

SHARP

Be Original.

Non-CE9: A speed up method of BDOPF JVET-O0211

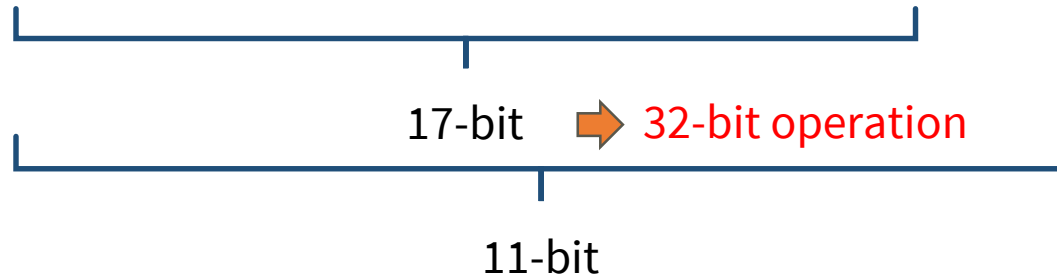
SHARP CORPORATION

Takeshi Chujoh and Tomohiro Ikai

- Current specification of deriving the gradient values for BDOF, 32-bit operation for each pixel is requested.
- In this contribution, by changing the order of subtraction and shift, 16-bit operations is realized.
 - When SIMD is used, the number of parallel operations become double and the speed of the processing is upped.
- As experimental results, the 1% encoding time has been improved and there was a little gain on average.

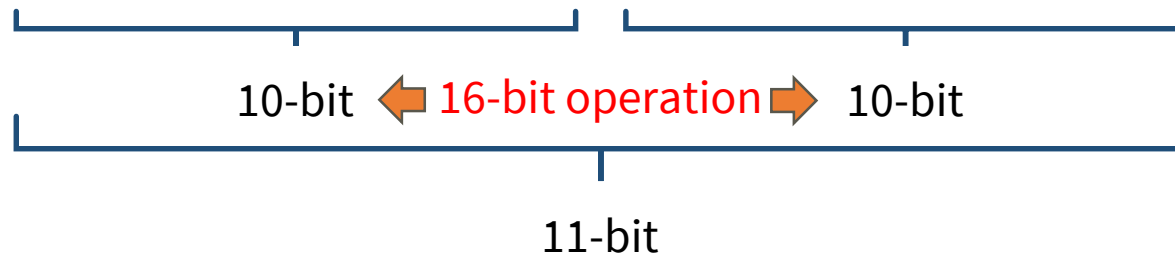
- The dynamic range of variable “predSamples” is 16-bit and the value of variable “shif1” is equal to 6 on CTC.

$$\text{gradientHL0}[x][y] = (\text{predSamplesL0}[hx + 1][vy] - \text{predSampleL0}[hx - 1][vy]) \gg \text{shift1} \quad (8-769)$$



- Change the order of subtraction and right shift

$$\text{gradientHL0}[x][y] = (\text{predSamplesL0}[hx + 1][vy] \gg \text{shift1}) - (\text{predSampleL0}[hx - 1][vy] \gg \text{shift1}) \quad (8-769)$$



- In case of 128-bit register, it can be improved from 4-parallel to 8-parallel.

- Change 4 lines.

8.5.6.4 Bidirectional optical flow prediction process

From

The variables $\text{gradientHL0}[x][y]$, $\text{gradientVL0}[x][y]$, $\text{gradientHL1}[x][y]$ and $\text{gradientVL1}[x][y]$ are derived as follows:

$$\text{gradientHL0}[x][y] = (\text{predSamplesL0}[h_x + 1][v_y] - \text{predSampleL0}[h_x - 1][v_y]) \gg \text{shift1} \quad (8-769)$$

$$\text{gradientVL0}[x][y] = (\text{predSampleL0}[h_x][v_y + 1] - \text{predSampleL0}[h_x][v_y - 1]) \gg \text{shift1} \quad (8-770)$$

$$\text{gradientHL1}[x][y] = (\text{predSamplesL1}[h_x + 1][v_y] - \text{predSampleL1}[h_x - 1][v_y]) \gg \text{shift1} \quad (8-771)$$

$$\text{gradientVL1}[x][y] = (\text{predSampleL1}[h_x][v_y + 1] - \text{predSampleL1}[h_x][v_y - 1]) \gg \text{shift1} \quad (8-772)$$

to

The variables $\text{gradientHL0}[x][y]$, $\text{gradientVL0}[x][y]$, $\text{gradientHL1}[x][y]$ and $\text{gradientVL1}[x][y]$ are derived as follows:

$$\text{gradientHL0}[x][y] = (\text{predSamplesL0}[h_x + 1][v_y] \gg \text{shift1}) - (\text{predSampleL0}[h_x - 1][v_y] \gg \text{shift1}) \quad (8-769)$$

$$\text{gradientVL0}[x][y] = (\text{predSampleL0}[h_x][v_y + 1] \gg \text{shift1}) - (\text{predSampleL0}[h_x][v_y - 1] \gg \text{shift1}) \quad (8-770)$$

$$\text{gradientHL1}[x][y] = (\text{predSamplesL1}[h_x + 1][v_y] \gg \text{shift1}) - (\text{predSampleL1}[h_x - 1][v_y] \gg \text{shift1}) \quad (8-771)$$

$$\text{gradientVL1}[x][y] = (\text{predSampleL1}[h_x][v_y + 1] \gg \text{shift1}) - (\text{predSampleL1}[h_x][v_y - 1] \gg \text{shift1}) \quad (8-772)$$

- The 1% encoding time has been improved.
- There was a little gain on average.

	Random Access Main 10				
	Over VTM-5.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	-0.20%	-0.04%	99%	100%
Class A2	-0.01%	-0.05%	-0.02%	100%	100%
Class B	0.00%	0.02%	0.02%	99%	100%
Class C	-0.02%	-0.02%	-0.07%	99%	100%
Class E					
Overall	-0.01%	-0.05%	-0.02%	99%	100%
Class D	0.00%	0.00%	0.15%	99%	99%
Class F	0.00%	0.00%	-0.01%	99%	100%

- The coding efficiency results were identical to ours.
- The 1% decoding time has been improved.

	Random access Main10				
	Over VTM-5.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	-0.20%	-0.04%	100%	100%
Class A2	-0.01%	-0.05%	-0.02%	99%	99%
Class B	0.00%	0.02%	0.02%	100%	100%
Class C	-0.02%	-0.02%	-0.07%	100%	99%
Class E					
Overall	-0.01%	-0.05%	-0.02%	100%	99%
Class D	0.00%	0.00%	0.15%	100%	97%
Class F	0.00%	0.00%	-0.01%	100%	99%

- In this contribution, we proposed a speed up method for BDOF.
- As an experimental result the encoding times became 1% faster and the average gains of coding efficiency were -0.01%/-0.05%/-0.02% for Y/Cb/Cr respectively.
- Recommend to fix this change to WD and VTM software.
- Thank Unno-san (KDDI, JVET-00982) for cross-checking.

SHARP

Be Original.

