Parametric Interpolation Filter for HD Video Coding

Jie Dong, *Member, IEEE,* and King Ngi Ngan, *Fellow, IEEE*

***Abstract*—Recently, adaptive interpolation filter (AIF) for motion-compensated prediction (MCP) has received increasing attention. This letter studies the existing AIF techniques, and points out that making tradeoff between the two conflicting aspects: the accuracy of coefficients and the size of side information, is the major obstacle to improving the performance of the AIF techniques that code the filter coefficients individually. To overcome this obstacle, parametric interpolation filter (PIF) is proposed for MCP, which represents interpolation filters by a function determined by five parameters instead of by individual coefficients. The function is designed based on the fact that high frequency energies of HD video source are mainly distributed along the vertical and horizontal directions; the parameters are calculated to minimize the energy of prediction error. The experimental results show that PIF outperforms the existing AIF techniques and approaches the efficiency of the optimal filter.**

***Index Terms*—Adaptive filter, H.264/AVC, interpolation, motion-compensated prediction, video coding.**

I. Introduction

Motion-compensated prediction (MCP) is the key to the success of the modern video coding standards. With MCP, the pixels to be coded are predicted from the temporally neighboring ones, and only the prediction errors and the motion vectors (MV) are transmitted. However, due to the finite sampling rate, the actual position of the prediction in the neighboring frames may be out of the sampling grid, where the intensity is unknown, so the intensities of the positions in between the integer pixels, called sub-positions, must be interpolated and the resolution of MV is increased accordingly.

In H.264/AVC [1], the interpolation filter is designed to fit the general statistics of various video sources, so the filter coefficients are fixed. For the resolution of MV is quarter-pixel, the reference frame is interpolated to be 16 times the size for MCP, 4 times both sides. As shown in Fig. 1(a), the interpolation defined in H.264 includes two stages, interpolating the half-pixel and quarter-pixel sub-positions, respectively.

Considering the time-varying statistics of video sources, some researchers propose adaptive interpolation filter (AIF), of which the coefficients are optimized and coded on a frame basis. Although using the common methodology to calculate the filter coefficients, the existing AIF techniques are distinguished from each other by the ways they reduce the side information, including reducing the support region, imposing the symmetry constraints, and coarsely quantizing the filter coefficients. AIF in [2] follows the process in Fig. 1(a), and only h_1 is modified. Owing to h_1's even symmetry, only three coefficients are analytically calculated and coded. Vatis *et al.* [3] developed a 2-D non-separable AIF, of which the interpolation process, shown in Fig.1(b), is the same as the optimal AIF. The spatial sampling rate is increased 16 times at one time, and each sub-position is interpolated directly by filtering the surrounding 6×6 integer pixels. h is circularly symmetric, because the spatial statistics are assumed to be isotropic. Directional AIF (D-AIF), proposed in [4], reduces the complexity, although also following the process in Fig. 1(b). Each sub-position is supported by at most 12 surrounding pixels in a form of diagonal cross. D-AIF is enhanced by the authors in [5], known as E-DAIF. In [5], the symmetry of the filter is flexible and the structure of the 12-tap filters is adaptively switched between diagonal cross and radial supports. The enhanced AIF (E-AIF) proposed in [6] uses 12-tap filters with a radial support to interpolate sub-positions, and adds a 5×5 filter for integer pixels. The horizontal and vertical statistical properties are thought different, so h in Fig. 1(b) is axisymmetric.

All coefficients in the existing AIF techniques are quantized to 512 levels, which are not enough to represent AIF coefficients. That is why the performances of the AIF techniques are very close to each other, whatever the size of the support region and the symmetry constraints. Nevertheless, the required bits for coding these coefficients are still significant especially at low bit-rates. Therefore, the conflict of the coefficients' precision and the size of side information is the major obstacle to improving the performance of the AIF techniques that code the coefficients individually.

To overcome this obstacle, parametric interpolation filter (PIF) is proposed for MCP, which represents interpolation filters by a function determined by five parameters instead of by individual coefficients. The function is suitable for HD video coding, because it is designed based on the fact that high frequency energies of HD video source are mainly distributed along the vertical and horizontal directions. The parameters are optimized on a frame basis in terms of minimum MCP error. With the parameters quantized to enough precision, the filter coefficients are calculated with negligible loss, but the side information reduces to exactly 68 bits per frame.

II. Optimal AIF and the Approximation Effects

As shown in Fig. 1(b), interpolation by definition comprises two steps: upsampling the frame to 16 times the size

Manuscript received November 2, 2009; revised April 20, 2010; accepted July 10, 2010. Date of publication October 14, 2010; date of current version January 22, 2011. This work was supported in part by a grant from the Chinese University of Hong Kong under the Focused Investment Scheme (Project 1903003). This paper was recommended by Associate Editor L. Chen.

The authors are with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: jdong@ee.cuhk.edu.hk; knngan@ee.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2010.2087530

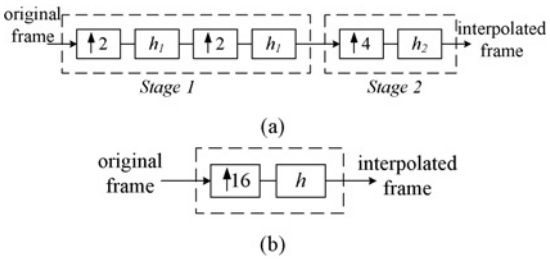

Fig. 1. Interpolation process of (a) filter in H.264/AVC and (b) optimal AIF.

by inserting zero-valued samples, which produces undesired spectra in the frequency domain, and removing the undesired spectra by a lowpass filter h. The optimal h, denoted as h_{opt}, should minimize the prediction error energy, given by

$$\sigma_e^2 = E\left[\left(\sum_{i,j} h(i,j)P_{16}(4x-i+d_x, 4y-j+d_y) - S(x,y)\right)^2\right] \quad (1)$$

where P_{16} is upsampled from the reference frame by a factor 16 using zero-insertion, S is the current frame to be coded, and d_x and d_y are the two components of MV. Letting $\partial\sigma_e^2/\partial h(m,n)$ equal to 0, one can derive the minimum σ_e^2 and the optimal interpolation filter h_{opt}; the solution converges to the Wiener-Hopf equations as in

$$\sum_{i,j} h_{opt}(i,j)R_{pp}(i-m, j-n) = R_{ps}(m,n) \quad (2)$$

where R_{pp} and R_{ps} represent the autocorrelation of P_{16} and the motion-compensated cross-correlation of P_{16} and S, respectively. R_{pp} and R_{ps} are calculated with all the MVs for the current frame known; therefore, motion estimation is performed before starting coding the current frame.

To be consistent with the interpolation in H.264/AVC, all the studies in this letter assume quarter-pixel MCP and each subposition is supported by the surrounding 6×6 integer pixels. Therefore the size of h in Fig. 1(b) is 23×23. If no symmetry constraint and quantization are imposed, h_{opt} obtained by (2) has 529 different real-valued coefficients, which are too expensive to be coded for each frame. The related work [2]–[6] makes effort to reduce the side information as introduced in Section I, including restricting the support region, imposing the symmetry constraints, and coarsely quantizing the coefficients, which lead to an approximation of h_{opt}, denoted as $\tilde{h}$, and larger prediction error compared with that produced by h_{opt}. The difference between h_{opt} and $\tilde{h}$ is denoted as h_Δ, equal to $h_{opt}-\tilde{h}$, and the increased energy of prediction error, introduced by h_Δ, is given in

$$\begin{aligned}\Delta err &= \sigma_e^2|_{\tilde{h}} - \sigma_e^2|_{h_{opt}} \\ &= \sum_{i,j}\sum_{m,n} h_\Delta(i,j)h_\Delta(m,n)R_{pp}(i-m, j-n) \\ &\quad -2\sum_{m,n} h_\Delta(m,n)\left(\sum_{i,j} h_{opt}(i,j)R_{pp}(i-m, j-n) - R_{ps}(m,n)\right).\end{aligned} \quad (3)$$

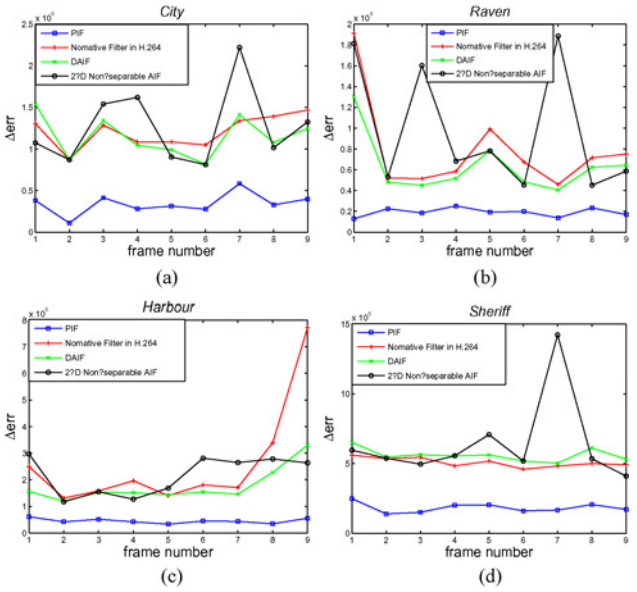

Fig. 2. Δerr introduced by different AIF techniques ($QP_P = 28$, $QP_B = 29$) in video sequences. (a) *City*. (b) *Raven*. (c) *Harbour*. (d) *Sheriff*.

Since h_{opt} is the solution of (2), the latter part in (3) equals zero and therefore Δerr, the increased energy of prediction error introduced by Δh, is finally expressed by

$$\Delta err = \sum_{i,j}\sum_{m,n} h_\Delta(i,j)h_\Delta(m,n)R_{pp}(i-m, j-n). \quad (4)$$

Δerr, introduced by different interpolation techniques, is studied based on the first ten frames (except I-frame) of a set of ten HD sequences. Fig. 2 gives the typical results on four out of the ten sequences. Generally, Δerr introduced by 2-D non-separable AIF and D-AIF is close to or even larger than that introduced by the standard filter in H.264/AVC. Especially, 2-D non-separable AIF, which has larger support region than D-AIF and is expected to perform better, is more likely to have particularly large Δerr. That is because R_{pp} in (2) is ill-conditioned, so any slight change in h_{opt}, having been denoted as h_Δ, will greatly increase Δerr. Furthermore, quantization error influences Δerr more significantly than other factors, e.g., the support region and the imposed symmetry constraint, and 2-D non-separable AIF has much more coefficients involved in quantization. Therefore, the 9-bit uniform quantization used in the existing AIF techniques cannot produce precise enough filter coefficients, which actually causes the value of Δerr to increase dramatically and frequently disables the AIF mode at the frame level during the encoding process, based on the R-D criterion [7].

III. Parametric Interpolation Filter (PIF)

In the related work, the accuracy of the coefficients and the size of the side information are conflicting, because the coefficients are coded individually. In this letter, the interpolation filters is represented by a function h_f determined by five parameters, and the filter coefficients are calculated as the

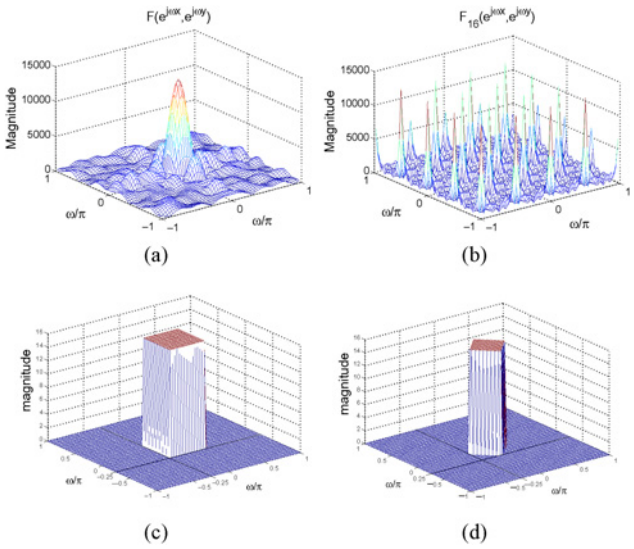

Fig. 3. Fourier transforms of (a) the original frame and (b) the upsampled frame, and the frequency responses of (c) the ideal interpolation filter and (d) the proposed interpolation filter.

function values. The five parameters should make the h_f approximate h_{opt} such that Δerr in (4) is minimized. Obviously, the side information for coding five parameters is very small. Yet the accuracy of the coefficients can also be guaranteed, if the parameters are quantized in enough precision. The form of h_f is presented in Section III-A; parameter determination and coding are introduced in Section III-B. Section III-C explains how to use h_f to interpolate certain sub-positions.

A. Function Representation of Interpolation Filters

Let $F(e^{j\omega_x}, e^{j\omega_y})$ be the Fourier transform of the reference frame P. After upsampling P using zero-insertion, shown in Fig. 1(b), the Fourier transform of the upsampled frame P_{16}, denoted as $F_{16}(e^{j\omega_x}, e^{j\omega_y})$, is given by

$$F_{16}(e^{j\omega_x}, e^{j\omega_y}) = F(e^{j4\omega_x}, e^{j4\omega_y}). \tag{5}$$

According to (5), F_{16} is a frequency-scaled version of F. Fig. 3(a) and (b) gives examples of F and its corresponding F_{16}, respectively. In Fig. 3(b), the undesired spectra centering at integer multiples of $(\pi/2, \pi/2)$, i.e., the original sampling rate, are introduced by the zero-insertion upsampling and should be removed. This requires a lowpass filter h [see Fig. 1(b)] with a gain of 16 and a cutoff frequency $\pi/4$, of which the ideal frequency response is shown in Fig. 3(c).

In this letter, the desired frequency response of h_f is proposed based on the special energy distribution of HD videos, represented by power spectral density. Our studies show that the high frequency energies in HD sequences are mainly distributed in the horizontal and vertical directions, whereas the high frequency energies in low resolution sequences are distributed in arbitrary directions. Therefore, the desired filter is proposed to have a diamond-shaped passband [see Fig. 3(d)], such that high frequency components neither in the horizontal direction nor in the vertical direction are filtered out. There are two reasons for proposing such a passband. First, interpolation in the context of video coding is for a better MCP, and the high frequency components are more likely to introduce large prediction error, thus exerting negative influence on MCP. Second, high frequency components outside the diamond-shaped passband have very low energy and contribute little to the content of the frame. This passband shape also agrees with our observation on a number of optimal AIFs.

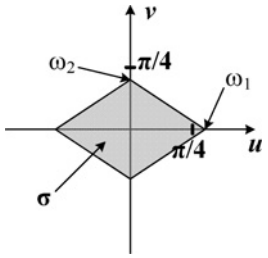

Fig. 4. Passband of the proposed ideal interpolation filter.

Theoretically, the cutoff frequency should be $\pi/4$, such as the one in Fig. 3(d), although, in practice, it may vary around $\pi/4$. As shown in Fig. 4, two parameters, ω_1 and ω_2, are used to denote the cutoff frequencies at two axes and the shaded diamond-shaped area, σ, represents the passband. The corresponding impulse response, h_d, can be obtained by inverse Fourier transform, as expressed in

$$h_d(m,n) = \frac{1}{4\pi^2}\int_{[-\pi,\pi]^2} H_d(e^{ju}, e^{jv}) e^{jmu+jnv} du dv \tag{6}$$

where H_d is given as

$$H_d(e^{ju}, e^{jv}) = \begin{cases} 16, & \text{if } (u,v) \in \sigma \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Substituting (7) into (6) and after some derivation steps, one can finally find the formula of h_d, as shown in

$$h_d(m,n) = \frac{8\omega_1\omega_2}{\pi^2}\text{Sinc}\left(\frac{\omega_1 m + \omega_2 n}{2}\right)\text{Sinc}\left(\frac{\omega_1 m - \omega_2 n}{2}\right). \tag{8}$$

h_d has infinite size and needs to be truncated by an appropriate window function w before being used for interpolation. In this letter, w, defined as in (9), is proposed, which is empirically better than other widely used ones, such as rectangular, triangular, Hanning, and Blackman window functions

$$w(m,n) = \begin{cases} a + \text{Sinc}(b|m| + c|n|), & \text{if } -11 \le m,n \le 11 \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Then, the approximation of the ideal desired filter, denoted as h_f, is obtained as the product of h_d and w, as in

$$h_f(m,n) = h_d(m,n)w(m,n) = N\text{Sinc}\left(\frac{\omega_1 m+\omega_2 n}{2}\right)\text{Sinc}\left(\frac{\omega_1 m-\omega_2 n}{2}\right)(a+\text{Sinc}(b|m|+c|n|)). \tag{10}$$

Therefore, h_f is determined by a parameter set, $\mathbf{x} = \{\omega_1, \omega_2, a, b, c\}$. The factor N guarantees the filter gain is 16. h_f is the interpolation filter proposed in this letter, which represents filters by parameters rather than individual coefficients and is named as PIF.

TABLE I
DECOMPOSE h_f TO 16 INTERPOLATION FILTERS FOR INTEGER AND SUB-POSITION PIXELS

Position	Filter	Scope of k	Scope of l	Position	Filter	Scope of k	Scope of l
Int.	$h_f(4k, 4l)$	$-2 \le k \le 2$	$-2 \le l \le 2$	h	$h_f(4k, 4l-2)$	$-2 \le k \le 2$	$-2 \le l \le 3$
a	$h_f(4k-1, 4l)$	$-2 \le k \le 3$	$-2 \le l \le 2$	i	$h_f(4k-1, 4l-2)$	$-2 \le k \le 3$	$-2 \le l \le 3$
b	$h_f(4k-2, 4l)$	$-2 \le k \le 3$	$-2 \le l \le 2$	j	$h_f(4k-2, 4l-2)$	$-2 \le k \le 3$	$-2 \le l \le 3$
c	$h_f(4k-3, 4l)$	$-2 \le k \le 3$	$-2 \le l \le 2$	k	$h_f(4k-3, 4l-2)$	$-2 \le k \le 3$	$-2 \le l \le 3$
d	$h_f(4k, 4l-1)$	$-2 \le k \le 2$	$-2 \le l \le 3$	l	$h_f(4k, 4l-3)$	$-2 \le k \le 2$	$-2 \le l \le 3$
e	$h_f(4k-1, 4l-1)$	$-2 \le k \le 3$	$-2 \le l \le 3$	m	$h_f(4k-1, 4l-3)$	$-2 \le k \le 3$	$-2 \le l \le 3$
f	$h_f(4k-2, 4l-1)$	$-2 \le k \le 3$	$-2 \le l \le 3$	n	$h_f(4k-2, 4l-3)$	$-2 \le k \le 3$	$-2 \le l \le 3$
g	$h_f(4k-3, 4l-1)$	$-2 \le k \le 3$	$-2 \le l \le 3$	o	$h_f(4k-3, 4l-3)$	$-2 \le k \le 3$	$-2 \le l \le 3$

B. Parameter Determination and Coding

As the form of h_f has been given in (10), the parameter set $\mathbf{x}$ has to be determined for each frame and coded. The optimal $\mathbf{x}$ should make h_f reduce as much prediction error as h_{opt} can, by achieving the minimum of Δerr in (4), that is

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \Delta err \tag{11}$$

where h_Δ in (4) can be expressed as in

$$h_\Delta(i, j) = h_{opt}(i, j) - h_f(i, j|\mathbf{x}). \tag{12}$$

The minimum is achieved by using the quasi-Newton method, which is based on Newton's method to find the stationary point of the objective function, where the gradient is zero. Different from Newton's method, which uses the first and second derivatives, i.e., gradient and Hessian, to find the stationary point, quasi-Newton method does not directly compute the Hessian of the objective function; instead, it updates the Hessian by analyzing successive gradient vectors. Here, BFGS algorithm is used to update the Hessian matrix. The detailed description is omitted here, as this method is not part of the contribution in this letter. Interested readers are referred to [8] for more in-depth expositions.

Since this numerical method solves the local minimum problem, the initial estimates of $\mathbf{x}$ become critical. At the beginning of coding a video sequence, the initial estimates of ω_1 and ω_2 are both 0.25π, and empirical initial values 0.1, 0.15, and 0.15 are assigned to a, b, and c, respectively. During the coding process, the initial estimate keeps updated by the value of $\hat{\mathbf{x}}$ of the latest P-frame using AIF mode.

Before coded, the five parameters are quantized. The magnitude of each parameter is uniformly quantized to 8196 steps and coded by 13-bit fixed length coding. To indicate the signs of a, b, and c, three additional bits are used. Hence, the side information is exactly 68 bits for each frame, about 20% of that required by the 2-D non-separable AIF [3].

C. Interpolation for Certain Positions

The perspective of interpolation in this letter is quite different from that in the related work [2]– [6]. In this letter, only one interpolation filter [see h in Fig. 1(b)] is studied, which can interpolate a frame to 16 times the size. In the related work, the missing pixels are classified into 15 categories, denoted as $a, b, \ldots, o$, as shown in Fig. 5, and each category of pixels is interpolated by its associated filter. Therefore, 15 independent filters are to be designed. Despite the different perspectives, the interpolation filters in this letter and in the related work are actually equivalent. In the related work, the 15 filters can construct an all-in-one filter, i.e., h in Fig. 1(b). Conversely, h_f proposed in Section III-A can be decomposed into 16 filters used for the integer pixels *Int*. and the sub-position pixels $a, b, \ldots, o$. Table I shows how h_f is decomposed. Note that k and l in the table are integers.

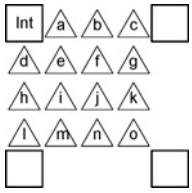

Fig. 5. Integer and sub-position pixels to be interpolated.

IV. EXPERIMENTAL RESULTS

A. Improvement of R-D Performance

The proposed PIF is integrated into the VCEG's reference software KTA2.3 and tested using the test conditions specified in [9]. In the encoder, the MVs estimated in the first step are reused in order to show PIF's performance with one-pass encoding strategy. All the filters in comparison, including the optimal filter, PIF, D-AIF [4], 2-D non-separable AIF [3], E-AIF [6], and E-DAIF [5], are implemented using 32-bit integer arithmetic. Note that the bitstreams coded using h_{opt} are non-decodable, because the filter coefficients are not quantized or transmitted. The AIF mode can be disabled based on the criterion of frame level R-D cost [7]. Table II compares the six interpolation techniques in terms of the bit-rate reduction at the same PSNR, denoted as ΔBR, and the frequency of occurrence, denoted as "Freq." The former uses H.264/AVC high profile as the benchmark and is estimated using the method proposed in [10].

Table II shows the experimental results, when the coded sequence structure is hierarchical B with GOP size equal to 8. The bit-rate reduction provided by h_{opt} is 9.46% on average, which can be taken as the upper bound of the performance that AIF techniques can bring. The performance of the proposed PIF is always better than other AIF techniques for all the test sequences. On average, the bit-rate reduction provided by PIF is 6.88%, about 2.58% less than the upper bound. For 2-D

TABLE II
BIT-RATE REDUCTION AND THE FREQUENCY OF OCCURRENCE OF DIFFERENT AIF TECHNIQUES WITH THE HIERARCHICAL B CODING STRUCTURE

Test Sequences	2-D NS AIF		D-AIF		E-AIF		E-DAIF		PIF		h_{opt}	
	ΔBR (%)	Freq. (%)	ΔBR (%)	Freq. (%)	ΔBR (%)	Freq. (%)	ΔBR (%)	Freq. (%)	ΔBR (%)	Freq. (%)	ΔBR (%)	Freq. (%)
Bigships	−2.76	20	−3.21	27	−4.98	78	−4.27	54	−11.53	77	−13.27	93
City	−4.10	35	−4.42	45	−7.13	85	−6.97	80	−13.52	90	−15.06	97
Crew	−2.14	13	−2.32	21	−4.74	52	−4.25	41	−6.54	74	−10.29	88
Harbour	−1.18	22	−0.98	27	−2.14	58	−1.76	49	−5.98	76	−8.05	91
Jets	0.34	1	0.26	2	−0.03	22	−0.03	15	−2.12	62	−2.54	72
Night	−0.52	2	−0.16	2	−2.25	55	−1.69	17	−3.98	74	−4.45	86
Optis	−2.39	38	−2.70	53	−4.92	78	−3.14	77	−9.64	90	−10.85	92
Raven	−3.24	24	−4.18	42	−5.99	73	−6.15	71	−11.44	88	−13.73	93
Sailormen	−0.83	33	−1.07	40	−1.95	59	−1.40	59	−5.86	77	−8.79	87
Sheriff	−2.53	27	−2.68	41	−3.76	67	−3.78	70	−7.68	81	−9.92	91
ShuttleStart	−2.55	12	−3.86	26	−5.77	66	−6.07	56	−12.12	76	−16.05	94
Average	−1.66	18	−1.74	23	−3.36	58	−2.99	47	−6.88	76	−9.46	89

non-separable AIF and D-AIF, the average bit-rate reductions are both less than 2% and their frequencies of occurrence are very low, because they cannot efficiently reduce the prediction error energy, as shown in Fig. 2. For E-AIF and E-DAIF, their frequencies of occurrence are remarkably increased, compared with 2-D non-separable AIF and D-AIF, but the performance improvements are trivial, i.e., less than 2% further bit-rate reduction on average. This phenomenon means that E-AIF and E-DAIF can reduce the prediction error energy in coding most of the frames, compared with using the standard filter in H.264/AVC, but the decrement is not significant.

B. Complexity Analysis

The complexities of the filter in H.264/AVC [1], 2-D non-separable AIF [3], D-AIF [4], E-AIF [6], E-DAIF [5], and the proposed PIF are quantitatively analyzed on the decoder side. The approach of complexity analysis follows what is proposed in [11], where the costs of multiplication, addition, and shifting are assumed to be equal. As shown in Table III, N_x ($x = Int, a, b, \ldots, o$) means the numbers of arithmetic operations, i.e., the summation of multiplications, additions, and shiftings, required to interpolate integer and sub-position pixels. The approach in [11] also assumes equipartition of the MVs with regard to integer and sub-positions, and therefore the average cost for one position can be calculated by

$$\bar{N} = \left(N_{Int} + 4N_a + 2N_b + N_j + 4N_e + 4N_f\right)/16. \quad (13)$$

PIF has the highest complexity, compared with other AIF techniques. First, PIF uses a 5×5-tap filter for integer pixels. This filter is axisymmetric and therefore is simpler than the 5×5-tap filter used in E-AIF, which has no symmetry imposed. Second, PIF uses 2-D non-separable filters for the sub-positions aligned with neighboring integer pixels, i.e., 6×5-tap filters for a, b, and c, and 5×6-tap filters for d, h, and l. In other AIF techniques, 1-D 6-tap filters are used for these six sub-positions. Third, for sub-positions j, e, g, m, and o, PIF has higher complexity than 2-D non-separable AIF, although both of them use 6×6-tap non-separable filters. That is because PIF is axisymmetric, whereas 2-D non-separable AIF has circular symmetry. Furthermore, PIF should be implemented by 32-bit integer arithmetic, in order to demonstrate one of its merits: representing the filter coefficients in high precision.

TABLE III
NUMBER OF ARITHMETIC OPERATIONS REQUIRED TO INTERPOLATE INTEGER AND SUB-POSITION PIXELS

Positions	H.264 Filter	2-D NS AIF	D-AIF	E-AIF	E-DAIF	PIF
N_{Int}	0	0	0	51	0	35
N_a, N_c, N_d, N_l	13	13	13	13	13	49
N_b, N_h	10	10	10	10	10	40
N_j	32.5	41	16	16	16	46
N_e, N_g, N_m, N_o	23	58	13	25	13	73
N_f, N_i, N_k, N_n	35.5	55	19	19	19	55
$\bar{N}$	**21.2**	**35.3**	**13.5**	**19.7**	**13.5**	**54.3**

On the encoder side, the complexity of implementing PIF mainly lies in the two-pass encoding strategy, just like implementing other AIF techniques. The additional complexity introduced by PIF includes the larger number of arithmetic operations required for interpolating reference frames (see Table III) and the iterative numerical algorithm, i.e., BFGS quasi-Newton algorithm (see Section III-B). Nevertheless, BFGS quasi-Newton algorithm is practical for a video encoder, because BFGS quasi-Newton algorithm does not need to compute the second derivative, i.e., the Hessian matrix, of the objective function or any matrix inversion.

V. CONCLUSION

In this letter, PIF for MCP was proposed, which represents interpolation filters by the function determined by five parameters, thus solving the conflict of the coefficients' accuracy and the overhead's size. The proposed PIF was efficient especially for HD video coding, because the function was designed based on the special statistical attributes of HD videos. The experimental results showed that PIF outperformed the other AIF techniques and approached the efficiency of the optimal AIF.

REFERENCES

[1] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264, Rev. 2, Mar. 2005.

[2] T. Wedi, "Adaptive interpolation filter for motion compensated prediction," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Sep. 2002, pp. 509–512.

[3] Y. Vatis and J. Ostermann, "Adaptive interpolation filter for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 179–192, Feb. 2009.

[4] D. Rusanovskyy, K. Ugur, A. Hallapuro, J. Lainema, and M. Gabbouj, "Video coding with low-complexity directional adaptive interpolation filters," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1239–1243, Aug. 2009.

[5] D. Rusanovskyy, K. Ugur, and M. Gabbouj, "Adaptive interpolation with flexible filter structures for video coding," in *Proc. IEEE ICIP*, Nov. 2009, pp. 1025–1028.

[6] M. Karczewicz, Y. Ye, P. Chen, and G. Motta, *Experimental Results of Interpolation Filters on High-Definition Sequences*, ITU-T Q.6/SG16 (VCEG) document VCEG-AJ30, Oct. 2008.

[7] Y. Vatis and J. Ostermann, *Rate-Distortion Optimized Coder Control for Adaptive Interpolation Filter in the KTA Reference Model*, ITU-T Q.6/SG16 (VCEG) document VCEG-AE16, Jan. 2007.

[8] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 2006.

[9] T. K. Tan, G. J. Sullivan, and W. Thomas, *Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 4*, ITU-T Q.6/SG16 (VCEG) document VCEG-AJ10_r1, Jul. 2008.

[10] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, ITU-T Q.6/SG16 (VCEG) document VCEG-M33, Apr. 2001.

[11] Y. Vatis and J. Ostermann, *Comparison of Complexity Between Two-Dimensional Non-Separable Adaptive Interpolation Filter and Standard Wiener Filter*, ITU-T Q.6/SG16 (VCEG) document VCEG-AA11, Apr. 2005.