A COMPARISON OF FRACTIONAL-PEL INTERPOLATION FILTERS IN HEVC AND H.264/AVC

Hao Lv[1,2], Ronggang Wang[1], Xiaodong Xie[2], Huizhu Jia[2] and Wen Gao[2]
[1]School of Computer & Information Engineering, Peking University Shenzhen Graduate School, Shenzhen, China
E-mail: hlv@pku.edu.cn, rgwang@pkusz.edu.cn
[2]National Engineering Laboratory of Video Technology, Peking University, Beijing, China
E-mail: donxie@pku.edu.cn, hzjia@pku.edu.cn, wgao@pku.edu.cn

ABSTRACT

The fractional-pel interpolation filter adopted in H.264/AVC improves motion compensation greatly. Recently, a new DCT-based fractional-pel interpolation filter is adopted in the oncoming standard HEVC. We are interested in the differences between these two types of fractional-pel interpolation filters. In this paper we describe the derivations of fractional-pel interpolation filters in HEVC and H.264/AVC in detail, and compare them on properties of frequency responses. We find that the half-pel interpolation filters in HEVC and H.264/AVC are very similar, but the low-pass properties of quarter-pel interpolation filters in HEVC are much better than those in H.264/AVC. Experimental results validate this phenomenon, the fractional-pel interpolation in H.264/AVC tends to increase BD-rates by more than 10% compared with that in HEVC, and this performance loss mainly comes from quarter-pel interpolation filters. On the other hand, the complexity of fractional-pel interpolation filtering in HEVC is greatly increased than that in H.264/AVC.

Index Terms**—**HEVC, H.264/AVC, fractional-pel, interpolation

1. INTRODUCTION

With the increasing popularity of high-definition video contents, video compression technology has received increased attention. Now, the Joint Collaborative Team on Video Coding (JCT-VC) is developing the next generation video coding standard, called High Efficiency Video Coding (HEVC). A number of new algorithmic tools are proposed, covering many aspects of video compression technology. Compared to AVC High Profile, the HEVC Working Draft 5 design reportedly provides bitrate savings at equal PSNR of about 23% for the high-efficiency all-intra configuration, 33% for the high-efficiency random-access configuration, and 41% for the high-efficiency low-delay configuration [1].

Motion compensation is the key factor for efficient video compression. Compensation for motion with fractional-pel accuracy requires interpolation of the reference pixels. H.264/AVC [2] uses the 6-tap FIR filter to perform half-pixel interpolation and the average filter to perform quarter-pixel interpolation for luma components. The DCTIF (DCT-based interpolation filter) was firstly used for motion compensation in Samsung's Call for Proposals (CfP) response [3]. Then the 12-tap DCTIF for HE (high efficiency) configuration and 6-tap directional interpolation filter for LC (low complexity) configuration for 1/4 luma sample were replaced by 8-tap DCTIF for both HE and LC configurations [4] in the HM2.0 because 8-tap DCTIF shows a good trade-off between complexity and performance for both HE and LC cases. In addition, the bilinear interpolation filter for 1/8 chroma sample was replaced by 4-tap DCTIF [5] in the HM2.0. HM5.0 adopts "7q(1/4)+8h" DCTIF for 1/4 luma sample which means 7-tap DCTIF is used for quarter-pel positions and 8-tap DCTIF is used for half-pel positions, because it can improve the compression efficiency for HD contents as well as all content sizes for LPHE (low delay P high efficiency) and LC cases if 7-tap interpolation filter is used for quarter-pel positions [6].

The rest of the paper is organized in five sections. In Section 2, we will present the derivation of the fractional-pel interpolation filter DCTIF in HEVC in detail. In Section 3, properties of frequency responses for the interpolation filters in HEVC and H.264/AVC are compared. The Section 4 will present the complexity comparison between the interpolation filtering in HEVC and H.264/AVC. In Section 5, the experimental results are given to show the performance gain of fractional-pel interpolation in HEVC over that in H.264/AVC. At last, we give a brief conclusion in Section 6.

2. LUMA INTERPOLATION METHOD IN HEVC

DCTIF is a 2D separable interpolation filter and easy to generate filter coefficients for arbitrary number of taps. The luma prediction values at fractional-pel positions are obtained by interpolation. Now HEVC supports quarter-pel motion vector accuracy for luma components.

2.1. DCTIF coefficients generation algorithm

DCT is one of the most popular transforms used in video signal processing applications, since DCT exhibits properties similar to the optimal Karhunen-Loeve Transform (KLT) [7]. The DCT-II used in image compression standard JPEG is defined by

$$X(k)=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}c_k x(n)\cos\left(\frac{(n+0.5)\pi k}{N}\right) \tag{1a}$$

$$x(n)=\sqrt{\frac{2}{N}}\sum_{k=0}^{N-1}c_k X(k)\cos\left(\frac{(n+0.5)\pi k}{N}\right) \tag{1b}$$

where

$$c_k=\begin{cases}1/\sqrt{2}\ , & k=0\\ 1\ , & otherwise\end{cases} \tag{2}$$

By substituting forward DCT-II in Eq.(1a) into inverse DCT-II in Eq.(1b), we can get the interpolation formula Eq.(3) [8].

$$x(i+p/L)=\sum_{m=0}^{N-1}x(m)\frac{2}{N}\sum_{k=0}^{N-1}\left(c_k^2\cos\left(\frac{(m+0.5)\pi k}{N}\right)\cos\left(\frac{(i+p/L+0.5)\pi k}{N}\right)\right) \tag{3}$$

2.1.1. Even taps DCTIF

When the number of taps in DCTIF is even, Eq.(4) can be derived from Eq.(3) by replacing N with 2M for integer i=0, 1,…, 2M-1.

$$x(i+\alpha)=\sum_{m=0}^{2M-1}x(m)\frac{1}{M}\sum_{k=0}^{2M-1}\left(c_k^2\cos\left(\frac{(m+0.5)\pi k}{2M}\right)\cos\left(\frac{(i+\alpha+0.5)\pi k}{2M}\right)\right) \tag{4}$$

Then, replace m with M-1+m and i with M-1+i to align the fractional-pel position $(i+\alpha)$ to that of p_α showed in Fig. 1. The values in integer positions p_l with l from –M+1 to M are known. To calculate p_α in fractional position α, we can use Eq.(5), by setting i to 0.

$$x(\alpha)=\sum_{m=-M+1}^{M}x(m)\frac{1}{M}\sum_{k=0}^{2M-1}\left(c_k^2\cos\left(\frac{(2m-1+2M)\pi k}{4M}\right)\cos\left(\frac{(2\alpha-1+2M)\pi k}{4M}\right)\right) \tag{5}$$

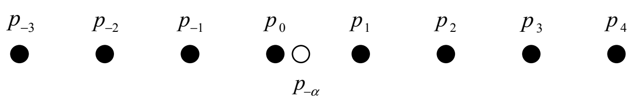


Fig. 1. Known integer-pel values (black) and predicted fractional-pel value (white) for N=8.

A fractional-pel value can be calculated by using the corresponding shift α. For example, we can get half-pel value by using Eq.(5) with α=1/2. The formula for even taps DCTIF coefficients generation is showed in Eq.(6), which is the product of forward and inverse transform DCT formulas mentioned above.

$$f_m(\alpha)=\frac{1}{M}\sum_{k=0}^{2M-1}\left(c_k^2\cos\left(\frac{(2m-1+2M)\pi k}{4M}\right)\cos\left(\frac{(2\alpha-1+2M)\pi k}{4M}\right)\right) \tag{6}$$

To smooth the properties of DCTIF, a smoothing-window [9] is added to Eq.(6), resulting in Eq.(7). Here N is the smoothing-window size, controlling the smoothing properties of windowing function. So, the filter coefficients of even taps DCTIF can be calculated by Eq.(7).

$$filter_m(\alpha)=\frac{1}{M}\cos\left(\pi\frac{m-\alpha}{N-1}\right)\sum_{k=0}^{2M-1}\left(c_k^2\cos\left(\frac{(2m-1+2M)\pi k}{4M}\right)\cos\left(\frac{(2\alpha-1+2M)\pi k}{4M}\right)\right) \tag{7}$$

2.1.2. Odd taps DCTIF

When the number of taps in DCTIF is odd, Eq.(8) can be derived from Eq.(3) by replacing N with 2M+1 for integer i=0, 1, …, 2M.

$$x(i+\alpha)=\sum_{m=0}^{2M}x(m)\frac{2}{2M+1}\sum_{k=0}^{2M}\left(c_k^2\cos\left(\frac{(m+0.5)\pi k}{2M+1}\right)\cos\left(\frac{(i+\alpha+0.5)\pi k}{2M+1}\right)\right) \tag{8}$$

Replace m with M+m and i with M+i to align the fractional-pel position $(i+\alpha)$ to that of p_α showed in Fig. 2. We can use Eq.(9) to calculate p_α in fractional position α, by setting i to 0.

$$x(\alpha)=\sum_{m=-M}^{M}x(m)\frac{2}{2M+1}\sum_{k=0}^{2M}\left(c_k^2\cos\left(\frac{(2m+1+2M)\pi k}{2(2M+1)}\right)\cos\left(\frac{(2\alpha+1+2M)\pi k}{2(2M+1)}\right)\right) \tag{9}$$

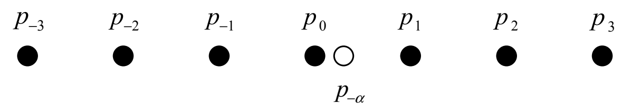


Fig. 2. Known integer-pel values (black) and predicted fractional-pel value (white) for N=7.

The formula for odd taps DCTIF coefficients generation is showed in Eq.(10).

$$f_m(\alpha)=\frac{2}{2M+1}\sum_{k=0}^{2M}\left(c_k^2\cos\left(\frac{(2m+1+2M)\pi k}{2(2M+1)}\right)\cos\left(\frac{(2\alpha+1+2M)\pi k}{2(2M+1)}\right)\right) \tag{10}$$

A smoothing-window is also added to Eq.(10), and the filter coefficients of odd taps DCTIF can be calculated by Eq.(11).

$$filter_m(\alpha)=\frac{2}{2M+1}\cos\left(\pi\frac{m-\alpha}{N-1}\right)\sum_{k=0}^{2M}\left(c_k^2\cos\left(\frac{(2m+1+2M)\pi k}{2(2M+1)}\right)\cos\left(\frac{(2\alpha+1+2M)\pi k}{2(2M+1)}\right)\right) \tag{11}$$

According to Eq.(7) and Eq.(11), we can derive that the filter coefficients for fractional positions $\alpha<1/2$ and the coefficients for $\alpha>1/2$ are mirror symmetry.

The interpolation filter coefficients calculated by Eq.(7) and (11) are real (not integer) numbers. In practical application, these filter coefficients are scaled by some factor (2^s is preferable) and rounded to integer, as described in Eq.(12). Here s represents the accuracy for filter coefficients.

$$Filter_m(\alpha)=\mathrm{int}\left(filter_m(\alpha)\times 2^s\right) \tag{12}$$

After scaling, the coefficients should keep the normalization condition as Eq.(13), which means the sum of filter coefficients is equal to 2^s. To keep the normalization some filter coefficients need to be adjusted by ±1.

$$\sum Filter_m(\alpha)=2^s \tag{13}$$

DCTIF can be derived to be any taps. It is clear that filters with more taps tend to provide more accurate interpolation results, but the interpolation complexity will be increased. According to JCTVC-D344, DCTIF8 (6 bits), namely s=6, shows a good trade-off between complexity and performance for both HE and LC cases, so it is adopted by new HEVC test model. Table 1 shows the "7q(1/4)+8h" DCTIF (6 bits) coefficients for luma interpolation in HM5.0 [10] and HM6.0 [11].

Table 1. the DCTIF coefficients for luma interpolation

Position	Filter coefficients
1/4	{ -1, 4, -10, 58, 17, -5, 1, 0 }
2/4	{ -1, 4, -11, 40, 40, -11, 4, -1 }
3/4	{ 0, 1, -5, 17, 58, -10, 4, -1 }

2.2. Luma interpolation process

Interpolation filter is applied in motion compensation for fractional position values generation. Current motion vector accuracy for luma components in HEVC is still quarter-pel, so 15 fractional-pel pixels will be interpolated showed in Fig. 3.

For fractional positions a, b and c, horizontal 1D filter is used. For fractional positions d, h and n, vertical 1D filter is used. For remaining positions, first horizontal 1D filter is applied for extended block and then vertical 1D filter is used. The block extension is 2M-1 (2M is the number of filter taps). For example, The fractional-pel pixels labeled $a_{0,0}$, $b_{0,0}$ and $c_{0,0}$ shall be derived by applying the 8-tap filter in horizontal direction to the adjacent integer pixels as described by Eq.(14a)~(14c). And the fractional-pel pixels labeled $d_{0,0}$, $h_{0,0}$ and $n_{0,0}$ shall be derived by applying the 8-tap filter in vertical direction.

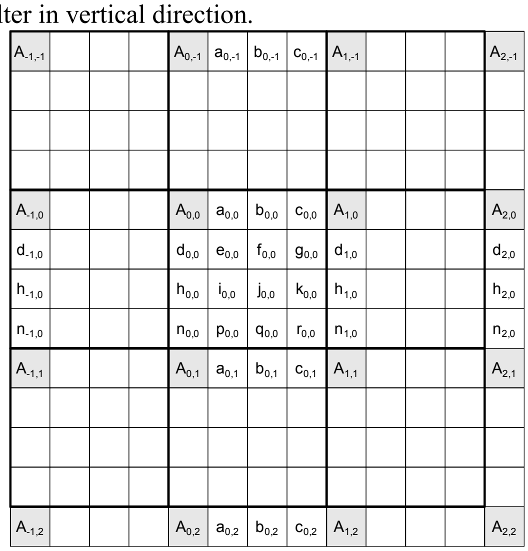


Fig. 3. Integer pixels (shaded blocks with upper-case letters) and fractional-pel pixels (un-shaded blocks with lower-case letters) for quarter-pel luma interpolation [11].

$$a_{0,0} = \left(-A_{-3,0} + 4\times A_{-2,0} - 10\times A_{-1,0} + 58\times A_{0,0} + 17\times A_{1,0} - 5\times A_{2,0} + A_{3,0}\right) >> shift1 \quad (14a)$$

$$b_{0,0} = \left(-A_{-3,0} + 4\times A_{-2,0} - 11\times A_{-1,0} + 40\times A_{0,0} + 40\times A_{1,0} - 11\times A_{2,0} + 4\times A_{3,0} - A_{4,0}\right) >> shift1 \quad (14b)$$

$$c_{0,0} = \left(A_{-2,0} - 5\times A_{-1,0} + 17\times A_{0,0} + 58\times A_{1,0} - 10\times A_{2,0} + 4\times A_{3,0} - A_{4,0}\right) >> shift1 \quad (14c)$$

The fractional-pel pixels labeled $e_{0,0}$, $i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $j_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$ and $r_{0,0}$ shall be derived by applying the 8-tap filter to the fractional-pel pixels $a_{0,i}$, $b_{0,i}$ and $c_{0,i}$ in vertical direction respectively, where $i = -3, \ldots, 4$. Eq.(15a)~(15c) show the interpolation formulas of the fractional-pel pixels $e_{0,0}$, $i_{0,0}$ and $p_{0,0}$.

$$e_{0,0} = \left(-a_{0,-3} + 4\times a_{0,-2} - 10\times a_{0,-1} + 58\times a_{0,0} + 17\times a_{0,1} - 5\times a_{0,2} + a_{0,3}\right) >> shift2 \quad (15a)$$

$$i_{0,0} = \left(-a_{0,-3} + 4\times a_{0,-2} - 11\times a_{0,-1} + 40\times a_{0,0} + 40\times a_{0,1} - 11\times a_{0,2} + 4\times a_{0,3} - a_{0,4}\right) >> shift2 \quad (15b)$$

$$p_{0,0} = \left(a_{0,-2} - 5\times a_{0,-1} + 17\times a_{0,0} + 58\times a_{0,1} - 10\times a_{0,2} + 4\times a_{0,3} - a_{0,4}\right) >> shift2 \quad (15c)$$

3. FREQUENCY PROPERTIES COMPARISON OF INTERPOLATION FILTERS IN H.264/AVC AND HEVC

We can get the frequency response curves of interpolation filters using MATLAB. And in this section the magnitude responses of the filters in HEVC and H.264/AVC will be analysed and compared.

3.1. Half-pel interpolation filter

The coefficients for half-pel luma interpolation are {2, -10, 40, 40, -10, 2} in H.264/AVC, {-1, 4, -11, 40, 40, -11, 4, -1} in HEVC (DCTIF8), and {2, -9, 39, 39, -9, 2} in DCTIF6.

The filter coefficients in H.264/AVC can be derived from the Lanczos filter as described in Eq.(16). The Lanczos filter can be obtained by adding a Lanczos window to the function *sinc*, which is the function of ideal low pass filter in time domain.

$$L_n(x) = \begin{cases} \mathrm{sinc}(x)\mathrm{sinc}(\frac{x}{n}) & , -n < x < n \\ 1 & , x = 0 \\ 0 & , otherwise \end{cases} \quad (16)$$

where

$$\mathrm{sinc}(x) = \frac{\sin \pi x}{\pi x} \quad (17)$$

When we set the parameter n of Lanczos filter to 3, and set the accuracy factor s for filter coefficients to 5, namely round the coefficients to integer by 2^5, the results are the filter coefficients for half-pel interpolation in H.264/AVC. If n and s are set to 3 and 6 respectively, the results are the same as the half-pel interpolation filter of DCTIF6. Similarly, if n and s are set to 4 and 6 respectively, the results are the same as the half-pel interpolation filter of DCTIF8. So, the differences of the half-pel interpolation filter between H.264/AVC and HEVC are the number of taps and the accuracy for filter coefficients.

Fig. 4 illustrates that DCTIF8 and DCTIF6 have much smaller ripples in the passband than H.264/AVC. And the transition band of DCTIF8 is much steeper than DCTIF6 and H.264/AVC.

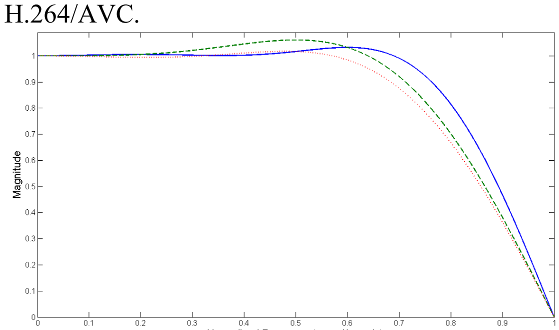
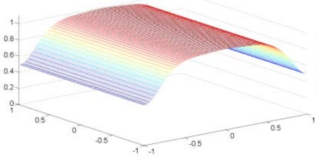


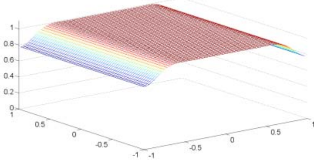
Fig. 4. Magnitude response curves of half-pel interpolation filters: DCTIF8 (solid), H.264/AVC (dashed) and DCTIF6 (dotted).

3.2. Fractional-pel interpolation filters

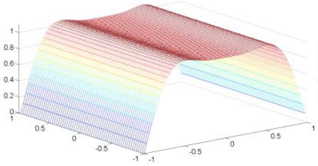
In fact, every fractional-pel interpolation filter can be generalized as a 2D filter. Fig. 5 shows all the magnitude responses of the filters for different fractional-pel positions in H.264/AVC and HEVC. The frequencies are normalized in the range -1.0 to 1.0, where 1.0 corresponds to half the sampling frequency or pi radians.



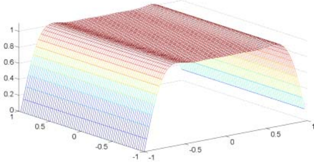
(a) a and c in H.264/AVC



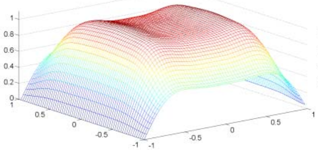
(b) a and c in HEVC



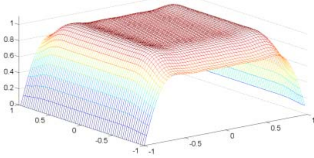
(c) b in H.264/AVC



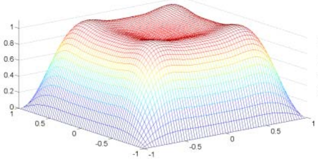
(d) b in HEVC



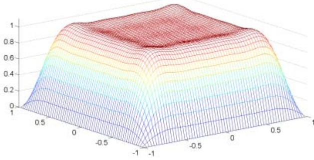
(e) f and q in H.264/AVC



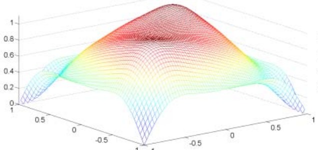
(f) f and q in HEVC



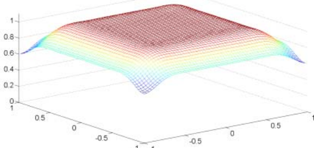
(g) j in H.264/AVC



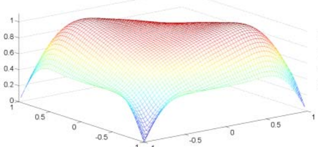
(h) j in HEVC



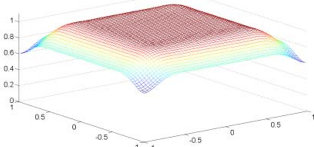
(i) e and r in H.264/AVC



(j) e and r in HEVC



(k) p and g in H.264/AVC



(l) p and g in HEVC

Fig. 5. Magnitude response curves of the interpolation filters in H.264/AVC and HEVC.

Due to the geometrical symmetry of filter coefficient matrices, the magnitude responses of some filters will be the same or symmetrical. Fig. 5(a), 5(c), 5(e), 5(g), 5(i) and 5(k) illustrate the magnitude responses of the filters in H.264/AVC. Specifically, the magnitude response of the interpolation filter for fractional-pel pixel a is the same as c, d is the same as n , f is the same as q, i is the same as k, e is the same as r and p is the same as g, respectively. Besides, d is symmetrical to a, h is symmetrical to b and i is symmetrical to f, respectively.

Fig. 5(b), 5(d), 5(f), 5(h), 5(j) and 5(l) illustrate the magnitude responses of the filters in HEVC. Also, the magnitude response of the interpolation filter for a is the same as c, d is the same as n, f is the same as q, i is the same as k and e is the same as p, g and r, respectively. Besides, d is symmetrical to a, h is symmetrical to b and i is symmetrical to f, respectively. As can be seen, the magnitude responses of interpolation filters in HEVC all have very square passbands and sharp transition bands, and are almost ideal for lowpass filtering.

By comparing the frequency responses of the filters in H.264/AVC and HEVC, obviously, in the passband the filters in HEVC are much flatter and have much smaller ripples than those in H.264/AVC. Especially for the fractional-pel pixels e, p, g and r in the diagonal direction, the responses for H.264/AVC are poor, however the responses for HEVC are evidently much better. To summarize, the frequency responses of quarter-pel filters in HEVC are obvious superior to those in H.264/AVC.

4. COMPLEXITY COMPARISON

Complexity is another important aspect for fractional-pel interpolation filter. In this section, we describe the complexity as memory accesses and arithmetic operations, and summarize the complexity of interpolating each fractional-pel pixels by the methods of H.264/AVC and HEVC as Table 2 respectively.

Table 2. Complexity analysis of fractional-pel interpolation filtering in H.264/AVC and HEVC

Pel	H.264/AVC			HEVC		
	Pixel accesses	Operations		Pixel accesses	Operations	
		mults	adds		mults	adds
A	1	0	0	1	0	0
a	6	6	6	7	7	6
b	6	6	5	8	8	7
c	6	6	6	7	7	6
d	6	6	6	7	7	6
e	6+5	6+6	5+5+1	7*7	7*7+7	7*6+6
f	6*6	6*6+6	6*5+6	7*8	7*8+7	7*7+6
g	6+5	6+6	5+5+1	7*7	7*7+7	7*6+6
h	6	6	5	8	8	7
i	6*6	6*6+6	6*5+6	8*7	8*7+8	8*6+7
j	6*6	6*6+6	6*5+5	8*8	8*8+8	8*7+7
k	6*6	6*6+6	6*5+6	8*7	8*7+8	8*6+7
n	6	6	6	7	7	6
p	6+5	6+6	5+5+1	7*7	7*7+7	7*6+6
q	6*6	6*6+6	6*5+6	7*8	7*8+7	7*7+6
r	6+5	6+6	5+5+1	7*7	7*7+7	7*6+6
avg	16.31	18.38	16.06	33.06	37.13	32.06

In H.264/AVC, the 6-tap FIR filter is used for half-pel interpolation, and the average filter is used for quarter-pel interpolation further. For example, to interpolate the fractional-pel pixel j, there are 6 pixel accesses, 6 multiplications and 5 additions in the first horizontal interpolation, and 6 fractional-pel pixel accesses, 6 multiplications and 5 additions in the second vertical interpolation, so the sum of pixel accesses, multiplications and additions are 6*6, 6*6+6 and 6*5+5, respectively. Thus, for the fractional-pel pixels i, k, f, and q, the sum of pixel accesses, multiplications and additions are 6*6, 6*6+6 and 6*5+6, respectively. Special attention is required that if the integer pixels are accessed in both the first and the second interpolation, we count the pixel accesses only once, but the multiplications and additions will be counted twice.

In HEVC, DCTIF is a 2D separable interpolation filter. For fractional positions a, b, c, d, h and n, 1D filter is used. For remaining positions, first horizontal 1D filter is applied and then vertical 1D filter is used. For example, to interpolate the fractional-pel pixel f, there are 8 pixel accesses, 8 multiplications and 7 additions in the first horizontal 1D interpolation, and 7 fractional-pel pixel accesses, 7 multiplications and 6 additions in the second vertical 1D interpolation, so the sum of pixel accesses, multiplications and additions are 7*8, 7*8+7 and 7*7+6, respectively.

The complexity analysis shows that the average numbers of pixel accesses, multiplications and additions of the interpolation in HEVC are almost twice as many as those in H.264/AVC, respectively. Table 2 gives a specific and theoretic calculation to illustrate a basic trend of the complexity of the interpolation from H.264/AVC to HEVC. And it will be a little different in terms of different platforms or implementation methods such as SIMD and hardware. For example, several words rather than individual pixels will be loaded from the memory every time the operation of reading data is executed when the interpolation is implemented by ARM NEON.

5. EXPERIMENTAL RESULTS

To compare the contribution to the compression performance by the interpolation filters in H.264/AVC and HEVC, various experiments were performed on the HEVC reference software HM5.2 [12] under the common test conditions defined in JCTVC-F900 [13], and HM5.2 is used as the anchor for all experiments. The configuration low delay P high efficiency (LPHE) is used, and only first picture is coded as I picture while the others are coded as P picture (IPPPPPPPPP…).

We have conducted four groups of experiments. In experiment 1, based on reference software HM5.2, the half-pel interpolation filter coefficients are changed to the coefficients in H.264/AVC, namely {2, -10, 40, 40, -10, 2}, and thus the fractional-pel pixels b, h, j, f, q, i and k are affected. In experiment 2, based on reference software HM5.2, the half-pel interpolation filter coefficients are changed to the coefficients of DCTIF6, namely {2, -9, 39, 39, -9, 2}. In experiment 3, in addition to the change in experiment 1, the interpolation methods of quarter-pel pixels in horizontal direction and in vertical direction are changed to those in H.264/AVC also. Thus the fractional-pel pixels b, h, j, a, c, d, n, f, q, i and k are affected. In experiment 4, besides the changes in experiment 3, the interpolation methods of remaining four quarter-pel pixels in the diagonal direction are changed to those in H.264/AVC, so all the interpolation methods of fractional-pel pixels are the same as H.264/AVC.

Tables 3~6 show the BD-rate increments for Y, U and V components for experiment 1~4 compared to the anchor HM5.2. Test results show that the average BD-rate increases on luma Y, chroma U and V are 1.0%, 0.6% and 0.7% in experiment 1, 1.1%, 0.4% and 0.9% in experiment 2, 7.0%, 5.4% and 6.1% in experiment 3, and 14.9%, 13.0% and 13.0% in experiment 4. Fig. 6 shows the distortion-rate curves for sequence BQSquare and BlowingBubbles.

Table 3. Results of experiment 1

Resolution	Sequence	BD-rates increase (%)		
		Y	U	V
416x240	BasketballPass	0.2	-0.2	0.2
	BQSquare	2.7	2.4	1.0
	BlowingBubbles	1.0	0.8	1.5
	RaceHorses	0.4	-0.4	0.2
	Average	1.0	0.6	0.7

Table 4. Results of experiment 2

Resolution	Sequence	BD-rates increase (%)		
		Y	U	V
416x240	BasketballPass	0.1	-0.3	0.0
	BQSquare	3.0	1.6	2.0
	BlowingBubbles	1.0	0.2	0.9
	RaceHorses	0.3	-0.1	0.6
	Average	1.1	0.4	0.9

Table 5. Results of experiment 3

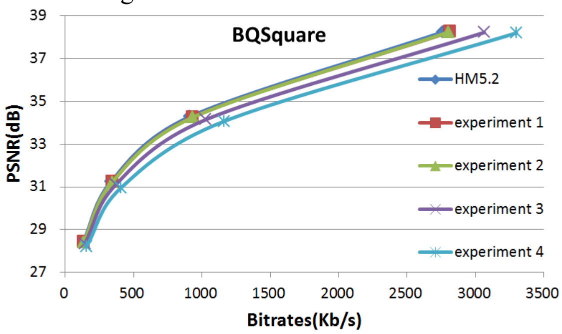
Resolution	Sequence	BD-rates increase (%)		
		Y	U	V
416x240	BasketballPass	3.0	2.5	2.3
	BQSquare	15.3	12.1	13.7
	BlowingBubbles	7.9	5.9	6.8
	RaceHorses	1.7	1.2	1.8
	Average	7.0	5.4	6.1

Table 6. Results of experiment 4

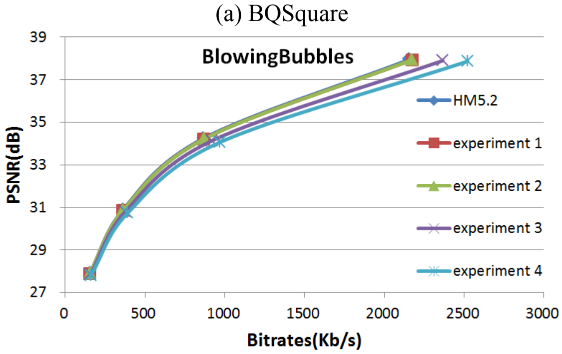
Resolution	Sequence	BD-rates increase (%)		
		Y	U	V
416x240	BasketballPass	6.1	5.9	5.5
	BQSquare	32.0	28.3	27.2
	BlowingBubbles	14.8	12.3	14.0
	RaceHorses	6.4	5.3	5.3
	Average	14.9	13.0	13.0

From results of experiment 1 and 2, we can see that there is almost no obvious difference in BD-rates, that is to say the influence on performance of the interpolation filter applied in half-pel interpolation either in H.264/AVC or DCTIF6 is relatively small. And the factor that impacts on performance is mainly the number of filter taps but the accuracy for filter coefficients.

By comparing the results of experiment 1, 3 and 4, we can draw a conclusion that the performance of the quarter-pel filters in H.264/AVC is relatively poor, especially the filters for the quarter-pel pixels e, g, p and r in the diagonal direction. About half of the performance losses come from the change of the filters (to those in H.264/AVC) for these four quarter-pel pixels. And the magnitude response properties in Fig. 5 tend to validate the conclusion.



(a) BQSquare



(b) BlowingBubbles

Fig. 6. Rate-PSNR curves for sequence BQSquare and BlowingBubbles.

6. CONCLUSION

Comprehensive comparison both in coding performance and complexity between the fractional-pel interpolation filters in HEVC and those in H.264/AVC is given by this paper. The evolution process of fractional-pel interpolation filter in HEVC is described in detail. We find that the frequency responses of quarter-pel interpolation filters in HEVC are much better than those in H.264/AVC. The experiment results testify this phenomenon, and almost all the performance gain (more than 10%) of interpolation filters in HEVC compared to H.264/AVC comes from the quarter-pel interpolations. However, the complexity of interpolation process in HEVC is much larger than that in H.264/AVC.

7. ACKNOWLEDGEMENTS

This work was partially supported by grants from the National Science & Technology Pillar Program 2011BAH08B03, the Chinese National Natural Science Foundation under contract No. 61171139 and No. 61035001, National Basic Research Program of China under contract No. 2009CB320907 and No. 2009CB320906, and Shenzhen Basic Research Program of JC201104210117A and JC201105170732A.

8. REFERENCES

[1] Bin Li, Gary J. Sullivan and Jizheng Xu, "Comparison of Compression Performance of HEVC Working Draft 5 with AVC High Profile", JCTVC-H0360, San José, CA, USA, Feb. 2012.

[2] ITU-T Recommendation and International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Oct. 2004.

[3] Ken McCann, Woo-Jin Han and Il-Koo Kim, "Samsung's Response to the Call for Proposals on Video Compression Technology", JCTVC-A124, Dresden, Germany, Apr. 2010.

[4] Elena Alshina, Jianle Chen, Alexander Alshin, Nikolay Shlyakhov and Woo-Jin Han, "CE3: Experimental results of DCTIF by Samsung", JCTVC-D344, Daegue, Korea, Jan. 2011.

[5] Jianle Chen, Elena Alshina and Woo-Jin Han, "CE4: Experimental results of DCTIF application for Chroma MC by Samsung", JCTVC-D347, Daegue, Korea, Jan. 2011.

[6] Elena Alshina, Alexander Alshin and JeongHoon Park, "CE3: 7 taps interpolation filters for quarter pel position MC from Samsung and Motorola Mobility", JCTVC-G778, Geneva, CH, Nov. 2011.

[7] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform", IEEE Trans. Computers, vol. C-23, pp. 90-93, Jan. 1974.

[8] Chien-Cheng Tseng and Su-Ling Lee, "Design of fractional delay FIR filter using discrete cosine transform", IEEE Asia Pacific Conference on Circuits and Systems, pp. 858-861, Nov. -Dec. 2008.

[9] A.Alshin and E.Alshina, "CE3: DCT derived interpolation filter test by Samsung", JCTVC-F247, Torino, Italy, Jul. 2011.

[10] Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan and Thomas Wiegand, "WD5: Working Draft 5 of High-Efficiency Video Coding", JCTVC-G1003, Geneva, CH, Nov. 2011.

[11] Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan and Thomas Wiegand, "High efficiency video coding (HEVC) text specification draft 6", JCTVC-H1003, San José, CA, USA, Feb. 2012.

[12] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[13] Frank Bossen, "Common test conditions and software reference configurations", JCTVC-F900, Torino, Italy, Jul. 2011.