electronics

MDPI

Article

Performance Comparison of Weak Filtering in HEVC and VVC

Junghyun Lee and Jechang Jeong *

Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, Korea; mizali@hanyang.ac.kr

* Correspondence: jjeong@hanyang.ac.kr

Received: 8 May 2020; Accepted: 5 June 2020; Published: 9 June 2020

Abstract: This study describes the need to improve the weak filtering method for the in-loop filter process used identically in versatile video coding (VVC) and high efficiency video coding (HEVC). The weak filtering process used by VVC has been adopted and maintained since Draft Four during H.265/advanced video coding (AVC) standardization. Because the encoding process in the video codec utilizes block structural units, deblocking filters are essential. However, as many of the deblocking filters require a complex calculation process, it is necessary to ensure that they have a reasonable effect. This study evaluated the performance of the weak filtering portion of the VVC and confirmed that it is not functioning effectively, unlike its performance in the HEVC. The method of excluding the whole of weak filtering from VVC, which is a non-weak filtering method, should be considered in VVC standardization. In experimental result in this study, the non-weak filtering method brings 0.40 Y-Bjontegaard-Delta Bit-Rate (BDBR) gain over VVC Test Model (VTM) 6.0.

Keywords: video signal processing; video compression; video coding; video codecs; HEVC; VVC; in-loop filter; deblocking filter

1. Introduction

The video compression standard in which block unit encoding takes place requires a deblocking filter to reduce blocking artifacts. In H.264 advanced video coding (AVC), the deblocking filter processes vertical and horizontal edge filtering sequentially in macroblock units [1]. In the high efficiency video coding (HEVC) standard, vertical edges are treated in picture units, followed by horizontal edges in picture units. Through this method, parallel processing is possible, which is also beneficial in terms of complexity [2–4]. Currently, the versatile video coding (VVC) standard uses the same process for the deblocking filter.

The deblocking filter of current video coding standards contains both strong and weak filtering processes. The degree of blocking artifacts is determined according to the conditions of the reconstructed pixels at the block boundary and the quantization parameter (QP) value used for encoding. Therefore, the deblocking filter is determined by adaptive strong or weak filtering. For blocks with weak filtering selected as the deblocking filter, the additional process of evaluating the situation of the pixels on the boundary determines whether the two or four pixels closest to the boundary are filtered. These current weak filtering methods have been adopted in H.265/AVC Draft Four and have remained the same to date [5,6].

Previous studies have shown that the process of filtering up to four pixels using weak filtering in the VVC test model (VTM) only adds complexity and thus reduces performance [7]. The simple weak filtering, omitting second stage of the weak filtering, showed better performance than the conventional weak filtering method in VVC. As well as part of the weak filtering, it is necessary to make sure that the whole of the weak filtering is contributing to the improvement of visual quality.

In this study, we experiment with non-weak filtering methods and measure the performance of the whole weak filtering in VVC. The logical proof of the method is the measured performance of the existing weak filtering method for the HEVC test model (HM) and the performance of the VTM. Experimental results show that in HEVC, weak filtering has an accurate deblocking effect; however, in VVC, weak filtering does not perform its role effectively. The non-weak filtering method provides a 0.40% average Bjontegaard-Delta Bit-Rate (BDBR) [8] gain for the luminance component in all intra (AI) mode.

This study is organized in five sections. In Section 2, we introduce the conventional deblocking filter, weak filter and simple weak filtering methods verified in previous studies. In Section 3, we present the non-weak filtering method to evaluate the performance of weak filtering in HEVC and VVC. The experimental results and comparisons with HEVC and VVC are presented in Section 3. Finally, we present some conclusions regarding our method in Section 4.

2. Related Works and Method

2.1. Related Works

2.1.1. Deblocking Filter

The conventional weak filtering method used is that proposed in Draft Four of the HEVC. At that time, several contributions regarding weak filtering methods were proposed [9]. Of these, two contributions were combined and adopted as the weak filtering method, which is divided into the current two parts [10–13]. Both contributions could be adopted, especially as assessments involving subjective visual quality were made.

To illustrate the conventional deblocking filter process, consider the example of a vertical edge consisting of reconstructed blocks on sides P and Q, as shown in Figure 1. Each side is a reconstructed block composed of 4 × 4 pixels. In this study, we will introduce weak filtering of luminance components, and each pixel, $p_{i,j}$ or $q_{i,j}$, represents the luminance pixel value corresponding to location (i, j) in each side.

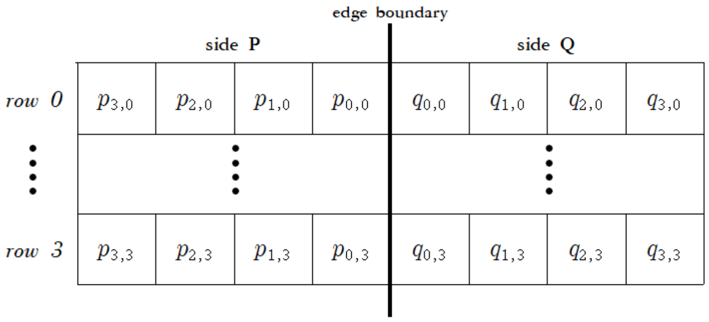


Figure 1. Example of the pixels of two blocks in side P and side Q relative to edge boundary for application of deblocking filter.

First, the boundary strength (BS) values are defined from a table in the relevant standard according to the prediction modes for side P and side Q and the results of the encoding (transform coefficient values, motion vectors, etc.).

For the deblocking of luminance components, the VVC, as with HEVC, calculates the d value according to (1) in four rows or four columns if the BS value is not zero. The true/false of condition (2) with the obtained d value determines if the deblocking filter is on or off.

$$d = |p_{2,0} - 2p_{1,0} + p_{0,0}| + |p_{2,3} - 2p_{1,3} + p_{0,3}| + |q_{2,0} - 2q_{1,0} + q_{0,0}| + |q_{2,3} - 2q_{1,3} + q_{0,3}|, \tag{1}$$

$$d < \beta, \tag{2}$$

The value of β in (2) is determined based on a table in the relevant standard and depends on the QP values used for side P and side Q. If "deblocking filter on" is determined, *d_strong* is calculated using (3) and the three conditions in (4)–(6) are checked for row 0 and row 3 in Figure 1. In this study, only the process of weak filtering needs to be modified, so the process description of strong filtering is omitted.

If any of the six conditions for the two rows are not satisfied, weak filtering is used.

$$d\_strong = \left|p_3 - p_0\right| + \left|q_3 - q_0\right|, \tag{3}$$

$$\left|p_{2,0} - 2p_{1,0} + p_{0,0}\right| + \left|q_{2,0} - 2q_{1,0} + q_{0,0}\right| < (\beta \gg 2), \tag{4}$$

$$d\_strong < (\beta \gg 3), \tag{5}$$

$$\left|p_0 - q_0\right| < (5 \times tc + 1) \gg 1, \tag{6}$$

Reconstructed pixels $p_{i,N}$, $q_{i,N}$ in the Nth row of Figure 1 are expressed as p_i, q_i respectively, in (3)–(6). The value of *tc* in (6) is determined based on a table the relevant standard and depends on the QP values used for side P and side Q in the same way as in the value of β. The decision table of β and *tc* for 8-bit depth video is shown in Table 1.

Table 1. Decision table for β and *tc* for 8-bit depth video.

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7
tc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
QP	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
β	8	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32
tc	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4
QP	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
β	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	-	-
tc	4	4	5	5	6	6	7	8	9	10	11	13	14	16	18	20	22	24

2.1.2. Weak Filter

If weak filtering is determined to be used by a deblocking filter, the Δ is calculated to check the extent to which the reconstructed pixels near the boundary target require weak filtering.

The reconstructed pixels around the boundary are used to calculate Δ with (7) to determine how the weak filtering should be applied.

$$\Delta = (9 \times (q_0 - p_0) - 3 \times (q_1 - p_1) + 8) \gg 4, \tag{7}$$

If the absolute value of Δ is greater than $10 \times tc$, weak filtering does not apply to that boundary. This means that no pixels are filtered. Otherwise, in Figure 1, the filtering is performed first for the two pixels p_0 and q_0 closest to the boundary. The weak filtering calculation method for these two pixels is as follows:

$$p_0{}' = Clip3(0, 255, p_0 + \Delta), \tag{8}$$

$$q_0{}' = Clip3(0, 255, q_0 - \Delta), \tag{9}$$

$Clip3(\min, \max, x)$ in (8) and (9) is a clipping function that clips the value of x to values between the minimum (min) and maximum (max).

The boundary conditions are then checked again to determine whether the second weak filtering target, pixels p_i and q_i, will be filtered, as follows:

$$\left|p_{2,0} - 2p_{1,0} + p_{0,0}\right| + \left|p_{2,3} - 2p_{1,3} + p_{0,3}\right| < (\beta + (\beta \gg 1)) \gg 3, \tag{10}$$

$$\left|q_{2,0} - 2q_{1,0} + q_{0,0}\right| + \left|q_{2,3} - 2q_{1,3} + q_{0,3}\right| < (\beta + (\beta \gg 1)) \gg 3, \tag{11}$$

If (10) is true, Δp is calculated for correction of p_i using (12) and $p_i{}'$, which is the filtered pixel value, is calculated using (13). Similarly, if (11) is true, q_i filtering is performed using (14) and (15).

$$\Delta p = Clip3(-(tc \gg 1),\ tc \gg 1,\ (((p_2 + p_0 + 1) \gg 1) - (p_1 + \Delta)) \gg 1), \tag{12}$$

$$p_1' = Clip3(0,\ 255,\ p_1 + \Delta p), \tag{13}$$

$$\Delta q = Clip3(-(tc \gg 1),\ tc \gg 1,\ (((q_2 + q_0 + 1) \gg 1) - (q_1 + \Delta)) \gg 1), \tag{14}$$

$$q_1' = Clip3(0,\ 255,\ q_1 + \Delta q), \tag{15}$$

2.1.3. Simple Weak Filter Method on VVC

The weak filtering method is divided into two stages. In the first step, the two pixels closest to the boundary, p_0 and q_0, are filtered, while the second step involves filtering the next two pixels, p_1 and q_1. Previous studies have suggested a simple weak filtering method to eliminate the second phase of the process. In that case, all of the processes from (7) to (15) that are calculated to filter p_1 and q_1 may be omitted.

The simple weak filtering method has shown an average 0.27% Y-BDBR gain for AI mode over VTM 5.0. The experimental results for VTM 6.0 with a simple weak filtering method are presented in Section 3.

2.2. Method

Going a step further from the simple weak filtering method, a non-weak filtering method is proposed in this study to ensure that the whole weak filtering process is valid for the VVC. The non-weak filtering method uses strong filtering according to the conditions in the deblocking filter or to omit filtering altogether in the block. By comparing the non-weak filtering method with the encoding results of conventional codec standards such as HEVC or VVC, the performance of the weak filtering method can be demonstrated in the conventional video codec standard.

Because the weak filtering method is coded identically in HEVC and VVC, we would like to compare the non-weak filtering method with two video codec standard test models. In the HEVC test model (HM) and VTM, weak filtering is performed using the following code (Algorithm 1).

Strong filtering is coded in Algorithm 1 (a) part. Strong filtering is performed if "sw" in the code, which acts as a flag through the "if" statement, is true. Strong filtering method has additional details in VVC than in HEVC, but the subject of this study is weak filtering, so this is omitted in Algorithm 1 (a) part.

Weak filtering was executed by the "else" syntax in Algorithm 1 (b) and Algorithm 1 (c) part. The detailed code for weak filtering could identify weak filtering structures that were divided into two parts by multiple "if" statements. The Algorithm 1 (b) part includes the calculation of delta values for the block boundary and the calculation of weak filtering for the two pixels p_0 and q_0 nearest the boundary. The Algorithm 1 (c) part includes the calculation that filters when each boundary condition was true for the two pixels p_1 and q_1 that were the second closest to the boundary. The flags, *bFilterSecondP* and *bFilterSecondQ*, were used identify boundary conditions.

Given that the simple weak filtering method to eliminate Algorithm 1 (c) part showed performance in VVC, it is necessary to ensure that the entire weak filtering is functioning correctly in VVC. Non-weak filtering is proposed to measure the performance of the entire weak filtering in VVC. The non-weak filtering method completely deletes the "else" syntax from the above code, Algorithm 1 (b) and Algorithm 1 (c) parts, and proceeds with encoding. In other words, strong filtering is performed based on the conditions as in the traditional method, and other methods do not apply deblocking filters. In addition to the filtering part, all process of calculating included flags, *bFilterSecondP* and *bFilterSecondQ*, can be omitted to further reduce complexity.

Algorithm 1: Weak Filter code in HM and VTM.

```
if (sw)
{
    Executable syntax for strong filtering              (a)
    (Skip detailed code)
}
else
{
  delta = (9 * (m4 − m3) − 3 * (m5 − m2) + 8) >> 4;

  if (abs(delta) < iThrCut)                              (b)
  {
    delta = Clip3(−tc, tc, delta);
    piSrc[-iOffset] = ClipPel(m3 + delta, clpRng);
    piSrc[0] = ClipPel(m4 − delta, clpRng);

    const int tc2 = tc >> 1;

    if(bFilterSecondP)
    {
      const int delta1=Clip3(−tc2, tc2, ((((m1+m3+1)>>1)−m2+delta)>>1));
      piSrc[−iOffset * 2] = ClipPel(m2 + delta1, clpRng);
    }                                                    (c)
    if(bFilterSecondQ)
    {
      const int delta2=Clip3(−tc2, tc2, ((((m6+m4+1)>>1)−m5−delta)>>1));
      piSrc[iOffset] = ClipPel(m5 + delta2, clpRng);
    }
  }
}
```

3. Results

This section describes the experimental conditions and results of the experiments performed by applying simple weak filtering methods and the non-weak filtering method in HEVC and VVC.

3.1. Experimental Conditions

This experiment used HM 16.20 [14] and the VTM 6.0 [15]. The 14 test video sequences were selected in class B—class E for 8-bit depth video that could be used in common with HEVC and VVC. All of the experiments were encoded in AI mode configuration for the simple weak filtering method and the non-weak filtering method according to common test conditions (CTC) [16].

The experimental results are summarized in Tables 2 and 3 in terms of the BDBR in the luminance component. The anchor for the experiments was HM 16.20 and VTM 6.0. A negative BDBR (%) in the resulted corresponds to a performance gain. However, because the non-weak filtering method removes the weak filtering process, if the non-weak filtering method provided a gain, this indicated that the weak filtering method was counterproductive.

Table 2. Encoding result comparison on high efficiency video coding (HEVC) in Bjontegaard-Delta Bit-Rate (BDBR) (%).

Video Sequence.		**Y-BDBR (%) Anchor: HEVC Test Model (HM) 16.20**	
		Simple Weak Filter Method [5]	**Non-Weak Filter Method**
Class B	Cactus	−0.16	0.96
	BasketballDrive	−0.02	1.45
	BQTerrace	−0.12	0.20
	Overall	−0.10	0.87
Class C	BQMall	−0.08	1.00
	RacehorsesC	−0.12	0.71
	ParkScene	−0.06	0.23
	BasketballDrill	−0.24	0.57
	Overall	−0.12	0.63
Class D	BasketballPass	−0.10	1.08
	BlowingBubbles	−0.09	0.49
	Racehorses	−0.21	0.68
	BQSquare	−0.04	0.08
	Overall	−0.11	0.58
Class E	FourePeople	−0.09	1.95
	Johnny	−0.05	1.85
	KristenAndSara	−0.03	1.67
	Overall	−0.06	1.82
Average		−0.10	0.92

Table 3. Encoding result comparison on Versatile Video Coding (VVC).

Video Sequence		**Y-BDBR (%) Anchor: VVC Test Model (VTM) 6.0**	
		Simple Weak Filter Method [5]	**Non-Weak Filter Method**
Class B	Cactus	−0.17	−0.59
	BasketballDrive	−0.21	−1.23
	BQTerrace	−0.10	−0.61
	Overall	−0.16	−0.81
Class C	BQMall	−0.09	−0.37
	RacehorsesC	−0.11	−0.23
	ParkScene	−0.04	−0.24
	BasketballDrill	−0.17	−0.52
	Overall	−0.11	−0.34
Class D	BasketballPass	−0.14	−0.51
	BlowingBubbles	−0.05	−0.24
	Racehorses	−0.11	−0.27
	BQSquare	−0.03	−0.18
	Overall	−0.08	−0.30
Class E	FourePeople	−0.13	−0.27
	Johnny	−0.14	−0.19
	KristenAndSara	−0.12	−0.14
	Overall	−0.13	−0.20
Average		−0.12	−0.40

3.2. Weak Filter on HEVC

Table 2 indicates that the simple weak filtering method, which removes the second stage of weak filtering, offers an average gain of 0.10%. This could be said to be counterproductive in terms of BDBR, where the second stage of weak filtering quantitatively exhibits compression performance. However, as mentioned earlier, the need for such a tool has been recognized because it is effective for improving subjective visual quality.

The non-weak filtering method with removal of the entire weak filtering process showed an average loss of 0.92%. The weak filtering originally improved the compression effect in the deblocking filter part, but the loss was due to its removal. In particular, in Class E, the weak filter caused an average BDBR gain of 1.82% for HEVC. This means that weak filtering was a necessary tool that also benefits from the compression performance in HEVC.

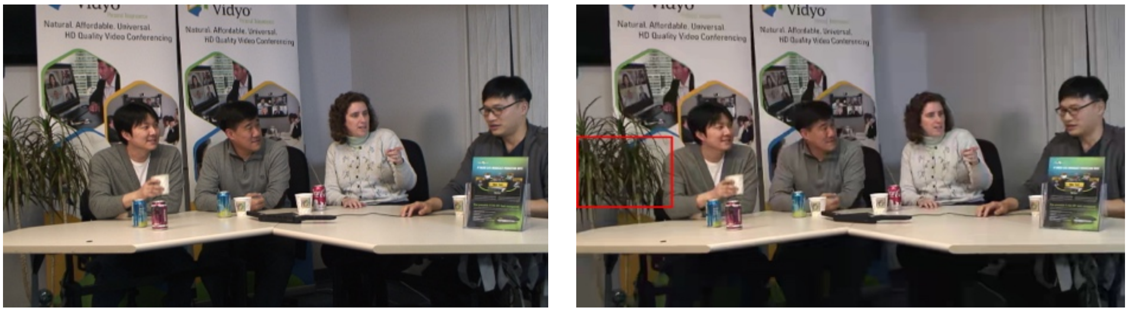
3.3. Weak Filter on VVC

From the experimental resulted in Table 3, the simple weak filtering method offered an average gain of 0.12%, while the non-weak filtering method offers an average gain of 0.40%. In VVC, not only the second step of weak filtering, but also the whole weak filtering process was ineffective as a deblocking filter.

A simple weak filtering method could identify a tendency for lower resolution and lower gain. It could be interpreted that the higher the resolution, the greater the boundary applied to the second weak filtering part were. Similarly, for the non-weak filtering method, the Y-BDBR gains tend to decrease as the resolution of the images decreased in Class B–D. However, even though Class E had higher resolution than Class C or Class D, the compression performance degradation due to weak filtering was low. This may be considered slightly misleading depending on the characteristics of the images; however, it still caused losses in terms of compression performance in a comprehensive manner.

3.4. Result Image Comparison

The *FourPeople* sequence shown Figure 2a had the greatest BDBR loss when the non-weak filtering method was applied in HM 16.20. For comparison under the most dramatic situation, each first frame was considered with QP = 37. Figure 3 shows that when the non-weak filtering method was applied to HEVC, there were certain blocking artifacts in several areas. The red box in Figure 2b marks the most noticeable location of the blocking artifacts and we could compare the result images of enlarged area in Figure 3. In Figure 3a, blocking artifacts were evident when using the non-weak filtering method on HM. But when using the traditional VVC or the non-weak filtering method on VVC, there were no block artifacts as shown Figure 3b,c.


Vidyo
Natural. Affordable. Universal.
HD Quality Video Conferencing

(**a**) (**b**)

Figure 2. First frame of The "*FourPeople*" sequence: (**a**) original image; (**b**) result image of the non-weak filtering method in HEVC test model (HM) 16.20 with QP = 37.

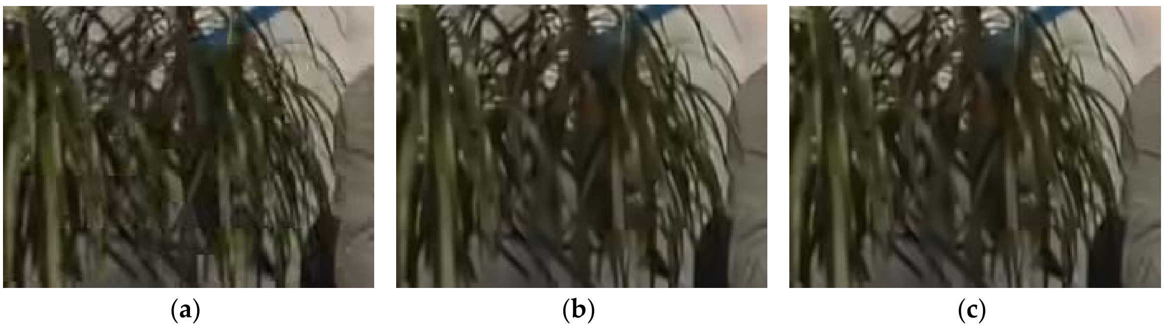


Figure 3. Comparison of the enlarged red box in Figure 2: (**a**) result for the decoded image using the non-weak filtering method on HM 16.20; (**b**) result for the decoded image with conventional versatile video coding (VVC) 6.0; (**c**) result for the decoded image using the non-weak filtering method on VVC 6.0.

4. Discussion and Conclusions

In this study, a method for eliminating the weak filtering process was proposed to measure the performance of weak filtering in VVC. The experimental results show that the non-weak filtering method provided a 0.40% Y-BDBR gain over VTM 6.0 in AI mode. No blocking artifacts were found in the comparison of the resulting images for the areas corresponding to the block boundaries. Blocking artifacts occur during the quantization process for the residual pixel value. If the predicted value was correct initially and the residual pixel value was reduced, the quantum error value can be reduced, and the blocking artifacts could also be reduced. Predictive technologies that vary and refine in VVC may reduce the need for weak filtering—or it may be necessary to incorporate a more sophisticated weak filtering method. Until improved weak filtering methods are developed, the VVC standard should consider excluding the current weak filtering method.

Author Contributions: Methodology, J.L.; software, J.L.; validation, J.L.; project administration, J.J.; test and writing, J.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported in part by the Research Fund of Signal Intelligence Research Center supervised by the Defense Acquisition Program Administration and in part by the Agency for Defense Development, South Korea.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *3*, 560–576. [CrossRef]
2. Sze, V.; Budagavi, M.; Sullivan, G.J. *High Efficiency Video Coding(HEVC): Algorithms and Architectures*; Springer: New York, NY, USA, 2014; pp. 171–208.
3. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1659–1668. [CrossRef]
4. Norkin, A.; Bjontegaard, G.; Fuldseth, A.; Narroschke, M.; Ikeda, M.; Anderson, K.; Zhou, M.; Van der Auwera, G. HEVC deblocking filter. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1746–1801. [CrossRef]
5. Bross, B.; Han, W.J.; Ohm, J.R.; Sullivan, G.J.; Wiegand, T. *WD4: Working Draft 4 of High-Efficiency Video Coding*; Document JCTVC-F803; JCTVC: Torino, Italy, 2011.
6. Bross, B.; Chen, J.; Liu, S. *Versatile Video Coding (Drafts 6)*; Document JVET-O2001; JVET: Gothenburg, Sweden, 2019.
7. Lee, J.; Jeong, J. Deblocking performance analysis of weak filter on versatile video coding. *Electron. Lett.* **2020**, *56*, 289–290. [CrossRef]
8. Bjontegaard, G. *Calculation of Average PSNR Differences between RD-Curves*; Document VCEG-M33; VCEG: Austin, TX, USA, 2001.

9. Yamakage, T.; Chono, K.; Chiu, Y.J.; Chong, I.S.; Narrashke, M. *JCT-VC AHG Report: In-Loop and Post-Processing Filtering (AHG 6)*; Document JCTVC-F006; JCTVC: Torino, Italy, 2011.
10. Norkin, A.; Guo, X.; Jeon, B.; Narroschke, M. *Description of Core Experiment 12: Deblocking Filtering*; Document JCTVC-F912; JCTVC: Torino, Italy, 2011.
11. Ikeda, M.; Tanaka, J.; Suzuki, T. *Parallel Deblocking Improvement*; Document JCTVC-F214; JCTVC: Torino, Italy, 2011.
12. Sadafale, M. *Improving Deblocking Filter Efficiency*; Document JCTVC-F256; JCTVC: Torino, Italy, 2011.
13. Hsu, C.W.; An, J.; Guo, X.; Lin, J.L.; Huang, Y.W.; Lei, S. *Deblocking Filter with Reduced Pixel Line Buffers for LCU-Based Processing*; Document JCTVC-F053; JCTVC: Torino, Italy, 2011.
14. HEVC Test Model (HM). Available online: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/ (accessed on 8 May 2020).
15. VVC Test Model (VTM). Available online: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/ (accessed on 8 May 2020).
16. Bossen, F.; Boyce, J.; Li, X.; Serejin, K.; Suhring, K. *JVET Common Test Conditions and Software Reference Configurations for SDR Video*; Document JVET-K1010; JVET: Ljubljana, Slovenia, 2018.