Simplification of BCW in Versatile Video Coding (VVC)

Dohyeon Park, Jae-Gon Kim, *Jinho Lee, and *Jungwon Kang
Korea Aerospace University, *ETRI
dhpark@kau.kr, *{jinosoul, jungwon}@etri.re.kr, jgkim@kau.ac.kr

Abstract

The emerging Versatile Video Coding (VVC) standard introduces Bi-prediction with CU-level Weights (BCW) to enhance the bi-predictive prediction. The syntax element of BCW index is adaptively coded according to the value of *NoBackwardPredFlag* which indicates if there is no future picture in the display order among the reference pictures, and it can violate the flexibility of codec and cause the dependency issue. This paper proposes BCW clean-up design that allows all weights can be parsed without any condition. The experimental results show negligible BD-rate losses while resolving the issues.

1. Introduction

More efficient video coding technologies are still required in the emerging immersive media services such as 4K/8K Ultra High-Definition (UHD) videos, and 360˚/Virtual Reality (VR) videos. Accordingly, Joint Video Experts Team (JVET) founded by ISO/IEC Moving Picture Experts Team (MPEG) and ITU-T Video Coding Experts Group (VCEG) is developing a new video coding standard named as Versatile Video Coding (VVC) beyond High-Efficiency Video Coding (HEVC) [1]. JVET has been releasing a draft document and the reference software called VVC Test Model (VTM) with new adopted coding tools in each meeting [2]. Currently, VVC Committee Draft (CD) and VTM6.0 are released after the 15th JVET meeting in Gothenburg, Sweden [3].

In VVC, the performance of inter prediction has been improved by allowing more flexible bi-predictive modes such as Triangle Partition Mode (TPM) and Bi-prediction with CU-level Weights (BCW) [4]. TPM is implemented to cover the defect of block-based coding with a sample-wise weighted bi-prediction that divides a rectangular coding block into two triangular blocks. On the other hand, the block-wise weight of BCW compensates for the temporal illumination change by weighted averaging of the two prediction signals obtained from two different reference pictures. In order to improve the weighted bi-predictive predictions at the aspect of coding performance and complexity, multiple contributions are positively discussed during the latest JVET meeting [5]. In this paper, the BCW design, which adaptively adjusts the weights according to the value of *NoBackwardPredFlag*, is introduced. Furthermore, the simplification method is proposed to resolve the flexibility and dependency issues of the design.

The rest of the paper is organized as follows. Section 2 describes the details of VTM6's BCW. Then, the raising issues and the proposed solution are presented in Section 3. After the performance analysis of the proposed method, this paper is concluded.

2. BCW in VTM6.0

In the VTM6, BCW is integrated to allow weighted averaging of the two inter-predictions, as shown in the following formula:

$$P \qquad =((8-w)\times P+w\times P+4)\gg 3 \qquad (1)$$

P_0 and P_1 are prediction blocks derived by bi-predictive motion information, and $P_{bi\text{-}pred}$ is the final motion-compensated block of BCW. For each bi-predictive CU, the weight w is determined among five weight values, $w \in$ {-2,3,4,5,10}. In the current design, some weights are restricted depending on the value of *NoBackwardPredFlag*. *bcw_idx* to be transmitted are also binarized differently. If *NoBackwardPredFlag* is equal to 1, which means there is no future picture in the display order among the reference pictures, all five weights are allowed. Otherwise, only three weights can be used. The weights and corresponding bin strings, according to *NoBackwardPredFlag*, are shown in Table 1.

Table 1. The weights and binarization for *bcw_idx*

bcw_idx	Weights	Bin string (*NoBackwardPredFlag*==1)	Bin string (*NoBackwardPredFlag*==0)
0	4	0	0
1	5	10	10
2	3	110	11
3	10	1110	Not allowed
4	-2	1111	Not allowed

3. Proposed method

The current syntax design of *bcw_idx* can cause some issues. Normative restriction on the possible weights can violate the flexibility of the codec. It is desirable that the adjustment on the number of available weights is handled at encoder, non-normatively. Besides, the condition check of *NoBackwardPredFlag*, which is determined by slice type and reference pictures, can cause a dependency on the parsing of *bcw_idx*. To solve these issues, we proposed a method to remove the condition check of *NoBackwardPredFlag* at the parsing of *bcw_idx*. In the proposed method, the selected weight among 5 weights ($w \in$ {-2,3,4,5,10}) can be parsed without any condition check. The adjustment on the number of available weights should be handled non-normatively.

The BCW weight index is binarized with truncated unary (TU), as shown in Table 2. The first bin of the bin string is context coded with its own context model, and rest bins are bypass coded.

Table 2. The proposed codeword for *bcw_idx*

bcw_idx	Weights	Bin string
0	4	0
1	5	10
2	3	110
3	10	1110
4	-2	1111

3. Experimental results

The proposed method is implemented on top of VTM6.0. It is evaluated in Random Access (RA) and Low-Delay B (LDB) configuration according to VTM Common Test Condition (CTC) [6]. The algorithm for weight searching in VTM6 is not changed (3 weights are searched in Random Access (RA) configuration), and only the part of *bcw_idx* coding is modified. All experimental results are compared over the anchor of VTM6.0. Table 3 shows the results of the proposed method in RA configuration. The experimental results reportedly show 0.02% luma BD-rate change. Since the encoding remains the same when *NoBackwardPredFlag* is 0, no BD-rate changes are also reported in LDB configuration.

Table 3. Experimental results in Random Access

Class	**Over VTM-6.0 in RA**				
	Y	U	V	EncT	DecT
Class A1	0.04%	0.05%	0.07%	100%	101%
Class A2	0.04%	0.06%	0.06%	100%	101%
Class B	0.02%	0.08%	-0.04%	100%	101%
Class C	0.02%	0.09%	0.03%	100%	102%
Overall	**0.02%**	**0.07%**	**0.02%**	**100%**	**101%**
Class D	-0.01%	-0.06%	0.12%	100%	102%
Class F	0.01%	0.02%	0.01%	100%	101%

Table 4. Experimental results in Low Delay B

Class	**Over VTM-6.0 in LDB**				
	Y	U	V	EncT	DecT
Class B	0.00%	0.00%	0.00%	100%	97%
Class C	0.00%	0.00%	0.00%	100%	99%
Class E	0.00%	0.00%	0.00%	100%	100%
Overall	0.00%	0.00%	0.00%	**100%**	**99%**
Class D	0.00%	0.00%	0.00%	99%	98%
Class F	0.00%	0.00%	0.00%	100%	95%

4. Conclusion

This paper presents a clean-up design for the syntax of BCW by removing the condition check of *NoBackwardPredFlag* in weight usage. In the proposed method, 5 BCW weights are used without any condition. The method resolves the potential issues of codec flexibility and decoding dependency of BCW with negligible coding loss.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00572).

References

[1] High Efficiency Video Coding, Version 1, Rec. ITU-T H.265, ISO/IEC 23008-2, Jan. 2013.

[2] [Online]. Available at: VVC Test Model (VTM), https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

[3] B. Bross, J. Chen, and S. Liu, “Versatile Video Coding (Draft 6),” 15th JVET meeting, Doc. JVET-O2001, Jul. 2019.

[4] W. Chien and J. Boyce, “JVET AHG report: Tool reporting procedure (AHG13),” 15th JVET meeting, Doc. JVET-O0013, Jul. 2019.

[5] D. Park, Y. Yoon, J. Do, J. Kim, J. Lee, and J. Kang, “Non-CE4: BCW clean-up for weight signaling,” 16th JVET meeting, Doc. JVET-P0317, Oct. 2019.

[6] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Suhring, “JVET common test conditions and software reference configurations for SDR video,” 14th JVET meeting, JVET-N1010, May. 2019.