
| } |  |
| if( cu\_scaling\_list\_enabled\_flag && log2CbSize >= Log2MinCuScalingListFlagSize ) { |  |
| IsCuScalingListFlagCoded = 0 |  |
| } |  |
| … |  |

#### 7.3.8.10 Transform unit syntax

|  |  |
| --- | --- |
| transform\_unit( x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx ) { | Descriptor |
| … |  |
| if( cu\_qp\_delta\_enabled\_flag && !IsCuQpDeltaCoded ) { |  |
| **cu\_qp\_delta\_abs** | ae(v) |
| if( cu\_qp\_delta\_abs ) |  |
| **cu\_qp\_delta\_sign\_flag** | ae(v) |
| } |  |
| if( cu\_scaling\_list\_enabled\_flag && !IsCuScalingListFlagCoded ) { |  |
| **cu\_scaling\_list\_flag** | ae(v) |
| } |  |
| … |  |

#### Picture parameter set RBSP semantics

**…**

**cu\_qp\_delta\_enabled\_flag** equal to 1 specifies that the diff\_cu\_qp\_delta\_depth syntax element is present in the PPS and that cu\_qp\_delta\_abs may be present in the transform unit syntax. cu\_qp\_delta\_enabled\_flag equal to 0 specifies that the diff\_cu\_qp\_delta\_depth syntax element is not present in the PPS and that cu\_qp\_delta\_abs is not present in the transform unit syntax.

**diff\_cu\_qp\_delta\_depth** specifies the difference between the luma coding tree block size and the minimum luma coding block size of coding units that convey cu\_qp\_delta\_abs and cu\_qp\_delta\_sign\_flag. The value of diff\_cu\_qp\_delta\_depth shall be in the range of 0 to log2\_diff\_max\_min\_luma\_coding\_block\_size, inclusive. When not present, the value of diff\_cu\_qp\_delta\_depth is inferred to be equal to 0.

The variable Log2MinCuQpDeltaSize is devived as follows:

Log2MinCuQpDeltaSize = CtbLog2SizeY − diff\_cu\_qp\_delta\_depth (7‑31)

**cu\_scaling\_list\_enabled\_flag** equal to 1 specifies that the diff\_cu\_scaling\_list\_depth syntax element is present in the PPS and that cu\_scaling\_list\_flag may be present in the transform unit syntax. cu\_scaling\_list\_enabbled\_flag equal to 0 specifies that the cu\_scaling\_list\_flag syntax element is not present in the PPS and that cu\_scaling\_list\_flag is not present in the transform unit syntax.

**diff\_cu\_scaling\_list\_depth** specifies the difference between the luma coding tree block size and the minimum luma coding block size of coding units that convey cu\_scaling\_list\_flag. The value of diff\_cu\_qp\_delta\_depth shall be in the range of 0 to log2\_diff\_max\_min\_luma\_coding\_block\_size, inclusive. When not present, the value of diff\_cu\_qp\_delta\_depth is inferred to be equal to 0.

The variable Log2MinCuScalingListFlagSize is devived as follows:

Log2MinCuScalingListFlagSize = CtbLog2SizeY − diff\_cu\_scaling\_list\_depth (7‑32)

**…**

#### 7.4.9.10 Transform unit semantics

…

**cu\_qp\_delta\_abs** specifies the absolute value of the difference CuQpDeltaVal between the luma quantization parameter of the current coding unit and its prediction.

**cu\_qp\_delta\_sign\_flag** specifies the sign of CuQpDeltaVal as follows:

* If cu\_qp\_delta\_sign\_flag is equal to 0, the corresponding CuQpDeltaVal has a positive value.
* Otherwise (cu\_qp\_delta\_sign\_flag is equal to 1), the corresponding CuQpDeltaVal has a negative value.

When cu\_qp\_delta\_sign\_flag is not present, it is inferred to be equal to 0.

When cu\_qp\_delta\_abs is present, the variables IsCuQpDeltaCoded and CuQpDeltaVal are derived as follows:

IsCuQpDeltaCoded = 1 (7‑66)

CuQpDeltaVal = cu\_qp\_delta\_abs \* ( 1 − 2 \* cu\_qp\_delta\_sign\_flag ) (7‑67)

The value of CuQpDeltaVal shall be in the range of −( 26 + QpBdOffsetY / 2 ) to +( 25 + QpBdOffsetY / 2 ), inclusive.

**cu\_scaling\_list\_flag** specifies whether the scaling list should be used for the current coding unit.

When cu\_scaling\_list\_flag is not present, it is inferred to be equal to 1.

When cu\_scaling\_list\_flag is present, the variable IsCuScalingListFlagCoded is set to 1.

### 8.6.3 Scaling process for transform coefficients

…

– The scaling factor m[ x ][ y ] is derived as follows:

– If scaling\_list\_enabled\_flag is 0 or cu\_scaling\_list\_flag is equal to 0,

m[ x ][ y ] = 16 (8‑282)

– Otherwise (scaling\_list\_enabled\_flag is equal to 1 and cu\_scaling\_list\_flag is equal to 1),

m[ x ][ y ] = ScalingFactor[ sizeId ][ matrixId ][ x ][ y ] (8‑283)

Where sizeId is specified in Table 7‑3 for the size of the quantization matrix equal to (nTbS)x(nTbS) and matrixId is specified in Table 7‑4 for sizeId, CuPredMode[ xTbY ][ yTbY ], and cIdx, respectively.

– The scaled transform coefficient d[ x ][ y ] is derived as follows:

d[ x ][ y ] = Clip3( coeffMin, coeffMax, ( ( TransCoeffLevel[ xTbY ][ yTbY ][ cIdx ][ x ][ y ] \* m[ x ][ y ] \*   
 levelScale[ qP%6 ]  <<  (qP / 6 ) ) + ( 1  <<  ( bdShift − 1 ) ) )  >>  bdShift )

# Patent rights declaration(s)

**Qualcomm Inc. 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).**