A Low Complexity and High Performance Interpolation Filter for MPEG IVC

Hao Lv, Ronggang Wang*, Yangang Cai
School of Electronic and Computer Engineering,
Peking University Shenzhen Graduate School,
Shenzhen, China
{hlv, xiaoc}@pku.edu.cn, rgwang@pkusz.edu.cn

Huizhu Jia, Xiaodong Xie and Wen Gao
National Engineering Laboratory of Video Technology,
Peking University,
Beijing, China
{hzjia, donxie, wgao}@pku.edu.cn

Abstract*—Fractional-pel motion compensation is widely adopted in the modern video coding standards such as H.264/AVC, AVS, and HEVC etc. The interpolation filter is a critical factor that influences the coding efficiency. In this paper, a generation algorithm of interpolation filter coefficients is utilized. Based on the coefficients generation algorithm, three different tap filters, namely 6 tap, 8 tap, 10tap, are tested. A combination of 6 tap and 8 tap interpolation filters is proposed and proved to be optimal scheme considering both performance and computational complexity. And it is beneficial to make software optimization more effective, especially when SIMD-like (Single Instruction Multiple Data) operation is used. Experiments show that the average BD-rate gains on luma Y, chroma U and V are 8.01%, 5.08% and 4.98% for CS1 (Constraint set 1), and 9.21%, 7.53% and 7.63% for CS2 (Constraint set 2) in MPEG IVC reference software ITM5.0, compared with the traditional IVC interpolation method. The coding efficiency gains are significant for some video sequences and can reach up to 28.7%. With the merits of high performance and low complexity, our proposed method is formally adopted by MPEG IVC.

Keywords*—MPEG, IVC, fractional-pel, interpolation, motion estimation

I. INTRODUCTION

The Call for Proposals (CfP) for Internet Video Coding (IVC) [1-3] was approved in order to address the diversified needs of the Internet in the 97th MPEG meeting (July 2011). Then the MPEG Ad-hoc Group on Internet Video Coding was established by ISO/IEC JTC1/SC29/WG11 (informally known as the Moving Picture Experts Group, MPEG) at the 98th MPEG meeting (November 2011). And the aim is to attain the compression capability that substantially outperforms MPEG-2 and is comparable to AVC Baseline Profile with royalty-free tools. Now the MPEG IVC group is committed to video coding standardization work. The block-based hybrid video coding framework is still adopted, and various new algorithmic tools are proposed covering many aspects of video compression technology to enhance the coding performance. With the increasing popularity of internet video contents and abundant multimedia applications, such as web video, VOD (Video on Demand) in PC, smart TV, Pad and Smartphone, the video compression technology specially designed for internet is necessary and promising in future.

The two main parts of fractional-pel motion estimation are the search method and fractional-pel interpolation. The fractional-pel motion search method in video codec generally contains two steps: first, interpolate the half pixels around the initial integer point provided by integer-pel motion search and then find the optimal half pixel which has the minimum cost; second, execute the same operation to the half pixel to get the best quarter pixel. To get the cost, SAD is used for integer-pel motion search and SA(T)D is used for half-pel and quarter-pel motion search. However the fractional-pel interpolation methods vary for different video compression standards. H.264/AVC [4] uses the 6-tap FIR filter to perform half-pixel interpolation and the average filter to perform quarter-pixel interpolation further for luma components. Similarly, in Chinese video coding standard AVS [5], a 4-tap filter is used for half-pel interpolation and partial quarter-pel interpolation, and the average filter is furtherly used for the remaining four quarter-pel pixels in the diagonal direction. From HM5.0, HEVC uses "7q(1/4)+8h" DCTIF (DCT-based interpolation filter) 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 [6, 7]. And this paper is mainly to propose a new interpolation method for MPEG IVC with low complexity and high performance.

The rest of the paper is organized in five sections. In Section 2, we will present the derivation of the proposed fractional-pel interpolation filter in detail, and the luma interpolation process. Then a comprehensive analysis of the fractional-pel interpolation is given, including coding performance and computational complexity. In Section 3, experimental results corresponding to different tap filters are provided. Finally, we give a brief conclusion in Section 4.

II. FRACTIONAL MOTION ESTIMATION

A. Proposed interpolation method

In order to obtain motion vector, block matching algorithm (BMA) is performed in encoder. For fractional-pel motion full search, all the sub-pixels need to be interpolated by applying interpolation filter and searched for each reference frame and every block size partition. In MPEG IVC, 15 fractional-pel

*The corresponding author.

pixels are required to be interpolated, so the filter coefficients are very important and certainly affect the coding efficiency.

We proposed a new interpolation filter for MPEG IVC based on the Lanczos filter, which is 2D separable. 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. Here, x means the fractional position, and n means the filter tap. Generally the n is odd.

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

Where

$$\mathrm{sinc}(x)=\frac{\sin \pi x}{\pi x} \quad (2)$$

According to (1) and (2), we can derive that the filter coefficients for fractional positions $x<1/2$ and then the coefficients for $x>1/2$ are derived by mirror symmetry feature. The interpolation filter coefficients calculated by (1) and (2) 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 (3). Here s represents the accuracy of filter coefficients. After scaling, the coefficients should keep the normalization condition as (4).

$$Filter_n(x)=round\left(L_n(x)\times 2^s\right) \quad (3)$$

$$\sum Filter_n(x)=2^s \quad (4)$$

According to the above formulas, two main factors that influence the coefficients of interpolation filter are the accuracy of coefficients and the number of filter tap. Generally, we set the accuracy factor s to 6, because when s is more than 6, say 8, there is no obvious performance gain in our experiment. Furthermore, since the filter input bit-depth (the video source) can be 8 or 10, the intermediate buffer can be reduced to 16 bits and all data after each of the vertical and horizontal filtering passes holds in 16-bit memory when coefficient accuracy is set to 6 [8]. For some application it is important to store intermediate data within 16 bits. For example, it is actually beneficial that the restriction of an intermediate buffer size from 32 bits to 16 bits can make software optimization more effective, especially when SIMD-like (Single Instruction Multiple Data) operation is used. When implemented with ARM NEON instructions, interpolation filters with no more than eight taps can achieve sufficient data level parallelism and use the SIMD capability of NEON effectively. Table I~IV show the filter coefficients for different taps. Note that Table II is the coefficients of 8 tap filter without optimization and we only analyze and test the 8 tap filter based on coefficients in Table III.

Considering both the complexity and coding efficiency, a combination of 6 tap and 8 tap interpolation filters is adopted in MPEG IVC [3], and this two kinds of filter are described in Table 1 and 3 respectively. 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 tap). For example, The fractional-pel pixels labeled $a_{0,0}$, $b_{0,0}$ and $c_{0,0}$ in Fig. 1 shall be derived by applying the 8 tap filter in horizontal direction to the adjacent integer pixels as described by (5a)~(5c). 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, as showed by (6a)~(6c).

$$a_{0,0}=\left(-A_{-3,0}+4\times A_{-2,0}-10\times A_{-1,0}+57\times A_{0,0}+18\times A_{1,0}-6\times A_{2,0}+3\times A_{3,0}-A_{4,0}\right)>>shift1 \quad (5a)$$

$$b_{0,0}=\left(-A_{-3,0}+4\times A_{-2,0}-11\times A_{-1,0}+40\times A_{0,0}+40\times A_{1,0}-11\times A_{2,0}+4\times A_{3,0}-A_{4,0}\right)>>shift1 \quad (5b)$$

$$c_{0,0}=\left(-A_{-3,0}+3\times A_{-2,0}-6\times A_{-1,0}+18\times A_{0,0}+57\times A_{1,0}-10\times A_{2,0}+4\times A_{3,0}-A_{4,0}\right)>>shift1 \quad (5c)$$

$$d_{0,0}=\left(-A_{0,-3}+4\times A_{0,-2}-10\times A_{0,-1}+57\times A_{0,0}+18\times A_{0,1}-6\times A_{0,2}+3\times A_{0,3}-A_{0,4}\right)>>shift1 \quad (6a)$$

$$h_{0,0}=\left(-A_{0,-3}+4\times A_{0,-2}-11\times A_{0,-1}+40\times A_{0,0}+40\times A_{0,1}-11\times A_{0,2}+4\times A_{0,3}-A_{0,4}\right)>>shift1 \quad (6b)$$

$$n_{0,0}=\left(-A_{0,-3}+3\times A_{0,-2}-6\times A_{0,-1}+18\times A_{0,0}+57\times A_{0,1}-10\times A_{0,2}+4\times A_{0,3}-A_{0,4}\right)>>shift1 \quad (6c)$$

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 6 tap filter to the fractional-pel pixels $a_{0,i}$, $b_{0,i}$ and $c_{0,i}$ in vertical direction respectively, where $i=-2$, ..., 3. Equation (7a)~(7c), (8a)~(8c) and (9a)~(9c) show the interpolation formulas of the fractional-pel pixels $e_{0,0}$, $i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $j_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$ and $r_{0,0}$.

$$e_{0,0}=\left(2\times a_{0,-2}-9\times a_{0,-1}+57\times a_{0,0}+17\times a_{0,1}-4\times a_{0,2}+a_{0,3}\right)>>shift2 \quad (7a)$$

$$i_{0,0}=\left(2\times a_{0,-2}-9\times a_{0,-1}+39\times a_{0,0}+39\times a_{0,1}-9\times a_{0,2}+2\times a_{0,3}\right)>>shift2 \quad (7b)$$

$$p_{0,0}=\left(a_{0,-2}-4\times a_{0,-1}+17\times a_{0,0}+57\times a_{0,1}-9\times a_{0,2}+2\times a_{0,3}\right)>>shift2 \quad (7c)$$

$$f_{0,0}=\left(2\times b_{0,-2}-9\times b_{0,-1}+57\times b_{0,0}+17\times b_{0,1}-4\times b_{0,2}+b_{0,3}\right)>>shift2 \quad (8a)$$

$$j_{0,0}=\left(2\times b_{0,-2}-9\times b_{0,-1}+39\times b_{0,0}+39\times b_{0,1}-9\times b_{0,2}+2\times b_{0,3}\right)>>shift2 \quad (8b)$$

$$q_{0,0}=\left(b_{0,-2}-4\times b_{0,-1}+17\times b_{0,0}+57\times b_{0,1}-9\times b_{0,2}+2\times b_{0,3}\right)>>shift2 \quad (8c)$$

$$g_{0,0}=\left(2\times c_{0,-2}-9\times c_{0,-1}+57\times c_{0,0}+17\times c_{0,1}-4\times c_{0,2}+c_{0,3}\right)>>shift2 \quad (9a)$$

$$k_{0,0}=\left(2\times c_{0,-2}-9\times c_{0,-1}+39\times c_{0,0}+39\times c_{0,1}-9\times c_{0,2}+2\times c_{0,3}\right)>>shift2 \quad (9b)$$

$$r_{0,0}=\left(c_{0,-2}-4\times c_{0,-1}+17\times c_{0,0}+57\times c_{0,1}-9\times c_{0,2}+2\times c_{0,3}\right)>>shift2 \quad (9c)$$

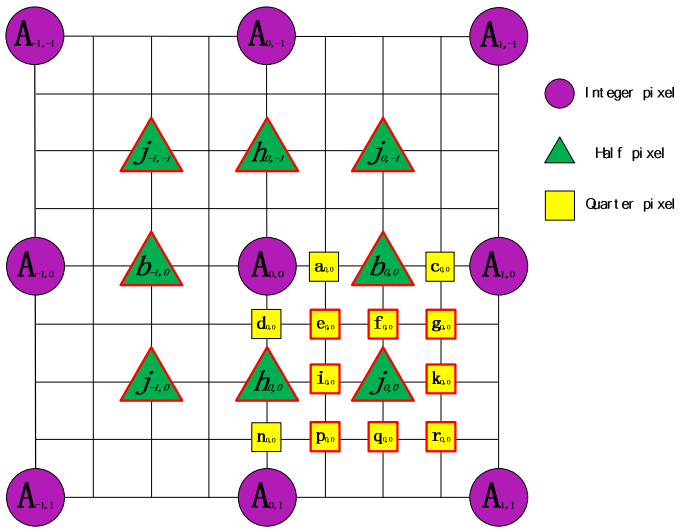


Figure 1. Fractional-pel interpolation: integer pixels (upper-case letters) and fractional-pel pixels (lower-case letters).

TABLE I. COEFFICIENTS OF 6 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{2, -9, 57, 17, -4, 1}	5	5
2/4	{2, -9, 39, 39, -9, 2}	6	5
3/4	{1, -4, 17, 57, -9, 2}	5	5

TABLE II. COEFFICIENTS OF 8 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{-1, 4, -10, 57, 18, -6, 2, 0}	6	6
2/4	{-1, 4, -11, 40, 40, -11, 4, -1}	6	7
3/4	{0, 2, -6, 18, 57, -10, 4, -1}	6	6

TABLE III. OPTIMIZED COEFFICIENTS OF 8 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{-1, 4, -10, 57, 18, -6, 3, -1}	6	7
2/4	{-1, 4, -11, 40, 40, -11, 4, -1}	6	7
3/4	{-1, 3, -6, 18, 57, -10, 4, -1}	6	7

TABLE IV. COEFFICIENTS OF 10 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{1, -2, 4, -10, 57, 19, -7, 3, -1, 0}	7	8
2/4	{1, -2, 5, -12, 40, 40, -12, 5, -2, 1}	8	9
3/4	{0, -1, 3, -7, 19, 57, -10, 4, -2, 1}	7	8

B. Complexity comparison of different fractional-pel interpolation schemes

Computation complexity is another important aspect for fractional-pel interpolation. We describe the complexity as memory accesses and arithmetic operations, and summarize the complexity of interpolating each fractional-pel pixel corresponding to different method. The analysis is necessary and beneficial to SIMD optimization and hardware optimization.

In fact, a block is regarded as a unit to be processed in terms of different platforms or implementation methods such as SIMD and hardware. For example, several words rather than individual pixels tend to be loaded from the memory each time the operation of reading data is executed when the interpolation process is implemented by ARM NEON instructions. Table V~ VII give a specific and theoretic calculation to illustrate the complexity of the interpolation including pixel accesses and arithmetical operations. The ratio of maximal pixel accesses of 8 tap interpolation to those of 6 tap interpolation is 1.5 at most and will decrease when the block becomes bigger. The maximal pixel accesses of proposed method are the same as 8 tap filter. For 10 tap interpolation, the ratio can be more than 2. For the arithmetical operations, the average numbers of multiplications and additions of 8 tap filter are about 1.5 times as many as those of 6 tap filter, respectively. For 10 tap filter the ratio is more than 2, however the proposed method have almost the same complexity as 6 tap filter.

TABLE V. MAXIMAL PIXEL ACCESSES ANALYSIS

Max	6tap (s=6)	8tap (s=6)	10tap (s=6)
	(L+5)×(W+5)	(L+7)×(W+7)	(L+9)×(W+9)
W=4,L=4	81	121	169
W=8,L=8	169	225	289
W=16,L=16	441	529	625
W=32,L=32	1369	1521	1681
W=64,L=64	4761	5041	5329

TABLE VI. MULTIPLICATION OPERATIONS ANALYSIS

Pel	Mults(×W×L)			
	6tap (s=6)	8tap (s=6)	10tap (s=6)	Proposed (s=6)
A	0	0	0	0
a	5	6	7	6
b	6	6	8	6
c	5	6	7	6
d	5	6	7	6
e	5×6+5	6×8+6	7×10+7	6×6+5
f	6×6+5	6×8+6	8×10+7	6×6+5
g	5×6+5	6×8+6	7×10+7	6×6+5
h	6	6	8	6
i	5×6+6	6×8+6	7×10+8	6×6+6
j	6×6+6	6×8+6	8×10+8	6×6+6
k	5×6+6	6×8+6	7×10+8	6×6+6
n	5	6	7	6
p	5×6+5	6×8+6	7×10+7	6×6+5
q	6×6+5	6×8+6	8×10+7	6×6+5
r	5×6+5	6×8+6	7×10+7	6×6+5
Avg	23.0	32.625	48.125	25.50

TABLE VII. ADDITION OPERATIONS ANALYSIS

Pel	Adds(×W×L)			
	6tap (s=6)	8tap (s=6)	10tap (s=6)	Proposed (s=6)
A	0	0	0	0
a	5	7	8	7
b	5	7	9	7
c	5	7	8	7
d	5	7	8	7
e	5×6+5	7×8+7	8×10+8	7×6+5
f	5×6+5	7×8+7	9×10+8	7×6+5
g	5×6+5	7×8+7	8×10+8	7×6+5
h	5	7	9	7
i	5×6+5	7×8+7	8×10+9	7×6+5
j	5×6+5	7×8+7	9×10+9	7×6+5
k	5×6+5	7×8+7	8×10+9	7×6+5
n	5	7	8	7
p	5×6+5	7×8+7	8×10+8	7×6+5
q	5×6+5	7×8+7	9×10+8	7×6+5
r	5×6+5	7×8+7	8×10+8	7×6+5
Avg	21.5626	38.0625	54.69	29.0625

III. EXPERIMENTAL RESULTS

The proposed method is implemented in the IVC reference software ITM5.0 [9]. Since ITM adopted a TSFT interpolation method [5] before our proposal, the TSFT method implemented in ITM5.0 is used as the anchor for all

experiments. The encoding configurations are based on the MPEG document N13354 [10], both Constraint set 1 (CS1) and Constraint set 2 (CS2) will be tested (QP=22, 27, 32, 37):

- CS1 is intended for enabling random access (<1.1 seconds), the encoder is consequently configured to make use of bi-prediction frames.
- CS2 is intended for low delay applications, with no reordering in the picture management. The encoder is therefore configured in IPPP mode, i.e. no B frame in use.

Several experiments corresponding to different tap filters are conducted, and Table VIII~X summarizes the BD-rate gains (%) for Y, U and V components. Four 832x480 sequences and four 416x240 are tested. Based on the results the average BD rate saving increases with the filter tap becomes bigger, and it is quite apparent by observing the sequences of PartyScene and BQSquare. Table XI gives the results when HEVC interpolation filter is implemented in ITM5.0, and the performance is a little poorer than that of 8 tap filter.

TABLE VIII. RESULTS OF 6 TAP INTERPOLATION FILTER IN ITM5.0

	CS1			CS2		
Sequence	Y	U	V	Y	U	V
BasketballDrill	-1.2	-0.8	-0.7	-1.7	-1.1	-1.4
BQMall	-3.4	-2.2	-2.3	-4.4	-3.9	-3.9
PartyScene	-13.4	-7.6	-7.9	-14.9	-11.7	-11.8
RaceHorses	0.2	-0.1	-0.3	-0.7	-1.0	-1.2
Average	-4.5	-2.6	-2.8	-5.4	-4.4	-4.6
BasketballPass	-0.5	0.0	-0.3	-0.6	-0.4	-0.6
BQSquare	-22.0	-15.8	-15.5	-22.1	-19.9	-19.7
BlowingBubbles	-9.1	-4.6	-4.7	-10.5	-7.7	-7.8
RaceHorses	-0.9	-0.6	-0.5	-2.2	-1.9	-2.0
Average	-8.1	-5.3	-5.3	-8.8	-7.5	-7.5
All	-6.29	-3.96	-4.03	-7.14	-6.0	-6.05

TABLE IX. RESULTS OF 8 TAP INTERPOLATION FILTER IN ITM5.0

	CS1			CS2		
Sequence	Y	U	V	Y	U	V
BasketballDrill	-1.6	-0.4	-0.5	-2.3	-0.9	-1.5
BQMall	-4.9	-3.0	-3.2	-6.6	-5.3	-5.6
PartyScene	-18.4	-10.5	-10.8	-20.7	-16.3	-16.2
RaceHorses	0.9	0.2	0.0	-0.5	-0.9	-1.1
Average	-6.0	-3.4	-3.6	-7.5	-5.9	-6.1
BasketballPass	-0.6	-0.2	-0.2	-0.7	-0.2	-0.5
BQSquare	-30.7	-22.1	-21.3	-32.5	-28.7	-28.7
BlowingBubbles	-12.6	-6.8	-6.6	-14.5	-10.4	-10.6
RaceHorses	-0.8	-0.5	-0.6	-2.6	-2.3	-2.4
Average	-11.2	-7.4	-7.2	-12.6	-10.4	-10.6
All	-8.59	-5.41	-5.40	-10.05	-8.13	-8.33

TABLE X. RESULTS OF 10 TAP INTERPOLATION FILTER IN ITM5.0

	CS1			CS2		
Sequence	Y	U	V	Y	U	V
BasketballDrill	-2.1	-0.4	-0.4	-3.4	-1.0	-1.3
BQMall	-4.8	-3.0	-3.2	-6.8	-5.6	-5.7
PartyScene	-20.0	-11.4	-11.6	-22.7	-17.4	-17.5
RaceHorses	1.4	0.4	0.2	0.0	-0.8	-0.9
Average	-6.4	-3.6	-3.7	-8.2	-6.2	-6.3
BasketballPass	-0.5	-0.3	-0.2	-0.8	-0.6	-0.6
BQSquare	-33.0	-23.7	-22.9	-35.2	-31.0	-30.7
BlowingBubbles	-13.5	-7.1	-7.2	-15.9	-11.2	-11.3
RaceHorses	-0.4	-0.3	-0.4	-2.6	-2.3	-2.2
Average	-11.9	-7.8	-7.7	-13.6	-11.3	-11.2
All	-9.11	-5.73	-5.71	-10.93	-8.74	-8.78

TABLE XI. RESULTS OF HEVC INTERPOLATION FILTER IN ITM5.0

	CS1			CS2		
Sequence	Y	U	V	Y	U	V
BasketballDrill	-2.0	-0.2	-0.2	-3.2	-0.8	-1.2
BQMall	-4.6	-2.7	-2.8	-6.7	-5.2	-5.3
PartyScene	-18.0	-9.9	-10.1	-20.3	-15.2	-15.2
RaceHorses	1.2	0.2	0.2	-0.3	-0.8	-1.0
Average	-5.8	-3.1	-3.2	-7.6	-5.5	-5.7
BasketballPass	-0.4	-0.1	-0.2	-0.6	-0.1	-0.5
BQSquare	-30.0	-21.2	-20.3	-31.5	-27.4	-27.1
BlowingBubbles	-12.1	-6.0	-6.2	-14.3	-10.0	-9.9
RaceHorses	-0.5	-0.3	-0.3	-2.9	-2.2	-2.3
Average	-10.8	-6.9	-6.7	-12.3	-9.9	-10.0
All	-8.30	-5.03	-4.99	-9.98	-7.71	-7.81

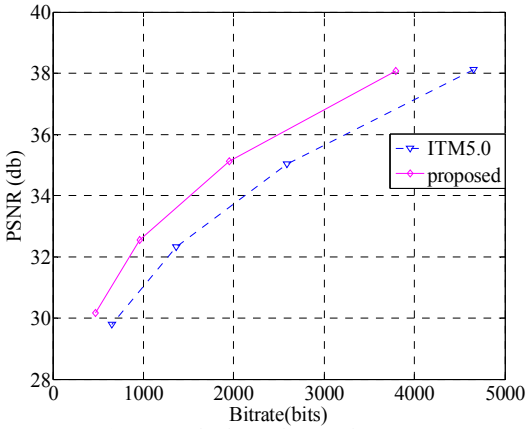
Table XII and XIII give the performance of proposed method implemented in ITM platform and JM platform. According to the results, the average BD-rate gains on luma Y, chroma U and V are 8.01%, 5.08% and 4.98% for CS1, and 9.21%, 7.53% and 7.63% for CS2 in ITM platform. Similarly, in JM platform (the number of reference frames is set to 2) the results are 4.53%, 3.63% and 3.64% for CS1, and 4.11%, 3.79% and 4.08% for CS2. For the four 416x240 sequences, the BD-rate saving can reach 10.4% for CS1 and 11.5% for CS2 in platform, and 6.3% for CS1 and 5.8% for CS2 in JM platform. Compared with the results in Table 8~10, the performance of proposed method is better than 6 tap filter and closes to 8 tap filter. To sum up, the proposed "8+6" filter is obviously better than the filters used in H.264/AVC, and can be comparable to HEVC which uses "7q+8h" filter. Fig. 2 shows the distortion-rate curves for sequence BQSquare when proposed scheme is implemented in ITM5.0 and JM.

TABLE XII. PROPOSED INTERPOLATION FILTER IN ITM5.0

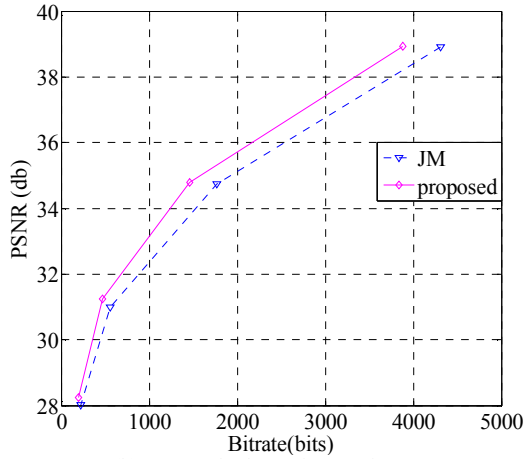
	CS1			CS2		
Sequence	Y	U	V	Y	U	V
BasketballDrill	-1.6	-0.7	-0.6	-2.3	-1.1	-1.5
BQMall	-4.4	-2.7	-2.7	-6.0	-5.1	-5.3
PartyScene	-17.1	-9.6	-9.7	-19.0	-14.9	-14.8
RaceHorses	0.4	0.0	-0.2	-0.6	-0.9	-1.3
Average	-5.6	-3.3	-3.3	-7.0	-5.5	-5.7
BasketballPass	-0.6	-0.2	-0.5	-0.8	-0.5	-0.7
BQSquare	-28.0	-20.4	-19.2	-28.7	-25.4	-25.3
BlowingBubbles	-12.0	-6.4	-6.5	-13.8	-10.1	-9.8
RaceHorses	-0.8	-0.6	-0.4	-2.5	-2.2	-2.3
Average	-10.4	-6.9	-6.7	-11.5	-9.6	-9.5
All	-8.01	-5.08	-4.98	-9.21	-7.53	-7.63

TABLE XIII. PROPOSED INTERPOLATION FILTER IN JM

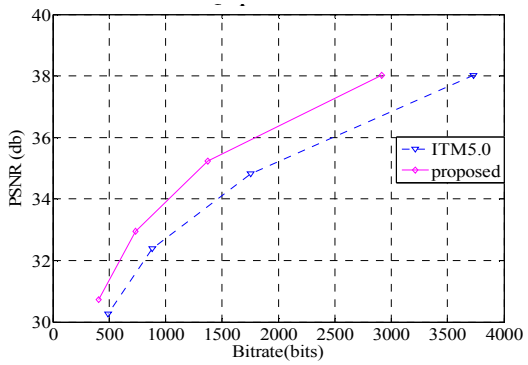
	CS1			CS2		
Sequence	Y	U	V	Y	U	V
BasketballDrill	-0.8	0.2	0.2	-0.1	0.5	0.2
BQMall	-1.0	-1.2	-1.4	-1.2	-1.7	-1.7
PartyScene	-10.3	-8.0	-7.9	-10.3	-9.5	-9.3
RaceHorses	1.2	0.9	0.6	2.1	0.6	0.1
Average	-2.7	-2.0	-2.1	-2.3	-2.5	-2.7
BasketballPass	-0.3	-0.5	0.0	0.9	0.2	-0.1
BQSquare	-18.3	-15.8	-15.5	-16.5	-13.5	-14.7
BlowingBubbles	-7.0	-4.5	-4.8	-7.8	-6.3	-6.6
RaceHorses	0.3	-0.1	-0.3	0.0	-0.6	-0.5
Average	-6.3	-5.2	-5.1	-5.8	-5.0	-5.5
All	-4.53	-3.63	-3.64	-4.11	-3.79	-4.08



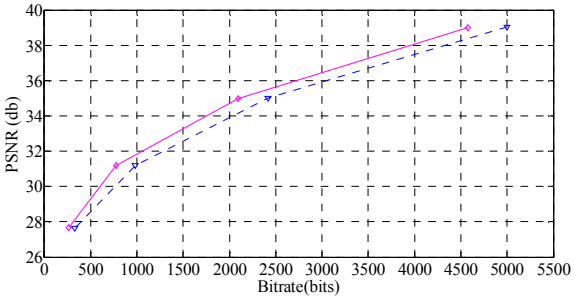
(a) Proposed scheme for CS1 in ITM5.0



(b) Proposed scheme for CS1 in JM



(c) Proposed scheme for CS2 in ITM5.0



(d) Proposed scheme for CS2 in JM

Figure 2. Distortion-rate curves for sequence BQSquare.

IV. CONCLUSION

The generation algorithm of interpolation filter coefficients for MPEG IVC is introduced in this paper. Different tap filters, namely 6 tap, 8 tap, 10tap, are tested and compared both in performance and computational complexity, and a combination of 6 tap and 8 tap interpolation filters is proved to be the optimal scheme for the low complexity and considerable coding efficiency. All the IVC test sequences with different resolutions are tested, and experimental results show that the average BD-rate gains on luma Y, chroma U and V are 8.01%, 5.08% and 4.98% for CS1, and 9.21%, 7.53% and 7.63% for CS2, compared to traditional MPEG IVC interpolation filter. Since the high performance of our proposed method, it is formally adopted in MPEG IVC ITM. Furthermore, it contributes to making software optimization more effective especially when SIMD-like operation is used.

ACKNOWLEDGMENT

This work is partially supported by the grants of National Science Foundation of China 61370115, Shenzhen Basic Research Program of JC201104210117A, JC201105170732A, and JCYJ2012061450301623, and grants from the Chinese National Natural Science Foundation under contract No.61171139 and No. 61035001, and National High Technology Research and Development Program of China (863 Program) under contract No.2012AA011703.

REFERENCES

[1] Call for Proposals (CfP) for Internet Video Coding Technologies, ISO/IEC JTC1/SC29/WG11 N12204, July 2011, Torino, Italy.

[2] Ronggang Wang, Xianguo Zhang, Hao Lv, etc, “RFM2.0 for internet video coding”, ISO/IEC JTC1/SC29/WG11/M26716, October 2012, Shanghai, China.

[3] Ronggang Wang, Xianguo Zhang, Siwei Ma, Jianwen Chen, Cliff Reader, “Internet Video Coding Test Model (ITM) Version 3.0”, ISO/IEC JTC1/SC29/WG11/N13162, October 2012, Shanghai, China.

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

[5] Ronggang Wang, Chao Huang, Jintao Li,Yanfei Shen, Sub-pixel Motion Compensation Interpolation Filter in AVS, IEEE International Conference on Multimedia and Expo (ICME), Jun. 2004.

[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] Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan, Ye-Kui Wang and Thomas Wiegand, High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent), JCTVC-L1003, Geneva, CH, Jan. 2013.

[8] Youngo Park, Sunyoung Jeon, Chanyul Kim, Kwang Pyo Choi, Unified design for motion compensation filter, JCTVC-F480, Torino, IT, July, 2011.

[9] Ronggang Wang, Lei Chen, Xufeng Li, Hao Lv, Siwei Ma, Tiejun Huang, Wen Gao, “IVC EE: Performance improvement of ITM5.0”, ISO/IEC JTC1/SC29/WG11/ M30187, July 2013, Vienna, AT.

[10] Video Subgroup, “IVC Core Experiment CE1: Overall Codec Testing”, ISO/IEC JTC1/SC29/WG11/N13354, January 2013, Geneva, CH.