



# Frequency-Based Adaptive Interpolation Filter in Intra Prediction

Su-Yeon Lim , Min-Kyeong Choi and Yung-Lyul Lee \* 

Department of Computer Engineering, Sejong University, Seoul 05006, Republic of Korea

\* Correspondence: yllee@sejong.ac.kr

**Abstract:** This paper proposes a method to improve the fractional interpolation of reference samples in the Versatile Video Coding (VVC) intra prediction. The proposed method uses additional interpolation filters which use more integer-positioned reference samples for prediction according to the frequency information of the reference samples. In VVC, a 4-tap Discrete Cosine Transform-based interpolation filter (DCT-IF) and 4-tap Smoothing interpolation filter (SIF) are alternatively performed on the block size and block directional prediction mode for reference sample interpolation. This paper uses four alternative interpolation filters such as 8-tap/4-tap DCT-IFs, and 4-tap/8-tap SIFs and an interpolation filter selection method using a high-frequency ratio calculated from one-dimensional (1D) transform of the reference samples are proposed. The proposed frequency-based Adaptive Filter allows to achieve the overall Bjøntegaard Delta (BD) rate gains of  $-0.16\%$ ,  $-0.13\%$ , and  $-0.09\%$  for Y, Cb, and Cr components, respectively, compared with VVC.

**Keywords:** discrete cosine transform (DCT); interpolation filter; intra prediction; versatile video coding (VVC)/H.266



**Citation:** Lim, S.-Y.; Choi, M.-K.; Lee, Y.-L. Frequency-Based Adaptive Interpolation Filter in Intra Prediction. *Appl. Sci.* **2023**, *13*, 1475. <https://doi.org/10.3390/app13031475>

Academic Editor: Dariusz Frejlichowski

Received: 29 December 2022

Revised: 17 January 2023

Accepted: 20 January 2023

Published: 22 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As video resolution increases, so does the need for a more efficient and high-compressed video codec. ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) formed the Joint Video Exploration Team (JVET) in October 2015 to develop the next generation video coding standard, and VVC/H.266 standardization [1] was completed in July 2020. VVC is a video codec developed after High-Efficiency Video Coding (HEVC/H.265) and achieves a 39% reduction in bit rate compared with HEVC. Similar to Advanced Video Coding (AVC/H.264) [2,3] and HEVC [4–7], VVC is also a block-based video codec. It was developed with a codec in mind that can be used for various types of videos such as higher resolution, screen content, and 360° video [8–10].

In VVC, the picture is first divided into coding tree units (CTUs), and then the CTUs are recursively divided into coding units (CUs) [11]. During CU encoding, intra or inter prediction is performed. Intra prediction [12] is a method that generates sample values of a prediction block using the spatial similarity between the current block and adjacent blocks. The adjacent blocks above and to the left of the current block are used as the prediction reference blocks. Intra prediction plays an important role in increasing coding efficiency. The inter prediction [13,14] utilizes temporal redundancy between the current block and reference blocks to improve coding efficiency. After the prediction blocks are generated of intra or inter prediction, the prediction block is subtracted from the current block to be coded, which is called the residual block. The transform [15] and quantization processes [16] are applied to the residual block. The transform process decorrelates the residual samples using basis vectors in the frequency domain and the quantization process discards the high-frequency data depending on the QP values. Since all CTUs are encoded in the picture, an in-loop filtering process, which consists of a deblocking filter, a sample adaptive offset, and an adaptive loop filter, is applied to the reconstructed block to reconstruct the filtered picture which reduces the coding noise.

VVC uses rectangular blocks of various shapes and advanced new tools are used for intra prediction. Once the block shape has become diversified, the wide-angle intra prediction mode with 65 directional prediction modes is available. Therefore, there are 87 intra prediction modes including DC and planar modes. In addition, the multiple reference line (MRL) has been added which uses two non-adjacent reference lines. A new intra sub-partition (ISP) mode is included in VVC, where the prediction block is divided into smaller blocks of the same size to perform prediction and transform on these small blocks. It splits  $4 \times 8$  and  $8 \times 4$  blocks into two small blocks and the remaining blocks into four small blocks, where the blocks are partitioned horizontally or vertically. Reference sample filtering is also performed to generate more accurate prediction values from reference samples of adjacent blocks of the current block. When the intra prediction mode has integer-slope modes, reference sample smoothing is applied to the integer reference samples. For fractional-slope modes, an interpolation filter is applied to integer reference samples to create fractional positioned reference samples. In Position Dependent Prediction Combination (PDPC), unfiltered or filtered reference samples are used according to the intra prediction mode and position-dependent weighting is applied to correct the prediction value of the current CU. In Matrix-Based Intra Prediction (MIP) using a predefined data-driven matrix in VVC, the upper reference samples and left CU reference samples are used to generate the current block prediction samples by matrix multiplication and linear interpolation. Since the Cross Component Linear Model (CCLM) uses the relationship between luma samples and chroma samples, the linear model parameters for chroma samples are obtained using the down-sampled luma sample values and the coding efficiency is improved due to the accurate prediction values of the chroma samples. The Most Probable Mode (MPM) list increased from the three (3) intra prediction modes in HEVC to the six (6) intra prediction modes in VVC.

In this paper, first, we design and propose two 8-tap interpolation filters with better high-pass filter (HPF) characteristics and low-pass filter (LPF) characteristics which are designed to generate the fractional prediction samples considering the frequency characteristic of the integer reference samples. One 8-tap DCT-IF with HPF characteristics is designed with DCT and the other 8-tap SIF with LPF characteristics is designed with the convolution of a linear filter and [1, 6, 15, 20, 15, 6, 1] low-pass filter which comes from a three times convolution of [1, 2, 1]. Second, a filter selection method is proposed, that uses transformed reference samples to investigate energy in the frequency domain. Finally, the proposed frequency-based Adaptive Filter, which uses the 8-tap DCT-IF and SIF and frequency domain filter selection, is combined with a conventional VVC filter. The proposed method brings better coding efficiency than VVC intra coding.

This paper is structured as follows: Section 2 provides an overview of reference sample filtering for video coding standards and related studies are described. The proposed method based on both the two 8-tap interpolation filters and the filter selection in the frequency domain is explained in Section 3. Then, the experimental results of the proposed method are reported in Section 4. Finally, Section 5 concludes the paper.

## 2. Previous Works

### 2.1. An Overview of Reference Sample Filtering for Video Coding Standards

In intra prediction, a prediction block is generated using the reference samples from adjacent blocks. During the prediction process, filtering is applied to remove discontinuities in the reference samples and to generate accurate prediction sample values. There are two types of reference sample smoothing on integer samples in HEVC [17], whether to use a strong reference filter or a weak reference filter. The filter is determined by the block size and the continuity of the reference samples. HEVC interpolation filters use a 2-tap linear interpolation filter to generate a fractional position sample value [18].

In VVC, reference sample smoothing is applied in the integer-slope mode and interpolation filtering is applied in the fractional-slope mode [9,10,12]. Figure 1 shows the integer-slope modes and fractional-slope modes in the VVC intra prediction modes. Out

of 87 intra prediction modes, the vertical and horizontal modes do not use filters when generating the prediction samples. Planar and  $-14, -12, -10, -6, 2, 34, 66, 72, 76, 78,$  or 80 modes corresponding to multiple 45 degrees angles with an integer slope use  $[1, 2, 1]$  reference sample filter without interpolation filter. The other fractional-slope modes generate fractional reference samples of the current prediction samples by applying an interpolation filter to integer position reference samples. In this case, 4-tap interpolation filters are used for the luma samples to generate fractional positioned samples. There are two interpolation filters in VVC: 4-tap DCT-IF and 4-tap SIF [19–21]. The 4-tap DCT-IF coefficients in VVC are derived from the DCT-II in Equation (1) and IDCT-II (Inverse DCT-II) in Equation (2).

$$X(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} c_k x(n) \cos \frac{(n + 1/2)\pi k}{N} \tag{1}$$

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c_k X(k) \cos \frac{(n + 1/2)\pi k}{N} \tag{2}$$

$$c_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & k = 1, 2, \dots, N - 1 \end{cases}$$

where  $X(k)$  is the  $N$ -point of DCT-II and  $x(n)$  is its inverse transform.

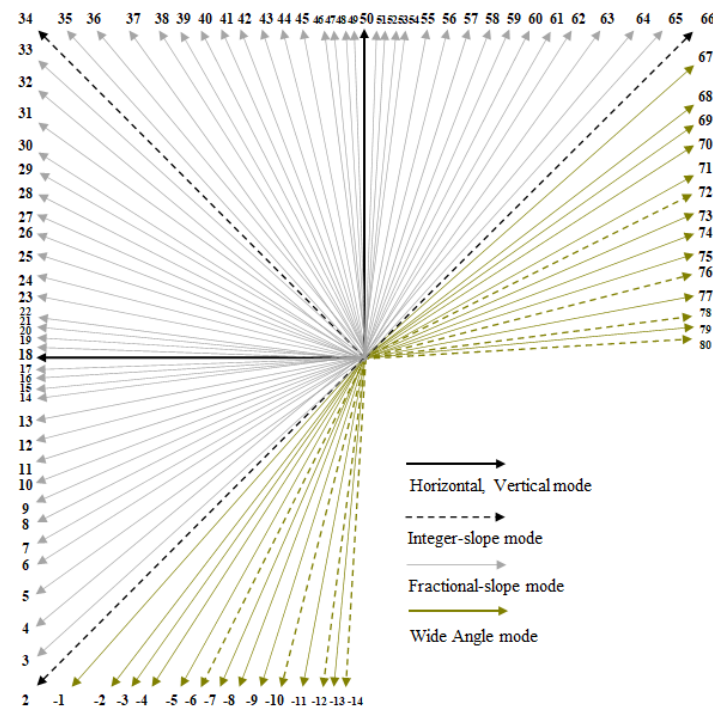


Figure 1. Angular intra prediction modes in VVC.

The DCT-IF coefficients [1] in Table 1 were obtained from Equations (1) and (2). The 4-tap SIF is derived from a convolution of  $[1, 2, 1]$  filter and a  $1/32$  linear interpolation on the reference samples as shown in Table 2 [1]. Tables 1 and 2 show the 4-tap DCT-IF coefficients and the 4-tap SIF coefficients, respectively. In Tables 1 and 2, the index  $I$  represents the position of the integer pixel. In VVC, the CU size and directional mode in Equations (3) and (4) are used to select a 4-tap DCT-IF or 4-tap SIF for fractional reference samples interpolation.

$$nTbS = ( \text{Log}_2 (W) + \text{Log}_2 (H) ) \gg 1 \tag{3}$$

$$minDistVerHor = Min \left( \begin{matrix} Abs(predModeIntra - 50), \\ Abs(predModeIntra - 18) \end{matrix} \right) \tag{4}$$

where  $W$  and  $H$  are the width and height of CU.  $nTbS$  is determined at Equation (3) according to  $W$  and  $H$ .  $predModeIntra$  represents the prediction mode of the current CU, and  $minDistVerHor$  is determined as the minimum difference value between the prediction mode of the current CU and the vertical mode 50 or between the prediction mode of the current CU and the horizontal mode 18.

**Table 1.** The 4-tap DCT Interpolation Filter (DCT-IF) coefficients in VVC.

| Index $i$                 | 0  | 1  | 2  | 3  |
|---------------------------|----|----|----|----|
| 0/32-pixel filter[ $i$ ]  | 0  | 64 | 0  | 0  |
| 1/32-pixel filter[ $i$ ]  | -1 | 63 | 2  | 0  |
| 2/32-pixel filter[ $i$ ]  | -2 | 62 | 4  | 0  |
| 3/32-pixel filter[ $i$ ]  | -2 | 60 | 7  | -1 |
| 4/32-pixel filter[ $i$ ]  | -2 | 58 | 10 | -2 |
| 5/32-pixel filter[ $i$ ]  | -3 | 57 | 12 | -2 |
| 6/32-pixel filter[ $i$ ]  | -4 | 56 | 14 | -2 |
| 7/32-pixel filter[ $i$ ]  | -4 | 55 | 15 | -2 |
| 8/32-pixel filter[ $i$ ]  | -4 | 54 | 16 | -2 |
| 9/32-pixel filter[ $i$ ]  | -5 | 53 | 18 | -2 |
| 10/32-pixel filter[ $i$ ] | -6 | 52 | 20 | -2 |
| 11/32-pixel filter[ $i$ ] | -6 | 49 | 24 | -3 |
| 12/32-pixel filter[ $i$ ] | -6 | 46 | 28 | -4 |
| 13/32-pixel filter[ $i$ ] | -5 | 44 | 29 | -4 |
| 14/32-pixel filter[ $i$ ] | -4 | 42 | 30 | -4 |
| 15/32-pixel filter[ $i$ ] | -4 | 39 | 33 | -4 |
| 16/32-pixel filter[ $i$ ] | -4 | 36 | 36 | -4 |

**Table 2.** The 4-tap Smoothing Interpolation Filter (SIF) coefficients in VVC.

| Index $i$                 | 0  | 1  | 2  | 3 |
|---------------------------|----|----|----|---|
| 0/32-pixel filter[ $i$ ]  | 16 | 32 | 16 | 0 |
| 1/32-pixel filter[ $i$ ]  | 16 | 32 | 16 | 0 |
| 2/32-pixel filter[ $i$ ]  | 15 | 31 | 17 | 1 |
| 3/32-pixel filter[ $i$ ]  | 15 | 31 | 17 | 1 |
| 4/32-pixel filter[ $i$ ]  | 14 | 30 | 18 | 2 |
| 5/32-pixel filter[ $i$ ]  | 14 | 30 | 18 | 2 |
| 6/32-pixel filter[ $i$ ]  | 13 | 29 | 19 | 3 |
| 7/32-pixel filter[ $i$ ]  | 13 | 29 | 19 | 3 |
| 8/32-pixel filter[ $i$ ]  | 12 | 28 | 20 | 4 |
| 9/32-pixel filter[ $i$ ]  | 12 | 28 | 20 | 4 |
| 10/32-pixel filter[ $i$ ] | 11 | 27 | 21 | 5 |
| 11/32-pixel filter[ $i$ ] | 11 | 27 | 21 | 5 |
| 12/32-pixel filter[ $i$ ] | 10 | 26 | 22 | 6 |
| 13/32-pixel filter[ $i$ ] | 10 | 26 | 22 | 6 |
| 14/32-pixel filter[ $i$ ] | 9  | 25 | 23 | 7 |
| 15/32-pixel filter[ $i$ ] | 9  | 25 | 23 | 7 |
| 16/32-pixel filter[ $i$ ] | 8  | 24 | 24 | 8 |

The filter is selected using  $nTbS$  and  $minDistVerHor$  of the CU calculated by Equations (3) and (4). If the  $minDistVerHor$  is greater than the  $intraHorVerDistThres[nTbS]$  in each  $nTbS$  in Table 3, a 4-tap SIF is used, otherwise, a 4-tap DCT-IF is used. If an MRL or ISP is used in the CU, only the 4-tap DCT-IF is applied for the CU reference samples.

**Table 3.** Specification of  $intraHorVerDistThres[nTbS]$  for various transform block size  $nTbS$  in VVC.

| $nTbS$                       | 2  | 3  | 4 | 5 | 6 |
|------------------------------|----|----|---|---|---|
| $intraHorVerDistThres[nTbS]$ | 24 | 14 | 2 | 0 | 0 |

The fractional sample position is derived from the angle of the intra prediction mode and the fractional position of the current predicted sample. The derivation of the prediction sample  $pred(x, y)$  on  $(x, y)$  position in the current CU using 4-tap intra fractional prediction is shown in Equation (5) as follows:

$$pred(x, y) = \left( \sum_{i=0}^3 f[p][i] \cdot r[i_0 + i] + 32 \right) \gg 6 \quad (5)$$

$$p = ((y + 1 + refldx) * intraPredAngle) \& 31 \quad (6)$$

where  $f[p][i]$  are the filter coefficients; where  $p, p = 0, 1, 2, \dots, 31$ , is the horizontal or vertical projection of the fractional part of the predicted sample; and  $i$  is the index of the integer reference sample in Tables 1 and 2.  $p$  is computed in Equation (6), where  $refIdx$  is the MRL index and  $intraPredAngle$  depends on the intra prediction mode of the current CU. When the intra prediction mode is less than 34,  $y$  is replaced by  $x$  in Equation (6).  $r[i + i_0]$ ,  $i = 0, 1, 2, 3$ , is the reference sample projected onto the CU boundary where  $i_0$  is the start integer reference sample index.

## 2.2. Related Studies

Various studies have been conducted to improve the performance of interpolation filters. There are several approaches to improving prediction performance by increasing the length of the filters or by suggesting a filter selection method. Matsuo et al. [22] proposed a 4-tap DCT-iFs for small prediction units (PUs) for intra prediction in HEVC. Kim et al. [23] used 12-point Discrete Sine Transform (DST) interpolation filters and 11-point DST interpolation filters to replace 8-point DCT-iFs and 7-point DCT-iFs for inter prediction in motion compensation in HEVC to improve inter coding efficiency. Zhao et al. [24] proposed 6-tap filters to replace 4-tap DCT-IF filters, where the coefficients of 6-tap filters were obtained from a polynomial regression model. Chang et al. [25] proposed 6-tap DCT-iFs and 6-tap Gaussian interpolation filters for blocks larger than  $32 \times 32$  blocks. In [22] and [23], the interpolation filters derived from DCT and DST highlight the high-frequency responses of integer samples in the prediction block. References [23–25] extend the interpolation filter to use more integer reference samples to apply long-tap DCT-IF and Gaussian IF compared with the existing interpolation filters in HEVC and VVC. Kim et al. [26] proposed a method of selecting interpolation filters, which consists of 4-tap interpolation filters and 3-tap intra smoothing filters according to the smoothness of the reference samples. In this method,  $[1, -2, 1]$  the HPF filter is used as the reference samples for the integer-slope modes to compute the difference between the filtered samples and the unfiltered reference samples, where the pixel domain difference value determines which interpolation filters are applied or if none are applied. In VVC, alternative interpolation filters for half-pixel position depending on the motion vector accuracy for inter prediction are proposed by Henkel et al. [27]. Kidani et al. [28] proposed a filter selection to characterize the reference samples based on the block-size and QP value for the 4-tap DCT-IF and SIF selection. Chiang et al. [29] proposed a method of selecting filters for horizontal and vertical directions in AV1. In [26–29], a filter selection method is proposed for selecting an interpolation filter that is better suited to the current block to improve the accuracy of the prediction block. Moreover, there are studies that apply filtering using a trained model based on Deep Convolutional Neural Network (CNN) rather than fixed filters. Pham et al. [30] proposed a CNN-based fractional interpolation filter for Luma and Chroma components for inter prediction in HEVC. Yan et al. [31] proposed an invertibility-driven interpolation filter using CNN in HEVC. The authors in [30,31] alternatively applied both DCT-IF and CNN-based iFs to generate fractional positioned

samples more accurately. They achieved better coding efficiency than using fixed filters in HEVC inter prediction with more parameters and complexity.

These studies show that the interpolation filter plays an important role in intra and inter coding to obtain better coding efficiency in video coding. Most of the methods applied long-tap filters or alternative filters based on filter selection methods in pixel domain to improve the video coding efficiency in intra and inter coding. However, in this study, we propose a new method that uses the newly derived 8-tap DCT-IF/SIF in this paper and the existing 4-tap DCT-IF/SIF which is used in the VVC standard based on high-frequency information after 1-D transform on reference samples. Finally, the proposed method improves coding efficiency in intra coding compared with the VVC standard.

### 3. Proposed Method

#### 3.1. Design of 8-Tap Interpolation Filter

A method is proposed that addresses more integer samples using 8-tap DCT-IF and 8-tap SIF to generate fractional reference samples based on the frequency information of the reference samples. The 8-tap DCT-IF coefficients are derived from DCT-II and IDCT-II (Inverse DCT-II) in Equations (1) and (2) of Section 2, where  $N$  is set to 8. The 8-tap DCT-IF is derived by substituting Equation (1) into Equation (2).

$p/32$ -pixel interpolation filter,  $p = 0, 1, 2, 3, \dots, 31$ , in the case of using  $1/32$  fractional samples, is derived by substituting  $n = 3 + p/32$  from 8-tap DCT-IF as a linear combination of discrete cosine coefficients and  $x(m)$ ,  $m = 0, 1, 2, \dots, 7$ . The 8-tap DCT-IF coefficients derived for  $(0/32, 1/32, 2/32, \dots, 16/32)$  fractional sample positions are shown in Table 4 by substituting  $n = 3 + (0/32, 1/32, 2/32, \dots, 16/32)$ , respectively, from 8-tap DCT-IF. The 8-tap DCT-IF coefficients for  $(17/32, 18/32, 19/32, \dots, 31/32)$  are obtained in the same way as above, where filter coefficients are scaled to an integer implementation.

**Table 4.** The 8-tap DCT Interpolation Filter (DCT-IF) coefficients for interpolation.

| Index $i$             | 0  | 1  | 2   | 3   | 4  | 5   | 6  | 7  |
|-----------------------|----|----|-----|-----|----|-----|----|----|
| 0/32-pixel filter[i]  | 0  | 0  | 0   | 128 | 0  | 0   | 0  | 0  |
| 1/32-pixel filter[i]  | 0  | 1  | -3  | 127 | 4  | -2  | 1  | 0  |
| 2/32-pixel filter[i]  | -1 | 3  | -7  | 127 | 8  | -4  | 2  | 0  |
| 3/32-pixel filter[i]  | -1 | 4  | -10 | 126 | 13 | -6  | 3  | -1 |
| 4/32-pixel filter[i]  | -1 | 5  | -12 | 124 | 17 | -7  | 3  | -1 |
| 5/32-pixel filter[i]  | -2 | 6  | -15 | 122 | 23 | -9  | 4  | -1 |
| 6/32-pixel filter[i]  | -2 | 7  | -17 | 120 | 28 | -11 | 5  | -2 |
| 7/32-pixel filter[i]  | -2 | 8  | -19 | 117 | 33 | -13 | 6  | -2 |
| 8/32-pixel filter[i]  | -3 | 9  | -21 | 114 | 38 | -14 | 7  | -2 |
| 9/32-pixel filter[i]  | -3 | 9  | -22 | 111 | 43 | -16 | 8  | -2 |
| 10/32-pixel filter[i] | -3 | 10 | -23 | 107 | 49 | -18 | 8  | -2 |
| 11/32-pixel filter[i] | -3 | 10 | -24 | 104 | 54 | -19 | 9  | -3 |
| 12/32-pixel filter[i] | -3 | 11 | -24 | 99  | 59 | -20 | 9  | -3 |
| 13/32-pixel filter[i] | -3 | 11 | -25 | 95  | 65 | -22 | 10 | -3 |
| 14/32-pixel filter[i] | -3 | 11 | -25 | 90  | 70 | -22 | 10 | -3 |
| 15/32-pixel filter[i] | -3 | 11 | -24 | 85  | 75 | -23 | 10 | -3 |
| 16/32-pixel filter[i] | -3 | 11 | -24 | 80  | 80 | -24 | 11 | -3 |

The 8-tap SIF coefficients are derived from the convolution of  $z[n]$  and  $1/32$  fractional linear filter, where  $z[n]$  in Figure 2 is derived from the convolution of  $h[n]$  and  $y[n]$  in Equations (7) and (8), in which  $h[n]$  is a 3-point  $[1, 2, 1]$  LPF. Equations (7) and (8) show the procedure for deriving  $y[n]$  and  $z[n]$ . Figure 2 shows  $h[n]$ ,  $y[n]$ , and  $z[n]$ , and the 8-tap SIF coefficients are obtained from linear interpolation of  $z[n]$  and  $1/32$  fractional linear filter.

$$y[n] = h[n] * h[n] = \sum_{k=-1}^1 h[k]h[n+k] \tag{7}$$

$$z[n] = h[n] * y[n] = \sum_{k=-1}^1 h[k]y[n+k] \tag{8}$$

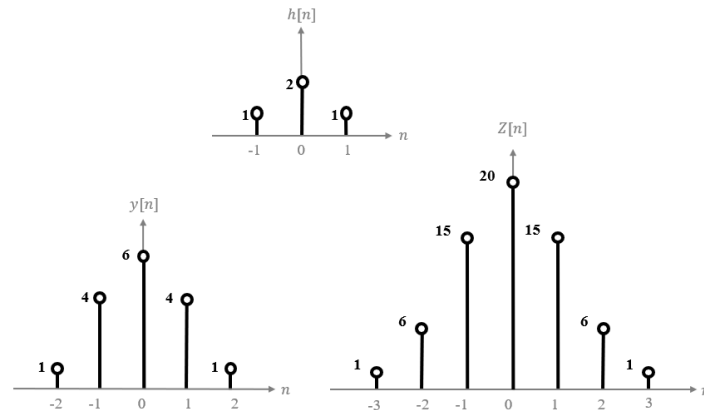


Figure 2. Derivation of  $z[n]$  from  $h[n]$ .

Equations (9) and (10) describe the procedure for calculating 8-tap SIF coefficients, where  $g[n] = z[n - 3]$ ,  $n = 0, 1, 2, \dots, 6$ .

$$\sum_{i=0}^6 g[i] \left( \frac{32-p}{32} r[i_0 + i] + \frac{p}{32} r[i_0 + 1 + i] \right) \ll 2 \tag{9}$$

Equation (10) shows an example of the SIF coefficient derivation at the position  $i_0 + 3 + 16/32$ -pixel, when  $p = 16$ . The bold integer numbers shown in Equation (10) are the filter coefficients, which are derived at  $16/32$ -pixel filter[i] in Table 5.

$$\begin{aligned} & \left( \begin{aligned} & \frac{16}{32} r[i_0] + \frac{112}{32} r[i_0 + 1] + \frac{336}{32} r[i_0 + 2] \\ & + \frac{560}{32} r[i_0 + 3] + \frac{560}{32} r[i_0 + 4] + \frac{336}{32} r[i_0 + 5] \\ & + \frac{112}{32} r[i_0 + 6] + \frac{16}{32} r[i_0 + 7] \end{aligned} \right) \ll 2 \\ & = \left( \begin{aligned} & 2 r[i_0] + 14 r[i_0 + 1] + 42 r[i_0 + 2] \\ & + 70 r[i_0 + 3] + 70 r[i_0 + 4] + 42 r[i_0 + 5] \\ & + 14 r[i_0 + 6] + 2 r[i_0 + 7] \end{aligned} \right) \end{aligned} \tag{10}$$

Table 5. The 8-tap Smoothing Interpolation Filter (SIF) coefficients for interpolation.

| Index $i$             | 0 | 1  | 2  | 3  | 4  | 5  | 6  | 7 |
|-----------------------|---|----|----|----|----|----|----|---|
| 0/32-pixel filter[i]  | 4 | 24 | 60 | 80 | 60 | 24 | 4  | 0 |
| 1/32-pixel filter[i]  | 4 | 24 | 60 | 80 | 60 | 24 | 4  | 0 |
| 2/32-pixel filter[i]  | 4 | 23 | 58 | 79 | 61 | 26 | 5  | 0 |
| 3/32-pixel filter[i]  | 4 | 23 | 58 | 79 | 61 | 26 | 5  | 0 |
| 4/32-pixel filter[i]  | 3 | 21 | 58 | 78 | 62 | 28 | 6  | 0 |
| 5/32-pixel filter[i]  | 3 | 21 | 58 | 78 | 62 | 28 | 6  | 0 |
| 6/32-pixel filter[i]  | 3 | 20 | 53 | 76 | 64 | 31 | 8  | 1 |
| 7/32-pixel filter[i]  | 3 | 20 | 53 | 76 | 64 | 31 | 8  | 1 |
| 8/32-pixel filter[i]  | 3 | 19 | 51 | 75 | 65 | 33 | 9  | 1 |
| 9/32-pixel filter[i]  | 3 | 19 | 51 | 75 | 65 | 33 | 9  | 1 |
| 10/32-pixel filter[i] | 3 | 18 | 49 | 74 | 66 | 35 | 10 | 1 |
| 11/32-pixel filter[i] | 3 | 18 | 49 | 74 | 66 | 35 | 10 | 1 |
| 12/32-pixel filter[i] | 3 | 17 | 47 | 73 | 67 | 37 | 11 | 1 |
| 13/32-pixel filter[i] | 3 | 17 | 47 | 73 | 67 | 37 | 11 | 1 |
| 14/32-pixel filter[i] | 2 | 15 | 44 | 71 | 69 | 40 | 13 | 2 |
| 15/32-pixel filter[i] | 2 | 15 | 44 | 71 | 69 | 40 | 13 | 2 |
| 16/32-pixel filter[i] | 2 | 14 | 42 | 70 | 70 | 42 | 14 | 2 |

The integer reference samples used to derive the 8-tap SIF coefficients are shown in Figure 3, where eight integer samples  $r[i_0]$  to  $r[i_0 + 7]$  to derive the filter coefficients for black  $i_0 + 3 + 16/32$ -pixel position are colored gray and  $r[i_0]$  is the start sample of eight

reference samples. Filter coefficients are scaled to an integer implementation. All  $p/32$  interpolation filter coefficients in Table 5 are derived by substituting  $p = 0, 1, 2, \dots, 15$  into Equation (9) in the same way as  $p = 16$  in Equation (10). Interpolation filters at  $p = 17, 18, \dots, 31$ , which are not shown in Table 5, can be easily obtained with even symmetry on  $i_0 + 3.5$ , reflecting the filter coefficients of  $p = 15, 14, \dots, 1$ , respectively.

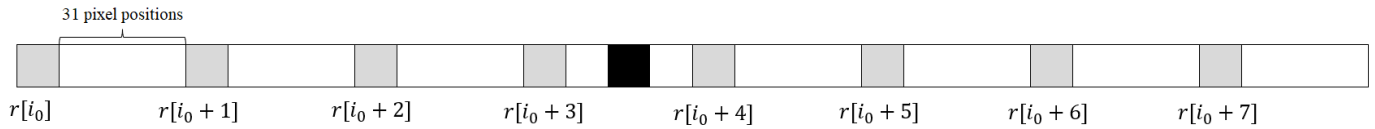


Figure 3. Integer reference samples used to derive 8-tap smoothing interpolation filter coefficients.

Tables 4 and 5 show 8-tap DCT-IF and 8-tap SIF coefficients, where index = 0 and index = 7 correspond to  $r[i_0]$  and  $r[i_0 + 7]$  in Figure 3, respectively. In Tables 4 and 5, index  $i$  is the position of the integer reference sample. Figure 4 shows the magnitude response at 16/32-pixel positions of 4-tap DCT-IF, 4-tap SIF, 8-tap DCT-IF, and 8-tap SIF, where the X-axis represents the normalized radian frequency and the Y-axis represents the magnitude response. The 8-tap DCT-IF has better HPF characteristics than the 4-tap DCT-IF and the 8-tap SIF has better LPF characteristics than the 4-tap SIF. Therefore, 8-tap SIF results in better interpolation than 4-tap SIF in low-frequency reference samples and 8-tap DCT-IF provides better interpolation than 4-tap DCT-IF in high-frequency reference samples.

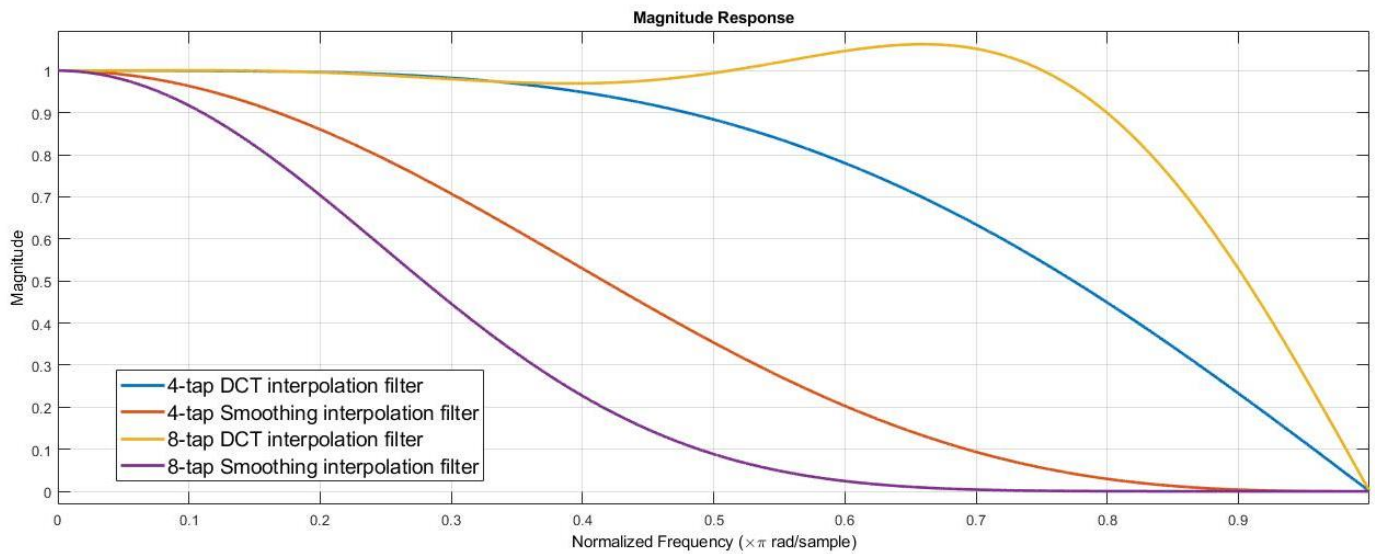


Figure 4. Magnitude response at 16/32-pixel positions of filters.

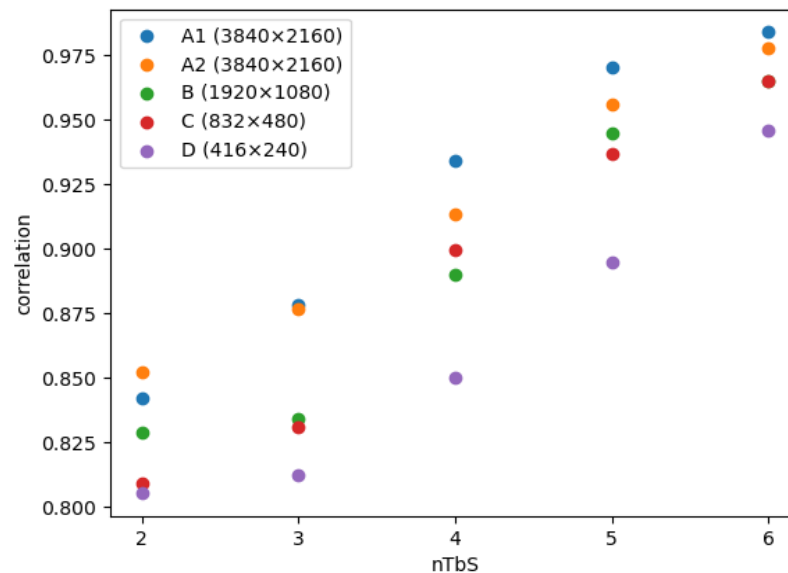
### 3.2. Frequency-Based Adaptive Interpolation Filter Selection

To determine the reference sample characteristics based on the CU size, the correlation at Equation (11) is computed from the above or left-hand reference samples of the current CU according to the intra prediction mode, where  $N$  is the width or height of the current CU. If the prediction mode of the current CU is greater than the diagonal mode 34 in Figure 1, then the reference samples located above the current CU are used in Equation (11). Otherwise, the reference samples located to the left of the current CU are used in Equation (11), where  $x_i$  is the reference samples,  $y_j$  is one sample right-shifted reference samples with respect to  $x_i$ , and  $\bar{x}$  and  $\bar{y}$  are the mean values of  $x_i$  and  $y_j$ , respectively.

$$correlation = \frac{\sum_{i=0}^{N-2} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-2} (x_i - \bar{x})^2} \sqrt{\sum_{i=0}^{N-2} (y_i - \bar{y})^2}} \tag{11}$$



Figure 5 shows the average correlation values of the reference samples for various video resolutions and each  $nTbS$  defined in Equation (3), which is determined according to the CU size at each picture resolution. The correlation increases as the CU size is larger and the video resolution is higher as shown in Figure 5, where video resolutions A1, A2, B, C, and D are shown in parentheses.



**Figure 5.** Correlation of reference samples according to  $nTbS$  and picture resolution A1, A2, B, C, and D.

The intra CU size partition in video coding depends on the prediction performance to enhance the coding in terms of bit rate and distortion. The prediction performance depends on the prediction errors between the prediction samples and the samples in the current CU. When the current block has many detailed areas including high frequencies, the CU size is partitioned into small blocks in the consideration of bit rate and distortion using boundary reference samples of small width and height. However, when the current block consists of a homogeneous area, the CU size is partitioned into large blocks in the consideration of bit rate and distortion using boundary reference samples of large width and height [32–34].

Figure 5 shows the correlation values of the reference samples according to the  $nTbS$  size and the video resolution shown as A1, A2, B, C, and D. The reference samples have a high correlation in the large  $nTbS$  and a low correlation in the small  $nTbS$ , respectively, which means that small  $nTbS$  have high-frequency characteristics consistent with a low correlation and large  $nTbS$  have low-frequency characteristics consistent with high correlation.

VVC uses two interpolation filters as explained in Section 2.1: 4-tap DCT-IF is used for all blocks when  $nTbS = 2$ , alternatively 4-tap DCT-IF or 4-tap SIF is used when  $nTbS = 3$  or 4 depending on  $minDistVerHor$  in Equation (4) and  $intraHorVerDistThres[nTbS]$  in Table 3, and 4-tap SIF applied to all blocks when  $nTbS \geq 5$ . In this paper, in addition to the proposed interpolation filters, we also propose a selection method of the proposed interpolation filters for generating accurate fractional boundary prediction samples using the frequency information of the integer reference samples.

Even if the CU reference samples have a low-frequency characteristic, DCT-IF is used for the CU reference samples with  $nTbS = 2$  in the VVC standard. However, from Figure 4, it is more effective to use SIF than DCT-IF when the CU reference samples have the low-frequency characteristics regardless of the  $nTbS$  size. Similarly, even if the CU reference samples have high-frequency characteristics, SIF is used for a CU with  $nTbS > 4$  in the VVC standard. However, it is more effective to use DCT-IF than SIF in Figure 4 when the CU reference samples have high-frequency characteristics regardless of the  $nTbS$  size. To solve this problem, a method was developed that selects two different filters which consist of

SIF and DCT-IF, according to the frequency characteristics of the reference samples. The reference samples are transformed using the scaled integer one-dimensional (1D) DCT-II kernel to detect the high-frequency energy of the reference samples. The scaled DCT-II coefficient  $X[k], k = 0, 1, 2, \dots, N - 1$ , is derived from Equation (12) as follows:

$$X[k] = \left( 2^{6+M/2} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} c_k x(n) \cos \frac{(n + 1/2)\pi k}{N} \right) \gg \text{shift} \tag{12}$$

where  $N$  is the number of reference samples necessary for  $X[k]$  and  $M$  is  $\log_2(N)$ , and shift is  $M + 1$ . After 1-D transform, high-frequency energy is observed in the transform domain. As the energy is concentrated in the low-frequency components, the reference samples are composed of homogeneous samples. However, as energy exists in the high-frequency components, the reference samples contain high-frequency samples, which indicates that the samples in the CU have high-frequency components. The transform size of the reference samples is set according to the intra prediction mode of the current block. If the intra prediction mode is greater than mode 34 (diagonal mode), the upper CU reference samples are transformed with the  $N = CU \text{ width}$  in Equation (12). Moreover, if the intra prediction mode is less than mode 34, the left reference samples are used with the  $N = CU \text{ height}$  in Equation (12).  $X[k]$  is used to measure the energy ratio of the high-frequency coefficients. If there exists energy in the high-frequency data, DCT-IF is used because the reference samples are composed of high-frequency data. In contrast, SIF is used for high-energy reference samples in low-frequency data.

The energy percentage of the high-frequency coefficients,  $high\_freq\_ratio$ , is calculated in Equation (13).

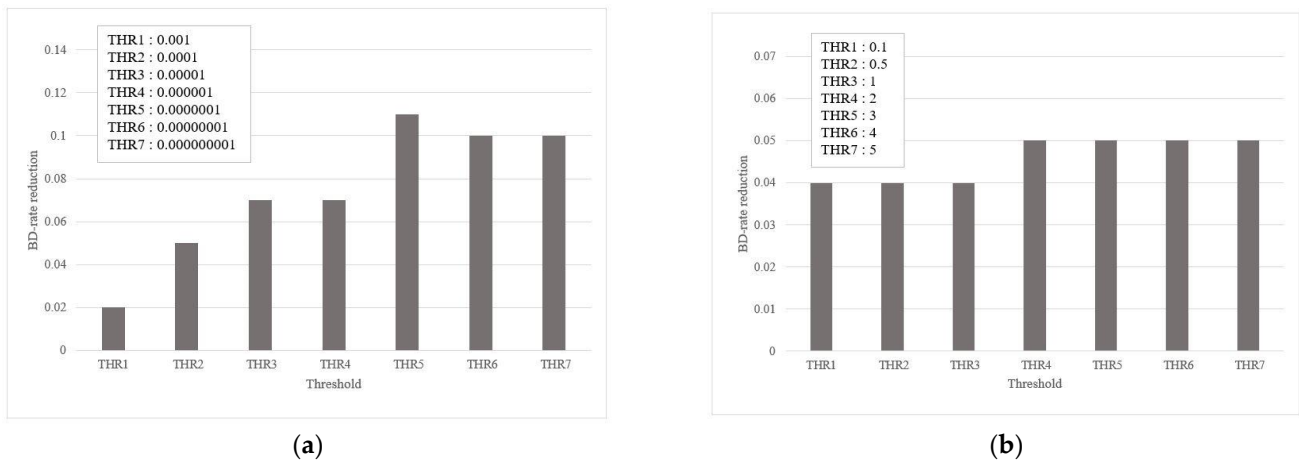
$$high\_freq\_ratio = \frac{\sum_{k=N*\frac{3}{4}}^{N-1} X[k] * X[k]}{\sum_{k=0}^{N-1} X[k] * X[k]} * 100 \tag{13}$$

Table 6 shows the proposed interpolation filter selection method by the threshold  $THR$  of  $high\_freq\_ratio$  and the interpolation filters applied to the selected method. The proposed method is only used when  $nTbS = 2$  and  $nTbS > 4$ . Otherwise, the existing VVC method is used when  $nTbS = 3, 4$ . The proposed method uses an 8-tap DCT-IF/SIF and a 4-tap DCT-IF/SIF according to  $nTbS$  and  $high\_freq\_ratio$ .

**Table 6.** Proposed interpolation filter selection method and interpolation filters applied to the selected method.

| $nTbS$                         | 2                      | 3          | 4          | 5                      | 6                      |
|--------------------------------|------------------------|------------|------------|------------------------|------------------------|
| <i>Filter Selection Method</i> | <i>high_freq_ratio</i> | VVC Method | VVC Method | <i>high_freq_ratio</i> | <i>high_freq_ratio</i> |
| $high\_freq\_ratio < THR$      | 4-tap SIF              | -          | -          | 8-tap SIF              | 8-tap SIF              |
| $high\_freq\_ratio \geq THR$   | 8-tap DCT-IF           | -          | -          | 4-tap DCT-IF           | 4-tap DCT-IF           |

The threshold ( $THR$ ) of  $high\_freq\_ratio$  in Table 6 was determined experimentally. To select the threshold, CIF sequences with  $352 \times 288$  resolution and FHD sequences with  $1920 \times 1080$  resolution were used. The CIF sequences used are Akiyo, Bridge-far, Highway, News, and Paris, and the FHD sequences used are IceRock, Market3, Netflix\_BarScene, Netflix\_Crosswalk, and Netflix\_FoodMarket. Figure 6 shows the Bjøntegaard Delta (BD) rate [35,36] reductions applying the proposed reference sample interpolations for each threshold,  $THR_1, THR_2, \dots, THR_7$ , according to  $nTbS$ . Figure 6a,b are the result of  $nTbS = 2$  and  $nTbS = 5$ , respectively.



**Figure 6.** BD rate reductions in the proposed reference sample interpolation for each threshold, THR1, THR2, . . . , THR7: (a) cUs with  $nTbS = 2$ ; (b) cUs with  $nTbS = 5$ .

In Figure 6a, since  $high\_freq\_ratio$  is less than the given threshold, a 4-tap SIF is used. Otherwise, the 8-tap DCT-IF is used. In Figure 6a, the most efficient BD rate reductions occurred with THR5 and THR6. Therefore, THR5 is selected as the  $THR$  of  $high\_freq\_ratio$  in this proposed method. In Figure 6b, when experimenting with  $high\_freq\_ratio$  at  $nTbS = 5$ , better coding efficiency was obtained with THR4 and a similar result was obtained when  $nTbS$  is 6. Therefore, when  $nTbS$  is 5 and 6, THR4 is selected as  $THR$  of  $high\_freq\_ratio$  in the proposed method. For CU with  $nTbS > 4$ , when  $high\_freq\_ratio < THR$ , the 8-tap SIF is used. Otherwise, the 4-tap DCT-IF is used. Correlation and filter have a close relationship. Small blocks have low correlation values with high-frequency characteristics. So, small blocks need to highlight high-frequency samples using strong HPF (8-tap DCT-IF). Since their average correlation value is relatively lower than large blocks, weak LPF (4-tap SIF) is better interpolation than strong LPF (8-tap SIF) when the block has low-frequency characteristics. Similarly, large blocks have high correlation values with low-frequency characteristics. So, large blocks need to highlight low-frequency samples using strong LPF (8-tap SIF). Since their average correlation value is relatively higher than small blocks, weak HPF (4-tap DCT-IF) has better interpolation than strong HPF (8-tap DCT-IF) when the block has high-frequency characteristics. For example, if the CU is  $4 \times 4$ , the  $nTbS$  value becomes 2. The 8-tap DCT-IF is used when  $high\_freq\_ratio \geq THR5$  in Figure 6a. Otherwise, 4-tap SIF is used for the CU. The proposed method depends on  $nTbS$  and  $high\_freq\_ratio$ . In the case where the CU has  $nTbS$  size of 2, if  $high\_freq\_ratio < THR$ , a 4-tap SIF with a weak LPF characteristic, as shown in Figure 4, is applied for the reference samples, because the correlation at  $nTbS = 2$  is relatively less than that at  $nTbS > 4$  as shown in Figure 5. Otherwise, if  $high\_freq\_ratio \geq THR$ , the 8-tap DCT-IF with strong HPF characteristic as shown in Figure 4 is applied for the reference samples.

In a similar way, in the case where a CU has  $nTbS$  size greater than 4, if  $high\_freq\_ratio < THR$ , an 8-tap SIF with a strong LPF characteristic, as shown in Figure 4, is applied to the reference samples because the correlation at  $nTbS > 4$  is relatively higher than with  $nTbS = 2$  as shown in Figure 5. Otherwise, if  $high\_freq\_ratio \geq THR$ , a 4-tap DCT-IF with weak HPF characteristic, as shown in Figure 4, is applied for the reference samples.

#### 4. Experimental Results

The proposed method was implemented in VTM-14.2 [37], the VVC reference software, and conducted in All Intra (AI) configurations under the JVET common test conditions (CTC) [38]. The sequences of classes A1, A2, B, C, and D were tested with Quantization Parameter (QP) values of 22, 27, 32, and 37, respectively. Table 7 shows the sequence name, picture size, picture rate, and bit depth of the CTC video sequences for each class.

**Table 7.** Information on video sequences for each class.

| Class | Sequence Name   | Picture Size | Picture Count | Picture Rate | Bit Depth |
|-------|-----------------|--------------|---------------|--------------|-----------|
| A1    | Tango2          | 3840 × 2160  | 294           | 60           | 10        |
|       | FoodMarket4     | 3840 × 2160  | 300           | 60           | 10        |
|       | Campfire        | 3840 × 2160  | 300           | 30           | 10        |
| A2    | CatRobot1       | 3840 × 2160  | 300           | 60           | 10        |
|       | DaylightRoad2   | 3840 × 2160  | 300           | 60           | 10        |
|       | ParkRunning3    | 3840 × 2160  | 300           | 50           | 10        |
| B     | MarketPlace     | 1920 × 1080  | 600           | 60           | 10        |
|       | RitualDance     | 1920 × 1080  | 600           | 60           | 10        |
|       | Cactus          | 1920 × 1080  | 500           | 50           | 8         |
|       | BasketballDrive | 1920 × 1080  | 500           | 50           | 8         |
|       | BQTerrace       | 1920 × 1080  | 600           | 60           | 8         |
| C     | RaceHorses      | 832 × 480    | 300           | 30           | 8         |
|       | BQMall          | 832 × 480    | 600           | 60           | 8         |
|       | PartyScene      | 832 × 480    | 500           | 50           | 8         |
|       | BasketballDrill | 832 × 480    | 500           | 50           | 8         |
| D     | RaceHorses      | 416 × 240    | 300           | 30           | 8         |
|       | BQSquare        | 416 × 240    | 600           | 60           | 8         |
|       | BlowingBubbles  | 416 × 240    | 500           | 50           | 8         |
|       | BasketballPass  | 416 × 240    | 500           | 50           | 8         |

Table 8 shows the interpolation filter selection method and the interpolation filters applied according to the selected method to test the efficiency of 8-tap/4-tap interpolation filter. Method A uses 8-tap DCT-IF for  $nTbS = 2$  and 4-tap SIF for  $nTbS > 4$  and Method B uses 8-tap SIF for  $nTbS > 4$  and 4-tap DCT-IF for  $nTbS = 2$ , while selecting DCT-IF or SIF in the same way as a VVC anchor. The difference between Method A and the VVC methods is that Method A uses 8-tap DCT-IF instead of 4-tap DCT-IF for  $nTbS = 2$  only. The difference between Method B and the VVC method is that Method B uses 8-tap SIF instead of 4-tap SIF only for  $nTbS > 4$ .

**Table 8.** Various interpolation filter selection methods and interpolation filters applied to the selected method.

| $nTbS$                  | 2                                 | 3                          | 4                          | 5                                 | 6                                 |
|-------------------------|-----------------------------------|----------------------------|----------------------------|-----------------------------------|-----------------------------------|
| Method A                | <b>8-tap DCT-IF</b>               | 4-tap DCT-IF,<br>4-tap SIF | 4-tap DCT-IF,<br>4-tap SIF | 4-tap SIF                         | 4-tap SIF                         |
| Method B                | 4-tap DCT-IF                      | 4-tap DCT-IF,<br>4-tap SIF | 4-tap DCT-IF,<br>4-tap SIF | <b>8-tap SIF</b>                  | <b>8-tap SIF</b>                  |
| Filter Selection Method | VVC Method                        | VVC Method                 | VVC Method                 | VVC Method                        | VVC Method                        |
| Method C                | <b>8-tap DCT-IF,</b><br>4-tap SIF | 4-tap DCT-IF,<br>4-tap SIF | 4-tap DCT-IF,<br>4-tap SIF | 4-tap SIF                         | 4-tap SIF                         |
| Filter Selection Method | <b>high_freq_ratio</b>            | VVC Method                 | VVC Method                 | VVC Method                        | VVC Method                        |
| Method D                | 4-tap DCT-IF                      | 4-tap DCT-IF,<br>4-tap SIF | 4-tap DCT-IF,<br>4-tap SIF | <b>8-tap SIF,</b><br>4-tap DCT-IF | <b>8-tap SIF,</b><br>4-tap DCT-IF |
| Filter Selection Method | VVC Method                        | VVC Method                 | VVC Method                 | <b>high_freq_ratio</b>            | <b>high_freq_ratio</b>            |

Method C uses 8-tap DCT-IF or 4-tap SIF depending on *high\_freq\_ratio* in Equation (13) for  $nTbS = 2$ , and 4-tap SIF for  $nTbS > 4$ . Method D uses 8-tap SIF or 4-tap DCT-IF depending on *high\_freq\_ratio* for  $nTbS > 4$ , and 4-tap DCT-IF for  $nTbS = 2$ . Tables 9 and 10 show simulation results of A, B, C, and D methods, in which the filter selection method and the interpolation filters used in the VVC anchor are employed for the CUs of  $nTbS = 3$  and  $nTbS = 4$ .

**Table 9.** Experimental results of Method A and Method B.

| Sequence | All Intra Main 10 |                |        |        |                |        |
|----------|-------------------|----------------|--------|--------|----------------|--------|
|          | Y                 | Method A<br>Cb | Cr     | Y      | Method B<br>Cb | Cr     |
| Class A1 | 0.01%             | −0.06%         | 0.06%  | −0.08% | −0.20%         | −0.15% |
| Class A2 | 0.02%             | −0.03%         | −0.02% | −0.01% | 0.00%          | −0.04% |
| Class B  | −0.01%            | 0.04%          | −0.01% | −0.01% | 0.05%          | −0.05% |
| Class C  | −0.33%            | −0.22%         | −0.27% | 0.00%  | −0.02%         | 0.07%  |
| Class D  | −0.28%            | −0.32%         | −0.15% | −0.01% | −0.07%         | 0.03%  |
| Overall  | −0.13%            | −0.12%         | −0.08% | −0.02% | −0.03%         | −0.02% |

**Table 10.** Experimental results of method C and method D based on frequency-based adaptive filter selection method.

| Sequence | All Intra Main 10 |                |        |        |                |        |
|----------|-------------------|----------------|--------|--------|----------------|--------|
|          | Y                 | Method C<br>Cb | Cr     | Y      | Method D<br>Cb | Cr     |
| Class A1 | 0.01%             | −0.18%         | 0.05%  | −0.09% | −0.10%         | −0.02% |
| Class A2 | 0.02%             | 0.08%          | 0.00%  | 0.00%  | 0.04%          | −0.04% |
| Class B  | 0.01%             | 0.09%          | 0.01%  | 0.01%  | 0.02%          | 0.05%  |
| Class C  | −0.40%            | −0.22%         | −0.36% | 0.01%  | 0.03%          | 0.02%  |
| Class D  | −0.30%            | −0.25%         | −0.23% | −0.01% | −0.06%         | 0.08%  |
| Overall  | −0.14%            | −0.09%         | −0.11% | −0.01% | −0.01%         | 0.03%  |

In the case of Method A, the overall increase in the BD rate by  $-0.13\%$ ,  $-0.12\%$  and  $-0.08\%$  is observed for the Y, Cb, and Cr components, respectively, where the sign ( $-$ ) means bit-savings. In the case of Method C, the overall increase in the BD rate by  $-0.14\%$ ,  $-0.09\%$ , and  $-0.11\%$  is observed for the Y, Cb, and Cr components, respectively. Particularly, increments of the Y component amounting to  $(-0.33\%, -0.28\%)$  in classes C with a resolution of  $832 \times 480$  and D with a resolution of  $416 \times 240$ , respectively, at low picture resolutions are achieved in the Method A, the component gain  $(-0.40\%, -0.30\%)$  in classes C and D is achieved in Method C. Methods A and C have in common that an 8-tap DCT-IF for  $nTbS = 2$  and a 4-tap SIF for  $nTbS > 4$  are applied to each CU regardless of the filter selection method. For Method B, the overall BD rate gains are  $-0.02\%$ ,  $-0.03\%$ , and  $-0.02\%$  for the Y, Cb, and Cr components, respectively. For Method D, the overall BD rate gains are  $-0.01\%$ ,  $-0.01\%$ , and  $0.03\%$  for the Y, Cb, and Cr components, respectively. Method B uses an 8-tap SIF for  $nTbS > 4$ , and a 4-tap DCT-IF for  $nTbS = 2$  and Method D uses an 8-tap SIF or 4-tap DCT-IF in line with the proposed *high\_freq\_ratio* for  $nTbS > 4$ , and 4-tap DCT-IF for  $nTbS = 2$ . Even though there is almost no overall increase in BD rate with Methods B and D, the increase in BD rate by  $(-0.08\%, -0.09\%)$  in Y components in class A1 with resolution  $3840 \times 2160$  is obtained in Methods B and D, respectively. The proposed frequency-based adaptive interpolation filtering using *high\_freq\_ratio* and  $nTbS$  and the existing VVC method in Table 6 were developed to take advantage of Methods C and D.

Table 11 shows the percentages of the CUs applying the 4-tap DCT-IF in the VVC anchor and the 8-tap DCT-IF based on the *high\_freq\_ratio* in the proposed method for all test sequences. For  $nTbS = 2$ , the 4-tap DCT-IF is selected 100% on  $4 \times 4$  CU,  $4 \times 8$  CU and  $8 \times 4$  CU in the VVC anchor, but the 8-tap DCT-IF is selected 97.16% on  $4 \times 4$  CU, 95.80% on  $4 \times 8$  CU, and 96.77% on  $8 \times 4$  CU in the proposed Adaptive Filter Method based on *high\_freq\_ratio*. The percentage of 4-tap SIF selection with  $nTbS = 2$  can be inferred from the DCT-IF selection percentages in Table 11.

**Table 11.** CUs percentage using DCT-IF in which H and W are the height and width of each CU.

| All Intra Main 10 |        |            |       |       |      |        |                             |        |       |        |       |  |       |       |      |    |
|-------------------|--------|------------|-------|-------|------|--------|-----------------------------|--------|-------|--------|-------|--|-------|-------|------|----|
| H \ W             | W      | VVC Anchor |       |       |      |        | VVC Anchor with MRL and ISP |        |       |        |       | Adaptive Filter Method Based on <i>high_freq_ratio</i> (Proposed Method) |       |       |      |    |
|                   |        | 4          | 8     | 16    | 32   | 64     | 4                           | 8      | 16    | 32     | 64    | 4  | 8     | 16    | 32   | 64 |
| 4                 | 100.00 | 100.00     | 85.67 | 94.70 | —    | 100.00 | 100.00                      | 82.25  | 89.16 | —      | 97.16 | 95.80  | 91.11 | 97.27 | —    |    |
| 8                 | 100.00 | 87.89      | 81.99 | 25.68 | —    | 100.00 | 91.35                       | 81.42  | 44.36 | —      | 96.77 | 91.36  | 86.69 | 53.43 | —    |    |
| 16                | 87.38  | 83.20      | 8.32  | 17.24 | —    | 86.26  | 82.59                       | 19.35  | 27.85 | 100.00 | 92.62 | 86.76  | 19.83 | 33.13 | 0.04 |    |
| 32                | 94.01  | 19.98      | 14.73 | 0.00  | —    | 90.45  | 36.35                       | 23.17  | 10.59 | —      | 97.06 | 42.55  | 27.34 | 0.07  | —    |    |
| 64                | —      | —          | —     | —     | 0.00 | —      | —                           | 100.00 | —     | 5.56   | —     | —  | 0.07  | —     | 0.07 |    |

The increases in percent selections of 4-tap SIF and 8-tap DCT-IF result in BD rate gains in Table 12. The use of 4-tap SIF with low LPF characteristics and 8-tap DCT-IF with strong HPF characteristics in line with *high\_freq\_ratio* with small CU helps to increase the BD rate in the proposed method. Moreover, the use of 8-tap SIF with strong LPF characteristics and 4-tap DCT-IF with weak HPF characteristics according to *high\_freq\_ratio* at a large CU helps to slightly increase the BD rate in the proposed method.

**Table 12.** Experimental results of the proposed filtering method.

| Adaptive Filter Method Based on <i>high_freq_ratio</i> (Proposed Method) |                 |        |        |        |      |      |
|--|-----------------|--------|--------|--------|------|------|
| Class  | Sequence Name   | Y      | Cb     | Cr     | EncT | DecT |
| A1   | Tango2          | −0.11% | −0.36% | −0.09% | 102% | 103% |
|  | FoodMarket4     | −0.11% | −0.06% | −0.22% | 102% | 106% |
|  | Campfire        | 0.02%  | 0.01%  | 0.17%  | 102% | 103% |
| A2   | CatRobot1       | −0.01% | 0.03%  | 0.07%  | 101% | 106% |
|  | DaylightRoad2   | 0.03%  | 0.34%  | 0.13%  | 103% | 105% |
|  | ParkRunning3    | −0.01% | −0.01% | 0.01%  | 101% | 103% |
| B  | MarketPlace     | −0.01% | −0.02% | −0.13% | 102% | 108% |
|  | RitualDance     | −0.01% | 0.08%  | 0.02%  | 103% | 104% |
|  | Cactus          | 0.00%  | −0.02% | 0.27%  | 102% | 105% |
|  | BasketballDrive | 0.03%  | 0.04%  | −0.05% | 102% | 108% |
|  | BQTerrace       | 0.06%  | 0.00%  | 0.10%  | 102% | 108% |
| C  | BasketballDrill | −1.20% | −0.78% | −0.68% | 101% | 110% |
|  | BQMall          | −0.18% | −0.26% | −0.12% | 101% | 106% |
|  | PartyScene      | −0.16% | −0.21% | −0.16% | 102% | 109% |
|  | RaceHorses      | −0.08% | −0.04% | −0.20% | 103% | 103% |
| D  | BasketballPass  | −0.29% | −0.25% | −0.42% | 102% | 104% |
|  | BQSquare        | −0.55% | −0.62% | −0.21% | 101% | 105% |
|  | BlowingBubbles  | −0.19% | −0.02% | −0.14% | 101% | 104% |
|  | RaceHorses      | −0.23% | −0.26% | 0.03%  | 101% | 103% |
| Class A1   |                 | −0.07% | −0.14% | −0.05% | 102% | 104% |
| Class A2   |                 | 0.00%  | 0.12%  | 0.07%  | 102% | 105% |
| Class B  |                 | 0.02%  | 0.01%  | 0.04%  | 102% | 107% |
| Class C  |                 | −0.41% | −0.32% | −0.29% | 101% | 107% |
| Class D  |                 | −0.31% | −0.29% | −0.18% | 101% | 104% |
| Overall  |                 | −0.16% | −0.13% | −0.09% | 102% | 105% |

For  $nTbS > 4$ , 4-tap DCT-IF is only applied to CUs using the MRL or ISP tool in the VVC anchor. However, 8-tap DCT-IF and 4-tap SIF based on *high\_freq\_ratio* are applied to

the CUs using the MRL or ISP in the proposed method so that the 8-tap DCT-IF is selected 0.07% in  $32 \times 32$  CU, 0.04% in  $16 \times 64$  CU, 0.07% in  $64 \times 16$  CU, and 0.07% in  $64 \times 64$  CU, compared with 10.59% in  $32 \times 32$  CU, 100% in  $16 \times 64$  CU, 100% in  $64 \times 16$  CU, and 5.56% in  $64 \times 64$  CU in the VVC anchor.

Table 12 shows the results of the proposed Adaptive Filter Method based on *high\_freq\_ratio*, in which EncT and DecT represent the total encoding and decoding time ratios compared with the VVC anchor for various test sequences of classes A1 to D in the AI Main 10 configuration. The proposed method allows to achieve the overall BD rate increase of  $-0.16\%$ ,  $-0.13\%$ , and  $-0.09\%$  for the Y, Cb, and Cr components, respectively, with an average increase in computational complexity by 2% and 5% in the encoder and decoder, respectively, compared with the VVC anchor. Particularly, the highest BD rate increases amounting to  $-0.41\%$ ,  $-0.32\%$ , and  $-0.29\%$  for the Y, Cb, and Cr components are obtained, respectively, in class C compared with the VVC anchor.

With a slight increase in computational complexity, the proposed method allows for achieving a reduction in the BD rate compared with the VVC anchor. The sequence that shows the highest BD rate reduction is the BasketballDrill sequence in class C, in which the proposed method yields a gain of the Y component of  $-1.20\%$ .

The proposed method did not obtain the BD rate gain in some sequences because we used CIF sequences and FHD sequences for determining only the threshold of *high\_freq\_ratio*. However, if the thresholds for correlation and *high\_freq\_ratio* are computed together for more various video sequences, the BD rate gain can be achieved in all test sequences with the computational complexity increases.

The proposed method shows slightly better subjective image quality than the VVC anchor when the BD rate gain is achieved, and the proposed method shows quite similar subjective quality to the VVC anchor even when the BD rate losses are 0.01% to 0.06%.

## 5. Conclusions

This paper proposes the Adaptive Filter Method to generate fractional reference samples for directional VVC intra prediction. With the *high\_freq\_ratio* derived from the 1-D scaled DCT, the 8-tap DCT-IFs and 8-tap SIFs in addition to the 4-tap DCT-IFs and the 4-tap SIFs are proposed to increase the precision of fractional reference samples. Depending on *high\_freq\_ratio* with respect to the block size, interpolation filters are applied to the reference samples. We conclude that in cases where the correlation between samples is high, 8-tap interpolation filters with strong HPF or strong LPF characteristics marginally affect the BD rate gain, but in cases where the correlation between samples is low, 8-tap interpolation filters with strong HPF or strong LPF characteristics affect the BD rate enhancement. Figure 5 shows the correlation graph for video resolutions and block size. Since high resolution videos show higher correlation value than low resolution videos, the sequences with low resolution achieve more coding gains than those with high resolution. For the proposed Adaptive Filter Method based on *high\_freq\_ratio*, overall BD rate gains of  $-0.16\%$ ,  $-0.13\%$ , and  $-0.09\%$  are observed for the Y, Cb, and Cr components, respectively, compared with the VVC anchor. We believe that the method of searching for high-frequency terms in the frequency domain helps video coding modules requiring strong/weak HPF and strong/weak LPF for next-generation video coding standards.

**Author Contributions:** Y.-L.L. and S.-Y.L. conceived and designed the experiments; Y.-L.L., S.-Y.L. and M.-K.C. derived mathematical interpolation equations; S.-Y.L. implemented software and performed the experiments; Y.-L.L. supervised the algorithm; and Y.-L.L. and S.-Y.L. wrote the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was in part supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (NRF-2018R1D1A1B07045156).

**Acknowledgments:** This research was in part supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2018R1D1A1B07045156).

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

## References

1. Bross, B.; Chen, J.; Liu, S.; Wang, Y.-K. Versatile Video Coding (Draft 10), document JVET-T2001, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29. In Proceedings of the 20th JVET Meeting, Online, 7–16 October 2020.
2. H.264 and ISO/IEC 14496-10; Advanced Video Coding (AVC), Standard ITU-T Recommendation. ITU-T: Geneva, Switzerland, 2003. Available online: <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=6312> (accessed on 20 January 2023).
3. 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**, *13*, 560–576. [[CrossRef](#)]
4. High Efficient Video Coding (HEVC), Standard ITU-T Recommendation, H.265 and ISO/IEC 23008-2, April 2013. Available online: <https://www.itu.int/rec/T-REC-H.265> (accessed on 28 April 2022).
5. Sze, V.; Budagavi, M.; Sullivan, G.J. *High Efficiency Video Coding: Algorithms and Architectures*; Springer: Berlin/Heidelberg, Germany, 2014.
6. Wien, M. *High Efficiency Video Coding: Coding Tools and Specification*; Springer: Berlin/Heidelberg, Germany, 2015.
7. Sullivan, G.J.; Ohm, J.; Han, W.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [[CrossRef](#)]
8. Bossen, F.; Li, X.; Suehring, K. AHG Report: Test Model Software Development (AHG3), document JVET-S0003, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. In Proceedings of the 19th JVET Meeting, Online, 22 June–1 July 2020.
9. Browne, A.; Chen, J.; Ye, Y.; Kim, S.H. Algorithm description for Versatile Video Coding and Test Model 14 (VTM 14), document JVET-W2002, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29. In Proceedings of the 23rd JVET Meeting, Online, 7–16 July 2021.
10. Bross, B.; Wang, Y.-K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.-R. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [[CrossRef](#)]
11. Huang, Y.-W.; An, J.; Haung, H.; Li, X.; Hsiang, S.-T.; Zhang, K.; Gao, H.; Ma, J.; Chubach, O. Block Partitioning Structure in the VVC Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3818–3833. [[CrossRef](#)]
12. Pfaff, J.; Filippov, A.; Liu, S.; Zhao, X.; Chen, J.; De-Luxán-Hernández, S.; Wiegand, T.; Rufitskiy, V.; Ramasubramonian, A.K.; Van der Auwera, G. Intra Prediction and Mode Coding in VVC. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3834–3847. [[CrossRef](#)]
13. Chien, W.-J.; Zhang, L.; Winken, M.; Li, X.; Liao, R.-L.; Gao, H.; Hsu, C.-W.; Liu, H.; Chen, C.-C. Motion Vector Coding and Block Merging in the Versatile Video Coding Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3848–3861. [[CrossRef](#)]
14. Yang, H.; Chen, H.; Chen, J.; Esenlik, S.; Sethuraman, S.; Xiu, X.; Alshina, E.; Luo, J. Subblock-Based Motion Derivation and Inter Prediction Refinement in the Versatile Video Coding Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3862–3877. [[CrossRef](#)]
15. Zhao, X.; Kim, S.-H.; Zhao, Y.; Egilmez, H.E.; Koo, M.; Liu, S.; Lainema, J.; Karczewicz, M. Transform Coding in the VVC Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3878–3890. [[CrossRef](#)]
16. Schwarz, H.; Coban, M.; Karczewicz, M.; Chuang, T.-D.; Bossen, F.; Alshin, A.; Lainema, J.; Helmrich, C.R.; Wiegand, T. Quantization and Entropy Coding in the Versatile Video Coding (VVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3891–3906. [[CrossRef](#)]
17. Ugur, K.; Alshin, A.; Alshina, E.; Bossen, F.; Han, W.-J.; Park, J.-H.; Lainema, J. Interpolation filter design in HEVC and its coding efficiency-complexity analysis. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013.
18. Lainema, J.; Bossen, F.; Han, W.-J.; Min, J.; Ugur, K. Intra coding of the HEVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1792–1801. [[CrossRef](#)]
19. Filippov, A.; Rufitskiy, V. Non-CE3: Cleanup of Interpolation Filtering for Intra Prediction, document JVET-P0599, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. In Proceedings of the 16th JVET Meeting, Geneva, Switzerland, 1–11 October 2019.
20. Filippov, A.; Rufitskiy, V.; Chen, J.; Van der Auwera, G.; Ramasubramonian, A.K.; Seregin, V.; Hsieh, T.; Karczewicz, M. CE3: A Combination of Tests 3.1.2 and 3.1.4 for Intra Reference Sample Interpolation Filter, document JVET-L0628, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. In Proceedings of the 12th JVET Meeting, Macao, China, 3–12 October 2018.
21. Filippov, A.; Rufitskiy, V.; Chen, J.; Alshina, E. Intra prediction in the emerging VVC video coding standard. In Proceedings of the 2020 Data Compression Conference (DCC), Snowbird, UT, USA, 24–27 March 2020.
22. Matsuo, S.; Takamura, S.; Jozawa, H. Improved intra angular prediction by DCT-based interpolation filter. In Proceedings of the 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012.
23. Kim, M.; Lee, Y.-L. Discrete Sine Transform-Based Interpolation Filter for Video Compression. *Symmetry* **2017**, *9*, 257. [[CrossRef](#)]
24. Zhao, X.; Seregin, V.; Karczewicz, M. Six tap intra interpolation filter, document JVET-D0119, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. In Proceedings of the 4th JVET Meeting, Chengdu, China, 15–21 October 2016.



25. Chang, Y.-J.; Chen, C.-C.; Chen, J.; Dong, J.; Egilmez, H.E.; Hu, N.; Haug, H.; Karczewicz, M.; Li, J.; Ray, B.; et al. Compression efficiency methods beyond VVC, document JVET-U0100, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29. In Proceedings of the 21st JVET Meeting, Online, 6–15 January 2021.
26. Kim, J.; Kim, Y.-H. Adaptive Boundary Filtering Strategies in VVC Intra-Prediction for Depth Video Coding. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Gangwon, Korea, 1–3 November 2021.
27. Henkel, A.; Zupancic, I.; Bross, B.; Winken, M.; Schwarz, H.; Marpe, D.; Wiegand, T. Alternative Half-Sample Interpolation Filters for Versatile Video Coding. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020.
28. Kidani, Y.; Kawamura, K.; Unno, K.; Naito, S. Blocksize-QP Dependent Intra Interpolation Filters. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019.
29. Chiang, C.-H.; Han, J.; Vitvitskiy, S.; Mukherjee, D.; Xu, Y. Adaptive interpolation filter scheme in AV1. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.
30. Pham, C.D.-K.; Zhou, J. Deep Learning-Based Luma and Chroma Fractional Interpolation in Video Coding. *IEEE Access* **2019**, *7*, 112535–112543. [[CrossRef](#)]
31. Yan, N.; Liu, D.; Li, H.; Li, B.; Li, L.; Wu, F. Invertibility-Driven Interpolation Filter for Video Coding. *IEEE Trans. Image Process.* **2019**, *28*, 4912–4925. [[CrossRef](#)] [[PubMed](#)]
32. Fan, Y.; Chen, J.; Sun, H.; Katto, J.; Jing, M. A Fast QTMT Partition Decision Strategy for VVC Intra Prediction. *IEEE Access* **2020**, *8*, 107900–107911. [[CrossRef](#)]
33. Li, W.; Fan, C.; Ren, P. Fast Intra-Picture Partitioning for Versatile Video Coding. In Proceedings of the 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), Nanjing, China, 23–25 October 2020.
34. Saldanha, M.; Sanchez, G.; Marcon, C.; Agostini, L. Analysis of VVC Intra Prediction Block Partitioning Structure. In Proceedings of the 2021 International Conference on Visual Communications and Image Processing (VCIP), Munich, Germany, 5–8 December 2021.
35. Bjøntegaard, G. Calculation of Average PSNR Differences Between RD-Curves, document VCEG-M33, ITU-T SG 16 Q 6 Video Coding Experts Group (VCEG). In Proceedings of the 13th VCEG Meeting, Austin, TX, USA, 2–4 April 2001.
36. Bjøntegaard, G. Improvements of the BD-PSNR Model, document VCEG-AI11, ITU-T SG 16 Q 6 Video Coding Experts Group (VCEG). In Proceedings of the 35th VCEG Meeting, Berlin, Germany, 16–18 July 2008.
37. Versatile Video Coding Test Model (VTM-14.2) Reference Software. Available online: [https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_VTM/tags/VTM-14.2](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-14.2) (accessed on 9 November 2021).
38. Bossen, F.; Boyce, J.; Suehring, K.; Li, X.; Seregin, V. JVET common test conditions and software reference configurations for SDR video, document JVET-T2010, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29. In Proceedings of the 20th JVET Meeting, Online, 7–16 October 2020.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.