symmetry

MDPI

Article

Discrete Sine Transform-Based Interpolation Filter for Video Compression

MyungJun Kim and Yung-Lyul Lee *

Department of Computer Engineering, Sejong University, Seoul 05006, Korea; mjkim@sju.ac.kr

* Correspondence: yllee@sejong.ac.kr

Received: 14 October 2017; Accepted: 30 October 2017; Published: 2 November 2017

Abstract: Fractional pixel motion compensation in high-efficiency video coding (HEVC) uses an 8-point filter and a 7-point filter, which are based on the discrete cosine transform (DCT), for the 1/2-pixel and 1/4-pixel interpolations, respectively. In this paper, discrete sine transform (DST)-based interpolation filters (DST-IFs) are proposed for fractional pixel motion compensation in terms of coding efficiency improvement. Firstly, a performance of the DST-based interpolation filters (DST-IFs) using 8-point and 7-point filters for the 1/2-pixel and 1/4-pixel interpolations is compared with that of the DCT-based IFs (DCT-IFs) using 8-point and 7-point filters for the 1/2-pixel and 1/4-pixel interpolations, respectively, for fractional pixel motion compensation. Finally, the DST-IFs using 12-point and 11-point filters for the 1/2-pixel and 1/4-pixel interpolations, respectively, are proposed only for bi-directional motion compensation in terms of the coding efficiency. The 8-point and 7-point DST-IF methods showed average Bjøntegaard Delta (BD)-rate reductions of 0.7% and 0.3% in the random access (RA) and low delay B (LDB) configurations, respectively, in HEVC. The 12-point and 11-point DST-IF methods showed average BD-rate reductions of 1.4% and 1.2% in the RA and LDB configurations for the Luma component, respectively, in HEVC.

Keywords: high efficiency video coding (HEVC); interpolation filter; sinc; DCT (discrete cosine transform); DST (discrete sine transform)

1. Introduction

The International Telecommunication Union-Telecommunication (ITU-T) Standardization Sector-Video Coding Expert Group (VCEG) and the Moving Picture Expert Group (ISO/IEC MPEG) organized the Joint Collaborative Team on Video Coding (JCT-VC) [1], and they jointly developed the next-generation video-coding standard HEVC/H.265. In high-efficiency video coding (HEVC) [2], motion-compensated prediction (MCP) is a significant video-coding function. MCP reduces the amount of information which should be transmitted to a decoder by using temporal redundancy in video signals [3–6]. In the MCP, each prediction unit (PU, block) in the encoder finds the best matching block that has the least SAD (sum of absolute difference) from the reference pictures in terms of the Lagrangian cost [7]. Using the best matching block, the motion vector that represents the movement from the current block to the best matching block is transmitted to the decoder with the residual signals that are the difference signals between the current block and the best matching block. Since the moving objects between two pictures are continuous, it is difficult to identify the actual motion vector in block-based motion estimation. In other words, the true displacements of moving objects between pictures are continuous and do not follow the sampling grid of the digitized video sequence. Hence, by utilizing fractional accuracy for motion vectors instead of integer accuracy, the residual error is decreased and coding efficiency of video compression is increased [4]. Therefore, the use of fractional pixels that have been derived from an interpolation filter for motion-vector searches can improve the precision of the MCP. The fractional interpolation filters in HEVC were discreetly considered with several factors such as coding efficiency, implementation complexity, and visual quality [8].

The sinc function is an ideal interpolation filter in terms of signal processing [9,10]. However, the sinc-interpolation filter is difficult to implement in HEVC because the sinc-interpolation filter needs to reference the neighbor pixels from $-\infty$ to ∞. Therefore, the finite filter lengths of interpolation filters are determined and motion vectors are supported with 1/4-pixel accuracy in HEVC. During the development of HEVC, there were several proposed interpolation filter techniques, such as switched interpolation filters with offset (SIFOs) [11], maximum order of interpolation with minimal support (MOMS) [12], one-dimensional directional interpolation filters (DIFs) [13], and DCT-based interpolation filters (DCT-IFs) [14]. As a result, the DCT-IFs are adopted in HEVC for the sake of coding efficiency. The HEVC interpolation filters are designed from the DCT type-II (DCT-II) transform [15–17] that reduces the bit-rate by approximately 4.0% for Luma and 11.3% for Croma components compared with the H.264/AVC (Advanced Video Coding) interpolation filters. The coding efficiency increments are very remarkable for some sequences and can reach a maximum coding gain of 21.7% [18]. The filter lengths of the DCT-II-based interpolation filter (DCT-IF) are 8-point and 7-point for the 1/2-pixel and 1/4-pixel interpolations, respectively. In the present paper, discrete sine transform [19] (DST)-based interpolation filters (DST-IFs) that use different interpolation filter lengths are proposed.

This paper is organized as follows. Section 2 presents the ideal interpolation filter, the sinc function, the DCT-IF, the proposed DST-IF, and an analysis of the interpolation filters. Section 3 presents the experiment results, and Section 4 concludes the paper.

2. Interpolation Filters for Generating Fractional Pixels

2.1. The Sinc-Based Interpolation Filter

The sinc-based interpolation filter is an ideal interpolation filter in terms of signal processing and its equation is as follows:

$$x(t) = \sum_{k=-\infty}^{\infty} x(kT_s) \frac{\sin \frac{\pi}{T_s}(t - kT_s)}{\frac{\pi}{T_s}(t - kT_s)} \tag{1}$$

where the sinc-based interpolation filter is defined as $x(t)$, t represents the locations of the subsamples, and k is the integer sample value, and T_s is the sampling period that is equal to 1. When the sinc-based interpolation filter is lengthened from $-\infty$ to ∞, it is the ideal interpolation filter to reconstruct all the samples. Although the sinc-based interpolation filter is ideal, it is not possible to implement it in HEVC. Since it is impossible to reference all of the neighbor pixels in a picture, the DCT-IF is adopted in HEVC, the filter lengths of which are restricted within 8-point and 7-point for the 1/2-pixel and 1/4-pixel interpolations, respectively.

2.2. The DCT-II Interpolation Filter (DCT-IF) in HEVC

The DCT-IF [9] in HEVC is designed in a different way, but it can be designed easily in this paper from the following forward/inverse DCT-II:

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

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c_k X(k) \cos \frac{(n + 1/2)\pi k}{N}. \tag{3}$$

In Equation (2), $X(k)$ is the DCT-II coefficients and the input pixel $x(n)$ is the IDCT-II (Inverse DCT-II) coefficients in Equation (3).

$$c_k = \begin{cases} \frac{1}{\sqrt{2}}\, , \ \mathrm{k} = 0 \\ 1\, , \ \text{otherwise} \end{cases} \tag{4}$$

where c_k is $1/\sqrt{2}$ at k = 0, and c_k is 1 at k $\neq$ 0. The substitution of Equation (2) into Equation (3) results in the following DCT-IF equation:

$$x(n) = \frac{2}{N}\sum\nolimits_{m=0}^{N-1} x(m)\sum\nolimits_{k=0}^{N-1} c_k^2 \cos\frac{(m+1/2)\pi k}{N}\cos\frac{(n+1/2)\pi k}{N}. \tag{5}$$

For example, the 1/2-pixel interpolation filter, when n = 3.5, in the 8-point DCT (N = 8) is derived as a linear combination of the cosine coefficients and $x(m)$, m = 0, 1, . . . , 7. Similarly, the 1/4-pixel interpolation filter, when n = 3.25, in the 7-point DCT (N = 7) is derived as a linear combination of the cosine coefficients and $x(m)$, m = 0, 1, . . . , 6. Lastly, the DCT-IFs that interpolate the 1/2-pixel and 1/4-pixel interpolations are shown as the integer numbers in Table 1. The filter-coefficient order of the 3/4-pixel interpolation filter is the reverse of the filter-coefficient order of the 1/4-pixel interpolation filter.

Table 1. 8-point and 7-point DCT-II based interpolation filter coefficients in high efficiency video coding (HEVC).

Index i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1/2-pixel filter[i]	−1	4	−11	40	40	−11	4	−1
1/4-pixel filter[i]	−1	4	−10	58	17	−5	1	

Figure 1 is an example of the integer- and fractional-pixel positions in the Luma motion compensation. In Figure 1, the capital letters (A_0 to A_7) indicate the integer-pixel position, the small letter b_0 is the 1/2-pixel position, and a_0 and c_0 are the 1/4-pixel and 3/4-pixel positions, respectively. For example, using the DCT-IF, the b_0 and a_0 are calculated from Table 1 as follows:

$$\begin{aligned} b_0 &= (-1\cdot A_0 + 4\cdot A_1 - 11\cdot A_2 + 40\cdot A_3 + 40\cdot A_4 - 11\cdot A_5 + 4\cdot A_6 - 1\cdot A_7 + 32) \gg 6 \\ a_0 &= (-1\cdot A_0 + 4\cdot A_1 - 10\cdot A_2 + 58\cdot A_3 + 17\cdot A_4 - 5\cdot A_5 + 1\cdot A_6 + 32) \gg 6 \end{aligned} \tag{6}$$

where the computation of a_0 is the same as that of b_0 from Table 1, the computation of c_0 is in the order that is the reverse of that of a_0, and the ">>" operation means the bit-wise shift right.

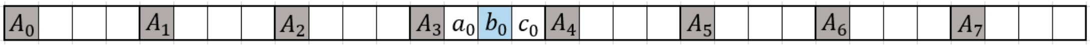

Figure 1. Fractional pixel position in Luma motion compensation.

2.3. The Proposed DST-VII Interpolation Filter (DST-IF)

The DST-IF for HEVC can easily be designed in this paper from the forward/inverse DST-VII. The DST-VII and inverse DST-VII are defined as follows:

$$X(k) = \sqrt{\frac{2}{N+\frac{1}{2}}}\sum\nolimits_{n=0}^{N-1} x(n)\sin\frac{(n+1)(k+\frac{1}{2})\pi}{N+\frac{1}{2}} \tag{7}$$

$$x(n) = \sqrt{\frac{2}{N+\frac{1}{2}}}\sum\nolimits_{k=0}^{N-1} X(k)\sin\frac{(n+1)(k+\frac{1}{2})\pi}{N+\frac{1}{2}} \tag{8}$$

where $X(k)$ is the DST-VII coefficient and $x(n)$ represents the input pixels. The substitution of Equation (7) into Equation (8) results in the following DST-IF equation:

$$x(n) = \frac{2}{N+\frac{1}{2}}\sum\nolimits_{m=0}^{N-1} x(m)\sum\nolimits_{k=0}^{N-1}\sin\frac{(m+1)(k+\frac{1}{2})\pi}{N+\frac{1}{2}}\sin\frac{(n+1)(k+\frac{1}{2})\pi}{N+\frac{1}{2}}. \tag{9}$$

In the similar way to obtain the DCT-IF coefficients, the DST-IF is derived from Equation (9). For example, the 1/2-pixel interpolation filter, when n = 3.5, in the 8-point DST (N = 8) is derived as a linear combination of the sine coefficients and $x(m)$, m = 0, 1, ... , 7. Similarly, the 1/4-pixel interpolation filter, when n = 3.25, in the 7-point DST (N = 7) is derived as a linear combination of the sine coefficients and $x(m)$, m = 0, 1, ... , 6. Lastly, the DST-IFs that interpolate the 1/2-pixel and 1/4-pixel interpolations are shown in Table 2. The filter-coefficient order of the 3/4-pixel interpolation filter is the reverse of the filter-coefficient order of the 1/4-pixel interpolation filter [20].

Table 2. 8-point and 7-point discrete sine transform (DST)-VII-based interpolation filter (DST-IF) coefficients.

Index i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1/2-pixel filter[i]	−2	6	−13	41	41	−13	6	−2
1/4-pixel filter[i]	−2	5	−11	58	18	−6	2	

In the given example, the 8-point and 7-point DST-IFs were derived, but the M-point and (M-1)-point DST-IFs, where M > 8, can be easily derived in a similar way for high-resolution sequences to improve the video-coding efficiency.

The 12-point and 11-point DST-IFs that interpolate the 1/2-pixel and 1/4-pixel interpolations are shown in Table 3. The 12-point and 11-point DST-IFs in Table 3 are derived in this paper from 10.3390/sym9110257 (9), where N = 12 and n = 5.5, and N = 11 and n = 5.25, respectively. The 12-point and 11-point DCT-IFs in Table 4 were derived in a similar way.

Table 3. 12-point and 11-point DST-VII-based interpolation filter (DST-IF) coefficients.

Index i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1/2-pixel filter[i]	−1	2	−4	7	**−13**	**41**	**41**	**−13**	7	−4	2	−1
1/4-pixel filter[i]	−1	2	−3	6	−11	58	**19**	**−8**	4	**−3**	1	

Table 4. 12-point and 11-point DCT-II-based interpolation filter (DCT-IF) coefficients.

Index i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1/2-pixel filter[i]	−1	2	−4	7	**−12**	**40**	**40**	**−12**	7	−4	2	−1
1/4-pixel filter[i]	−1	2	−3	5	−11	58	**18**	**−7**	4	**−2**	1	

2.4. Analysis of the Interpolation Filters

Figure 2 shows all of the different graphs of the magnitude responses of the 1/2-pixel interpolation filters. In the x-axis, the discrete time frequency $\hat{\omega}$ is normalized in the range of 0 to 1, where 1 corresponds to the π radian. The y-axis is the magnitude response. Figure 2 illustrates the magnitude-response graphs of five (5) interpolation filters reconstructing the 1/2-pixel position. The sinc function, which is assumed to be the ideal interpolation filter, is designed with a 48-point interpolation filter and represented by a dot-line. The 48-point sinc interpolation filter has relatively high frequency response even around $\hat{\omega} = 0.9\pi$ compared with other interpolation filters such as 8-point DCT-IF, 8-point DST-IF, 12-point DCT-IF, and 12-point DST-IF and it comprises many more ripples at high frequencies compared with the other interpolation filters. In particular, in the low frequency responses when $\hat{\omega} < 0.5\pi$, all interpolation filters have similar responses. It can be interpreted that all five (5) interpolation filters have similar low frequency responses, but the high frequency responses are different. Comparing the 8-point DCT-IF drawn in a gray line and the 8-point DST-IF drawn in a black line, the 8-point DST-IF has relatively high frequency responses compared with the 8-point DCT-IF around $\hat{\omega} = 0.9\pi$ even if the low frequency responses are quite similar. In case of the 12-point DST-IF and 12-point DCT-IF, which are represented by a green and red line, two interpolation filters have relatively higher frequency responses than the 8-point DST-IF and 8-point DCT-IF even if the low frequency responses are quite similar. The 12-point DST-IF and the 12-point DCT-IF have similar high frequency responses because they have almost similar interpolation filter coefficients as shown in Tables 3 and 4, where only the filter coefficients of integer pixel positions 4, 5, 6, and 7 are different in 1/2-pixel filter coefficients. This means that 12-point DST-IF and 12-point DCT-IF are similar when they are derived mathematically. Therefore, comparing the 12-point DST-IF with 8-point DCT-IF and DST-IF and 12-point DCT-IF in Figure 2, the 12-point DST-IF shows relatively high frequency responses, even though the 48-point sinc interpolation filter shows better high frequency responses than the other four (4) interpolation filters.

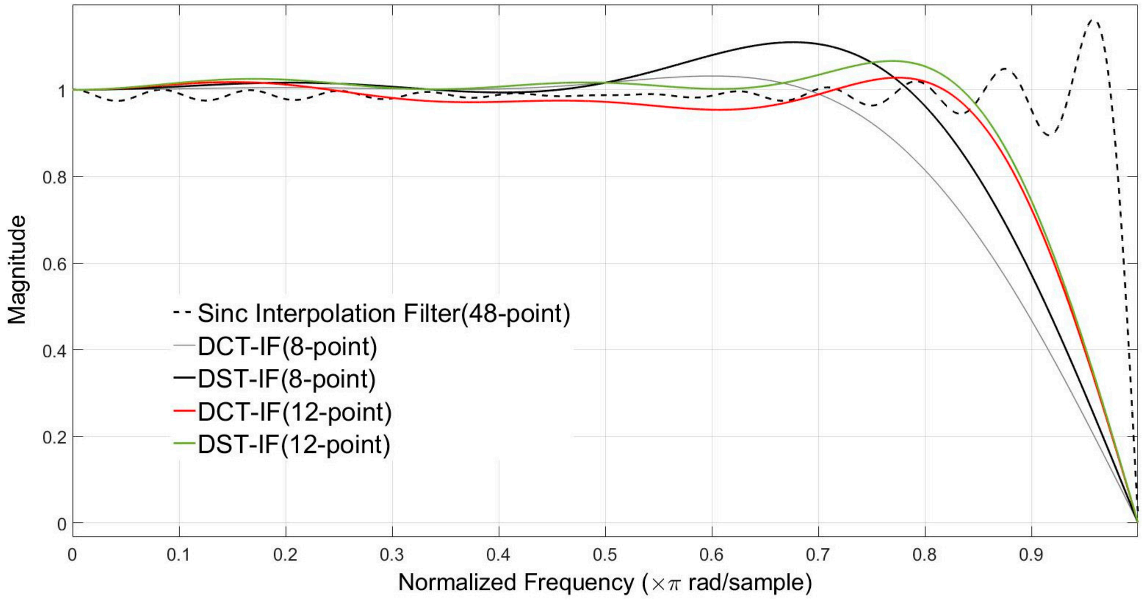

Figure 2. Magnitude responses of interpolation filters for the 1/2-pixel position in the Luma component.

3. Experimental Results

3.1. Experimental Conditions

The proposed DST-IF was implemented in the HEVC reference software, HM (HEVC test Model)-16.6 [21], according to the HEVC common-test conditions. Table 5 shows the test sequences where the sequences of the classes B, C, D, and E comprise the resolutions of 1080p, 832 × 480, 416 × 240, and 720p, respectively, and the proposed method was applied when the quantization-parameter (QP) values were 22, 27, 32, and 37, respectively. Tables 6 and 7 show the test sequences and the BD-rate gain compared with those of HM-16.6 for the Luma component in the low delay B (LDB), low delay P (LDP), and RA configurations, respectively. The random access configuration has hierarchical B pictures (IBBBBBBBP) which have a GOP (group of pictures) size of eight (8). The low delay structure is composed of the first I (intra) picture and the following P (predictive) pictures (IPPPPP . . .). The P pictures in the low delay structure are GPBs (generalized P and B pictures), in which the P pictures are replaced by B pictures having the same two reference pictures.

The negative sign of the BD-rate represents the bit-saving of the proposed method compared with that of HM-16.6 in the same PSNR (peak signal-to-noise ratio) [22].

Table 5. Test sequences used in HEVC common-test conditions.

Class	Sequence Name	Frame Count	Frame Rate	Bit Depth
B	Kimono	240	24 fps	8
B	ParkScene	240	24 fps	8
B	Cactus	500	50 fps	8
B	BQTerrace	600	60 fps	8
B	BasketballDrive	500	50 fps	8
C	RaceHorses	300	30 fps	8
C	BQMall	600	60 fps	8
C	PartyScene	500	50 fps	8
C	BasketballDrill	500	50 fps	8
D	RaceHorses	300	30 fps	8
D	BQSquare	600	60 fps	8
D	BlowingBubbles	500	50 fps	8
D	BasketballPass	500	50 fps	8
E	FourPeople	600	60 fps	8
E	Johnny	600	60 fps	8
E	KristenAndSara	600	60 fps	8

Table 6. DST-IF bit-saving results applied to uni- and bi-directional prediction.

Class	Sequence Name	Saving Bits (%)					
		8-Point and 7-Point DST-IF			12-Point and 11-Point DST-IF/12-Point and 11-Point DCT-IF		
		LDB	LDP	RA	LDB	LDP	RA
B	Kimono	0.3	1.2	0.2	0.6/0.5	2.5/0.5	0.2/0.3
B	ParkScene	0.8	2.1	0.3	1.7/1.3	3.9/1.6	0.5/0.9
B	Cactus	0.8	2.3	0.2	1.1/1.2	3.6/1.6	0.0/0.8
B	BasketballDrive	0.1	1.2	0.1	0.3/0.4	2.3/0.6	0.3/0.3
B	BQTerrace	1.5	5.3	1.0	2.7/3.4	8.6/4.4	1.5/2.3
C	RaceHorses	−0.9	0.3	−0.2	−1.2/−0.7	0.6/−0.1	−0.5/−0.2
C	BQMall	−0.2	1.3	−0.5	−0.5/−0.6	1.8/−0.2	−1.0/−0.5
C	PartyScene	−1.7	−0.2	−2.5	−3.5/−4.4	−1.7/−3.6	−4.5/−3.8
C	BasketballDrill	0.6	1.7	0.4	1.2/0.9	2.9/1.1	0.8/0.6
D	RaceHorses	0.1	0.8	−0.1	0.0/−0.2	1.2/0.0	−0.3/−0.2
D	BQSquare	−4.1	−0.4	−5.2	−7.5/−7.2	−2.9/−4.9	−9.0/−7.4
D	BlowingBubbles	−1.5	0.0	−1.8	−2.8/−3.1	−0.9/−2.2	−3.1/−2.4
D	BasketballPass	0.5	1.2	0.2	0.9/0.9	2.0/1.1	0.4/0.4
E	FourPeople	0.6	2.4	x	1.2/1.2	4.7/1.6	x
E	Johnny	0.5	4.4	x	1.0/1.9	9.0/2.3	x
E	KristenAndSara	0.6	2.4	x	1.2/1.5	5.5/1.4	x
	Overall	−0.1	1.6	−0.6	−0.2/−0.2	2.7/0.3	−1.1/−0.7

Table 7. DST-IF bit-saving results applied to bi-directional prediction.

Class	Sequence Name	Saving Bits (%)			
		8-Point and 7-Point DST-IF		12-Point and 11-Point DST-IF/12-Point and 11-Point DCT-IF	
		LDB	RA	LDB	RA
B	Kimono	0.1	0.1	0.1/0.1	0.0/0.2
B	ParkScene	0.2	0.1	0.0/0.3	0.0/0.6
B	Cactus	0.2	0.0	−0.4/0.2	−0.3/0.6
B	BasketballDrive	0.0	0.0	−0.1/0.1	−0.1/0.2
B	BQTerrace	1.1	0.8	0.4/2.3	0.8/2.0
C	RaceHorses	−0.7	−0.2	−1.3/−0.5	−0.6/−0.2
C	BQMall	−0.3	−0.6	−1.2/−0.9	−1.3/−0.5
C	PartyScene	−1.5	−2.4	−3.8/−3.7	−4.4−3.5
C	BasketballDrill	0.2	0.2	0.3/0.2	0.2/0.3
D	RaceHorses	−0.1	−0.2	−0.3/−0.2	−0.5/−0.2
D	BQSquare	−3.7	−5.0	−8.3/−6.3	−9.1/−6.9
D	BlowingBubbles	−1.2	−1.7	−2.9/−2.4	−3.0/−2.1
D	BasketballPass	0.0	−0.1	−0.1/0.1	−0.1/0.2
E	FourPeople	0.4	x	−0.1/0.6	x
E	Johnny	−0.4	x	−1.5/0.2	x
E	KristenAndSara	0.2	x	−0.2/0.5	x
	Overall	−0.3	−0.7	−1.2/−0.6	−1.4/−0.7

3.2. Experimental Results

HM-16.6 uses an 8-point filter and a 7-point filter for the 1/2-pixel and 1/4-pixel interpolations, respectively. From Table 6, the average bit-saving (BD-rate gain) in the RA configuration was improved by 0.6% with the use of the 8-point DST-IF for 1/2-pixel and 7-point DST-IF for 1/4-pixel. Especially, the result of BQSquare in Class D achieved a bit-saving up to 5.2% in the RA configuration. The average bit-savings of 0.6% and 0.1% were achieved in the RA and LDB configurations, respectively. However, the average bit-saving was decreased by 1.6% in the LDP configuration. In Table 6, the 12-point and 11-point DST-IFs that were applied to HM-16.6 also showed bit-saving in the RA and LDB configurations and bit-increasing (BD-rate loss) in the LDP configuration. In Table 6, Class E sequences in the RA configuration are not experimented because they are not experimental condition in the HEVC test. Those sequences are marked as x.

Interestingly, the DST-IFs in the LDP configuration show bit increments (BD-rate loss), while the DST-IFs in the RA and LDB configurations show bit-savings. It is because the backward (uni-directional) prediction using the decoded past pictures provides the incomplete motion-compensated block compared with the bi-directional prediction that utilizes the average pixel values of two different blocks that were derived by the forward and backward motion-compensations for subsample interpolation. Therefore, the proposed 12-point and 11-point DST-IFs are applied only on the bi-directional motion-compensated blocks. The 12-point and 11-point DST-IFs, which are almost the same filter coefficients as the 12-point and 11-point DCT-IFs, are effective on the bi-directional prediction. Table 7 shows the results of the DST-IF bit-saving results applied only on the bi-directional prediction. In the RA and LDB configurations, the 8-point and 7-point DST-IFs achieved bit-savings of 0.7% and 0.3% compared with HM-16.6, respectively, and the 12-point and 11-point DST-IFs achieved bit-savings of 1.4% and 1.2% compared with HM-16.6, respectively. Table 7 shows the results of the 12-point and 11-point DCT-IFs as well. It shows bit-savings of 0.6% and 0.7% in the LDB and RA configurations compared with HM-16.6, respectively.

Table 8 shows the computational complexity results. As the 12-point and 11-point DST-IFs reference four additional neighbor pixels compared with the 8-point and 7-point DST-IFs in HEVC, when both the uni-directional and bi-directional predictions were applied, the computational complexities in the encoding process and the decoding process were increased by 118% and 113%, respectively. However, the 12-point and 11-point DST-IFs, which were applied on only the bi-directional prediction, increased the computational complexity in the encoding process by 104% and in the decoding process by 107%. The computational complexity of the 12-point and 11-point DCT-IFs is almost same as that of the 12-point and 11-point DST-IFs. Even if the complexity of the proposed 12-point and 11-point DST-IFs is increased compared with that of the existing 8-point and 7-point DCT-IFs in HEVC, the proposed method gives better bit-saving results than the existing method.

Table 8. Results of the computational complexity of the proposed method in the low delay B (LDB) configuration.

Computational Complexity		
Proposed Methods	**Encoding Time (%)**	**Decoding Time (%)**
HM-16.6 vs. 8- and 7-point DST-IFs (uni- and bi-directional predictions)	101	101
HM-16.6 vs. 8- and 7-point DST-IFs (bi-directional prediction only)	97	99
HM-16.6 vs. 12- and 11-point DST-IFs (uni- and bi-directional predictions)	118	113
HM-16.6 vs. 12- and 11-point DST-IFs (bi-directional prediction only)	104	107

For an alternative method, one interpolation filter was chosen between the DCT-IF and the DST-IF, and this experiment has been tested using the coding unit-level rate-distortion optimization [23], but the results are worse than those of Tables 6 and 7 because one signaling bit is needed to indicate which interpolation filter is used in the decoder side. An alternative interpolation method selecting the DCT-IF and DST-IF based on Coding Tree Unit (CTU) will be explored in a future study.

4. Conclusions

In this paper, DST-IF pairs of 12-point and 11-point filter lengths are proposed to achieve a bit-rate reduction compared with the 8-point and 7-point DCT-IFs. Interestingly, the 12-point DST-IF and the 12-point DCT-IF have similar high frequency responses because the 12-point DST-IF and 12-point DCT-IF derived have almost similar interpolation filter coefficients as shown in Tables 3 and 4. The experiment results show that the proposed DST-IF pairs achieved coding gains in the RA and LDB configurations. However, as the bit-rate was increased in the LDP configuration using the uni-directional prediction, the proposed DST-IF method was applied only on the bi-directional prediction. Overall, the proposed 12-point and 11-point DST-IFs achieved average BD-rate reductions of 1.4% and 1.2% compared with the 8-point and 7-point DCT-IFs in the RA and LDB configurations of the Luma component, respectively. We believe this method can be considered in the next video coding standard.

Acknowledgments: This research was in part supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science, ICT and Future Planning) (NRF-2015R1A2A2A01006085).

Author Contributions: MyungJun Kim and Yung-Lyul Lee conceived and designed the experiments; MyungJun Kim performed the experiments; Yung-Lyul Lee wrote the paper.

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

References

1. Bros, B.; Han, W.-J.; Ohm, J.-R.; Sulivan, G.J.; Wang, Y.-K.; Wiegand, T. High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent). *J. Inst. Telev. Eng. Jpn.* **2013**, *67*, 244–247.
2. Sullivan, G.J.; Ohm, J.-R.; Han, W.-J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst.* **2012**, *22*, 1649–1668. [CrossRef]
3. Jain, J.R.; Jain, A.K. Displacement measurement and its application in interframe image coding. *IEEE Trans. Commun.* **1981**, *29*, 1799–1804. [CrossRef]
4. Girod, B. Motion-compensating prediction with fractional-pel accuracy. *IEEE Trans. Commun.* **1993**, *41*, 604–612. [CrossRef]
5. Wiegand, T.; Zhang, X.; Girod, B. Long-term memory motion-compensated prediction. *IEEE. Trans. Circuits Syst. Video Technol.* **1999**, *9*, 70–84. [CrossRef]
6. Flierl, M.; Wiegand, T.; Girod, B. Rate-constrained multihypothesis prediction for motion-compensated video compression. *IEEE Trans. Circuits Syst. Video Technol.* **2002**, *12*, 957–969. [CrossRef]
7. Rosewarne, C.; Bross, B.; Naccari, M.; Sharman, K.; Sullivan, G.J. *High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 2*; JCTVC-T1002; ITU-T/ISO/IEC Jt. Collab. Team Video Coding (JCT-VC): New York, NY, USA, February 2015.
8. Wedi, T. Motion compensation in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 577–586. [CrossRef]
9. McClellan, J.H.; Schafer, R.W.; Yoder, M.A. *Signal Processing First*; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2003.
10. Haykin, S.; Van Veen, B. *Signals and Systems*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2003.
11. Karczewicz, M.; Chen, P.; Joshi, R.L.; Wang, X.; Chien, W.J.; Panchal, R.; Reznik, Y.; Coban, M.; Chong, I.S. A hybrid video coder based on extended macroblock sizes, improved interpolation, and flexible motion representation. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 1698–1708. [CrossRef]

12. Marpe, D.; Schwarz, H.; Bosse, S.; Bross, B.; Helle, P.; Hinz, T.; Kirchhoffer, H.; Lakshman, H.; Nguyen, T.; Oudin, S.; et al. Video compression using nested quadtree structures, leaf merging, and improved techniques for motion representation and entropy coding. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 1698–1708. [CrossRef]
13. Ugur, K.; Andersson, K.; Fuldseth, A.; Bjøntegaard, G.; Endresen, L.P.; Lainema, J.; Hallapuro, A.; Ridge, J.; Rusanovskyy, D.; Zhang, C.; et al. High performance, low complexity video coding and the emerging HEVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 1698–1708. [CrossRef]
14. Han, W.J.; Min, J.; Kim, I.K.; Alshina, E.; Alshin, A.; Lee, T.; Chen, J.; Seregin, V.; Lee, S.; Hong, Y.M.; et al. Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 1698–1708. [CrossRef]
15. Ugur, K.; Alshin, A.; Alshina, E.; Bossen, F.; Han, W.J.; Park, J.H.; Lainema, J. Motion Compensated Prediction and Interpolation Filter Design in H.265/HEVC. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 946–956. [CrossRef]
16. Wien, M. *High Efficiency Video Coding—Coding Tools and Specification*; Springer: Berlin/Heidelberg, Germany, 2015.
17. Sze, V.; Budagavi, M.; Sullivan, G.J. *High Efficiency Video Coding (HEVC)—Algorithms and Architectures*; Springer: Heidelberg, Germany, 2014.
18. Wiegand, T.; Sullivan, G.J.; Bjøntegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst.* **2003**, *13*, 560–576. [CrossRef]
19. Stitch Dicrete Sine Transform. 2013. Available online: http://planetmath.org/sites/default/files/texpdf/39764.pdf (accessed on 1 September 2016).
20. Kim, M.J.; Kim, N.; Lee, Y. Investigation on interpolation filters in HEVC. In Proceedings of the International Workshop on Advanced Image Technology, Penang, Malaysia, 8–10 January 2017.
21. Bossen, F. *Common HM Test Conditions and Software Reference Configurations*; JCTVC-H1100; ITU-T/ISO/IEC Jt. Collab. Team Video Coding (JCT-VC): New York, NY, USA, 2012.
22. Bjøntegaard, G. Calculation of Average PSNR Differences between RD-curves. In Proceedings of the ITU-T VCEG Meeting, Austin, TX, USA, 2–4 April 2001.
23. Sullivan, G.J.; Wiegand, T. Rate-Distortion Optimization for Video Compression. *IEEE Signal Process. Mag.* **1998**, *15*, 74–90. [CrossRef]