Generalized Interpolation-Based Fractional Sample Motion Compensation

Haricharan Lakshman, Heiko Schwarz, and Thomas Wiegand, *Fellow, IEEE*

***Abstract*—Fractional sample interpolation with finite impulse response (FIR) filters is commonly used for motion-compensated prediction (MCP). The FIR filtering can be viewed as a signal decomposition using restricted basis functions. The concept of generalized interpolation provides a greater degree of freedom for selecting basis functions. We developed a generalized interpolation framework for MCP using fixed-point infinite impulse response and FIR filters. An efficient multiplication-free design of the algorithm that is suited for hardware implementation is shown. A detailed analysis of average and worst case complexities compared to FIR filter-based interpolation techniques is provided. Average bitrate savings of around 2.0% compared to an 8-tap FIR filter are observed over the high-efficiency video coding dataset at a similar worst case complexity.**

***Index Terms*—B-splines, interpolation, motion-compensated prediction, video coding.**

I. Introduction

In motion-compensated prediction (MCP), the current image is predicted from already reconstructed images and the prediction error is coded along with motion information. MCP using fractional sample accurate displacements is an established technique in video coding. When the accessed position does not fall on the integer–sample grid, it is interpolated using the neighboring samples. In the H.264/AVC standard [1], a 2-D separable architecture is employed, in which the horizontal and vertical filtering stages are performed sequentially, depending on the required fractional position. For each direction, a 6-tap finite impulse response (FIR) filter is used to generate the half-sample positions, followed by a 2-tap FIR filter for generating quarter-sample positions.

A theoretical analysis of fractional-sample MCP using Gaussian power spectral density (PSD) models is provided in [2]. The effects of different interpolation kernels and noise in pictures were extensively studied. Adapting the interpolation filter coefficients was proposed in [3]. It was shown that changing the filter coefficients for each picture could account for nonstationarities of the video signals. In today's scenario, there are two major changes from the settings that were previously analyzed. First, with the advancement of camera technology, the noise in the captured video is maintained relatively low. The edges and fine spatial features of a scene are acquired well, resulting in relevant signal components even in the higher frequencies of the signal spectrum. Second, the coding tools of new video codecs, such as the upcoming high-efficiency video coding (HEVC) [4], have improved significantly and generate reference samples more suitable for MCP. After the differential pulse-code modulation (DPCM) reconstruction, a deblocking filter and an optional adaptive loop filter can be used in HEVC. In the adaptive loop filter stage, coefficients are estimated and transmitted in the bitstream to reduce the distortion (due to quantization, etc.) in reconstructed pictures. The role of smoothing the reference samples during interpolation filtering, see Wiener filtering in [2], has been moved to the deblocking and adaptive loop filters. Hence, it is important not to further blur the spectral components of the reference pictures during interpolation filtering in order to be able to predict and cancel the corresponding components in subsequent pictures.

Some of the main techniques proposed for improving the interpolation filtering in MCP are using FIR filters with higher number of taps [5], 1-D directional filters for reducing complexity [6], switched interpolation filters [8], etc. A 12-tap filter was used for MCP in HM 1.0, the first test model [4], of the ongoing HEVC standardization. Although increasing the filter support can improve the quality of interpolation in general, it increases the computational complexity and may introduce ringing artifacts around edges. A portion of the gains in HM 1.0 relative to H.264/AVC filter structure stems from the use of higher precision arithmetic [8]. In [6], an approach to achieve low-complexity interpolation using 1-D directional filters is presented. Although the complexity of 1-D directional filtering is low, it is reported to show significant losses for sequences with considerable high-frequency (HF) content. Switched interpolation filtering [8] consists of a set of filters, for each fractional-sample position, known to both encoder and decoder. The index of the specific filter to be used for MCP of the current slice is transmitted in the bitstream. With such a scheme, many filters with different cutoff characteristics can be designed, but to have improved filters that reproduce low frequencies well and at the same time do not blur the edges, the

Manuscript received November 29, 2011; revised April 26, 2012; accepted May 23, 2012. Date of publication July 5, 2012; date of current version March 7, 2013. This paper was recommended by Associate Editor H. Gharavi.

H. Lakshman and H. Schwarz are with the Image and Video Coding Group, Fraunhofer Institute for Telecommunications-Heinrich Hertz Institute, Berlin 10587, Germany (e-mail: haricharan.lakshman@hhi.fraunhofer.de; heiko.schwarz@hhi.fraunhofer.de).

T. Wiegand is with the Department of Image Processing, Fraunhofer Institute for Telecommunications-Heinrich Hertz Institute, Berlin 10587, Germany, and also with the Technical University of Berlin, Berlin 10587, Germany (e-mail: thomas.wiegand@hhi.fraunhofer.de).

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.2012.2207273

number of taps of the FIR filters has to be increased, thereby increasing the overall complexity.

In this paper, we use the concept of generalized interpolation and show that significant gains can be achieved even when using fixed filters with small number of taps, e.g., 4-tap FIR filters used along with motion information-independent 2-tap infinite impulse response (IIR) filters. In the classical approach to interpolation, a discrete signal is mapped onto a continuous signal using interpolating basis functions. In generalized interpolation, the constraints on the set of usable basis functions are relaxed, which can improve the approximation quality of the prediction [9]. The signal is expressed in terms of weighted summation of generalized basis functions, where the weights are called expansion coefficients. The expansion coefficients are derived using IIR filters so that the known signal at integer locations can be perfectly reconstructed. Using the determined signal expansion, the values at arbitrary fractional positions can be generated using FIR filtering. Due to the continuous nature of the underlying basis functions, such a framework can be used to generate values at arbitrary fractional positions (e.g., 1/8, 1/16, or others) and can be used for translational as well as higher-order motion models. Methods to design simple IIR filter stages are shown and a multiplication-free design of the IIR and FIR stages is provided, which is more suitable for a hardware implementation. Furthermore, the complexity in terms of the average and worst case number of operations and the overall memory accesses are analyzed in detail and compared to the commonly used FIR filter-based MCP.

II. Review of Motion Compensation With Fractional-Sample Resolution

A continuous scene gets captured in the form of a set of discrete samples by a sampling stage in cameras. The true displacements of objects from one frame to another are independent of the sampling grid of cameras. Therefore in MCP, the samples of a block are predicted using fractional-sample displacements of the reference picture samples to better capture the continuous motion.

A. Ideal Interpolation Filters

Consider the z-transform of the reference signal by $R(z)$, the z-transform of the prediction signal by $P(z)$, and the desired fractional displacement as D. If we require the prediction signal to be an *ideally* displaced version of the reference signal, then

$$P(z) = z^{-D} \cdot R(z). \tag{1}$$

Therefore, the transfer function of the ideal displacement filter can be written as $H_{\text{ideal}}(z) = z^{-D}$. For the case of fractional displacement of discrete data, it is useful to analyze the spatial displacement process in the Fourier domain by setting $z = e^{j\omega}$, where ω denotes the angular frequency. The magnitude response of the ideal displacement filter becomes

$$|H_{\text{ideal}}(j\omega)| = |e^{-j\omega D}| = 1 \tag{2}$$

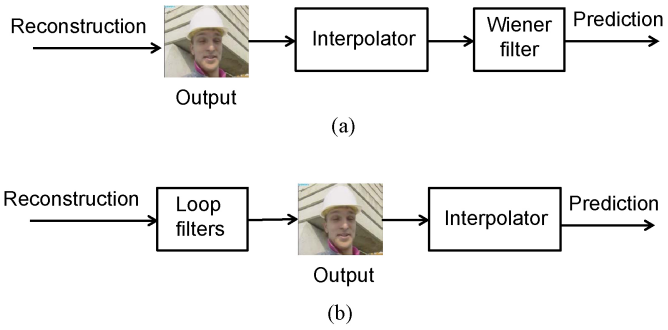

Fig. 1. Comparison of two structures for MCP. (a) Prediction structure considered in [2]. A setup in which the reference signal is interpolated to produce phase shift and then Wiener filtered. For implementation, the combination of interpolator and Wiener filter is simulated using a single FIR filter that produces phase delay in passband and smoothing in higher frequencies. (b) Prediction structure used in HEVC draft [4]. The DPCM reconstructed picture is filtered using loop filters to produce the output picture and filtering information are transmitted as side information. The loop filters consist of deblocking and adaptive filters that decide whether to smooth the reconstruction edges or not. The role of the interpolator is mainly to produce the required phase delay, without much smoothing in the HF components.

with a phase delay of

$$\tau = -\frac{\arg\{H(j\omega)\}}{\omega} = \frac{\omega D}{\omega} = D. \tag{3}$$

Hence, the ideal filter retains all the frequencies in the reference signal and introduces a constant phase delay of D to each frequency, thereby producing samples at a displaced location in the reference picture. In spite of the simple and elegant theoretical formulation of the ideal fractional displacement filter, a fundamental problem arises when trying to implement the filter. If we compute the impulse response of the filter by taking the inverse Fourier transform, we get $\text{sinc}(n - D)$. Unfortunately, the resulting function is infinitely long and cannot be used in practice. The task of designing a filter for fractional sample displacement is to create an implementable solution that comes close to the ideal solution with limited amount of resources (complexity, delay, memory, etc).

B. Interaction of Loop Filters and Interpolator

The various stages of an image processing chain, such as image acquisition, transform domain quantization, and so on, add noise to the pictures and therefore limit the potential of a reference picture to predict the subsequent pictures. Girod [2] provided a theoretical analysis of fractional-sample MCP using Gaussian PSD models and close-to-ideal filters simulated using long FIR filters. A Wiener filter [2] was introduced to smooth the displaced signal before using it for prediction. It was shown that smoothing the high frequencies after a nearly ideal filter was beneficial to counter the displacement estimation errors and noise in the pictures. Considering this result, the combination of interpolator and Wiener filter was jointly approximated using a single filter using a minimum mean square estimation (MMSE). In the last decade, there has been significant work on loop filters to spatially filter the signal after the DPCM stage in video codecs. Blocking artifacts due to transform domain quantization are reduced using deblocking filters in H.264/AVC and more recently during HEVC

development, a Wiener filter is introduced in the reconstruction loop, as shown in Fig. 1(b). This dedicated loop filter is also designed using the MMSE technique and handles the signal nonstationarities using block adaptive processing. Hence, the traditional role of smoothing in interpolation filtering has been moved to the loop filters in the reconstruction stage.

For reference pictures that are deblocked and adaptively loop filtered, the main challenge is to design MCP filters that produce constant magnitude response and phase delay for a larger range of frequencies with limited filter complexity. A straightforward way to produce better filters is to use longer number of FIR taps, which would require very high complexity as the filter support becomes large. Apart from blurring high frequencies, other common interpolation artifacts include ringing, aliasing, and blocking.

C. Approximation Order

The distortions introduced by interpolation filters to each frequency component of a signal can be categorized as magnitude and phase distortions. Using approximation theory, Blu *et al.* [10] studied the effects due to interpolation filters generated by sampling basis functions with a given support, using a metric known as the approximation order. The approximation order can be viewed as a summary of the errors to the individual frequency components over the entire frequency range. It was reported that the approximation order inherent in the basis functions is important to limit the interpolation artifacts [10]. With the advances in sampling and interpolation theory [9], the freedom in choosing the basis functions to model the sampled data has led to a generalized interpolation framework. This paper focuses on constructing filters using the generalized framework and investigates the effects on the rate-distortion (RD) performance and computational complexity in the context of video coding.

III. Structure of Generalized Interpolation

The process of interpolation can be viewed as fitting a continuous curve through a given set of samples and determining the values of the curve at new points of interest. A commonly employed setup is to choose the continuous curve to be formed by a linear combination of shifted basis functions. Among the possible basis functions, those that pass through zero at all integer locations except the origin, where they have a value of unity, are known to be restricted under the interpolating constraint (e.g., sinc function).

A. Interpolation Formulation

Given a set of samples $s[k]$ corresponding to integer locations $k \in \mathbb{Z}$, the task of interpolation is to estimate the sample value $g(x)$ at a fractional location x. The traditional interpolation formula is of the form

$$g(x) = \sum_{k\in\mathbb{Z}} s[k] \cdot \phi_{\text{int}}(x-k) \tag{4}$$

where ϕ_{int} is chosen to satisfy the interpolating condition. Equation (4) can be viewed as a signal expansion where the expansion coefficients are the samples themselves. Depending on the required fractional position x, the basis ϕ_{int} can be sampled to generate an FIR filter. The interpolation process is then a convolution of the reference picture samples with the corresponding FIR filter.

In generalized interpolation [9], the zero crossing constraint is not imposed on the basis functions and the problem is reformulated as

$$f(x) = \sum_{k\in\mathbb{Z}} c[k] \cdot \phi(x-k) \tag{5}$$

where $\phi(x)$ are basis functions with basic constraints for stability and unambiguous reconstruction and $c[k]$ are the expansion coefficients. The major difference to the expression in (4) is that the expansion coefficients are not the signal samples $s[k]$ anymore. The formulation in (5) gives an extended choice of basis functions $\phi(x)$ with better properties than the case of $\phi_{\text{int}}(x)$ restricted under the interpolating condition or, equivalently, restricted under the condition $c[k] = s[k]$. This increase in the choice of basis functions comes at the expense of an additional step of calculating the expansion coefficients $c[k]$.

B. Determination of Expansion Coefficients

A well-known example in the category of generalized basis functions is the cubic spline function. The problem of determining the spline expansion coefficients was approached using a matrix inversion. It was recognized in [11] that this problem could also be solved using simpler digital filtering techniques. Based on [12], we briefly describe the computation of the expansion coefficients in this section for the sake of better understanding of the generalized interpolation framework. The unknown parameters c in the signal model from (5) can be determined by evaluating the continuous function at integer values of x. The values resulting from the continuous model $f(n)$ at positions $n \in \mathbb{Z}$ should equal the known discrete samples $s[n]$ of the signal for perfect reconstruction, that is

$$s[n] = \sum_{k\in\mathbb{Z}} c[k] \cdot \phi(n-k), \quad \forall n \in \mathbb{Z}. \tag{6}$$

In order to solve for $c[k]$, a transform domain formulation is employed. Denoting the z-transforms of $s[n]$, $c[n]$, and $\phi(n)$ as $S(z)$, $C(z)$, and $\Phi(z)$, respectively, (6) can be written as

$$S(z) = C(z) \cdot \Phi(z) \implies C(z) = S(z)/\Phi(z). \tag{7}$$

Hence, the unknown coefficients $c[n]$ can be obtained by filtering $s[n]$ using

$$H(z) = 1/\Phi(z) = \frac{1}{\sum_{n\in\mathbb{Z}} \phi(n) z^{-n}}. \tag{8}$$

This IIR filter used for determining the expansion coefficients is also known as a prefilter in the signal processing community. For basis satisfying the interpolating constraint, this prefilter reduces to $H(z) = 1$ because $\phi_{\text{int}}(0) = 1$ and $\phi_{\text{int}}(n) = 0$ for $n \neq 0$. Last, in order to take care of the samples at the boundaries, $s[k]$ and $c[k]$ are extended using mirror-symmetry [12]. To implement the IIR prefilter, it can be

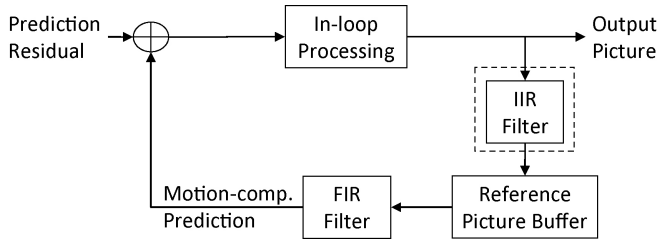

Fig. 2. MCP using generalized interpolation. The IIR filtering block in the dashed box constitutes the main difference to the standard techniques.

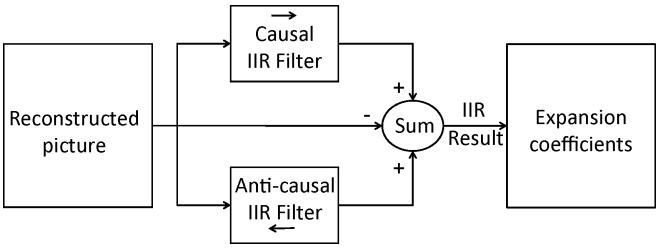

Fig. 3. Computation of expansion coefficients from reconstructed picture. The IIR prefilter has causal, anticausal, and direct terms.

factorized using its poles. The spline basis functions, for instance, result in both causal and anticausal poles for the prefilter.

IV. Integration Into Hybrid Video Codec

A method for integrating the generalized interpolation scheme into a hybrid video codec is described in this section. In typical hybrid video codecs, the reconstructed picture is obtained after completing the in-loop processing steps like deblocking, and so on. The reconstructed pictures go into temporary buffers and act as references for predicting subsequent frames, hence they are also called reference pictures at this stage. In this paper, we propose a design in which the reconstructed picture is IIR filtered using the prefilter in (8) and the result, which has the same dimensions as the input picture, is stored, as illustrated in Fig. 2. In other words, the reference buffers contain the expansion coefficients. During the MCP stage, the integer part of the motion vectors (MVs) are used to access the expansion coefficients and the fractional part is used to select the FIR coefficients for filtering.

The prefilter described in (8) is a 1-D filter and can be easily extended for using with images. The computation of expansion coefficients for a reconstructed image in a video codec can be done by 1-D filtering successively along the rows and columns. Here, we consider the case of a prefilter with a causal and a symmetric anticausal pole. Hence, (8) can be written as

$$H(z) = \frac{1}{a_1 \cdot z^{-1} + a_0 + a_1 \cdot z^{1}} \tag{9}$$

where $a_0 = \phi(0)$ and $a_1 = \phi(1) = \phi(-1)$. This can be factorized into partial fractions using its causal pole p_1 as

$$H(z) = g \cdot \left\{ \frac{1}{1 - p_1 \cdot z^{-1}} + \frac{1}{1 - p_1 \cdot z} - 1 \right\}. \tag{10}$$

The factorization shown above has a causal component with a z^{-1} term, an anticausal component with z term, and a direct component, along with a normalization factor, denoted as g. An example implementation of the IIR prefiler step along the rows of a reconstructed picture is depicted in Fig. 3. Another possible implementation is to factorize (9), using a cascade of a causal and an anticausal filter. The anticausal filter can then operate on the result of the causal filter. This structure can operate on the buffers in place and does not require additional temporary memory but has two important disadvantages.

1) The errors from the first filter due to a finite precision implementation can get amplified by the second filter because of the sequential application of the filters.
2) The latency of the prefiltering stage increases because the second filter has to wait for the completion of the first filter. This latency can be significant, especially for the vertical anticausal filter, where the entire frame buffer needs to be reloaded for filtering.

Due to these reasons, in this paper, the parallel application of the causal and anticausal filters is preferred to the serially cascaded version.

Consider a 2-D coordinate system with axes (m, n) and an image size of M columns and N rows. The following notation for the different stages of filtering are employed: reconstructed picture $r[m, n]$, causal horizontal filter output $h_c[m, n]$, anticausal horizontal filter output $h_a[m, n]$, final horizontal output $h[m, n]$, causal vertical filter output $v_c[m, n]$, anticausal vertical filter output $v_a[m, n]$, and final vertical output $v[m, n]$. Converting (10) into the spatial domain leads to the following steps for the horizontal filtering case

$$\begin{aligned} h_c[m, n] &= r[m, n] + p_1 \cdot h_c[m-1, n], \forall m \in [1, M-1] \\ h_a[m, n] &= r[m, n] + p_1 \cdot h_a[m+1, n], \forall m \in [M-2, 0] \\ h[m, n] &= h_c[m, n] + h_a[m, n] - r[m, n], \forall m \in [0, M-1]. \end{aligned}$$

Next, the result of the horizontal filtering is operated vertically using the same structure as above

$$\begin{aligned} v_c[m, n] &= h[m, n] + p_1 \cdot v_c[m-1, n], \forall n \in [1, N-1] \\ v_a[m, n] &= h[m, n] + p_1 \cdot v_a[m+1, n], \forall n \in [N-2, 0] \\ v[m, n] &= v_c[m, n] + v_a[m, n] - h[m, n], \forall n \in [0, N-1] \end{aligned}$$

The initialization for each filtering step can be done using mirror-symmetric boundary conditions, according to [12]. The final result of the IIR filtering is $v[m, n]$, which constitutes the expansion coefficients for the 2-D separable generalized interpolation. As shown in Fig. 2, the coefficients are stored in the picture buffers for subsequent MCP.

The actual prediction of a set of samples based on the values stored in the reference buffers happens according to (5), which is extended to 2-D using tensor product as

$$f(x, y) = \sum_{m=m_1}^{m_1+T-1} \sum_{n=n_1}^{n_1+T-1} v[m, n] \cdot \phi(x - m)\phi(y - n) \tag{11}$$

where T denotes the number of nonzero taps of the FIR filter resulting from sampling the basis functions at the desired

fractional locations. At each location (x, y) pointed by the MVs, the corresponding integer part is used to compute position (m_1, n_1) in the expansion coefficients buffer and the interpolated value is calculated according to (11).

V. Selection of Basis for MCP

Interpolation for MCP constitutes a significant amount of decoder complexity in a hybrid video codec. In order to employ the concepts of generalized interpolation for MCP, it has to be ensured that the resulting scheme is attractive for a practical implementation. Specifically, it should be possible to implement the design using simple fixed-point arithmetic and preferably it should involve multiplication-free filtering for hardware implementation. The well-known B-spline basis functions $\beta(x)$ do fall under the category of generalized basis functions, but as shown in [10], the approximation abilities can be improved, for the same computational cost, by sacrificing the regularity property of splines. Additionally, the prefilters for cubic and quintic splines have poles with floating point values that do not have simple binary representation. Hence, the rounding errors in a fixed-point implementation of the prefilters would prove detrimental to the interpolation quality because of the recursive nature of the IIR structures.

The choice of $\phi(x)$ directly influences the length of the resulting filter support. Using approximation theory, an expression for the ϕs that have minimal support for a given approximation order L has been derived in [10]. This class of functions, which are called maximal-order minimal support (MOMS) functions, is made of linear combinations of the B-spline of order r, denoted as $\beta^r(x)$, and its derivatives. The expression for MOMS is of the form

$$\phi(x) = \sum_{n=0}^{L-1} \lambda_n \frac{d^n}{dx^n} \beta^{L-1}(x - t) \tag{12}$$

where $\lambda_0 = 1$ and t is a shift parameter. Within this class, the minimization of the asymptotic constant [10] gives rise to the optimal MOMS, also known as O-MOMS.

The IIR prefilter for O-MOMS of third order has a causal and an anticausal pole. When increasing the order, the number of poles of the prefilter for O-MOMS increases, e.g., the fifth-order O-MOMS has two causal and two anticausal poles. It can then be implemented as two sets of first-order causal and anticausal IIR filters. In order to keep the IIR complexity and memory bandwidth under control and enable a multiplication-free filtering, we propose to approximate the frequency response of the O-MOMS prefilter, using only one causal and one corresponding anticausal pole that is a binary fraction, even for higher-order basis. Once the causal pole is chosen to be a simple binary fraction, the implementation of the causal and the corresponding anticausal IIR filters is guaranteed to be straightforward in fixed-point arithmetic. After selecting the prefilter, we use Theorem 1 from [14] that states that there exists a unique basis in the MOMS family that corresponds to the chosen prefilter. It involves the determination of weighting factors λ_n so that the FIR filter generated by sampling $\phi(x)$ at integer locations, when applied on the IIR filtered result, gives back the reconstructed picture samples at integer locations. Given the L and λ_n, the basis function $\phi(x)$ in (12) gets fixed and can be used to generate FIR filters for the desired fractional positions.

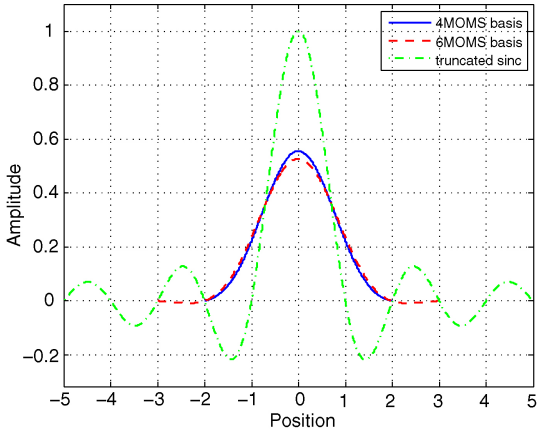

Fig. 4. Plot of MOMS and sinc basis functions. Blue solid: 4-tap MOMS. Red dash: 6-tap MOMS. Green dash-dot: truncated sinc. The compact support of MOMS is evident in comparison to the slow decay of sinc.

A. Design Procedure

The design procedure consists of the following steps:

1) denote the required number of FIR taps as $L + 1$;
2) obtain the poles p_i of Lth-order O-MOMS IIR prefilter [10];
3) choose a simple binary fraction b_1 close to the actual dominant causal pole p_i;
4) represent the resulting IIR filter in the form
$$H(z) = h_0 \cdot \frac{1}{(1 - b_1 z^{-1})(1 - b_1 z^1)}$$
where h_0 is a normalization factor. Rewriting it, we get
$$H(z) = \frac{1}{a_1 \cdot z^{-1} + a_0 + a_1 \cdot z^1}.$$
5) The FIR filter $Q(z)$ needed to cancel the effect of the IIR filter should satisfy $Q(z)\, H(z) = 1$, yielding
$$\Longrightarrow \; Q(z) = a_1 \cdot z^{-1} + a_0 + a_1 \cdot z^1$$
which when inverse transformed gives
$$q[k] = \{\cdots, 0, 0, a_1, a_0, a_1, 0, 0, \cdots\}.$$
6) Compute the unknown weights λ_n in (12) such that $\phi(k) = q[k]$ by solving a system of linear equations.
7) Given λ_n, compute the FIR coefficients by sampling $\phi(x)$ at the required fractional positions.

In this paper, we design two variants of the filter, namely, 4MOMS and 6MOMS, that use 4-tap FIR and 6-tap FIR, respectively. For 4MOMS, the pole b_1 is chosen to be equal to -0.5 and for 6MOMS, it is chosen to be equal to -0.625. The values of the basis functions $\phi(x)$ become very small as the position x tends toward the end of the support. Therefore, quantizing the FIR coefficients to the commonly used 8-bits per sample may result in zero coefficients. To counter this

TABLE I
FIR FILTER COEFFICIENTS CORRESPONDING TO 4-TAP AND 6-TAP MOMS BASIS RESULTING FROM THE DESIGN ALGORITHM IN SECTION V-A

MOMS Type	Frac. Position	FIR Coefficients
4-tap	Half sample	[7, 57, 57, 7]/128
4-tap	Quarter sample	[16, 67, 43, 2]/128
6-tap	Half sample	[−6, 77, 484, 484, 77, −6]/256
6-tap	Quarter sample	[−7, 156, 560, 377, 26, −3]/256

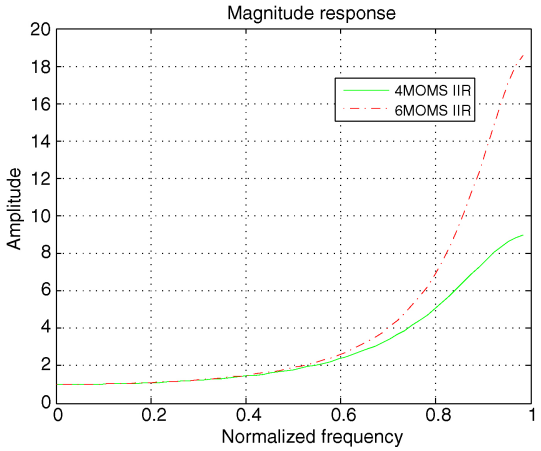

Fig. 5. Magnitude response of IIR prefilters. Green solid: 4MOMS. Red dash-dot: 6MOMS. The HF components are amplified by the prefilters, but the low-frequency components are relatively unaltered. The phase delay of the filters is zero due to symmetry of their impulse responses around origin.

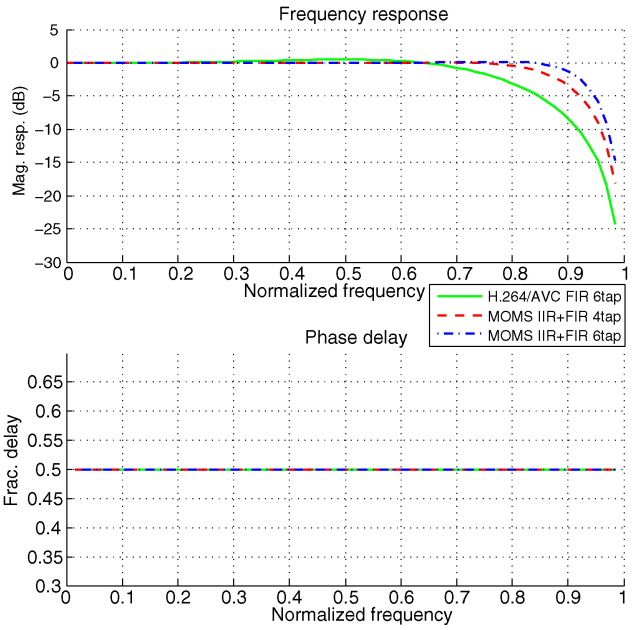

Fig. 6. Frequency response of MOMS and H.264/AVC 6-tap half-sample filter. The phase delay is equal to 0.5 for these filters and the phase plots overlap each other. The MOMS filters retain more HF components without causing distortions in the low-frequency regions in comparison to the H.264/AVC filter.

numerical accuracy effect, the value of L is increased until we get the required number of nonzero 8-bit coefficients. A plot of the 4MOMS basis with a support from -2 to 2, 6MOMS basis with a support from -3 to 3, and the sinc basis truncated to range -5 to 5 is shown in Fig. 4. The FIR coefficients generated using the described algorithm are given in Table I.

The magnitude response of the IIR prefilters for 4MOMS and 6MOMS are depicted in Fig. 5. It can be seen that the IIR prefilters have a HF boosting effect. The pole value of -0.625 for the 6MOMS prefilter, being closer to the unit circle than the pole of -0.5 in the case of 4MOMS, has even more HF amplification. The normalization factor g in (10) is not included in the magnitude response depicted in Fig. 5. The phase responses of both 4MOMS and 6MOMS prefilters have zero phase delay as they are symmetric around origin, due to the symmetric causal and anticausal filtering. Therefore, the fractional displacement (or phase delay) is solely caused by the FIR stage.

The magnitude and phase responses of the entire system, i.e., the combination of IIR and FIR components, are depicted in Figs. 6 and 7, corresponding to half-sample and quarter-sample shifts, respectively. For comparison, the frequency responses of the half-sample and quarter-sample filters in H.264/AVC are also shown. The quarter-sample H.264/AVC filter coefficients are computed by combining the half-sample filter coefficients $[1, -5, 20, 20, -5, 1]/32$ with an implicit integer-sample filter $[0, 0, 32, 0, 0, 0]/32$ using a linear interpolation $[1, 1]/2$ to result in $[1, -5, 52, 20, -5, 1]/64$. The phase delay of all the half-sample filters depicted in Fig. 6 are equal to the ideal phase delay of 0.5 due to the symmetry of the filter coefficients. However, as can be seen from Fig. 7, none of the filters provide a perfect quarter-sample shift over the entire spectrum. Nevertheless, in each of their passbands, the filters approximately provide a fractional delay of 0.25.

B. From Generalized to Classical Interpolation

In order to show the connection between the design in the previous section and the typical FIR-only interpolation systems, consider the values of the basis functions at integer locations resulting from the design. From step 5 of the algorithm in Section V-A, it can be noted that the basis function has a value of $\phi(1) = \phi(-1) = a_1$. A choice of $a_1 = 0$ makes the dependency on the z and z^{-1} terms vanish and in order to maintain a unit norm in the denominator, a_0 is set to unity. The IIR filter transfer function becomes $H(z) = 1$, which corresponds to no IIR prefiltering, i.e., the expansion coefficients are the same as the sample values. The FIR filter required to produce integer samples is then $Q(z) = 1/H(z) = 1$, i.e., no filtering of expansion coefficients to generate samples at integer locations. The sampling of the resulting basis at the desired fractional locations generate the FIR coefficients for producing the required phase delays. Hence, with $a_1 = 0$ and $a_0 = 1$, the generalized interpolation algorithm maps to an FIR-only system in accordance to the previous definition of interpolating basis functions that required the basis functions to be zero at all integer locations except at origin.

As examples, consider the third-order and seventh-order MOMS basis. The unknown weights λ_n of (12) have to be determined such that sampling these bases at integer positions result in $q[k]$ equal to $\{0, 0, 1, 0, 0\}$ for third order and

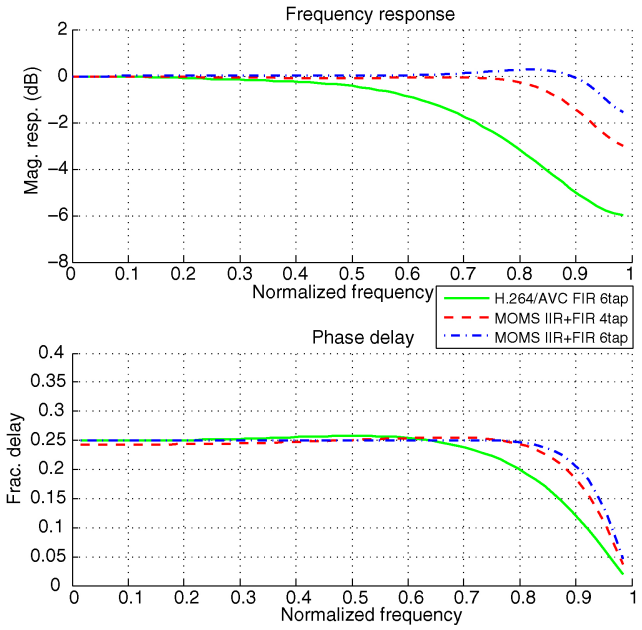

Fig. 7. Frequency response of MOMS and H.264/AVC 6-tap quarter-sample filter. The phase delay is approximately equal to 0.25 in the passbands. Like the half-sample case, the MOMS quarter-sample filters retain more HF components without causing distortions in the low-frequency regions in comparison to the H.264/AVC filter.

TABLE II
FILTER COEFFICIENTS CORRESPONDING TO 4-TAP AND 6-TAP MOMS-BASED FIR FILTERS DESCRIBED IN SECTION V-B

Filter Type	Frac. Position	Coefficients
4-tap FIR	Half sample	[−4, 36, 36, −4]/64
4-tap FIR	Quarter sample	[−4, 53, 17, −2]/64
6-tap FIR	Half sample	[2, −8, 38, 38, −8, 2]/64
6-tap FIR	Quarter sample	[1, −7, 56, 18, −5, 1]/64

$\{0, 0, 0, 0, 1, 0, 0, 0, 0\}$ for seventh order. To determine the weights, the values of the B-spline and its derivatives at integer positions are stored in the columns of a matrix $\mathbf{A}$. The values of $q[k]$ are used to form a column vector $\mathbf{b}$. Then, the solution for λ_n is computed as $\mathbf{A}^{-1}\mathbf{b}$. The resulting λ_n for n in range $[0, 3]$ are equal to $\{1, 0, -1/6, 0\}$ in case of third order and λ_n for n in range $[0, 7]$ are equal to $\{1, 0, -1/3, 0, 7/120, 0, -1/140, 0\}$ in case of seventh order. The FIR coefficients scaled to 6 bits are given in Table II.

VI. COMPLEXITY AND MEMORY BANDWIDTH ANALYSIS

Some important parameters for implementing a video codec are computational complexity, memory bandwidth, latency, etc. In order to understand the resource requirements of the generalized interpolation framework, a step-by-step analysis of the operations in various stages is provided. A major parameter affecting the computational complexity of an interpolation scheme is the support of the filter. In the proposed MCP approach, the FIR filters have short support but need an additional filtering stage: the IIR prefiltering. In this section, the complexity of the IIR and FIR stages are first analyzed separately and then an estimate of total complexity is obtained.

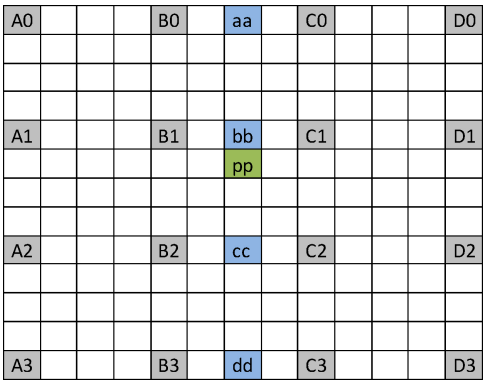

Fig. 8. Notation of different positions in a quarter luma sample resolution. Capital letters denote samples at integer positions. Fractional position of interest is shown as `pp`. Intermediate positions to be computed are `aa`, `bb`, `cc`, and `dd`.

The proposed scheme is compared to the case of FIR-only interpolation system with 6 taps, 8 taps, and 12 taps.

A. Complexity of the IIR Stage

The IIR filter resulting from the design process described in Section V has one causal and one anticausal pole and can be factorized as shown in (10). This filter can be implemented in a 2-D separable way using fixed-point arithmetic. Denote the reconstructed picture as `s[x, y]`, where `[x, y]` indicates the spatial position. The exact expressions of these filters, for the case of 4MOMS with $b_1 = -0.5$, without using multiplications are shown in Table III in the form of pseudocode. The causal and anticausal filter operations are separately tabulated for the horizontal and vertical directions. To compute the number of IIR operations, the number of shifts and adds in each stage of IIR filtering is counted, which lead to six shifts and eight adds per sample of a reconstructed picture. A bit depth of 8 bits/sample of a reconstructed picture and 16 bits/sample for filter outputs is employed. This results in 10 bytes of memory read and 8 bytes of memory write, totaling to 18 bytes, as shown in Table III. Assuming a memory access rate of 4 bytes/cycle, the number of cycles for the memory accesses can be estimated as $18/4 = 4.5$ cycles. Furthermore, assuming single arithmetic operation per cycle, the number of arithmetic cycles required can be calculated as $6 + 8 = 14$ cycles. Finally, the total number of cycles is computed as the sum of the arithmetic and memory access cycles, resulting in $4.5 + 14 = 18.5$ cycles per sample, for the entire IIR process.

B. Complexity of the FIR Stage

In a video codec with flexible block sizes, each block gets predicted using the MV that is indicated in the bitstream. Typically, an MV uses quarter-sample accuracy, resulting in 16 possible positions including the full-sample position. Let the desired location within a grid of fractional samples be denoted by `pp`, as shown in Fig. 8. Let `(xoffset, yoffset)` be the fractional shift relative to the integer position B1, where `xoffset` and `yoffset` are specified in quarter-sample units. In case of 4MOMS, the following horizontal filtering is per-

TABLE III
COMPLEXITY ANALYSIS OF THE IIR PREFILTER FOR 4-TAP MOMS

Filter Type	Filter Process	Shift	Add	Load	Store
Left to right	`tmp=(src[i]«2)-(tmp»1)` `coeff[i]=tmp`	2	1	1	2
Right to left	`tmp=(src[i]«2)-(tmp»1)` `coeff[i]=coeff[i]+tmp-(src[i]«2)`	2	3	3	2
Top to bottom	`tmp=src[i]-(tmp»1)` `coeff[i]=tmp`	1	1	2	2
Bottom to top	`tmp=src[i]-(tmp»1)` `coeff[i]=coeff[i]+tmp-src[i]`	1	3	4	2
Overall	Total cycles IIR	6	8	10	8

Cycles for the arithmetic operations in each stage are counted in terms of the number of shift/add and the memory bandwidth usage is counted in terms of the number of bytes accessed in load/store operations. Input to IIR filter is assumed to be 8 bits and the output is 16 bits. Detailed discussion of the operations can be found in Section VI-A.

formed to generate intermediate samples `aa`, `bb`, `cc`, and `dd`:

```
aa = fir4tap ( A0, B0, C0, D0, xoffset)
bb = fir4tap ( A1, B1, C1, D1, xoffset)
cc = fir4tap ( A2, B2, C2, D2, xoffset)
dd = fir4tap ( A3, B3, C3, D3, xoffset)
```

where the function `fir4tap()` performs a 4-tap FIR filtering with the coefficients selected from a table according to the specified `xoffset`. The intermediate values are then used for vertical interpolation

```
pp = fir4tap ( aa, bb, cc, dd, yoffset).
```

The filter `fir4tap` is called even when `xoffset` or `yoffset` is equal to 0, in which case it reduces to a 3-tap filter because of symmetry. The operations of 6MOMS are similar to that of 4MOMS. The case of offset equal to 0 is still a symmetric 3-tap filter, but the values at other offsets are generated using 6-tap filters.

Consider a block size of 4×4 to be interpolated using 4MOMS. For vertically filtering each sample position, the current position along with one sample above and two samples below are required for the FIR operation. Hence, for the top row, one row above the top row is required and for the bottom row, two rows below it are required. This means that 7×4 samples are required for the vertical interpolation, which needs to be generated during the horizontal filtering step. The filter to generate half-sample positions is symmetric, hence, in an implementation with multiplications (MUL) and additions (ADD), 2 MUL and 3 ADD are needed, leading to 56 MUL and 84 ADD for the entire horizontal filtering of 7×4 samples. The coefficients for the quarter-sample vertical filtering are, however, not symmetric due to the position of `pp` in the grid of `aa`, `bb`, `cc`, and `dd`. Therefore, 4 MUL and 3 ADD are needed for each filtering, resulting in 64 MUL and 48 ADD for the entire 4×4 block. The total of horizontal and vertical filtering would then be 120 MUL and 132 ADD.

C. Total Complexity of IIR and FIR Stages

The complexity estimation is performed for all the 16 positions in a quarter-sample grid. It is then repeated for block sizes of 8×8 to 64×64, the prediction sizes in the current HEVC test model. Note that the overhead due to

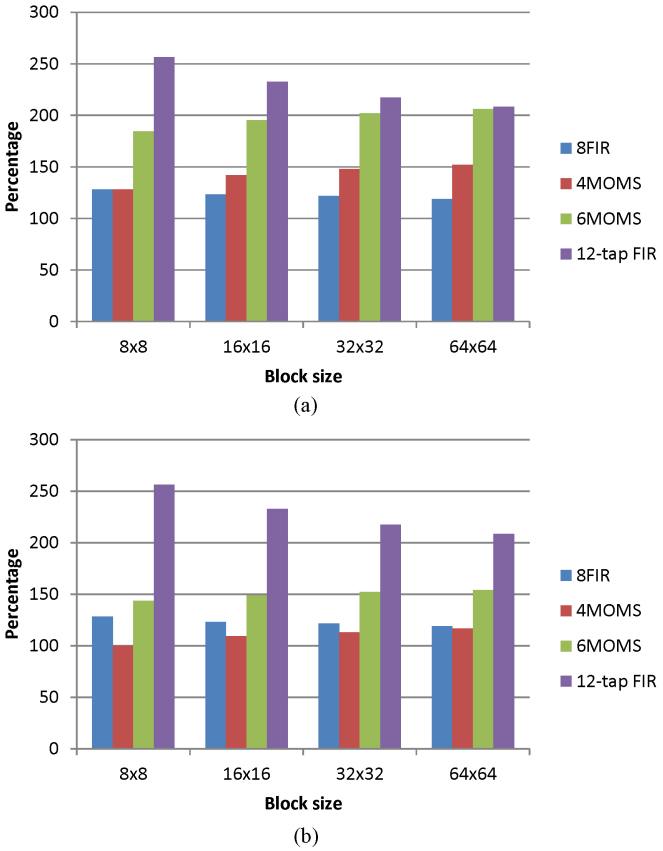

Fig. 9. Comparison of 4MOMS, 6MOMS, 8-tap FIR, and 12-tap FIR filter relative to a 6-tap FIR filter. The worst case cycles for each approach is expressed as a percentage of the worst case cycles of a 6-tap FIR filter for each block size. (a) Worst case complexity estimates for single hypotheses prediction. (b) Worst case complexity estimates for two hypotheses prediction.

samples required for vertical filtering that go beyond the block boundary decreases as the block size increases. The worst case cycles, including memory accesses, for computing one prediction sample, is then expressed as a percentage of the worst case cycles that would be needed for a 6-tap FIR-only system. The result of this computation is depicted in Fig. 9(a). The complexity estimate is based on shifts and additions for the IIR operations and multiplications and additions for the FIR operations. However, the FIR coefficients have been chosen so as to enable a simple implementation using shifts and additions. The worst case analysis shows that a 12-tap FIR

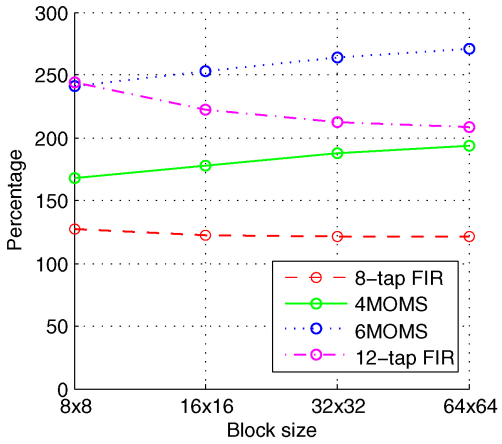

Fig. 10. Ratio of the estimated average complexities of 4MOMS, 6MOMS, 8-tap FIR, and 12-tap FIR relative to a 6-tap FIR filter. The average complexity of 8-tap FIR is around 25% higher than a 6-tap FIR. The average complexity of 6MOMS is higher than a 12-tap FIR even though the worst case complexity (Fig. 9) is lower.

TABLE IV
LD CODING TEST: AVERAGE BITRATE SAVINGS OF 6MOMS COMPARED TO 6-TAP, 8-TAP, AND 12-TAP FILTERS

LD	Y-BD Bitrate %		
	Ref. 6-Tap FIR	Ref. 8-Tap FIR	Ref. 12-Tap FIR
Class B	−1.2	−0.7	−0.9
Class C	−5.2	−4.0	−2.5
Class D	−9.4	−7.6	−5.1
Class E	1.4	2.2	1.4
Average	−3.8	−2.7	−1.9

filter has a high overhead for small block sizes in comparison to both 4MOMS and 6MOMS.

The complexity analysis is extended to B-pictures, which can have up to two hypotheses for predicting each block. The complexity of FIR filtering is doubled for the case of two hypotheses prediction. However, the IIR prefiltering needs to be performed only once for each reference picture. Hence, the complexity advantage of generalized interpolation increases with an increasing number of hypothesis, as can be seen in Fig. 9(b) for biprediction. Fig. 10 provides a ratio of the average complexity estimates assuming a uniform distribution of fractional-sample positions for MCP for one and two hypotheses predictions. In all the cases, the highest per sample complexity occurs for the 12-tap FIR when operating on small block sizes because of the overhead due to the samples required beyond block borders.

Notwithstanding the lower arithmetic complexity of MOMS, the inclusion of a prefilter can affect the overall latency of the MCP process. Also, the computed expansion coefficients are saved using a bit depth of 16-bits per coefficient, which increases the overall memory access of the prefiltering and MCP process. Although the additional stage of IIR filtering can increase the latency due to the accesses to entire reconstructed picture, this issue can be controlled because the entire IIR process is independent of the motion information of the subsequent pictures. The IIR process can be started

TABLE V
LD CODING TEST: AVERAGE BITRATE SAVINGS OF 4MOMS COMPARED TO 6-TAP, 8-TAP, AND 12-TAP FILTERS

LD	Y-BD Bitrate %		
	Ref. 6-Tap FIR	Ref. 8-Tap FIR	Ref. 12-Tap FIR
Class B	−0.6	−0.1	−0.3
Class C	−3.9	−2.7	−1.2
Class D	−6.9	−5.0	−2.3
Class E	0.7	1.5	0.8
Average	−2.7	−1.7	−0.8

TABLE VI
RA CODING TEST: AVERAGE BITRATE SAVINGS OF 6MOMS COMPARED TO 6-TAP, 8-TAP, AND 12-TAP FILTERS

RA	Y-BD Bitrate %		
	Ref. 6-Tap FIR	Ref. 8-Tap FIR	Ref. 12-Tap FIR
Class A	0.1	−0.1	0.0
Class B	−0.4	0.0	−0.2
Class C	−3.2	−1.6	−0.9
Class D	−5.8	−3.0	−1.7
Average	−2.5	−1.2	−0.7

immediately after a picture is reconstructed, without waiting for the MVs of the subsequent pictures. On the other hand, the FIR filtering is not independent of the motion information for MCP. Therefore, increasing the FIR filter length, has a direct impact on the MCP latency. However, the impact of using FIR-only filters with long support can be reduced using single-instruction multiple data architecture compared to sequential execution. Therefore, the complexity of generalized interpolation can be analyzed with more details about the underlying architecture. However, the operations on different rows or columns are independent of each other in each direction IIR prefiltering. Hence, the prefiltering of different rows or columns can be done in parallel to reduce the latency.

VII. SIMULATION RESULTS

The RD performance of the proposed generalized interpolation is evaluated using test sequences defined in the call for proposals (CfPs) issued by the Joint Collaborative Team on Video Coding (JCT-VC) [13]. Two constraint sets restricting the coding structures are defined in the CfP as follows.

1) *Low delay (LD)*: no picture reordering in decoder.
2) *Random access (RA)*: structural delay not larger than eight pictures.

The software used for the comparison of RD performances of different fractional-sample interpolation schemes is the first test model [4] of the ongoing HEVC standardization. All 18 sequences from the HEVC test dataset are used for testing and all frames of each sequence are used for the RD performance comparison. The prediction block size goes up to 64×64 with rectangular shapes like 8×16 enabled. RD-optimized mode selection, adaptive loop filter, deblocking filter, context-adaptive binary arithmetic coding, and quarter-sample accuracy MCP with generalized biprediction are used.

TABLE VII
RA CODING TEST: AVERAGE BITRATE SAVINGS OF 4MOMS COMPARED TO 6-TAP, 8-TAP, AND 12-TAP FILTERS

RA	Y-BD Bitrate %		
	Ref. 6-Tap FIR	Ref. 8-Tap FIR	Ref. 12-Tap FIR
Class A	0.1	0.2	0.1
Class B	−0.3	0.2	0.0
Class C	−2.4	−0.9	−0.9
Class D	−4.2	−1.3	0.0
Average	−1.9	−0.5	0.0

TABLE VIII
SEQUENCES WITH LARGE BITRATE SAVINGS WITH 6MOMS INTERPOLATION IN COMPARISON TO 6-TAP, 8-TAP, AND 12-TAP FIR FILTERS IN LD AND RA CODING

6MOMS	Y-BD Bitrate %					
	Ref. 6-Tap		Ref. 8-Tap		Ref. 12-Tap	
	LD	RA	LD	RA	LD	RA
Blowingbubbles	−9.1	−5.0	−8.1	−2.6	−3.7	−0.9
BQSquare	−26.1	−16.7	−21.3	−9.0	−16.7	−6.3
Partyscene	−13.5	−8.2	−11.7	−4.5	−7.1	−3.0

The 4-tap and 6-tap MOMS are compared with three interpolation methods: the 12-tap 2-D separable FIR filters of the first HEVC draft [4], the 8-tap 2-D separable filters of HM-6.0, and 6-tap 2-D separable FIR filters with quarter-sample coefficients as $[8, -35, 227, 73, -23, 6]/256$ and half-sample coefficients as $[5, -33, 156, 156, -33, 5]/256$. The number of reference pictures for MCP is set as two, as per the JCT-VC common coding conditions for comparison [16]. The performance improvements are measured in terms of the average bitrate difference over all rate points using the Bjøntegaard delta bitrate (BDBR) metric [17]. Negative BDBR indicates an improvement in the RD performance. The 6-tap filter used for reference performs approximately 3%–4% better than the 6-tap interpolation scheme in H.264/AVC mainly because of a higher precision implementation of the 2-D separable filtering, which does not involve any rounding after the first 1-D filtering. The simulation results averaged over the entire testset for the LD cases are shown in Tables IV and V. The GOP8 RA cases with hierarchical B pictures can be found in Tables VI and VII.

The sequences are grouped into five classes according to [16]. In the LD scenario, 6MOMS gives an average BD rate reduction of 1.9% compared to the 12-tap filter, 2.7% compared to the 8-tap filter, and 3.8% compared to the 6-tap filter. The gains due to MOMS filtering in the GOP8 RA scenario are slightly less: 6MOMS gives an average BD rate reduction of 0.7% compared to the 12-tap filter, 1.2% compared to the 8-tap filter, and 2.5% compared to the 6-tap filter. Compared to the 12-tap FIR filter, 4MOMS provides a moderate gain, whereas the gains compared to 6-tap and 8-tap filters are higher. Table VIII lists some sequences that show significant bitrate savings by 6MOMS interpolation. It is interesting to see that the sequence *BQSquare* shows a reduction in BD rate of around 26% compared to the 6-tap FIR case. In addition to the BD rate savings, RD curves for two sequences are depicted in Fig. 11(a) and (b).

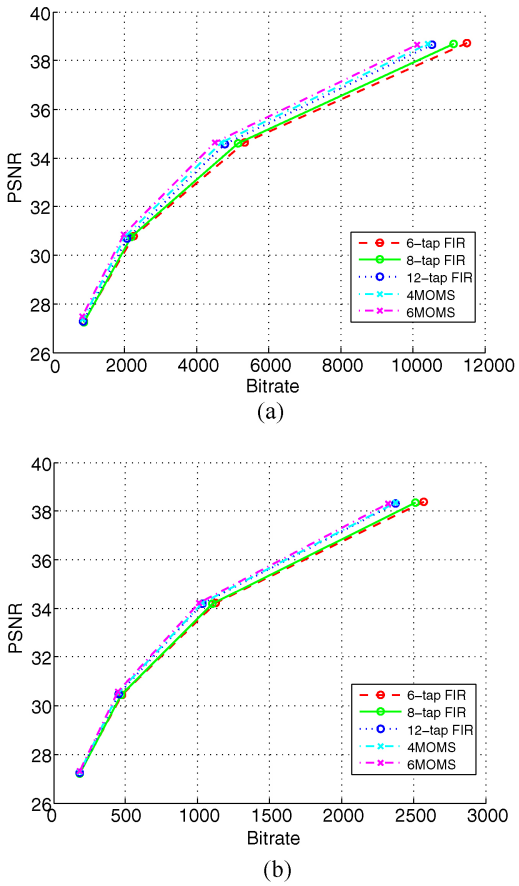

Fig. 11. RD comparison of 6-tap, 8-tap, and 12-tap FIR with 4MOMS and 6MOMS for two sequences. (a) RD curves for the sequence *Partyscene*. (b) RD curves for the sequence *Blowingbubbles*.

Performance Analysis: The observed RD performance gains in Tables IV–VII show large sequence-to-sequence variation. It is known that the efficiency of MCP depends on the spectral content of original video, noise, and other unmodeled causes. In order to quantify these factors for the tested video sequences, we performed an experiment to measure signal and noise content. Each original sequence was filtered using a 17-tap FIR low-pass filter with a magnitude cutoff of 0.8π and zero phase delay. The resulting signal was subtracted from the original to yield components in HF range. To estimate signal and noise components, we consider small variation in the HF range as noise. To this end, a thresholding operation was performed on the HF signal to categorize each amplitude. Values that were below a threshold of ± 2 were categorized as noise components and higher values were treated as signal components. The estimated signal and noise power for the video sequences in the HEVC dataset is plotted in Fig. 12.

It can be seen from Fig. 12 that the HF signal and noise content show a large variation from sequence to sequence. All tested filters, i.e., 6-tap, 8-tap, 12-tap, 4MOMS, and 6MOMS have good frequency response in the frequency range below 0.8π. Due to the fact that the main difference in the tested filters is in the HF range, the RD gain changes with the signal-to-noise ratio (SNR) in the HF range. We can observe

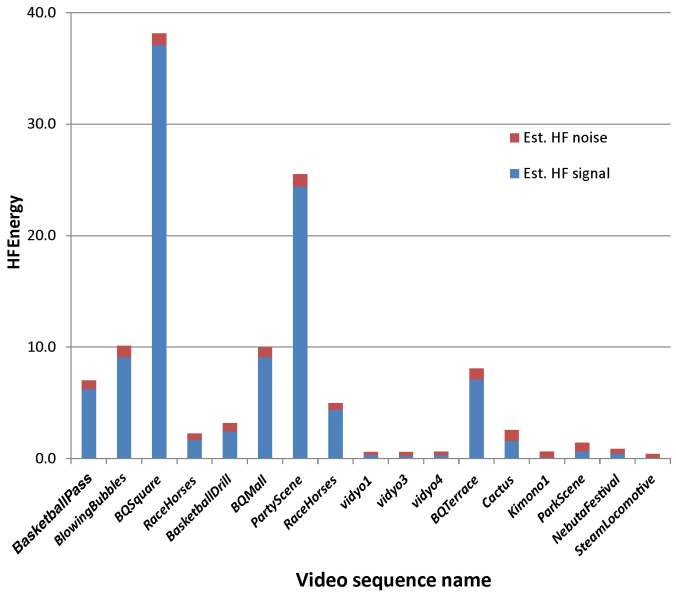

Fig. 12. Estimated signal and noise power in HF range for the video sequences in the HEVC dataset. Sequences like *BQSquare* and *Partyscene* have high SNR in the considered HF range, whereas sequences like *Vidyo* and *Kimono* have low SNR in the considered range.

TABLE IX
AVERAGE BITRATE SAVINGS OVER VCEG SEQUENCES IN LD CODING

LD	Y-BD Bitrate %					
	Ref. 6-Tap FIR		Ref. 8-Tap FIR		Ref. 12-Tap FIR	
	6MOMS	4MOMS	6MOMS	4MOMS	6MOMS	4MOMS
Foreman	−2.0	−1.5	−0.3	0.2	−0.4	0.0
Mobile	−8.4	−5.6	−7.3	−4.6	−3.5	−0.6
Tempete	−6.1	−4.4	−5.1	−3.5	−3.5	−1.8
Flower	−6.9	−4.6	−5.6	−3.3	−3.9	−1.6
Nuts	−0.3	1.0	0.5	1.8	0.2	1.6
Average	−4.7	−3.0	−3.6	−1.9	−2.2	−0.5

Comparison of 6MOMS and 4MOMS to 6-tap, 8-tap, and 12-tap filters.

a broad correlation of the RD gains in Tables IV–VII and the SNR plot in Fig. 12. Sequences like *BQSquare* and *Partyscene* have a SNR much higher than other sequences due to the spatial details in the videos. Therefore, an improved interpolation filter that has the desired frequency response for a larger range of frequencies shows such a high gain (see Table VIII). For other sequences, with moderate SNR in HF range, the RD gains are correspondingly moderate. For the Class E sequences, however, a marginal loss is observed. These sequences have low SNR (*Vidyo1*, *Vidyo3*, and *Vidyo4*) as evident from Fig. 12 and hence an interpolation filter retaining more HF content results in a noisy prediction signal, leading to RD performance loss. It can also be noted that the video sequences with high resolution (Classes A and B) are less sensitive to interpolation filtering. This can be explained by the fact that these sequences are sampled with a high spatial sampling frequency for generating high-resolution videos. This can lead to less signal content, resulting in low SNR in the HF range. Apart from the video sequences in the HEVC dataset, we also evaluated the RD performance on the sequences defined in the VCEG-KTA dataset. The RD performance gains observed on these sequences can be found in Table IX. The proposed schemes provide overall bitrate reduction for the VCEG sequences as well.

The source code of our implementation, complexity analysis, and the test results can be downloaded from [15].

VIII. CONCLUSION

Fractional-sample estimation for MCP plays a critical role in a hybrid video codec. For improving the fractional-sample estimation, a generalized interpolation framework using MOMS basis was described. It involved breaking the processing into IIR and FIR filters. The worst case complexity of MOMS was shown to be less than increasing the support of FIR filters to length 12. The generalized interpolation with MOMS basis showed improved RD performance compared to a 12-tap FIR filter and significantly better performance compared to 6-tap filters. The improvements were mainly observed for sequences with fine spatial details. However, the latency and memory access involved in MCP may increase due to the additional IIR prefiltering stage. Parallel processing can be very useful to reduce the prefiltering latency by simultaneously operating the 1-D filters on independent rows and columns.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[2] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 604–612, Apr. 1993.
[3] T. Wedi, "Adaptive interpolation filter for motion compensated hybrid video coding," in *Proc. Picture Coding Symp.*, Apr. 2001, pp. 509–512.
[4] T. Wiegand, W.-J. Han, J.-R. Ohm, and G. J. Sullivan, "High efficiency video coding (HEVC) text specification working draft 1," document JCT-VC C403, Guangzhou, China, 2010.
[5] K. McCann, W.-J. Han, I.-K. Kim, J.-H. Min, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y.-M. Hong, M.-S. Cheon, and N. Shlyakhov, "Video coding technology proposal by Samsung (and BBC)," document JCT-VC A124, Dresden, Apr. 2010.
[6] K. Ugur, K. Andersson, A. Fuldseth, G. Bjøntegaard, L. P. Endresen, J. Lainema, A. Hallapuro, J. Ridge, D. Rusanovskyy, C. Zhang, A. Norkin, C. Priddle, T. Rusert, J. Samuelsson, R. Sjoberg, and Z. Wu, "High performance, low complexity video coding and the emerging HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1688–1697, Dec. 2010.
[7] T. Wedi, "Adaptive interpolation filters and high-resolution displacements for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 484–491, Apr. 2006.
[8] M. Karczewicz, Y. Ye, and P. Chen, "Switched interpolation filter with offset," document COM16-C463, ITU-T, Apr. 2008.
[9] M. Unser, "Sampling: 50 years after Shannon," *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.
[10] T. Blu, P. Thévenaz, and M. Unser, "MOMS: Maximal-order interpolation of minimal support," *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1069–1080, Jul. 2001.
[11] A. Goshtasby, F. Cheng, and B. A. Barsky, "B-spline curves and surfaces viewed as digital filters," *Comput. Vision, Graphics, Image Process.*, vol. 52, no. 2, pp. 264–275, Nov. 1990.
[12] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.
[13] ITU-T Q6/16, "Joint call for proposals on video compression technology," document VCEG-AM91, ITU-T, Kyoto, Japan, Jan. 2010.
[14] T. Blu, P. Thévenaz, and M. Unser, "High-quality causal interpolation for online unidimensional signal processing," in *Proc. EUSIPCO*, Sep. 2004, pp. 1417–1420.
[15] H. Lakshman, H. Schwarz, D. Marpe, and T. Wiegand, "CE3: Luma interpolation using MOMS," document JCT-VC D056, Jan. 2011.
[16] F. Bossen, "Common test conditions and software reference configurations," document JCT-VC C500, Oct. 2010.
[17] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," document VCEG-M33, ITU-T, Apr. 2001.

Haricharan Lakshman received the B.E. degree from the National Institute of Technology Karnataka, Surathkal, Karnataka, India, in 2002, and the M.S. degree from the University of Erlangen-Nuremberg, Erlangen, Germany, in 2008. He is currently pursuing the Ph.D. degree with the Technical University of Berlin, Berlin, Germany.

He is currently a Research Associate with Fraunhofer HHI, Berlin. From 2002 to 2006, he was an Engineer with Ittiam Systems, Bangalore, India. From 2006 to 2007, he was a Research Associate with Fraunhofer IIS, Erlangen. From 2011 to 2012, he was a Visiting Researcher with Stanford University, Stanford, CA. His current research interests include image processing, video coding, and 3-D video and semantic analysis.

Heiko Schwarz received the Dipl.-Ing. degree in electrical engineering and the Dr.-Ing. degree from the University of Rostock, Rostock, Germany, in 1996 and 2000, respectively.

In 1999, he joined the Image and Video Coding Group, Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, Germany. Since then, he has been contributing successfully to the standardization activities of the ITU-T Video Coding Experts Group (ITU-T SG16/Q.6-VCEG) and the ISO/IEC Moving Pictures Experts Group (ISO/IEC JTC 1/SC 29/WG 11-MPEG). During the development of the scalable video coding extension of H.264/AVC, he was the Co-Chair of several ad hoc groups of the Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, investigating particular aspects of the scalable video coding design. He is a co-editor of ITU-T H.264 and ISO/IEC 14496-10 and a Software Coordinator of the SVC reference software.

Thomas Wiegand (M'05–SM'08–F'11) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Hamburg, Germany, in 1995, and the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Erlangen, Germany, in 2000.

He is currently a Professor with the Department of Electrical Engineering and Computer Science, Berlin Institute of Technology, Berlin, Germany, chairing the Image Communication Laboratory, and is jointly heading the Department of Image Processing, Fraunhofer Institute for Telecommunications-Heinrich Hertz Institute, Berlin. In 2000, he was the Head of the Image Communication Group, Department of Image Processing, Heinrich Hertz Institute. From 1993 to 1994, he was a Visiting Researcher with Kobe University, Kobe, Japan. In 1995, he was a Visiting Scholar with the University of California at Santa Barbara, Santa Barbara. From 1997 to 1998, he was a Visiting Researcher with Stanford University, Stanford, CA, and a consultant with 8×8, Inc., Santa Clara, CA. From 2006 to 2008, he was a consultant with Stream Processors, Inc., Sunnyvale, CA. From 2007 to 2009, he was a consultant with Skyfire, Inc., Mountain View, CA. Since 2006, he has been a Technical Advisory Board Member with Vidyo, Inc., Hackensack, NJ. From 2011 to 2012, he was a Visiting Professor with Stanford University. His current research interests include video processing and coding, multimedia transmission, and computer vision and graphics.

Dr. Wiegand was the recipient of the SPIE VCIP Best Student Paper Award in 1998, and the Fraunhofer Award and the ITG Award of the German Society for Information Technology in 2004. The projects that he co-chaired for the development of H.264/MPEG-4 AVC standard received the ATAS Primetime Emmy Engineering Award and two NATAS Technology and Engineering Emmy Awards in 2008. He received the Innovations Award of the Vodafone Foundation, the EURASIP Group Technical Achievement Award, and the Best Paper Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2009, the Eduard Rhein Technology Award in 2010, the Best Paper Award of EURASIP and the Karl Heinz Beckurts Award in 2011, and the IEEE Masaru Ibuka Technical Field Award in 2012. He was a Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY for its special issue on the "H.264/AVC video coding standard" in July 2003, its special issue on "Scalable video coding-standardization and beyond" in September 2007, and its special section on the "Joint call for proposals on high efficiency video coding (HEVC) standardization." Since January 2006, he has been an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. Since 1995, he has been an active participant in standardization for multimedia with successful submissions to ITU-T VCEG, ISO/IEC MPEG, 3GPP, DVB, and IETF. In October 2000, he became an Associate Rapporteur of ITU-T VCEG. In December 2001, he became an Associate Rapporteur or Co-Chair of the Joint Video Team. In February 2002, he became an editor of the H.264/MPEG-4 AVC video coding standard and its extensions (FREXt and SVC). From 2005 to 2009, he was the Co-Chair of MPEG Video.