

JVET-P0167

CE7-related: Improved coding of user defined quantization matrices

Authors:

Olena Chubach, Chen-Yen Lai, Ching-Yeh Chen, Tzu-Der Chuang, Chih-Wei Hsu, Yu-Wen Huang, Shaw-Min Lei

Presenter: Olena Chubach

Overall Summary

- Simple changes suggested on top of the current design to improve coding efficiency of user-defined QMs:
 - **[Inter QM extended copy mode]**
Allow referencing a previously coded QM with the same base size as the current QM, where the reference QM size¹ can be the same as or smaller than the current QM size¹.
 - **[Inter QM prediction mode]**
Allow coding element-to-element differences between the current QM and the reference QM.
 - **[Intra QM prediction mode]**
When the previous two modes are impossible or inefficient, the original DPCM coding of elements within the current QM can still be applied.
- On average -43% bitrate reduction can be achieved

Introduction

- In VVC Draft 6, when user defined QMs are specified, current coding procedure is limited to following two methods:
 - **[Inter QM copy mode]** reference one of the previously decoded QMs if it is equal to the current to-be-coded QM and has the same size.
 - **[Intra QM prediction mode]** apply DPCM coding of elements within the current to-be-coded QM.
- VVC, having a larger number of block sizes compared to HEVC, requires coding of more QMs.

Inter QM extended copy mode

- In case of VVC Draft 6, QM for INTRA8X8_CHROMAU can be decoded by copying a previously decoded QM for INTRA8X8_LUMA.
- However, QM for INTRA16X16_LUMA cannot be decoded same way, since prediction between QMs with different sizelds is not allowed in VVC Draft 6.
- It is suggested to allow referencing QMs with the same base size, while reference QM sizeld can be smaller than or equal to the current QM sizeld.

Table 1. Selected QMs for block sizes 8x8 and 16x16 (HEVC default)

INTRA8X8_LUMA =	INTRA8X8_CHROMAU =	INTRA16X16_LUMA =
16, 16, 16, 16, 17, 18, 21, 24	16, 16, 16, 16, 17, 18, 21, 24	16, 16, 16, 16, 17, 18, 21, 24
16, 16, 16, 16, 17, 19, 22, 25	16, 16, 16, 16, 17, 19, 22, 25	16, 16, 16, 16, 17, 19, 22, 25
16, 16, 17, 18, 20, 22, 25, 29	16, 16, 17, 18, 20, 22, 25, 29	16, 16, 17, 18, 20, 22, 25, 29
16, 16, 18, 21, 24, 27, 31, 36	16, 16, 18, 21, 24, 27, 31, 36	16, 16, 18, 21, 24, 27, 31, 36
17, 17, 20, 24, 30, 35, 41, 47	17, 17, 20, 24, 30, 35, 41, 47	17, 17, 20, 24, 30, 35, 41, 47
18, 19, 22, 27, 35, 44, 54, 65	18, 19, 22, 27, 35, 44, 54, 65	18, 19, 22, 27, 35, 44, 54, 65
21, 22, 25, 31, 41, 54, 70, 88	21, 22, 25, 31, 41, 54, 70, 88	21, 22, 25, 31, 41, 54, 70, 88
24, 25, 29, 36, 47, 65, 88, 115	24, 25, 29, 36, 47, 65, 88, 115	24, 25, 29, 36, 47, 65, 88, 115

Inter QM prediction mode

- Implies using any of the previously coded QM for decoding of the current QM.
 - For that, a flag is sent to indicate whether inter QM prediction mode is enabled.
 - Then, the reference sizeId and matrixId specified in Tables 2 and 3 are coded together with the delta QM. The delta QM is equal to the difference between the current QM and the reference QM. DPCM is applied for coding delta QM.
- The top-left value of the reference QM is used to predict the separate DC value of the current QM, if the QM used for prediction does not have a separate DC value but the current QM has a separate DC value.
- Default QMs can also be used for predicting current matrices. No additional flags need to be signalled for this.

Table 2. Size identifier (VTM6.1)

Luma	Chroma	sizeId
-	-	0
-	2x2	1
4x4	4x4	2
8x8	8x8	3
16x16	16x16	4
32x32	32x32	5
64x64	-	6

Table3. Matrix type identifier (VTM6.1)

CuPredMode	cldx (Colour component)	matrixId
MODE_INTRA	0 (Y)	0
MODE_INTRA	1 (Cb)	1
MODE_INTRA	2 (Cr)	2
MODE_INTER	0 (Y)	3
MODE_INTER	1 (Cb)	4
MODE_INTER	2 (Cr)	5

Intra QM prediction mode

- When both previous methods are either impossible or inefficient, the original DPCM coding of elements within the current QM can be applied.
 - This mode already exists in VVC Draft 6.
- The method requiring minimum number of bits will be chosen for coding of the current QM.

Additional optimizations

- sizeld of the reference QM is signalled only for QMs with sizeld >3
- For 2x2 IntraChromaU, 4x4 IntraLuma, 8x8 IntraLuma, reference QM matrixId is not signalled, and inferred to 0.
 - Referencing the default QMs is still possible in this case. The number of bits required will be compared to the number of bits after DPCM coding, and the method requiring minimum number of bits will be chosen.
- Delta values are signalled for sizeld and matrixId when inter QM extended copy mode or inter QM prediction mode is used.
- ~~• If sps_max_luma_transform_size_64_flag equal to 0 (i.e., maximum transform size in luma samples is equal to 32), no QM for 64x64 has to be decoded.~~



This will make APS parsing be dependent on SPS, which is not desirable. Therefore, this is removed in the updated JVET-P0167. Removing this aspect has very tiny impact on bit rates of QMs.

Simulation Results

- The QM sets used for testing are taken from JVET-00223 (VVC test set).

Test name	VTM6.1	VTM6.1 + ext. copy mode	VTM6.1 + ext. copy mode vs. VTM6.1	Proposed	Proposed vs. VTM6.1
JPEG-like	4748	2459	-2289(-48%)	1840	-2908(-61%)
MPEG2-like	911	592	-319(-35%)	472	-439(-48%)
h264-like	1224	802	-422(-34%)	662	-562(-46%)
DTT-like	2940	1896	-1044(-36%)	1484	-1456(-50%)
HEVC	1504	530	-974(-65%)	532	-972(-65%)
symmetry1	2658	1584	-1074(-40%)	1116	-1542(-58%)
symmetry2	1620	1292	-328(-20%)	1118	-502(-31%)
symmetry3	3220	2560	-660(-20%)	2210	-1010(-31%)
symmetry4	3146	2418	-728(-23%)	2351	-795(-25%)
symmetry7	4264	3252	-1012(-24%)	2678	-1586(-37%)
asymmetry1	3078	1806	-1272(-41%)	1233	-1845(-60%)
asymmetry2	1778	1404	-374(-21%)	1208	-570(-32%)
asymmetry3	3624	2842	-782(-22%)	2414	-1210(-33%)
asymmetry4	4344	3344	-1000(-23%)	3354	-990(-23%)
asymmetry7	5406	3316	-2090(-39%)	2737	-2669(-49%)
Average	2964	2006	-957(-33%)	1693	-1270(-43%)

Conclusions

- Simple and easy for understanding coding scheme for encoding user-defined QMs is proposed
- Coding efficiency impact: -43% on average over VTM6.1
- Proposed to adopt and include into the next VVC WD
- Thanks to Panasonic for cross-checking