

JCTVC-H0275 non-CE10: Introduction of strong filter clipping in deblocking filter

Masaru Ikeda, Teruhiko Suzuki (Sony)

- Summary
- Introduction
- Proposed method
- Experimental results
- Conclusions

- This contribution presents the introduction of strong filter clipping in deblocking filter
 - HM5.0:
 - Strong filter does NOT have clipping process
 - Proposed method:
 - Strong filter has clipping whose value is set to $2 \cdot t_c$
 - In order NOT to be filtered too much without someone's intent
 - With consideration of weak filter clipping with t_c
- BD-bitrates are around 0.0% in almost classes and improved a little bit in class-F
 - In class-F, BD-bitrates are improved as follows
 - AI_HE:-0.6%, RA_HE:-0.4%, LB_HE:-0.4%, LP_HE:-0.4%
 - AI_LC:-0.5%, RA_LC:-0.4%, LB_LC:-0.7%, LP_LC:-0.5%

- Problem statement:
 - Maximum pixel values changed by strong filter are shown in below

	AI_HE				RA_HE				LB_HE				LP_HE			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
Class A	16	25	23	29	12	19	19	18								
Class B	9	37	42	50	15	29	25	84	9	55	53	73	12	38	50	82
Class C	29	40	54	109	9	28	34	35	27	15	34	44	5	24	34	49
Class D	7	13	57	54	6	9	40	42	3	8	20	22	4	8	18	20
Class E	15	16	20	28					5	7	15	17	6	11	16	20
Class F	190	190	190	191	178	189	191	190	191	188	189	182	191	188	181	169
	AI_LC				RA_LC				LB_LC				LP_LC			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
Class A	16	19	23	29	16	19	21	31								
Class B	7	28	36	58	9	17	36	42	7	50	63	47	8	19	67	47
Class C	10	32	52	109	23	31	34	41	5	22	40	38	7	21	40	48
Class D	6	34	56	49	10	9	20	35	4	7	17	19	6	9	19	28
Class E	14	16	29	32					5	10	15	22	5	8	15	19
Class F	189	189	184	190	188	188	181	171	164	189	175	165	182	189	166	164

- Problem statement:
 - Maximum pixel values changed by strong filter are shown in below
 - In some cases, especially class-F, the pixel value is largely changed by strong filter

	AI_HE				RA_HE				LB_HE				LP_HE			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
Class A	16	25	23	29	12	19	19	18								
Class B	9	37	42	50	15	29	25	84	9	55	53	73	12	38	50	82
Class C	29	40	54	109	9	28	34	35	27	15	34	44	5	24	34	49
Class D	7	13	57	54	6	9	40	42	3	8	20	22	4	8	18	20
Class E	15	16	20	28					5	7	15	17	6	11	16	20
Class F	190	190	190	191	178	189	191	190	191	188	189	182	191	188	181	169
	AI_LC				RA_LC				LB_LC				LP_LC			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
Class A	16	19	23	29	16	19	21	31								
Class B	7	28	36	58	9	17	36	42	7	50	63	47	8	19	67	47
Class C	10	32	52	109	23	31	34	41	5	22	40	38	7	21	40	48
Class D	6	34	56	49	10	9	20	35	4	7	17	19	6	9	19	28
Class E	14	16	29	32					5	10	15	22	5	8	15	19
Class F	189	189	184	190	188	188	181	171	164	189	175	165	182	189	166	164

- Problem statement:
 - Maximum pixel values changed by strong filter are shown in below
 - In some cases, especially class-F, the pixel value is largely changed by strong filter
- Therefore adding clipping to strong filter is proposed in order to save this cases

	AI_HE				RA_HE				LB_HE				LP_HE			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
Class A	16	25	23	29	12	19	19	18								
Class B	9	37	42	50	15	29	25	84	9	55	53	73	12	38	50	82
Class C	29	40	54	109	9	28	34	35	27	15	34	44	5	24	34	49
Class D	7	13	57	54	6	9	40	42	3	8	20	22	4	8	18	20
Class E	15	16	20	28					5	7	15	17	6	11	16	20
Class F	190	190	190	191	178	189	191	190	191	188	189	182	191	188	181	169
	AI_LC				RA_LC				LB_LC				LP_LC			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
Class A	16	19	23	29	16	19	21	31								
Class B	7	28	36	58	9	17	36	42	7	50	63	47	8	19	67	47
Class C	10	32	52	109	23	31	34	41	5	22	40	38	7	21	40	48
Class D	6	34	56	49	10	9	20	35	4	7	17	19	6	9	19	28
Class E	14	16	29	32					5	10	15	22	5	8	15	19
Class F	189	189	184	190	188	188	181	171	164	189	175	165	182	189	166	164

- Strong filter in HM5.0

- $p0_i' = (p2_i + 2*p1_i + 2*p0_i + 2*q0_i + q1_i + 4) >> 3$
- $p1_i' = (p2_i + p1_i + p0_i + q0_i + 2) >> 2$
- $p2_i' = (2*p3_i + 3*p2_i + p1_i + p0_i + q0_i + 4) >> 3$
- $q0_i' = (p1_i + 2*p0_i + 2*q0_i + 2*q1_i + q2_i + 4) >> 3$
- $q1_i' = (p0_i + q0_i + q1_i + q2_i + 2) >> 2$
- $q2_i' = (p0_i + q0_i + q1_i + 3*q2_i + 2*q3_i + 4) >> 3$

- Proposed strong filter with clipping

- $p0_i' = p0_i + \text{Clip}_{(-2*tc)-(-2*tc)} ((p2_i + 2*p1_i - 6*p0_i + 2*q0_i + q1_i + 4) >> 3)$
- $p1_i' = p1_i + \text{Clip}_{(-2*tc)-(-2*tc)} ((p2_i - 3*p1_i + p0_i + q0_i + 2) >> 2)$
- $p2_i' = p2_i + \text{Clip}_{(-2*tc)-(-2*tc)} ((2*p3_i - 5*p2_i + p1_i + p0_i + q0_i + 4) >> 3)$
- $q0_i' = q0_i + \text{Clip}_{(-2*tc)-(-2*tc)} ((p1_i + 2*p0_i - 6*q0_i + 2*q1_i + q2_i + 4) >> 3)$
- $q1_i' = q1_i + \text{Clip}_{(-2*tc)-(-2*tc)} ((p0_i + q0_i - 3*q1_i + q2_i + 2) >> 2)$
- $q2_i' = q2_i + \text{Clip}_{(-2*tc)-(-2*tc)} ((p0_i + q0_i + q1_i - 5*q2_i + 2*q3_i + 4) >> 3)$

Experimental results

	All Intra HE			All Intra LC			All Intra HE-10		
	Y	U	V	Y	U	V	Y	U	V
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Enc Time[%]	100%			98%					
Dec Time[%]	100%			99%					
	Random Access HE			Random Access LC			Random Access HE-10		
	Y	U	V	Y	U	V	Y	U	V
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%			100%		
Dec Time[%]	99%			99%			100%		
	Low delay B HE			Low delay B LC			Low delay B HE-10		
	Y	U	V	Y	U	V	Y	U	V
Overall	0.0%	-0.1%	0.2%	0.0%	-0.1%	-0.2%			
	0.0%	-0.1%	0.2%	0.0%	-0.1%	-0.1%			
Enc Time[%]	100%			100%					
Dec Time[%]	100%			99%					
	Low delay P HE			Low delay P LC			Low delay P HE-10		
	Y	U	V	Y	U	V	Y	U	V
Overall	0.0%	0.1%	0.1%	0.0%	-0.1%	-0.1%			
	0.0%	0.1%	0.0%	0.0%	-0.1%	0.0%			
Enc Time[%]	101%			100%					
Dec Time[%]	99%			99%					

- BD-bitrates are around 0.0% in almost classes and improved in class-F without unchanged run-time

	All Intra HE			All Intra LC			All Intra HE-10		
	Y	U	V	Y	U	V	Y	U	V
Class A	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Class E	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Class F	-0.6%	0.0%	0.0%	-0.5%	0.0%	0.0%			
Enc Time[%]	100%			98%					
Dec Time[%]	100%			99%					
	Random Access HE			Random Access LC			Random Access HE-10		
	Y	U	V	Y	U	V	Y	U	V
Class A	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class B	0.0%	-0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class C	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%			
Class D	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%			
Class E									
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class F	-0.4%	-0.1%	-0.1%	-0.4%	-0.1%	-0.1%			
Enc Time[%]	100%			100%			100%		
Dec Time[%]	99%			99%			100%		
	Low delay B HE			Low delay B LC			Low delay B HE-10		
	Y	U	V	Y	U	V	Y	U	V
Class A									
Class B	0.0%	0.1%	0.5%	0.0%	-0.2%	-0.3%			
Class C	0.0%	-0.2%	-0.1%	0.0%	0.1%	-0.1%			
Class D	0.0%	-0.2%	0.4%	-0.1%	0.2%	-0.1%			
Class E	0.1%	0.1%	-0.2%	0.0%	-0.5%	0.0%			
Overall	0.0%	-0.1%	0.2%	0.0%	-0.1%	-0.2%			
	0.0%	-0.1%	0.2%	0.0%	-0.1%	-0.1%			
Class F	-0.4%	0.1%	-0.5%	-0.7%	-0.5%	-0.7%			
Enc Time[%]	100%			100%					
Dec Time[%]	100%			99%					
	Low delay P HE			Low delay P LC			Low delay P HE-10		
	Y	U	V	Y	U	V	Y	U	V
Class A									
Class B	0.0%	0.0%	-0.1%	0.0%	0.0%	-0.1%			
Class C	0.1%	0.1%	0.0%	0.0%	0.0%	0.1%			
Class D	0.0%	0.2%	0.3%	-0.1%	0.1%	0.1%			
Class E	-0.1%	0.1%	0.2%	-0.1%	-0.6%	-0.5%			
Overall	0.0%	0.1%	0.1%	0.0%	-0.1%	-0.1%			
	0.0%	0.1%	0.0%	0.0%	-0.1%	0.0%			
Class F	-0.4%	0.1%	0.0%	-0.5%	-0.3%	-0.5%			
Enc Time[%]	101%			100%					
Dec Time[%]	99%			99%					

- Sony propose to introduce strong filter clipping in deblocking filter
- The clipping can save the cases to be filtered too much without someone's intent.



"SONY" or "make.believe" is a registered trademark and/or trademark of Sony Corporation.

Names of Sony products and services are the registered trademarks and/or trademarks of Sony Corporation or its Group companies.

Other company names and product names are the registered trademarks and/or trademarks of the respective companies.

- Strong filter in HM5.0

$$- p0_i' = (p2_i + 2*p1_i + 2*p0_i + 2*q0_i + q1_i + 4) >> 3$$

$$= \textcolor{red}{p0_i} + (p2_i + 2*p1_i - \textcolor{red}{6*p0_i} + 2*q0_i + q1_i + 4) >> 3$$


1st part: Input pixel

2nd part: Pixel changed by strong filter

- Proposed strong filter with clipping

$$- p0_i' = p0_i + \textcolor{red}{Clip}_{(-2*tc)-(-2*tc)} ((p2_i + 2*p1_i - 6*p0_i + 2*q0_i + q1_i + 4) >> 3)$$