

SHARP

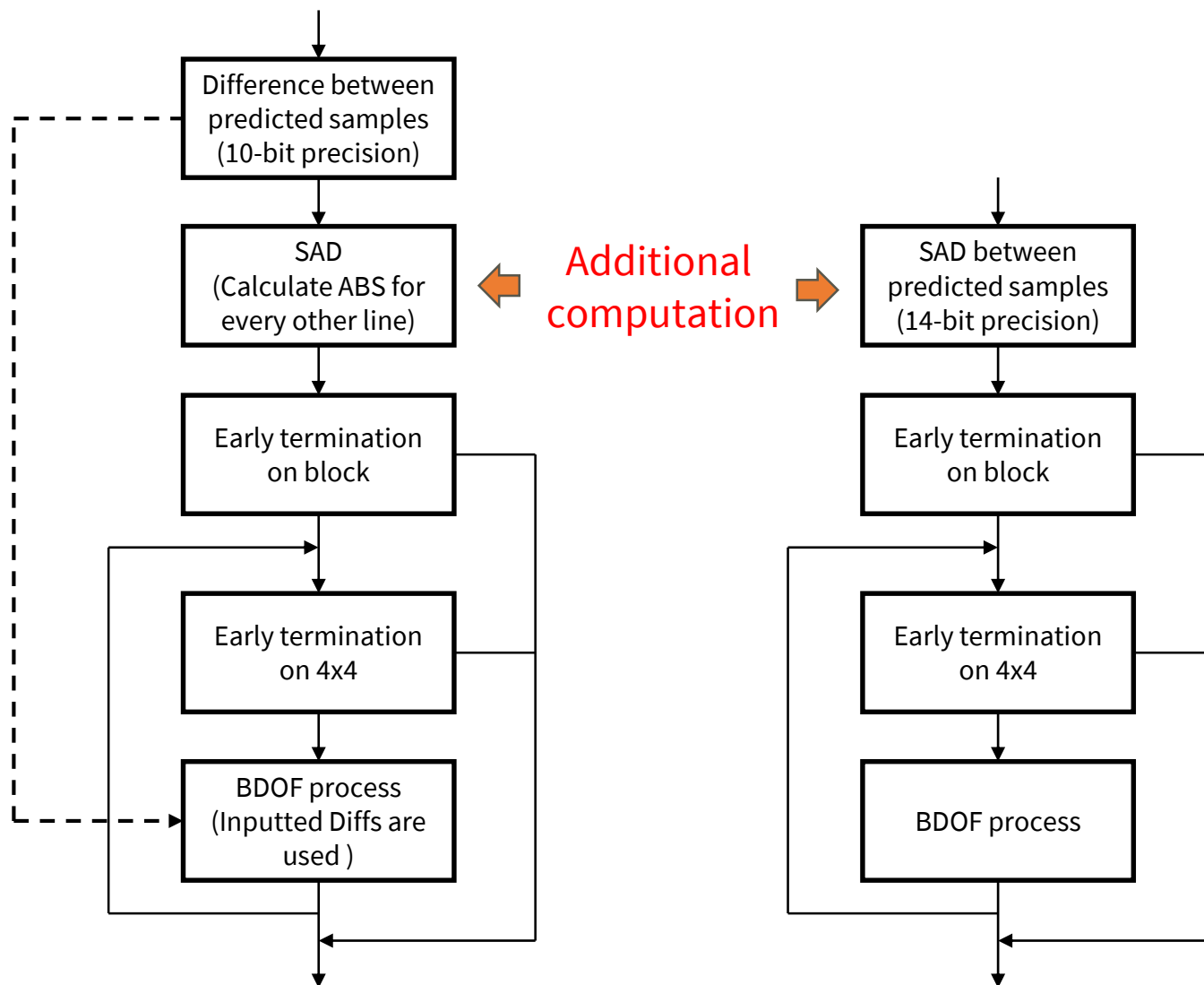
Be Original.

Non-CE9: An improvement of early termination for BDOF JVET-00208

SHARP CORPORATION

Takeshi Chujoh, Eiich Sasaki and Tomohiro Ikai

- Early termination of BDOF cannot reduce the worst of computational complexity.
 - A problem from the viewpoint of hardware design.
 - CE9-1.2 can remove the early termination of BDOF, but what should be done when DMVR is off?
- In this contribution, a method to make the difference for calculating SAD by using derivation process of BDOF is proposed.
 - SAD calculation for every other line the same as the early termination of DMVR is introduced.
 - The worst case of complexity has been reduced.
- As experimental results, the 1% decoding time was improved and there was a little gain.



Proposed method

Current method

• Change 5 lines

8.5.6 Decoding process for inter blocks

8.5.6.1 General

If bdofFlag is equal to TRUE, the following applies:

The variable shift is set equal to ~~Max(2, 14 - BitDepth_y)~~ **Max(4, bitDepth - 8).**

The variables sbDiffThres, bdofBlkDiffThres, and sbSumDiff are derived as follows:

$$\text{sbDiffThres} = \text{Max}(\text{Max}(2, 14 - \text{BitDepth}_y) - 8 + \text{shift}) * \text{sbWidth} * \text{sbHeight} \quad (8-705)$$

$$\text{bdofBlkDiffThres} = \text{Max}(\text{Max}(2, 14 - \text{BitDepth}_y) - 3 + \text{shift}) * 64 \quad (8-706)$$

$$\text{sbSumDiff} = 0 \quad (8-707)$$

For xldx = 0..(sbWidth >> 2) - 1 and yldx = 0..(sbHeight >> 2) - 1, the variables bdofBlkSumDiff and the bidirectional optical flow utilization flag bdofUtilizationFlag[xldx][yldx] are derived as follows:

$$\begin{aligned} \text{bdofBlkSumDiff} &= \sum_{i=0}^3 \sum_{j=0}^1 \text{Abs}((\text{predSamplesL0}_L[(\text{xldx} \ll 2) + 1 + i][(\text{yldx} \ll 2) + 1 + 2*j] \gg \text{shift}) - \\ &\quad (\text{predSamplesL1}_L[(\text{xldx} \ll 2) + 1 + i][(\text{yldx} \ll 2) + 1 + 2*j] \gg \text{shift})) \end{aligned} \quad (8-708)$$

$$\text{bdofUtilizationFlag}[\text{xldx}][\text{yldx}] = \text{bdofBlkSumDiff} \geq \text{bdofBlkDiffThres} \quad (8-709)$$

$$\text{sbSumDiff} += \text{bdofBlkSumDiff} \quad (8-710)$$

The variable sbBdofFlag is derived as follows:

If sbSumDiff is less than sbDiffThres, sbBdofFlag is set equal to FALSE.

Otherwise, sbBdofFlag set equal to TRUE.

As an implementation, Equation (8-708) can use the result of Equation (8-773) in BDOF prediction process .

$$\text{diff}[\text{x}][\text{y}] = (\text{predSamplesL0}[\text{h}_x][\text{v}_y] \gg \text{shift2}) - (\text{predSamplesL1}[\text{h}_x][\text{v}_y] \gg \text{shift2}) \quad (8-773)$$

- The 1% decoding time has been improved.
- There was a little gain.

	Random Access Main 10				
	Over VTM-5.0				
	Y	U	V	EncT	DecT
Class A1	-0.01%	-0.17%	-0.04%	100%	99%
Class A2	0.01%	0.02%	0.04%	100%	99%
Class B	0.00%	0.04%	0.01%	100%	99%
Class C	-0.03%	-0.02%	-0.10%	100%	99%
Class E					
Overall	-0.01%	-0.02%	-0.02%	100%	99%
Class D	0.04%	-0.03%	-0.14%	99%	99%
Class F	0.00%	0.03%	0.02%	100%	99%

- The coding efficiency results were identical to ours.
- The 1% encoding and decoding times have been improved.

	Random Access Main 10				
	Over VTM-5.0				
	Y	U	V	EncT	DecT
Class A1	-0.01%	-0.17%	-0.04%	99%	99%
Class A2	0.01%	0.02%	0.04%	100%	99%
Class B	0.00%	0.04%	0.01%	99%	99%
Class C	-0.03%	-0.02%	-0.10%	98%	99%
Class E					
Overall	-0.01%	-0.02%	-0.02%	99%	99%
Class D	0.04%	-0.03%	-0.14%	98%	99%
Class F	0.00%	0.03%	0.02%	97%	99%

- In this contribution, we proposed a method to make the difference for calculating SAD by using derivation process of BDOF.
 - The worst case of complexity has been reduced.
- As an experimental result, the decoding times became 1% faster and the average gains of coding efficiency were -0.01%/-0.02%/-0.02% for Y/Cb/Cr, respectively.
- Recommend to fix this change to WD and VTM software.
- Thank Kondo-san (Sony, JVET-O0987) for cross-checking.

SHARP

Be Original.

