Interpolation Scheme for Fractional-Pel Motion Compensation Using Multi-Directional Filters

Do Quan and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea
E-mails: {viequando, hoyo}@gist.ac.kr

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, the two most important criteria are computational complexity and coding efficiency. 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 both complexity and coding efficiency.

Keywords: Interpolation; Directional interpolation filter; H.264/AVC sub-pixel interpolation; Video coding; H.264/AVC.

1. INTRODUCTION

The latest 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 but also result in spatial error concealment. 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]. Due 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 [2]. 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 [2].

In order to simplify the implementation of the two-dimensional non-separable 6-tap adaptive interpolation scheme, a separable adaptive interpolation scheme was proposed [3]. Instead of using two-dimensional filter, these authors successively used 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 [4], a simple modification of this idea, named adaptive-fixed scheme [5], and a low complex AIF scheme [6], were proposed. These schemes use one-dimensional 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 [2], these schemes are still complicated. The process for minimizing the prediction error energy causes high delay. In addition, the rate-distortion (RD) performances of these schemes in [4] and [5] 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. Through analysis of interpolation process, filter coefficients, and complexity of our proposed interpolation scheme, we see 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.

ISBN: 978-602-8944-11-3

2. PROPOSED INTERPOLATION SCHEME USING 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.

2.1 Interpolation Process and Interpolation Filter Coefficients

In order to simplify the separable and non-separable adaptive interpolation filter, directional adaptive interpolation filter was proposed [4], [5]. 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 much iteration for each sub-pixel. We will reduce this high complexity by employing fixed filter coefficients.

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. 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*.

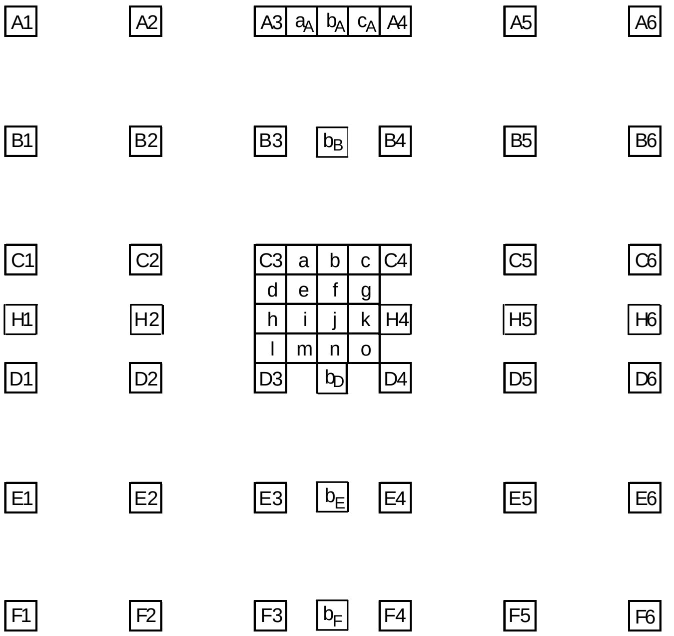


Fig. 1: Structure of different decompositions

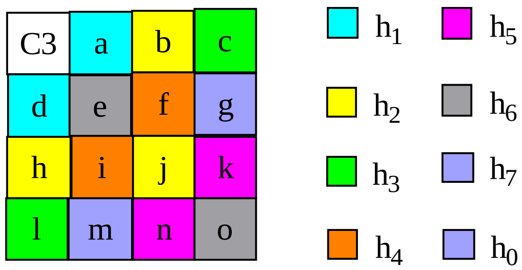


Fig. 2: Arrangement of filters for our interpolation process to obtain sub-pixels.

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*.

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

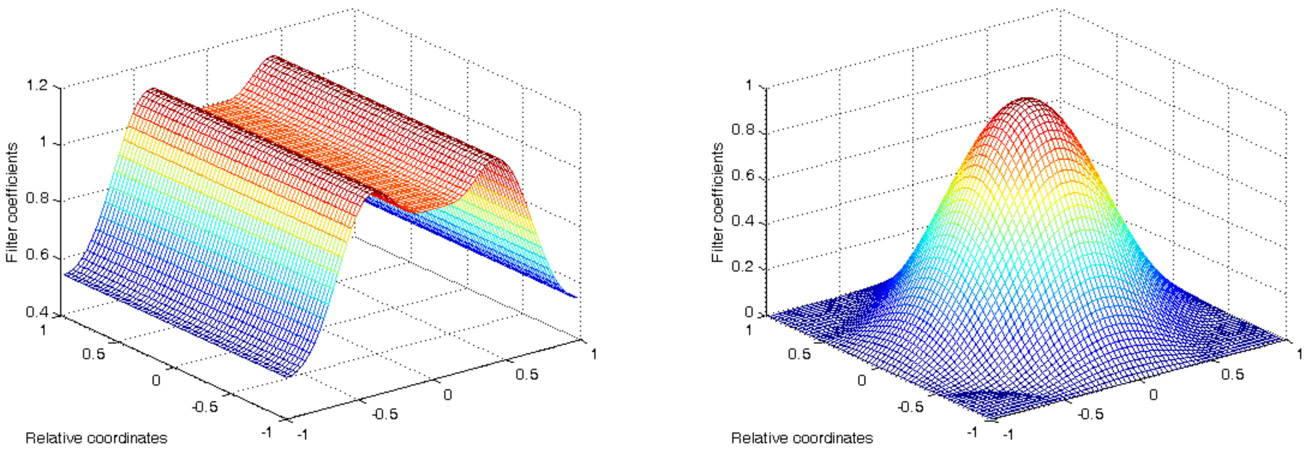
We use three basic filters which used in [6]. Filter coefficients of these filters were defined:

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

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

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

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,



(a) (b)

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

Table 1: Number of operations for each sub-pixel in various interpolation schemes

Sub-pixel	Proposed scheme	Adaptive-fixed scheme [5]	Directional AIF scheme [4]	Separable AIF scheme [3]	2D non-separable AIF scheme [2]
{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

respectively. From the three above-defined filters, we combine h_1, h_2, and h_3 to obtain 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 [6]. 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.

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.

Using fixed filter coefficients, the interpolation process is improved by adding fixed offsets [6]. 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.

2.2 Complexity Analysis

We calculate the number of operations to obtain each sub-pixel sample for 4×4 block in various interpolation schemes. 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 [6]. 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 Table 1. Table 1 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 1.

Based on the number of operations in various interpolation schemes in Table 1, the number of operations of our proposed interpolation scheme is similar to that of the directional AIF scheme [4] with our proposed scheme's resulting in a marginal complexity increment due to ours using a strong filter. We obtain the average and the greatest number of operations for each scheme. Both our and AIF interpolation schemes 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.

Table 2: Bit-rate savings in percentage of all schemes compared to the H.264/AVC standard

Sequence	Proposed scheme	AIF scheme [8]	Low complex AIF scheme [6]	Adaptive-fixed scheme [5]	Directional AIF scheme [4]	Separable AIF scheme [3]	2D non-separable scheme [2]
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**

3. 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 [7], 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 with high profile, number of reference frames: 4, search range: 32, GOP structure: IBBP, and QPISlice: 22, 27, 32, 37 and other QP values for P and B frames successively increase one value. Other conditions were defined in [9].

3.1 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 [10] with the same coding conditions, the KTA reference software, and all required VCEG test sequences. Table 2 shows percentage of bit-rate savings. 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.

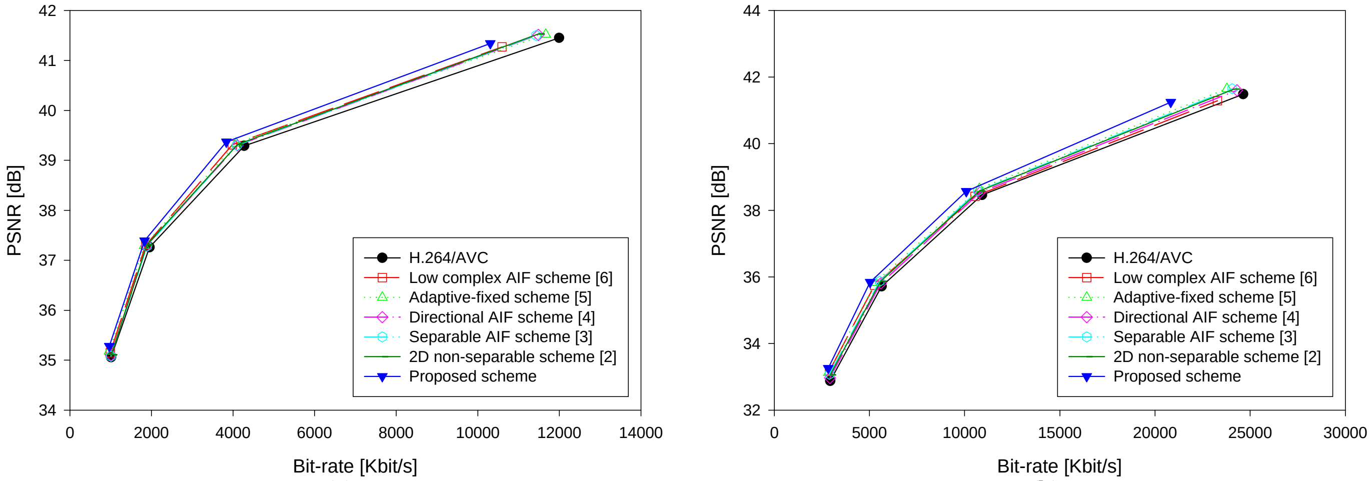
Our proposed scheme achieves 4.5%, 4.5% and 8% bit-rate savings on average for QCIF, CIF and HDTV sequences, respectively. Table 2 illustrates that our proposed scheme results in a stable coding efficiency improvement. Furthermore, it also illustrates that compared to other schemes; our proposed scheme 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 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 2, 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 [2], [3], [4], [5], and [8] while our proposed scheme perform similarly the scheme in [6]. At the high resolutions, our proposed scheme is much better than those in [5] and [6].

From our experimental results in Table 2, the performance of the separable AIF scheme [3] is similar to that of the two-dimensional non-separable scheme [2]. As mentioned in Section I, the adaptive-fixed scheme [5] is a simple modification of the directional AIF scheme [4]. 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 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%.



(a) (b)

Fig. 4: RD curves for high resolution sequences: (a) Crew (720p) and (b) Tractor (1080p).

3.2 Rate-distortion Curves

In this subsection, we determine these differences based on rate-distortion (RD) curves. Fig. 4 displays the RD curves which clearly indicate the significant coding improvements of our proposed scheme over the other schemes.

Compared to RD curves for the high resolutions in Fig. 4, 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.

3.3 Visual Quality Comparison

Fig. 5 and Fig. 6 compare output frame qualities of different schemes. We use “Crew” and “Tractor” sequences as representative sequences for this comparison. We observed the output frame 0 of “Crew” and “Tractor” sequences. From the comparison in Fig. 5 and Fig. 6, the output frame qualities of different schemes, such as H.264/AVC scheme, adaptive-fixed scheme [5], low complex AIF scheme [6], and proposed scheme, are quite similar in terms of visual quality. That is confirmed through PSNR values in Fig. 4.

4. CONCLUSIONS

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. In addition, it 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 H.264/AVC video coding standard and the latest interpolation schemes.

ACKNOWLEDGEMENTS

This work was supported by the IT R&D program of Ministry of Knowledge Economy/Institute for Information Technology Advancement, Republic of Korea under “Development of the Rich UCC Technology”

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, Feb. 2009.

[2] 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, April 2005.

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

[4] D. Rusanovskyy, K. Ugur, and J. Lainema, “Adaptive interpolation with directional filters,” ITU-T SG16/Q6, AG21, Oct. 2007.

[5] Z. Wu and Y. He, “Combined adaptive-fixed interpolation with multi-directional filters,” ITU-T SG16/Q6, AI16, July 2008.

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

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

[8] 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, April 2009.

[9] T.K Tan, G. Sullivan, and T. Wedi, “Recommended simulation common conditions for coding efficiency experiments,” ITU-T SG16/Q6, AH10, Jan. 2008.

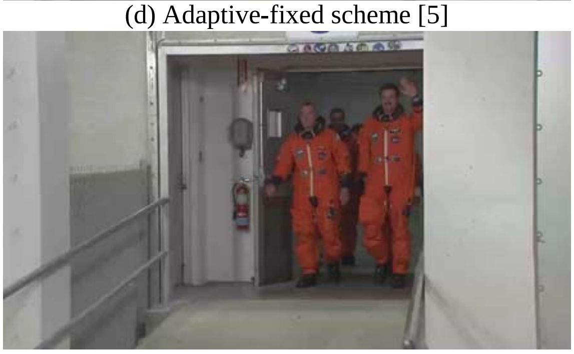
[10] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” ITU-T SG16/Q6, M33, April 2001.

(a) Original frame

(b) H.264/AVC

(c) Low complex AIF scheme [6]

(d) Adaptive-fixed scheme [5]



(e) Proposed scheme

Fig. 5: Visual quality comparison for frame 0 of “Crew” (720p) sequence with QP = 37.

(a) Original frame

(b) H.264/AVC

(c) Low complex AIF scheme [6]

(d) Adaptive-fixed scheme [5]

(e) Proposed scheme

Fig. 6: Visual quality comparison for frame 0 of “Tractor” (1080p) sequence with QP = 37.