



Method and Syntax for Quantization Matrices Representation

Ximin Zhang, Shan Liu and Shawmin Lei



Outline

- Summary of the contribution
- Challenge and motivation
- Implicit matrices generation scheme
- Results
- Conclusions

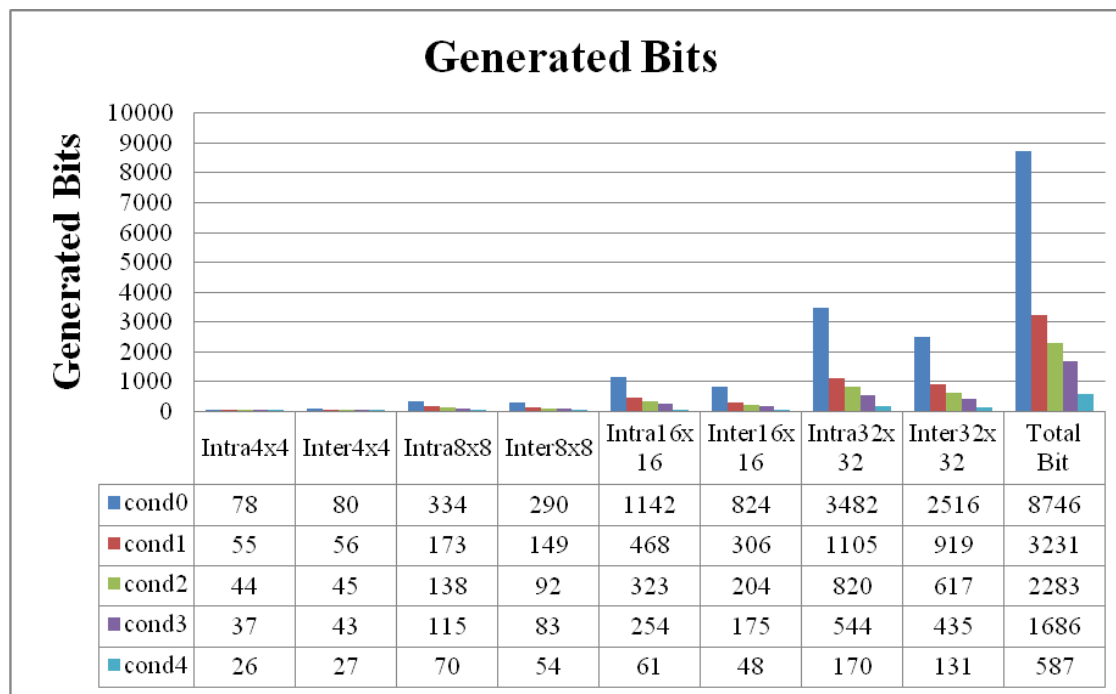
Summary of the Contribution

- **General contribution description**

- Quantization matrices coding in SPS, PPS and APS
- Support 4x4, 8x8, 16x4, 4x16, 16x16, 32x8, 8x32, 32x32 matrices
- Support explicit encoding of 4x4 and 8x8 matrices
- Implicit generation of 16x4, 4x16, 16x16, 32x8, 8x32, 32x32 matrices

Challenge and Motivation

- **Observation:** Quantization matrices of multiple/large block size could lead to a significant overhead with existing approach



	AVC Method	Sony Proposal Method
ClassA	25.1%	3.7%
ClassB	59.5%	8.6%
ClassC	138.5%	19.9%
ClassD	553.1%	79.3%
Total Average	194.0%	27.9%

Statistic data is based on JCTVC-F475 (Enhancement of quantization matrix for HEVC)

Challenge and Motivation

- **Challenge:** Compared with AVC, the memory requirement for 16x4, 4x16, 16x16, 32x32, 8x32, 32x8 quantization matrices is increased to another level.
- **Motivation:** A simple quantization matrix encoding method with minimum overhead.
- **Reasoning:**
 - The smaller quantization matrices can be treated as a subset of the bigger quantization matrices.
 - The coefficients of a smaller quantization matrices can be mapped to the coefficients of a bigger quantization matrices if they have the similar frequency.

Implicit Large Square Matrices Generation

- Generate $2N \times 2N$ from $N \times N$ based on interpolation

6	10	13	16	18	23	25	27
10	11	16	18	23	25	27	29
13	16	18	23	25	27	29	31
16	18	23	25	27	29	31	33
18	23	25	27	29	31	33	36
23	25	27	29	31	33	36	38
25	27	29	31	33	36	38	40
27	29	31	33	36	38	40	42



6	8	10	11	12	13	15	16	17	18	21	23	24	25	26	27
8	10	11	12	13	15	16	17	19	21	23	24	25	26	27	28
10	11	11	12	14	16	17	18	21	23	24	25	26	27	28	29
11	12	12	13	14	16	18	19	21	23	24	25	26	27	28	29
12	13	14	15	16	17	19	21	23	24	25	26	27	28	29	30
13	15	16	16	17	18	21	23	24	25	26	27	28	29	30	31
15	16	17	18	19	21	23	24	25	26	27	28	29	30	31	32
16	17	18	19	21	23	24	25	26	27	28	29	30	31	32	33
17	19	21	22	23	24	25	26	27	28	29	30	31	32	34	35
18	21	23	23	24	25	26	27	28	29	30	31	32	33	35	36
21	23	24	24	25	26	27	28	29	30	31	32	34	35	36	37
23	24	25	25	26	27	28	29	30	31	32	33	35	36	37	38
24	25	26	26	27	28	29	30	31	32	34	35	36	37	38	39
25	26	27	27	28	29	30	31	32	33	35	36	37	38	39	40
26	27	28	28	29	30	31	32	34	35	36	37	38	39	40	41
27	28	29	29	30	31	32	33	35	36	37	38	39	40	41	42

Generate 16X16 from 8x8

Implicit Non-square Matrices Generation

- Generate $0.5N \times 2N / 2N \times 0.5N$ from $2N \times 2N$

6	8	10	11	12	13	15	16	17	18	21	23	24	25	26	27
8	10	11	12	13	15	16	17	19	21	23	24	25	26	27	28
10	11	11	12	14	16	17	18	21	23	24	25	26	27	28	29
11	12	12	13	14	16	18	19	21	23	24	25	26	27	28	29
12	13	14	15	16	17	19	21	23	24	25	26	27	28	29	30
13	15	16	16	17	18	21	23	24	25	26	27	28	29	30	31
15	16	17	18	19	21	23	24	25	26	27	28	29	30	31	32
16	17	18	19	21	23	24	25	26	27	28	29	30	31	32	33
17	19	21	22	23	24	25	26	27	28	29	30	31	32	34	35
18	21	23	23	24	25	26	27	28	29	30	31	32	33	35	36
21	23	24	24	25	26	27	28	29	30	31	32	34	35	36	37
23	24	25	25	26	27	28	29	30	31	32	33	35	36	37	38
24	25	26	26	27	28	29	30	31	32	34	35	36	37	38	39
25	26	27	27	28	29	30	31	32	33	35	36	37	38	39	40
26	27	28	28	29	30	31	32	34	35	36	37	38	39	40	41
27	28	29	29	30	31	32	33	35	36	37	38	39	40	41	42

16x16 to 4x16

6	13	21	27
8	15	23	28
10	16	24	29
11	16	24	29
12	17	25	30
13	18	26	31
15	21	27	32
16	23	28	33
17	24	29	35
18	25	30	36
21	26	31	37
23	27	32	38
24	28	34	39
25	29	35	40
26	30	36	41
27	31	37	42

16x16 to 16x4

6	8	10	11	12	13	15	16	17	18	21	23	24	25	26	27
13	15	16	16	17	18	21	23	24	25	26	27	28	29	30	31
21	23	24	24	25	26	27	28	29	30	31	32	34	35	36	37
27	28	29	29	30	31	32	33	35	36	37	38	39	40	41	42

Results

	All Intra HE			All Intra LC		
	Y	U	V	Y	U	V
Class A	-5.6%	-2.1%	-1.0%	-6.2%	-0.1%	1.0%
Class B	-6.7%	-6.0%	-6.0%	-6.7%	-5.4%	-5.4%
Class C	-15.0%	-13.9%	-14.0%	-14.0%	-13.0%	-13.1%
Class D	-32.4%	-30.4%	-30.7%	-31.0%	-29.2%	-29.4%
Class E	-17.8%	-16.4%	-16.1%	-16.4%	-14.8%	-14.7%
Class F						
Overall	-14.9%	-13.3%	-13.1%	-14.4%	-12.0%	-11.9%
	-14.9%	-13.3%	-13.1%	-14.4%	-12.1%	-12.0%
Enc Time[%]		99%			99%	
Dec Time[%]		98%			98%	

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	-21.0%	-17.3%	-14.7%	-19.9%	-15.0%	-12.2%
Class B	-35.0%	-32.1%	-30.8%	-33.4%	-30.3%	-29.1%
Class C	-52.3%	-50.2%	-50.2%	-51.2%	-49.1%	-49.1%
Class D	-72.6%	-71.1%	-71.2%	-71.9%	-70.6%	-70.8%
Class E						
Class F						
Overall	-44.6%	-42.0%	-41.1%	-43.5%	-40.6%	-39.6%
	-44.6%	-42.1%	-41.2%	-43.5%	-40.7%	-39.7%
Enc Time[%]		100%			100%	
Dec Time[%]		98%			99%	

	Low delay B HE			Low delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	-37.1%	-33.0%	-31.9%	-35.7%	-31.5%	-30.0%
Class C	-52.5%	-50.0%	-49.8%	-51.4%	-48.9%	-49.0%
Class D	-72.2%	-70.4%	-70.6%	-71.7%	-69.9%	-70.2%
Class E	-75.6%	-73.6%	-73.4%	-74.3%	-72.0%	-72.4%
Class F						
Overall	-57.0%	-54.2%	-53.9%	-55.9%	-53.1%	-52.8%
	-56.9%	-54.2%	-54.0%	-55.9%	-53.1%	-52.9%
Enc Time[%]		100%			100%	
Dec Time[%]		98%			99%	

	Low delay P HE			Low delay P LC		
	Y	U	V	Y	U	V
Class A						
Class B	-36.2%	-32.3%	-31.1%	-34.5%	-30.4%	-29.0%
Class C	-51.8%	-49.1%	-48.9%	-50.7%	-48.1%	-48.2%
Class D	-71.7%	-69.7%	-69.9%	-71.2%	-69.2%	-69.7%
Class E	-75.0%	-72.9%	-72.8%	-73.6%	-71.3%	-71.8%
Class F						
Overall	-56.2%	-53.5%	-53.1%	-55.0%	-52.2%	-52.0%
	-56.2%	-53.5%	-53.2%	-55.0%	-52.2%	-52.1%
Enc Time[%]		100%			100%	
Dec Time[%]		98%			99%	

Proposed SPS syntax

seq_parameter_set_rbsp() {	Descriptor
...	
seq_quant_matrix_present_flag	u(1)
if(seq_quant_matrix_present_flag)	
for(quant_matrix_id = 0; quant_matrix_id < 36; quant_matrix_id++) {	
seq_quant_matrix_present_idx[quant_matrix_id]	u(1)
if(seq_quant_matrix_present_idx[quant_matrix_id]) {	
if(quant_matrix_id < 12){	
quant_matrix_coding(quant_matrix_id)	
}else{	
implicit_quant_matrix_present_idx[quant_matrix_id]	u(1)
if(implicit_quant_matrix_present_idx[quant_matrix_id]) {	
quant_matrix_derivation(quant_matrix_id)	
}else{	
quant_matrix_coding(quant_matrix_id)	
}	
}	
}	
}	
...	
rbsp_trailing_bits()	
}	

Proposed PPS Syntax

pic_parameter_set_rbsp() {	Descriptor
...	
pic_quant_matrix_present_flag	u(1)
if(pic_quant_matrix_present_flag)	
for(quant_matrix_id = 0; quant_matrix_id < 36; quant_matrix_id++) {	
pic_quant_matrix_present_idx [quant_matrix_id]	u(1)
if(pic_quant_matrix_present_idx [quant_matrix_id]) {	
if(quant_matrix_id <12){	
quant_matrix_coding (quant_matrix_id)	
}else{	
implicit_quant_matrix_present_idx [quant_matrix_id]	u(1)
if(implicit_quant_matrix_present_idx [quant_matrix_id]) {	
quant_matrix_derivation (quant_matrix_id)	
}else{	
quant_matrix_coding (quant_matrix_id)	
}	
}	
}	
...	
}	

Conclusion

- It is proposed to support quantization matrices in HEVC.
- It is proposed to derive large square quantization matrices (32x32, 16x16) from 4x4 and 8x8 quantization matrices.
- It is proposed to derive non-square quantization matrices from 4x4 and 8x8 quantization matrices.

- Recommend to support quantization matrices in HEVC.
- Recommend to support derivation of large square (32x32, 16x16) and non-square quantization matrices from 4x4 and 8x8 quantization matrices.
- Recommend to adopt or further study the proposed methods for generating large square (32x32, 16x16) and non-square quantization matrices.

MEDIATEK

www.mediatek.com

