Interpolation Scheme using Simple Multi-Directional Filters for Fractional-Pel Motion Compensation

Do Quan and Yo-Sung Ho, *Senior Member*, IEEE

**Abstract — ***The H.264/AVC video coding standard supports motion-compensated prediction with quarter-pel accuracy of motion vectors. In the case of fractional-pel motion vector, pixels at fractional positions have to be interpolated. In an interpolation scheme, computational complexity and coding efficiency are the most important criteria. In order to enhance both the computational complexity and coding efficiency for various textures of videos, we propose a new interpolation scheme using multi-directional filters. Experimental results show that our proposed scheme outperforms other schemes, including the H.264/AVC standard and the latest interpolation schemes, in terms of complexity and coding efficiency.*[1]

Index Terms — Interpolation, Directional interpolation filter, H.264/AVC sub-pixel interpolation, Video coding, H.264/AVC.

I. INTRODUCTION

The H.264/AVC video coding standard [1] employs the block-based hybrid coding with motion compensated prediction. For each block with a given resolution, a motion vector is estimated to discover hidden information inside an already reconstructed reference image. Therefore, the reference image has to be interpolated. Based on design of the sub-pixel interpolation filter, we not only balance the tradeoff between memory accessing scheme and interpolation filtering [2] but also result in spatial error concealment [3]. There are two criteria to evaluate an interpolation scheme: computational complexity and coding efficiency. This section describes a progress to address interpolation problems and improvements in coding efficiency to obtain our proposed interpolation scheme.

H.264/AVC uses fixed coefficient interpolation at half-pel and quarter-pel accuracy. It uses a 6-tap Wiener interpolation filter [1]. In the first step of the interpolation process, as shown in Fig. 1, b_A, b_B, b, b_D, b_E, b_F half-pel positions are calculated using a horizontal 6-tap Wiener filter while H_1, H_2, h, H_4, H_5, H_6 half-pel positions are calculated by using a vertical 6-tap Wiener filter. In the second step, the quarter-pel positions are calculated by using a bilinear filter which uses the given full-pel positions and the above-calculated half-pel positions. Due

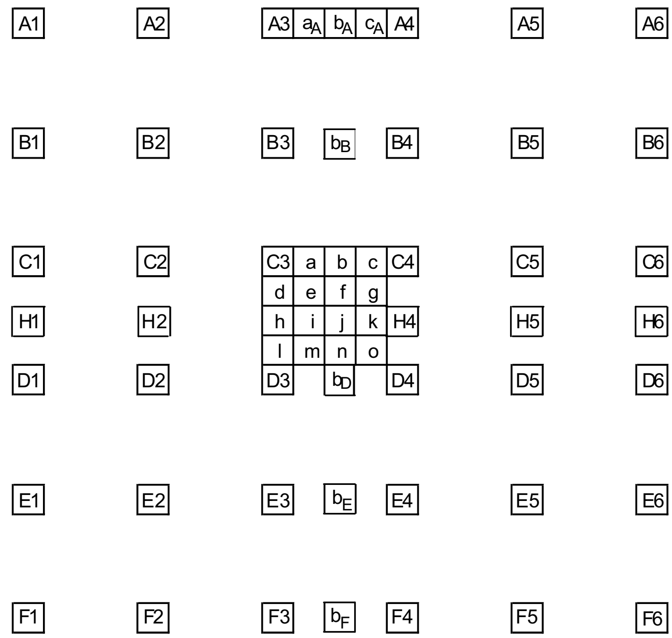


Fig. 1. Integer and fractional positions

to the use of fixed coefficient interpolation filter for all pictures of sequences, the coding efficiency is limited. Since this interpolation uses only horizontal and vertical directions, it is not suitable for other textural sequences. These problems are motivations for other works.

In order to improve coding efficiency and reduce prediction error energy, the two-dimensional non-separable 6-tap adaptive interpolation filter (AIF) scheme was proposed [4]. How minimizing the prediction error energy of each coded frame helps to obtain coefficients of each filter. In order to simplify the interpolation process, the interpolation filter is symmetric. However, the interpolation process is still complicated, approximated three times higher interpolation complexity than the fixed coefficient interpolation of the H.264/AVC standard [4].

In order to simplify the implementation of the two-dimensional non-separable 6-tap adaptive interpolation scheme, a separable adaptive interpolation scheme was proposed [5]. Instead of using two-dimensional filter, these authors successively use a horizontal interpolation filtering and a vertical interpolation filtering for the interpolation process. In this way, they get similar results as using the two-dimensional non-separable 6-tap adaptive interpolation scheme with only half of the time delay.

An adaptive interpolation with directional filters [6] and a simple modification of this idea, named adaptive-fixed scheme [7], were proposed. These schemes use one-dimensional

[1] This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-(C1090-1011-0003)).

Do Quan and Yo-Sung Ho are with the School of Information and Mechatronics, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, Korea (e-mails: viequando@gist.ac.kr, hoyo@gist.ac.kr).

Contributed Paper
Manuscript received 10/13/10
Current version published 12/23/10
Electronic version published 12/30/10.

filters and six given full-pel positions to obtain all sub-pixels. Since the obtaining of the filter coefficients requires minimizing the prediction error energy of each coded frame using iterations as given in [4], these schemes are still complicated. The process for minimizing the prediction error energy is a reason of high delay. In addition, the rate-distortion (RD) performances of these schemes in [6] and [7] lose much for detailed texture pictures, noised pictures, and other 45^0-based directional pictures compared to the other mentioned schemes.

From the above-mentioned review of representative interpolation schemes, we propose an interpolation scheme to solve the above-mentioned problems. In our proposed interpolation scheme, we employ multi-directional filters, so that the interpolation process has low computational complexity and low delay. Since our proposed interpolation process takes into account multi-directional filtering, it can be suitable for various textural sequences. In addition, it does not need any iteration for each sub-pixel to find the filter coefficients and it uses only one-dimensional filter operations with fixed filter coefficients.

II. Proposed Interpolation Scheme using Simple Multi-directional Filters

In this section, we describe the process of sub-pel interpolation and the process of obtaining interpolation filter coefficients. Then, we analyze the complexity of our proposed interpolation scheme.

A. Sub-pel Interpolation Process and Interpolation Filter Coefficients

In order to simplify the separable and non-separable adaptive interpolation filter, directional adaptive interpolation filter was proposed [6], [7]. This interpolation scheme uses directional filters and corresponding directional integer pixels to obtain sub-pixels. It takes into account directional filtering; hence, it offers a significant loss reduction and coding efficiency. However, this scheme is still complicated, because the filter coefficients are obtained by many iterations for each sub-pixel. We will reduce this high complexity by employing fixed filter coefficients.

An efficient set of directional fixed coefficient filters were proposed in [8]. These filters have low complexity and low delay, since they use fixed coefficients. We employ these filters for the interpolation process in various directions to solve the above-mentioned problems.

The following is our interpolation process. In this process, we employ integer pixels to obtain the corresponding sub-pixels, as shown in Fig. 1. Furthermore, we interpolate sub-pixels based on given integer pixels and take filtering along with horizontal, vertical, 45^0-based, -45^0-based, 135^0-based, and -135^0-based directions. In order to save bits and obtain extra gain, the filters may be symmetric or asymmetrical filters. We determine sets of sub-pixels which share the same filter as follows:

- Sub-pixels $\{a, d\}$ have the same filter h_1.
- Sub-pixels $\{b, h, j\}$ have the same filter h_2.
- Sub-pixels $\{c, l\}$ have the same filter h_3.
- Sub-pixels $\{f, i\}$ have the same filter h_4.
- Sub-pixels $\{k, n\}$ have the same filter h_5.
- Sub-pixels $\{e, o\}$ have the same filter h_6.
- Sub-pixels $\{g, m\}$ have the same filter h_7.

The sub-pixels in our interpolation process are obtained in detail using the following formulas:

$$a = [C1h_1(1) + C2h_1(2) + C3h_1(3) + C4h_1(4) + C5h_1(5) + C6h_1(6) + 128] \gg 8 \quad (1)$$

$$d = [A3h_1(1) + B3h_1(2) + C3h_1(3) + D3h_1(4) + E3h_1(5) + F3h_1(6) + 128] \gg 8 \quad (2)$$

$$b = [(C1 + C6)h_2(1) + (C2 + C5)h_2(2) + (C3 + C4)h_3(3) + 128] \gg 8 \quad (3)$$

$$h = [(A3 + F3)h_2(1) + (B3 + E3)h_2(2) + (C3 + D3)h_3(3) + 128] \gg 8 \quad (4)$$

$$j = [(A1 + A6 + F1 + F6)h_2(1) + (B2 + B5 + E2 + E5)h_2(2) + (C3 + C4 + D3 + D4)h_3(3) + 128] \gg 8 \quad (5)$$

$$c = [C1h_3(1) + C2h_3(2) + C3h_3(3) + C4h_3(4) + C5h_3(5) + C6h_3(6) + 128] \gg 8 \quad (6)$$

$$l = [A3h_3(1) + B3h_3(2) + C3h_3(3) + D3h_3(4) + E3h_3(5) + F3h_3(6) + 128] \gg 8 \quad (7)$$

$$f = [(A1 + A6)h_4(1) + (B2 + B5)h_4(2) + (C3 + C4)h_4(3) + (D3 + D4)h_4(4) + (E2 + E5)h_4(5) + (F1 + F6)h_4(6) + 128] \gg 8 \quad (8)$$

$$i = [(A1 + F1)h_4(1) + (B2 + E2)h_4(2) + (C3 + D3)h_4(3) + (C4 + D4)h_4(4) + (B5 + E5)h_4(5) + (A6 + F6)h_4(6) + 128] \gg 8 \quad (9)$$

$$n = [(A1 + A6)h_5(1) + (B2 + B5)h_5(2) + (C3 + C4)h_5(3) + (D3 + D4)h_5(4) + (E2 + E5)h_5(5) + (F1 + F6)h_5(6) + 128] \gg 8 \quad (10)$$

$$k = [(A1 + F1)h_5(1) + (B2 + E2)h_5(2) + (C3 + D3)h_5(3) + (C4 + D4)h_5(4) + (B5 + E5)h_5(5) + (A6 + F6)h_5(6) + 128] \gg 8 \quad (11)$$

$$e = [A1h_6(1) + B2h_6(2) + C3h_6(3) + D4h_6(4) + E5h_6(5) + F6h_6(6) + 128] \gg 8 \quad (12)$$

$$o = [A6h_6(1) + B5h_6(2) + C4h_6(3) + D3h_6(4) + E2h_6(5) + F1h_6(6) + 128] \gg 8 \quad (13)$$

$$m = [A6h_7(1) + B5h_7(2) + C4h_7(3) + D3h_7(4) + E2h_7(5) + F1h_7(6) + 128] \gg 8 \quad (14)$$

$$g = [A1h_7(1) + B2h_7(2) + C3h_7(3) + D4h_7(4) + E5h_7(5) + F6h_7(6) + 128] \gg 8 \quad (15)$$

where h_i is filter coefficient of filter i.

We can show the interpolation process in terms of filtering by using Fig. 2. Note that each color in this figure represents one filter with the individual filter coefficients. Filter h_0 is a special filter for particular position g.

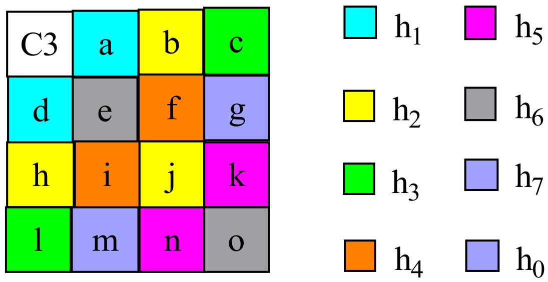


Fig. 2. Arrangement of filters for our interpolation process to obtain sub-pixels.

The above-mentioned formulas show how to determine the sub-pixels with the given integer pixels in one block. In the interpolation process, we use seven filters, whose filter coefficients of these filters will be described as follows.

We use three basic filters which used in [8]. Filter coefficients of these filters were defined:

$$h_1 = \{3,-15,111,37,-10,2\}/128 \quad (16)$$

$$h_2 = \{3,-17,78,78,-17,3\}/128 \quad (17)$$

$$h_3 = \{2,-10,37,111,-15,3\}/128 \quad (18)$$

We arrange these filters for fractional positions as follows. h_1, h_2, and h_3 are applied into (0, 1/4) and (1/4, 0), (0, 1/2) and (1/2, 0), and (0, 3/4) and (3/4, 0) fractional positions, respectively. From the three above-defined filters, we combine h_1, h_2, and h_3 to obtain other directional filters. The remaining filters for the other fractional positions are determined as follows:

$$h_4 = \{3,3,-17,-15,78,111\}/128 \quad (19)$$

$$h_5 = \{2,3,-10,-17,37,78\}/128 \quad (20)$$

$$h_6 = \{3,3,-15,-15,111,111\}/128 \quad (21)$$

$$h_7 = \{3,2,-15,-10,111,37\}/128 \quad (22)$$

At sub-pel position *g*, we apply a special filter [8]. This filter is called a strong filter.

$$h_0 = \{\{0,5,5,0\},\{5,22,22,5\},\{5,22,22,5\},\{0,5,5,0\}\}/128 \quad (23)$$

From the two-dimensional frequency responses of the directional filter and that of the strong filter shown in Fig. 3, we determine the strong filter, because it has a narrowed pass-band compared to the corresponding directional filter. In other words, the strong filter attenuates faster than the directional filter. This is the reason why we use the strong filter instead of the variety of filter responses to choose from during motion vector selection.

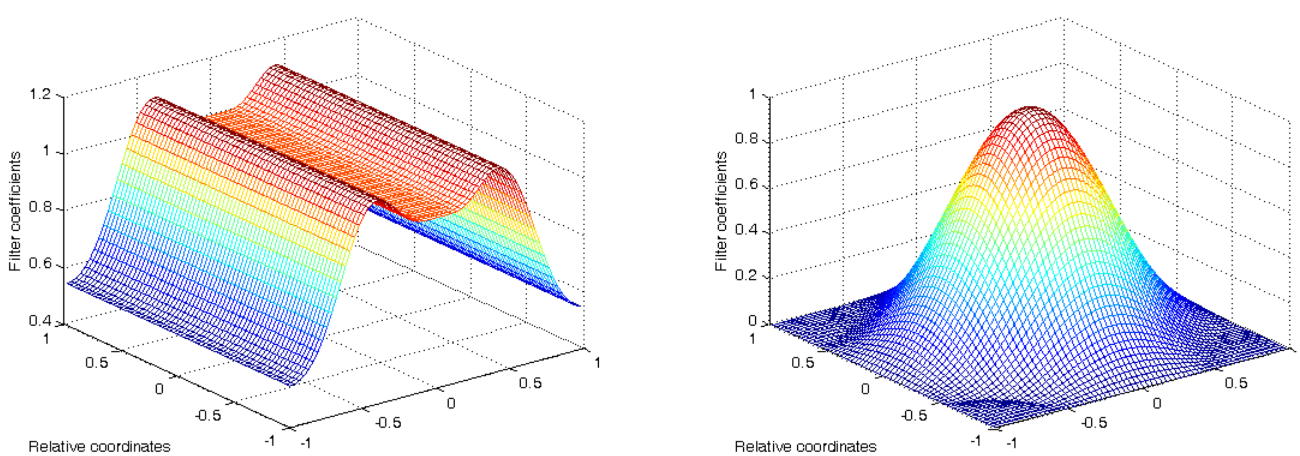
Using fixed filter coefficients, the interpolation process is improved by adding fixed offsets [8]. The fixed offsets are called local DC offsets. The local DC offset has the same video characteristics as weighted prediction offset without adding complexity and delay.

There are 49 coefficients to transmit to the decoder. Instead of using many iterations to find the filter coefficients, in our proposed interpolation scheme, we use fixed filter coefficients but strong filter. Hence, the complexity of the interpolation process is minimum compared to that of existing interpolation schemes.

The interpolation coefficients of [8] and [6] are calculated by minimizing the prediction error energy. These schemes use the same analytical process as described in [4]. However, as we discussed in Section I, the scheme in [4] is still complicated. In our proposed scheme, we have a combination of three basic filters as h_1, h_2, and h_3 to determine the other filters for the remaining fractional positions. In this way, we reduce the complexity of minimizing the prediction error energy but we still keep advantages of multi-directional filters, low complexity, low delay, strong filter position, and local DC offset.

B. Complexity Analysis

As we have discussed above, our proposed interpolation scheme is simple, because we use fixed filter coefficients for the interpolation process. Moreover, we do not need any iteration for each sub-pixel to find the filter coefficients. In addition, these filters have low complexity and low delay, because we do not need to store the intermediate values at high precision in the interpolation process.



(a) **(b)**

Fig. 3. Two-dimensional frequency response of (a) directional filter and (b) strong filter.

TABLE I
NUMBER OF OPERATIONS FOR EACH SUB-PIXEL IN VARIOUS INTERPOLATION SCHEMES

Sub-pixel	Proposed scheme	Adaptive-fixed scheme [7]	Directional AIF scheme [6]	Separable AIF scheme [5]	2D non-separable AIF scheme [4]
{a,c,d,l}	13	13	13	13	13
{b,h}	10	10	10	10	10
{e,g,m,o}	13	13	13	42.25	58
{f,n}	19	33.25	19	35.5	55
{i,k}	19	33.25	19	39.25	55
{j}	16	30.25	16	32.25	41

TABLE II
AVERAGE INTERPOLATION COMPLEXITY

	Average number of operations
Proposed scheme	14.4
Adaptive-fixed scheme [7]	19.15
Directional AIF scheme [6]	14.4
Separable AIF scheme [5]	28.18
2D non-separable AIF scheme [4]	37.67

TABLE III
THE HIGHEST INTERPOLATION COMPLEXITY

	Greatest number of operations
Proposed scheme	19
Adaptive-fixed scheme [7]	33.25
Directional AIF scheme [6]	19
Separable AIF scheme [5]	42.25
2D non-separable AIF scheme [4]	58

We calculate the number of operations to obtain each sub-pixel sample for the 4×4 block in various interpolation schemes using the similar concept as described in [9]. We use the same complexity analysis that was used in the existing schemes. In addition, the strong filter process does not seriously affect the complexity of our proposed interpolation scheme [8]. In other words, it only brings a marginal complexity increment.

Using the above-mentioned concept, we obtain the analysis of the complexity of the existing interpolation schemes, as shown in Tables I-III. Table I shows in detail the number of operations for each sub-pixel in various interpolation schemes. There are 15 sub-pixels whose numbers of operations are determined in Table I. From Table I, we can easily obtain the average number of operations for each scheme. Table II presents the average interpolation complexity. From Table I, we also choose the highest interpolation complexity. In this case, the number of operations is the greatest number. Table III describes the highest interpolation complexity.

Based on the number of operations in various interpolation schemes in Tables I-III, the number of operations of our proposed interpolation scheme is similar to that of the directional AIF scheme [6] with our proposed scheme's resulting in a marginal complexity increment due to ours using a strong filter. Both our scheme and the directional AIF interpolation scheme involve smaller numbers of operations than the other schemes do. Moreover, in our proposed scheme, we use fixed filter coefficients without any iteration to find filter coefficients as using in the directional AIF scheme; therefore, our proposed scheme is simpler than the directional AIF scheme.

III. EXPERIMENTAL RESULTS AND ANALYSIS

In order to evaluate the coding efficiency of our proposed scheme and compare our proposed scheme to the other schemes, we use the KTA reference software, version KTA1.9 [10], which is the official Video Coding Experts Group (VCEG) exploration software. We experimented on several QCIF, CIF, and HDTV sequences which are VCEG test sequences. For easy comparison with other reference schemes, we use the same test sequences for other reference schemes. We describe the experimental conditions in Table IV. Other conditions were defined in [11].

In this section, we first compare the coding efficiency of all schemes, including our proposed scheme, to the well-known H.264/AVC standard scheme. These schemes are integrated into the reference software with the above-mentioned coding conditions as mentioned above. Then, we directly compare the performance of our proposed scheme to those of other reference schemes. We compare our proposed scheme to three schemes: H.264/AVC scheme [1], adaptive-fixed scheme [7], and low complex AIF scheme [8], since those schemes are much related to our proposed scheme. Finally, rate-distortion (RD) curves show the difference between our proposed scheme and the other schemes.

A. Coding Efficiency of All Schemes vs. The H.264/AVC Standard

In this subsection, we use a reference scheme and compare it to other schemes. The reference is the H.264/AVC standard [1]. We compare the coding efficiency of all schemes to that of the H.264/AVC standard using the Bjontegaard tool [13] with the same coding conditions, the KTA reference software, and all required VCEG test sequences. Table V shows bit-rate savings in percentage. In this table, the negative sign denotes bit-rate savings of a test scheme while the positive sign shows that compared to the reference scheme, the test scheme needs more bit-rates.

Our proposed scheme achieves 4.5%, 4.5% and 8% bit-rate savings on average for QCIF, CIF and HDTV sequences, respectively. Table V illustrates that our proposed scheme results in a stable coding efficiency improvement. Furthermore, it also illustrates that compared to other schemes; our proposed scheme has significant improvements even in low resolutions while most schemes do not have significant improvements at these resolutions.

The lower resolution results in a lower coding efficiency. In other words, bit-rate savings are much better for higher

TABLE IV
MAIN EXPERIMENTAL CONDITIONS

Parameter	Settings	Parameter	Settings
Profile	High	Frame skip	2
Number of reference frames	4	Number of B frames	2
Search range	32	GOP structure	IBBP
Intra period	0	B List 0 references	2
CABAC	On	B List 1 references	1
8×8 transform	1	Quantization parameter	QPISlice = 22, 27, 32, 37
Rate-distortion optimization	On		QPPSlice = 23, 28, 33, 38
Weighted prediction	Off		QPBSlice = 24, 29, 34, 39

TABLE V
BIT-RATE SAVINGS IN PERCENTAGE OF ALL SCHEMES COMPARED TO THE H.264/AVC STANDARD

Sequence	Proposed scheme	AIF scheme [12]	Low complex AIF scheme [8]	Adaptive-fixed scheme [7]	Directional AIF scheme [6]	Separable AIF scheme [5]	2D non-separable scheme [4]
Foreman (QCIF)	-4.87	-5.19	-4.91	0.18	-0.66	-0.55	-0.85
Container (QCIF)	-4.7	-	-4.42	-0.43	0.84	1.18	1.93
Silent (QCIF)	-3.89	-0.51	-3.89	0.23	0.6	0.35	0.5
QCIF Average	**-4.49**	**-2.85**	**-4.41**	**-0.01**	**0.26**	**0.33**	**0.53**
Foreman (CIF)	-4.95	-5.57	-5.1	-0.47	-1.44	-2.53	-2.21
Mobile (CIF)	-5.11	-2.09	-5.12	-0.59	-0.83	-2.14	-1.35
Paris (CIF)	-3.58	0.19	-3.6	-0.27	0.37	0	-0.04
Tempete (CIF)	-4.26	-1.36	-4.5	0.16	-0.21	-1.21	-1.31
CIF Average	**-4.48**	**-2.21**	**-4.58**	**-0.29**	**-0.53**	**-1.47**	**-1.23**
Bigships (720p)	-4.61	-3.39	-3.9	-0.98	-6.13	-5.37	-6.21
City (720p)	-7.66	-10.04	-3.76	-4	-8.95	-9.23	-9.22
Crew (720p)	-11.07	-3.11	-5.88	-7.17	-4.37	-4.25	-4.94
Night (720p)	-5.51	-4.03	-4.13	-2.02	-2.2	-2.04	-2.81
Shuttlestart (720p)	-7.07	-9.23	-5.01	-1.79	-7.76	-6.66	-7.94
Tractor (1080p)	-11.3	-6.77	-4.49	-7.32	-2.96	-5.37	-4.5
HD Average	**-7.87**	**-6.1**	**-4.53**	**-3.88**	**-5.4**	**-5.49**	**-5.94**
AVERAGE	**-6.04**	**-4.26**	**-4.52**	**-1.88**	**-2.59**	**-2.91**	**-3**

resolution sequences. For instance, HDTV sequences have much better coding efficiency than QCIF or CIF sequences do, and two HDTV sequences have the highest bit-rate savings. Since our proposed scheme uses directional interpolation scheme with strong and efficient filters, we obtain good results for sequences, which contain rich texture details. For example, using our proposed scheme, the bit-rate saving of “Tractor” (1080p) sequence with rich texture details is 11%, compared to the H.264/AVC standard.

As shown in Table V, our proposed scheme outperforms the other schemes in terms of bit-rate savings. On average, bit-rate savings obtained by our proposed scheme are twice as that obtained by other schemes. Moreover, for low resolution sequences, our proposed scheme saves much more bit-rate than those in [4], [5], [6], [7], and [12] while our proposed scheme perform similarly the scheme in [8]. At the high resolutions, our proposed scheme is much better than those in [7] and [8].

From our experimental results in Table V, the performance of the separable AIF scheme [5] is similar to that of the two-dimensional non-separable scheme [4]. This is also confirmed in [14]. As mentioned in Section I, the adaptive-fixed scheme [7] is a simple modification of the directional AIF scheme [6]. That is the reason why the performance of adaptive-fixed scheme is slightly better than that of the directional AIF scheme.

Comparing our proposed scheme and other schemes to the H.264/AVC standard, we notice the significant improvement of our proposed scheme compared to the others. With the same coding conditions and the reference software, the bit-rate saving obtained by our proposed scheme is 6% while those obtained by other schemes are around 2-3% or less than that.

B. Coding Efficiency of Our Proposed Scheme vs. Other Schemes

In this subsection, we directly compare our proposed scheme to other related schemes. The direct comparison is the main difference from the previous comparison with the H.264/AVC. Table VI compares the coding efficiency of our proposed scheme to that of other schemes. We employ the same coding conditions, integrate all schemes into the reference software, and use the required VCEG test sequences and add some high resolution sequences. Through Table VI, we not only compare the bit-rate savings but also consider the differences among the peak signal-to-noise ratio (PSNR) gains by using the Bjontegaard tool.

Compared to the H.264/AVC standard scheme [1], as mentioned above, our proposed scheme performs better than other schemes in terms of bit-rate savings. In addition, our proposed scheme results in significant PSNR for various resolution sequences. On average, the PSNR gain for all test sequences is 0.27 dB.

TABLE VI
CODING EFFICIENCY OF OUR PROPOSED SCHEME COMPARED TO OTHER SCHEMES

Sequence	Proposed scheme vs. H.264/AVC scheme [1]		Proposed scheme vs. Low complex AIF scheme [8]		Proposed scheme vs. Adaptive-fixed scheme [7]	
	BD PSNR [dB]	BD bit-rate [%]	BD PSNR [dB]	BD bit-rate [%]	BD PSNR [dB]	BD bit-rate [%]
Foreman (QCIF)	0.29	-4.87	0	0.04	0.29	-5.04
Container (QCIF)	0.32	-4.7	0.02	-0.31	0.29	-4.27
Silent (QCIF)	0.26	-3.89	0	0	0.27	-4.09
QCIF Average	**0.29**	**-4.49**	**0.01**	**-0.09**	**0.28**	**-4.47**
Foreman (CIF)	0.23	-4.95	-0.01	0.16	0.21	-4.49
Mobile (CIF)	0.32	-5.11	0	0.02	0.29	-4.53
Paris (CIF)	0.25	-3.58	0	0.02	0.23	-3.33
Tempete (CIF)	0.22	-4.26	-0.02	0.27	0.23	-4.41
CIF Average	**0.26**	**-4.48**	**-0.01**	**0.12**	**0.24**	**-4.19**
Bigships (720p)	0.16	-4.61	0.03	-0.75	0.12	-3.65
City (720p)	0.32	-7.66	0.17	-4.11	0.16	-3.87
Crew (720p)	0.31	-11.07	0.15	-5.49	0.12	-4.2
Night (720p)	0.24	-5.51	0.06	-1.43	0.16	-3.56
Shuttlestart (720p)	0.19	-7.07	0.06	-2.19	0.15	-5.3
Rolling tomatoes (1080p)	0.04	-3.01	0.01	-1.89	0.03	-1.65
Sunflower (1080p)	0.46	-11.44	0.27	-6.89	0.15	-4.19
Toys and calendar (1080p)	0.2	-7.69	0.08	-2.92	0.11	-4.24
Tractor (1080p)	0.48	-11.3	0.3	-7.2	0.18	-4.3
HD Average	**0.27**	**-7.71**	**0.13**	**-3.65**	**0.13**	**-3.88**
AVERAGE	**0.27**	**-6.3**	**0.07**	**-2.04**	**0.19**	**-4.07**

Using the low complex AIF scheme [8] as a reference scheme, compared to the reference scheme, our proposed scheme performs equally well at the low resolutions but better at the high resolutions. Our proposed scheme has significant improvements in terms of both PSNR and bit-rates at the full HD. We confirm these results further by illustrating RD curves in the next subsection.

The PSNR gains and bit-rate savings show the improvements of our proposed scheme, compared to the adaptive-fixed scheme [7]. These improvements are less than those of our proposed scheme if we use H.264/AVC as a reference, but much better than those of ours if we use the low complex AIF scheme. On average, the coding efficiency of our proposed scheme compared to the adaptive-fixed scheme is 0.19 dB PSNR or 4.07% bit-rate savings.

Using the direct comparison between our proposed scheme and other schemes, our proposed scheme outperforms other schemes both in terms of PSNR values and bit-rate savings. Our proposed scheme is better than not only the H.264/AVC standard but also other schemes. In addition, from Section II, our proposed scheme is simpler than the directional AIF scheme [6]. As such, we can conclude that our proposed scheme is valuable in practice, since it is a simple and effective scheme.

C. Rate-distortion Curves

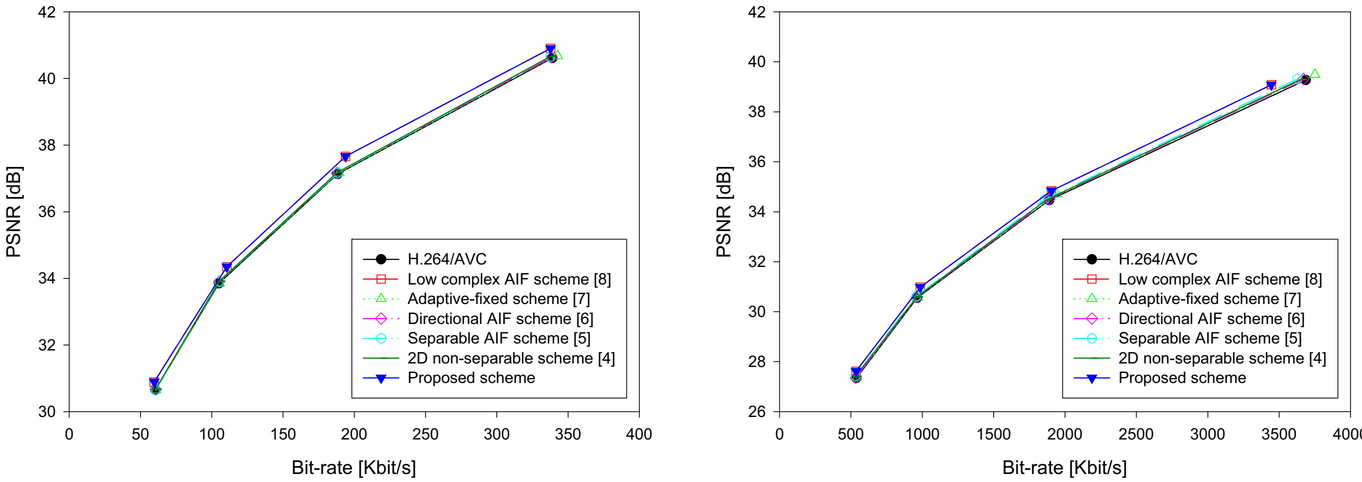
Table V illustrates the performance differences among various schemes. In this subsection, we determine these differences based on rate-distortion (RD) curves. Fig. 4 and Fig. 5 display the RD curves which clearly indicate the significant coding improvements of our proposed scheme over the other schemes. Fig. 4 illustrates the performance difference between our proposed scheme and the other schemes for the QCIF and CIF sequences. Fig. 5 shows the RD performances for HDTV sequences.

Compared to RD curves for low resolutions in Fig. 4, our proposed scheme performs slightly better than low complex AIF scheme [8] does. However, both schemes outperform the other schemes. Especially, the two above-mentioned schemes result in improvements in terms of PSNR for “Foreman” (QCIF) and “Mobile” (CIF) sequences, and “Mobile” (CIF) sequence has additional improvement in terms of bit-rate savings.

Compared to RD curves for the high resolutions in Fig. 5, the performance of our proposed scheme outperforms the other schemes in terms of bit-rate savings and PSNR. For HDTV sequences, at high bit-rate, both the bit-rate savings and PSNR gains from our proposed scheme are significant. For “Tractor” (1080p) sequence, the improvement obtained by our proposed scheme is especially considerable.

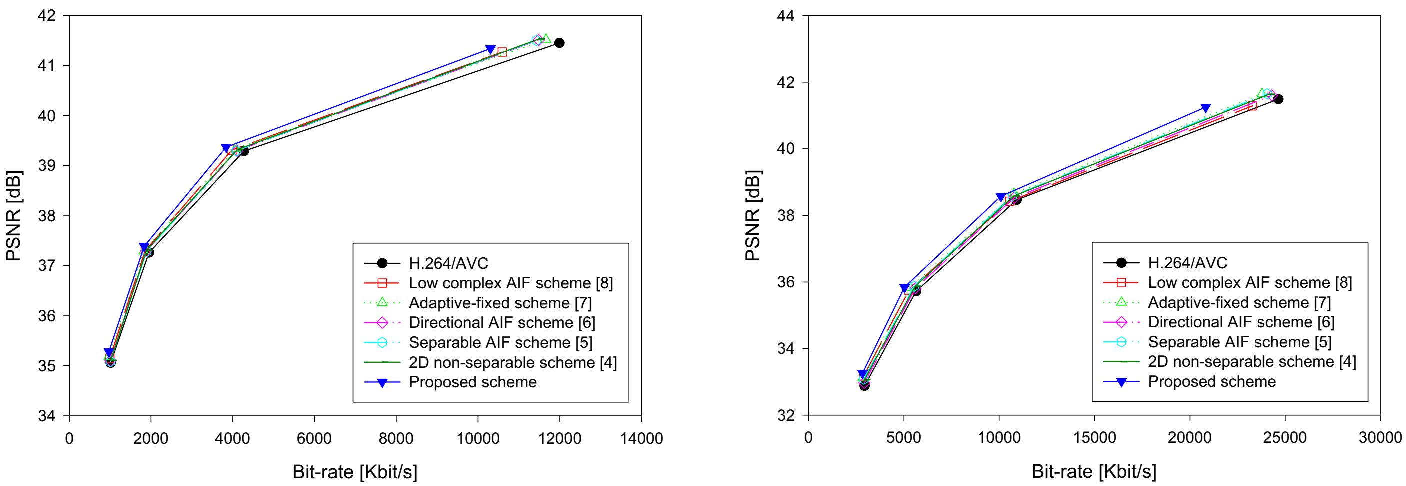
IV. CONCLUSION

In this paper, we proposed an efficient interpolation scheme using multi-directional filters. Our proposed scheme uses fixed filter coefficients with low complexity and low delay, strong filter position, and local DC offset. Our complexity analysis has illustrated that our proposed scheme is simpler than the other schemes. Our proposed scheme is also suitable for various textures of videos by using multi-directional filters. Experiment results demonstrated that the coding efficiency of our proposed scheme outperforms that of the other reference schemes including the well-known video coding H.264/AVC standard and the latest interpolation schemes.



(a) **(b)**

Fig. 4. RD curves for (a) Foreman (QCIF) and (b) Mobile (CIF).



(a) **(b)**

Fig. 5. RD curves for high resolution sequences: (a) Crew (720p) and (b) Tractor (1080p).

REFERENCES

[1] G.J. Sullivan, T. Wiegand, and H. Schwarz, "Editors' draft revision to ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding," JVT of ISO/IEC MPEG & ITU-T VCEG, JVT-AD205, Geneva, Switzerland, Feb. 2009.

[2] R. Wang, M. Li, J. Li, and Y. Zhang, "High throughput and low memory access sub-pixel interpolation architecture for H.264/AVC HDTV decoder," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 1006-1013, Aug. 2005.

[3] M. Kim, H. Lee, and S. Sull, "Spatial error concealment for H.264 using sequential directional interpolation," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1811-1818, Nov. 2008.

[4] Y. Vatis, B. Edler, D. N. Nguyen, and J. Ostermann, "Two-dimensional non-separable adaptive Wiener interpolation filter for H.264/AVC," ITU-T SG16/Q6, Z17, Busan, Korea, Apr. 2005.

[5] S. Wittmann and T. Wedi, "Separable adaptive interpolation filter," ITU-T SG16/Q6, C-0219, Geneva, Switzerland, July 2007.

[6] D. Rusanovskyy, K. Ugur, and J. Lainema, "Adaptive interpolation with directional filters," ITU-T SG16/Q6, AG21, Shenzhen, China, Oct. 2007.

[7] Z. Wu and Y. He, "Combined adaptive-fixed interpolation with multi-directional filters," ITU-T SG16/Q6, AI16, Berlin, Germany, July 2008.

[8] A. Fuldseth, G. Bjontegaard, D. Rusanovskyy, K. Ugur, and J. Lainema, "Low complextity directional interpolation filter," ITU-T SG16/Q6, AI12, Berlin, Germany, July 2008.

[9] Z. Wu and Y. He, "Combined adaptive-fixed interpolation with multi-directional filters," *Signal Processing: Image Communication*, vol. 24, no. 4, pp. 277-286, Apr. 2009.

[10] KTA software, Ver. JM11.0 KTA1.9 June 2008 [Online]. Available: http://iphome.hhi.de/suehring/tml/download/KTA/jm11.0kta1.9r1.zip

[11] T.K. Tan, G. Sullivan, and T. Wedi, "Recommended simulation common conditions for coding efficiency experiments," ITU-T SG16/Q6, AH10, Antalya, Turkey, Jan. 2008.

[12] Y. Vatis and J. Ostermann, "Adaptive interpolation filter for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 179-192, Apr. 2009.

[13] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T SG16/Q6, M33, Austin, USA, Apr. 2001.

[14] S. Wittmann and T. Wedi, "Separable adaptive interpolation filter for video coding," International Conference on Image Processing, pp. 2500-2503, Oct. 2008.

BIOGRAPHIES

Do Quan received the B.S. degree in Electrical Engineering at Hanoi University of Technology (HUT), Vietnam, in 2001, and M.S. degree in Electrical Engineering at Information and Communications University (ICU), now called "IT Convergence Campus" (ICC) of Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2007. From 2001 to 2004, he worked as a researcher at Vietnam Research Institute of Electronics, Informatics, and Automation (VIELINA), Vietnam. He is currently pursuing the Ph.D. degree at Gwangju Institute of Science and Technology (GIST), Korea. His current research interests include digital image and video coding, H.264/AVC, next generation video coding.

Yo-Sung Ho received both B.S. and M.S. degrees in electronic engineering from Seoul National University, Korea, in 1981 and 1983, respectively, and Ph.D. degree in Electrical and Computer Engineering from the University of California, Santa Barbara, in 1990. He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, New York, where he was involved in development of the advanced digital high-definition television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in development of the Korea direct broadcast satellite (DBS) digital television and high-definition television systems. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), where he is currently a professor in the Information and Communications Department. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, 3-D television, and realistic broadcasting.