

GENERALIZED INTERPOLATION FOR MOTION COMPENSATED PREDICTION

Haricharan Lakshman¹, Heiko Schwarz¹, Thierry Blu² and Thomas Wiegand^{1,3}

¹Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, Germany.

²Department of Electronic Engineering, The Chinese University of Hong Kong, HK.

³Image Communication Chair, Technical University of Berlin, Germany

ABSTRACT

Fractional sample interpolation with 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 implemented generalized interpolation using a combination of short IIR and FIR filters. An efficient multiplication-free design of the algorithm that is suited for hardware implementation is shown. Compared to a 6-tap FIR interpolation filter, average rate savings of 3.1% are observed. A detailed analysis of the complexity and memory bandwidth cycles compared to existing interpolation techniques for MCP is provided.

Index Terms— video coding, motion compensated prediction, interpolation, B-spline

1. INTRODUCTION

Motion compensated prediction (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 neighboring samples. In the H.264/AVC standard [1], a 6-tap filter is used to generate the half-sample positions followed by a 2-tap filter for generating the quarter-sample positions.

Several techniques for improving the interpolation quality in MCP have been investigated, e.g., Adaptive Interpolation Filters (AIF) [2], Switched Interpolation Filters with Offset (SIFO) [3], 1D Directional filters [4]. It was also identified that a portion of the gains relative to H.264/AVC stems from the fact that they use higher precision arithmetic. Another approach to improve the quality is to use filters with longer support like 8-tap or 12-tap filters that better approximate the ideal low pass response. For example, a 12-tap filter is used for MCP in the first test model of the ongoing HEVC standardization [9]. Although increasing the filter support improves the quality of interpolation in general, it increases the computational complexity and can introduce ringing artifacts around edges. In [5], we proposed a framework for Generalized Interpolation based MCP (Fig. 1). We showed

that significant gains can be achieved even when using non-adaptive short support filters. The reference picture samples were modelled using O-MOMS basis [6] that have good approximation properties for a given support length. The computation of model parameters involved IIR filtering and the fractional sample estimation was achieved 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.

In this paper, we propose an improved design of generalized interpolation for MCP by exploiting the available degrees of freedom in order to increase the compression efficiency and reduce the computational complexity compared to the previous design [5]. Specifically, we reduce the complexity of the IIR filter stage and provide a multiplication-free design of the IIR and FIR stages, which is more suitable for hardware implementation. Furthermore, we provide a detailed analysis of the complexity in terms of the worst case and average number of operations and cycles for memory accesses and compare it to the case of commonly used FIR filter based MCP.

2. STRUCTURE OF GENERALIZED INTERPOLATION

In the classical approach to interpolation, a discrete signal is mapped onto a continuous signal using interpolating basis functions, i.e. the basis functions pass through zero at all integer locations except the origin, where they have a value of unity (e.g. *sinc* function). Given a set of samples $s[k]$ corresponding to integer locations $k \in Z$, the task of interpolation is to estimate the sample value $g(x)$ at a fractional location x . The classical interpolation formula is of the form,

$$g(x) = \sum_{k \in Z} s[k] \cdot \phi_{\text{int}}(x - k), \quad (1)$$

where ϕ_{int} is chosen to satisfy the interpolating condition. Eq. (1) 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

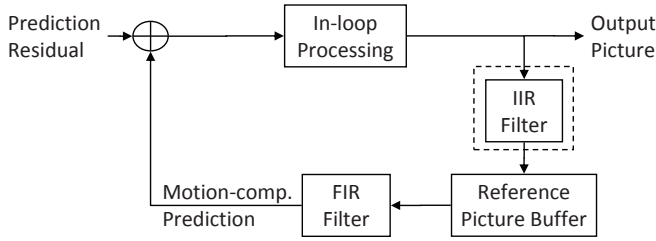


Fig. 1. Motion-compensated prediction using generalized interpolation. The IIR filtering block in dashed lines constitutes the main difference to the standard techniques.

sampled to generate an FIR filter. The interpolation process is then a convolution of the reference picture samples with the corresponding FIR filter. In spite of this elegant setup, there are many practical issues, most important being the slow decay of the ideal *sinc* function.

In generalized interpolation [7], the zero crossing constraint is not imposed on the basis functions and the problem is reformulated as,

$$f(x) = \sum_{k \in \mathbb{Z}} c[k] \cdot \phi(x - k), \quad (2)$$

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 Eq. (1) is that the expansion coefficients are not the signal samples $s[k]$ anymore. The expansion coefficients $c[k]$ are determined so that the values generated at integer locations using Eq. (2) match exactly with the signal samples $s[k]$. This can be done using the inverse filtering approach [8], where the signal samples $s[k]$ are filtered using the IIR filter, $H(z) = 1/(\sum_{n \in \mathbb{Z}} \phi(n)z^{-n})$.

3. SELECTION OF BASIS FOR MOTION COMPENSATED PREDICTION

The choice of $\phi(x)$ directly influences the length of the resulting filter support. Using approximation theory, an expression for the ϕ 's that have minimal support for a given approximation order L has been derived in [6]. This class of functions, which are called Maximal Order Minimal Support functions (MOMS) is made of linear combinations of the B-spline and its derivatives. The expression for MOMS is of the form,

$$\phi(x) = \sum_{n=0}^{L-1} \lambda_n \frac{d^n}{dx^n} \beta^{L-1}(x - d), \quad (3)$$

where $\lambda_0 = 1$ and d is a shift parameter corresponding to the support of $\phi(x)$. Within this class, the minimization of the asymptotic constant [6] gives rise to the optimal MOMS, also known as O-MOMS.

Design procedure

1. Denote the required number of FIR taps as $L + 1$.
2. Obtain the poles p_i of L th order O-MOMS IIR pre-filter [6].
3. Choose a simple binary fraction b_1 close to 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 constant. 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 IIR filter should satisfy $Q(z) H(z) = 1$, yielding

$$\Rightarrow Q(z) = a_1 \cdot z^{-1} + a_0 + a_1 \cdot z^1,$$

which when inverse transformed gives,

$$q[k] = \{\dots, 0, 0, a_1, a_0, a_1, 0, 0, \dots\}.$$

6. Compute the unknown weights λ_n in Eq. 3 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.

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, we have to ensure that the resulting scheme is attractive for a practical implementation. Specifically, the design should involve simple fixed point arithmetic and preferably have multiplication-free filtering for hardware implementation.

The IIR filter, also denoted as prefilter, for O-MOMS of 3rd order has a causal and an anti-causal pole. When increasing the order, the number of poles of O-MOMS increases, e.g. the 5th order O-MOMS has 2 causal and 2 anti-causal poles. It can then be implemented as 2 sets of 1st order causal and anti-causal 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 anti-causal pole that is a binary fraction, even for higher order basis. After selecting the prefilter, we use the Theorem 1 from [11], that states that there exists a unique basis in the MOMS family that corresponds to the chosen prefilter. Considering the above factors, the filter design is done according to the algorithm shown in the box above. 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, pole b_1 is chosen to be -0.5 and for 6MOMS, the pole b_1 is chosen to be -0.625 .

4. COMPLEXITY AND MEMORY BANDWIDTH ANALYSIS

One of the important factors affecting the complexity of an interpolation scheme is the support of the filter. In our proposed approach, the FIR filters have short support but need an additional stage of IIR prefiltering. In this section, we analyze the complexity of IIR and FIR separately to compute the overall complexity and compare it to the case of FIR-only system with a long support to achieve a similar performance.

The IIR filter resulting from the design process described in Sec. 3 has one causal and one anti-causal pole. This filter is implemented in a 2D separable way using fixed point arithmetic. The output picture is denoted as $s[x, y]$, where $[x, y]$ indicate the spatial position. The exact representation of these filters, for the case of 4MOMS with $b_1 = -0.5$, without using multiplications are,

Causal horizontal filter:

$$h1[x] = (s[x] \ll 2) - (h1[x-1] \gg 1)$$

Anti-causal horizontal filter:

$$h2[x] = (s[x] \ll 2) - (h2[x+1] \gg 1)$$

$$h[x] = h1[x] + h2[x] - (s[x] \ll 2)$$

Causal vertical filter:

$$v1[y] = h[y] - (v1[y-1] \gg 1)$$

Anti-causal vertical filter:

$$v2[y] = h[y] - (v2[y+1] \gg 1)$$

$$v[y] = v1[y] + v2[y] - h[y].$$

To compute the number of IIR operations, we counted the number of shifts and adds in each stage of IIR filtering, which lead to 6 shifts and 8 adds per sample of a reconstructed picture. We use 16-bit representations for filter outputs resulting in 10 bytes of memory read and 8 bytes of memory write, totalling to 18 bytes. Assuming a memory access of 4 bytes/cycle, it needs $18/4 = 4.5$ cycles of memory access and $6 + 8 = 14$ cycles for arithmetic operations, resulting in 18.5 cycles per sample for the entire IIR process.

In a video codec with flexible block sizes, each block gets predicted using the motion vector (MV) that is indicated in the bitstream. Typically, a MV uses quarter-sample accuracy, resulting in 16 possible positions including the full-sample position. Let the sample location of pp in Fig. 2 be $(xoffset, yoffset)$ with respect to position $B1$, where $xoffset$ and $yoffset$ are specified in quarter sample units. The following horizontal filtering is performed to generate intermediate samples aa, bb, cc, dd :

$$aa = \text{fir_filt}(A0, B0, C0, D0, xoffset)$$

$$bb = \text{fir_filt}(A1, B1, C1, D1, xoffset)$$

$$cc = \text{fir_filt}(A2, B2, C2, D2, xoffset)$$

$$dd = \text{fir_filt}(A3, B3, C3, D3, xoffset),$$

where the function $\text{fir_filt}()$ performs a FIR filtering with the coefficients selected from a table according to the used offset. The intermediate values are then used for vertical interpolation, expressed as,

$$pp = \text{fir_filt}(aa, bb, cc, dd, yoffset).$$

A0			B0	aa	C0			D0
A1			B1	bb	C1			D1
				pp				
A2			B2	cc	C2			D2
A3			B3	dd	C3			D3

Fig. 2. 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’, ‘dd’.

The function $\text{fir_filt}()$ is called even when either $xoffset$ or $yoffset$ is equal to 0, in which case it is a symmetric 3-tap filter. The operations of 6MOMS are similar to that of 4MOMS. The zero offset case is still a symmetric 3-tap filter but other offsets employ 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 the bottom row are required. This means that 7×4 samples are to be generated during the horizontal filtering. The filter to generate half-sample positions are symmetric, hence, in an implementation with multiplies (MUL) and adds (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, 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 horizontal and vertical operations would then be 120 MUL and 132 ADD. This procedure of complexity calculation is performed for all the 16 positions in a quarter-sample grid for block sizes from 4×4 to 64×64 . Notice that the overhead due to samples required for vertical filtering that go beyond the block boundary decreases as the block size increases. The worst case cycles, including memory accesses, are then expressed as a percentage of worst case cycles that would be needed for a 6-tap FIR-only system, depicted in Fig. 3.

Notwithstanding the lower arithmetic complexity, the inclusion of a prefilter can affect the overall latency of the MCP process. The impact of using FIR-only filters with long support can be reduced using SIMD architecture compared to sequential execution. Therefore, the complexity of generalized interpolation has been analyzed with more details about the underlying architecture. However, the operations on different rows or columns are independent of each other in each direction prefiltering. Hence, the IIR filtering of different rows or columns can be done in parallel.

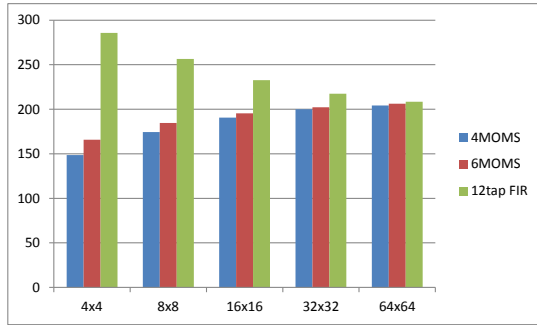


Fig. 3. Worst case cycles of 4MOMS, 6MOMS and 12-tap FIR per fractional sample, expressed as a percentage of worst case cycles of a 6-tap FIR filter. Each category in the x-axis represents a block size.

5. SIMULATION RESULTS

The RD performance of the proposed generalized interpolation is evaluated using test sequences defined in the Call for Proposals (CfP) issued by the JCT-VC [10]. Two constraint sets restricting the coding structures are defined in CfP as follows: a) GOP8 random access: structural delay not larger than 8-pictures, b) Low delay: no picture reordering in decoder. The software base for our experiments is the Test Model under Consideration [9] for the ongoing HEVC standardization. All the sequences from the dataset are used for testing and the entire sequences with up to 500 frames are coded.

We compare the 4MOMS and 6MOMS cases with two reference interpolation methods, the 6-tap and the 12-tap DCT-based FIR filters of the current HEVC draft [9] in the high efficiency setting. The improvements are measured in terms of the average bit rate difference over all rate points using the Bjøntegaard Delta Bit Rate [13] metric. The 6-tap filter used is around 3% better compared to the interpolation filter in H.264/AVC. The simulation results for GOP8 with hierarchical B pictures can be found in Tab. 1 and the low delay case is shown in Tab. 2. In low delay scenario, 6MOMS gives an average BD rate reduction of 1.9% compared to the 12-tap filter and 3.8% compared to the 6-tap filter. Gains up to 16.5% are observed compared to the 12-tap filter for the sequence BQSquare. Some sequences like PartyScene, BQTerrace, BQMall also show large performance gains, whereas the gains for other sequences are moderate. For the Class E sequences, in the low delay scenario, however, a marginal loss is observed. These sequences are noisy and an improved interpolation results in noisy prediction signal leading to a performance loss. The source code of our implementation, complexity analysis and the test results are available online [12].

6. CONCLUSION

Generalized interpolation using MOMS basis for MCP can provide improved coding efficiency compared to FIR filter based interpolation. The improvements are mainly observed for video sequences with fine spatial details. The worst case

GOP8	Y-BD bit rate %			
	Ref 12-tap FIR		Ref 6-tap FIR	
	6MOMS	4MOMS	6MOMS	4MOMS
Class A	0.0	0.1	0.1	0.1
Class B	-0.2	0.0	-0.4	-0.3
Class C	-0.9	-0.1	-3.2	-2.4
Class D	-1.7	-0.0	-5.8	-4.2
Average	-0.7	0.0	-2.5	-1.9

Table 1. GOP8 random access coding: BD rate comparison of 6MOMS and 4MOMS to 12-tap and 6-tap filters.

Low Delay	Y-BD bit rate %			
	Ref 12-tap FIR		Ref 6-tap FIR	
	6MOMS	4MOMS	6MOMS	4MOMS
Class B	-0.9	-0.3	-1.2	-0.6
Class C	-2.5	-1.2	-5.2	-3.9
Class D	-5.1	-2.3	-9.4	-6.9
Class E	1.4	0.8	1.4	0.7
Average	-1.9	-0.8	-3.8	-2.7

Table 2. Low delay coding: BD rate comparison of 6MOMS and 4MOMS to 12-tap and 6-tap filters.

complexity of bidirectional IIR along with 6-tap FIR is shown to be less than increasing the FIR support to 12. However, the latency involved in the MCP may increase due to an additional IIR prefiltering stage. Parallel processing can be used to independently 1D filter different rows or columns to reduce the latency.

7. 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, pp. 560–576, Jul. 2003.
- [2] T. Wedi, "Adaptive Interpolation Filters and High-Resolution Displacements for Video Coding", *IEEE Trans. CSVT*, vol. 16, no. 4, Apr. 2006.
- [3] M. Karczewicz, Y. Ye, P. Chen, "Switched Interpolation Filter with Offset", *ITU-T COM16 C463*, Apr. 2008.
- [4] K. Ugur, et. al., "High Performance, Low Complexity Video Coding and the Emerging HEVC Standard", *IEEE Tran. on CSVT*, vol. 20, no. 12, Dec. 2010.
- [5] H. Lakshman, B. Bross, H. Schwarz, and T. Wiegand, "Fractional-Sample Motion Compensation using Generalized Interpolation", *Picture Coding Symposium, Nagoya, Dec. 2010*.
- [6] T. Blu, P. Thévenaz, M. Unser, "MOMS: Maximal-Order Interpolation of Minimal Support", *IEEE Trans. Img. Proc.*, vol. 10, no. 7, Jul. 2001.
- [7] M. Unser, "Sampling - 50 Years After Shannon", *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.
- [8] M. Unser, "Splines: A Perfect Fit for Signal and Image Processing", *IEEE Signal Proc. Magazine*, vol. 16, no. 6, pp. 22–38, Nov. 1999.
- [9] T. Wiegand, W.-J. Han, J.-R. Ohm, G. J. Sullivan, "High Efficiency Video Coding (HEVC) text specification Working Draft 1", JCT-VC C403, Guangzhou 2010.
- [10] ITU-T Q6/16, "Joint Call for Proposals on Video Compression Technology", ITU-T VCEG-AM91, Kyoto, Japan, Jan. 2010.
- [11] T. Blu, P. Thévenaz, M. Unser, "High-Quality Causal Interpolation for Online Unidimensional Signal Processing", *Proc. EUSIPCO*, Wien, Austria, September 6–10, 2004, pp. 1417–1420.
- [12] H. Lakshman, H. Schwarz, D. Marpe, T. Wiegand, "CE3: Luma Interpolation using MOMS", JCT-VC D056, Korea, Jan 2011. http://phenix.int-evry.fr/jct/doc_end_user/documents/4_Daegu/wg11/JCTVC-D056-v2.zip
- [13] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," ITU-T VCEG-M33, April 2001.