

JCTVC-N0245

MV data buffer size analysis

Picture based buffer size estimation

- Picture based buffer size estimation was provided in JCTVC-M0142
- For each of the motion information, the following data needs to be stored:
 - Mvx (16bits)
 - Mvy (16bits)
 - Refidx (4bits)
- So totally $16+16+4 = 36\text{bits}$ will be necessary.
 - If byte-alignment is considered, **40bits** will be necessary

Buffer size estimation

- Assuming that
 - W: EL Width
 - H: EL Height
 - R: Scalability Ratio (2, 1.5 or 1)
 - D: Decimation Ratio of colblmv (4, 2 or 1)
 - N: Number of DPB
- Then (bi-directional case)
 - $\text{EL_pixel_buf_size} = W * H * 8 \text{ [bit]} * 1.5 = 12 * W * H$
 - $\text{BL_mv_buf_size} = (W/R) * (H/R) * (1/4D * 1/4D) * 40 \text{ [bit]} * 2 = (5 / (D^2 R^2)) * W * H$
 - $\text{EL_mv_buf_size} = W * H * (1/16 * 1/16) * 40 \text{ [bit]} * 2 * N = 5/16 * N * W * H$
- Only L0 and L1 motion data of the current frame need to be stored for the base layer,
- whereas L0 and L1 data for all reference frames need to be stored for the enhancement layer as it may be used for encoding/decoding of the future frames within the same layer.

Buffer size estimation

- Assuming that N=6 (bi-directional case)
 - $\text{EL_mv_buf_size} / \text{EL_pixel_buf_size} = (5/16) / 12 * 6 = 15.6\%$
 - $\text{BL_mv_buf_size} / \text{EL_pixel_buf_size} = (5 / D^2R^2) / 12 = 5 / (12 * D^2 * R^2)$
 - =0.65% (R=2; D=4)
 - =1.16% (R=1.5; D=4)
 - =2.6% (R=1; D=4)
 - =2.6% (R=2; D=2)
 - =4.6% (R=1.5; D=2)
 - =10.4% (R=1; D=2)
 - =10.4% (R=2; D=1)
 - =18.5% (R=1.5; D=1)
 - =41.7% (R=1; D=1)

								MVsizeInPic x DPBnum		
		W	H	R	D	DPB	BL_mv_buf_size[bit]	EL_mv_buf_size[bit]	total MV data [byte]	diff [byte]
4KSNR	curr. (4:1)	3840	2160	1	4	6	2592000	15552000	2268000	
4KSNR	2:2 (2:1)	3840	2160	1	2	6	10368000	15552000	3240000	972000 (+42.8%)
4K SNR	Uncompress	3840	2160	1	2	6	41472000	15552000	57024000	54756000 (+241%)

PU/LCU based decoding

- Base layer MV can be used on the fly
- Then required MV data size is smaller
- Worst case PU size is 64x64 (=LCU)

- **2x Spa. Scalability**
 - Stored by 16x16:
 - $(64/2/16) * (64/2/16) * 2 * 40\text{bits} = 320\text{bits}$
 - Stored by 8x8:
 - $(64/2/8) * (64/2/8) * 2 * 40\text{bits} = 1280\text{bits}$
 - Stored by 4x4
 - $(64/2/4) * (64/2/4) * 2 * 40\text{bits} = 5120\text{bits}$
- **SNR Scalability**
 - Store by 16x16
 - $(64/16) * (64/16) * 2 * 40\text{bits} = 1280\text{bits}$
 - Stored by 8x8
 - $(64/8) * (64/8) * 2 * 40\text{bits} = 5120\text{bits}$
 - Stored by 4x4
 - $(64/4) * (64/4) * 2 * 40\text{bits} = 20480\text{bits}$

	2X Spaial	SNR
4:1 compression	-	-
2:1 compression	960 bit	4800 bit
Uncompress	3840 bit	19200 bit