28th Picture Coding Symposium, PCS2010, December 8-10, 2010, Nagoya, Japan

FRACTIONAL-SAMPLE MOTION COMPENSATION USING GENERALIZED INTERPOLATION

Haricharan Lakshman[1], *Benjamin Bross*[1], *Heiko Schwarz*[1], *and Thomas Wiegand*[1,2]

[1]Image Processing Department
Fraunhofer Institute for Telecommunications -
Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany

[2]Image Communication Chair
Department of Telecommunication Systems
Technical University of Berlin
Einsteinufer 17, 10587 Berlin, Germany

ABSTRACT

Typical interpolation methods in video coding perform filtering of reference picture samples using FIR filters for motion-compensated prediction. This process can be viewed as a signal decomposition using basis functions which are restricted by the interpolating constraint. Using the concept of generalized interpolation provides a greater degree of freedom for selecting basis functions. We implemented generalized interpolation using a combination of IIR and FIR filters. The complexity of the proposed scheme is comparable to that of an 8-tap FIR filter. Bit rate savings up to 20% compared to the H.264/AVC 6-tap filter are shown.

Index Terms— video coding, motion-compensated prediction, reference picture upsampling, B-splines

1. INTRODUCTION

Block-based motion-compensated prediction (MCP) using fractional-accuracy is widely used in video coding. In MCP, a block in an already reconstructed frame forms the prediction for the current block. The spatial displacement between these blocks is estimated by the encoder and transmitted to the decoder as side information. 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 6-tap filter is used to generate the half-sample positions followed by a bi-linear filter to generate the quarter-sample positions.

There have been many proposals to modify the interpolation filter for MCP to improve the quality of the prediction signal. The concept of Adaptive Interpolation Filter (AIF), adapted to consider nonstationary statistical properties of the video signal like aliasing, quantization errors and displacement estimation errors was proposed in [2]. Various refinements to this basic idea, e.g. non-separable AIF, separable AIF, directional AIF, etc have also been presented. It was also identified that a portion of the gains achieved by the AIF schemes stems from the fact that they use higher precision arithmetic and that for many sequences, similar gains can be achieved by using the H.264/AVC 6-tap coefficients without intermediate rounding [3]. This filter is further referred to as High Precision Filter (HPF) in this paper. The use of Switched Interpolation (SIFO), in which the indices of the best filters from a set of filters are signaled for each fractional position, was demonstrated in [3]. The SIFO filter set included the H.264/AVC HPF and another filter that was obtained by applying the analytical process used in AIF schemes to the training video set, instead of single frame. Another approach to improve quality is to use filters with longer support like 8-tap or 12-tap filters that better approximate the *sinc* response. Although increasing the filter support improves the quality of interpolation in general, it increases the memory bandwidth/computational complexity and can introduce ringing artifacts around edges.

In this paper, we use the concept of generalized interpolation and show that significant gains can be achieved even when using fixed filters with short support, e.g. fixed 4-tap filters. This is motivated by advances in sampling theory [4]. 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 interpolation [4]. The expansion coefficients for the new basis are derived 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.

2. OVERVIEW OF GENERALIZED INTERPOLATION

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,

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

where ϕ_{int} is chosen to satisfy the interpolating condition, i.e. it passes through zero at all integer locations except ori-

978-1-4244-7135-5/10/$26.00 ©2010 IEEE

gin, where it has a value of unity. Eq. (1) can be viewed as a signal expansion where the expansion coefficients are the samples themselves. The typical example of $\phi_{\text{int}}(x)$ is a windowed *sinc* function and the interpolation process is a simple convolution (or a low-pass filtering when viewed in frequency domain). In spite of this elegant setup, there are many practical issues, most important being the slow decay of the ideal *sinc* function. In practice, it has therefore been tried to approximate the ideal low pass response using finite support filters. As shown in [4], the short support and zero crossing constraints for the basis functions are the reasons for the suboptimality of classical interpolation.

In generalized interpolation [4], the zero crossing constraint is not imposed on the basis functions and the problem is reformulated as,

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

where $\phi(x)$ are basis functions with basic constraints for stability and unambiguous reconstruction. The crucial difference between classical interpolation (Eq. 1) and generalized interpolation (Eq. 2) is that the expansion coefficients $c[k]$ are not directly the data samples $s[k]$ anymore. This apparent drawback however offers new possibilities — it allows an extended choice of basis functions with better properties than that of the restricted basis functions in classical interpolation. Traditionally, the problem of determining the expansion coefficients has been approached using a matrix inversion. It was recognized in [5] that this problem could also be approached using simpler digital filtering techniques. We use the inverse filtering solution provided by Unser in [6]. In terms of digital filtering, the computation of coefficients from input signal is an IIR filtering and the estimation of value at a fractional position x using Eq. 2 is an FIR filtering step.

3. MOTION-COMPENSATED PREDICTION USING GENERALIZED INTERPOLATION

In the MCP loop of a hybrid video codec, the reconstructed picture is stored in the reference picture buffer after in-loop processing steps like deblocking are completed. In this paper, we propose a design in which the reconstructed picture is IIR filtered and the resulting array of expansion coefficients (which has the same dimensions as the input picture) is stored instead, as illustrated in Fig. **??**. Then, for motion compensation, this array is FIR filtered when a particular location is accessed (Eq. 2).

3.1. Basis function examples

There are several desirable properties for selecting a set of basis functions for the generalized interpolation, namely, short support, separability, symmetry, regularity, etc. The set of B-spline functions is a good example for such a basis. They

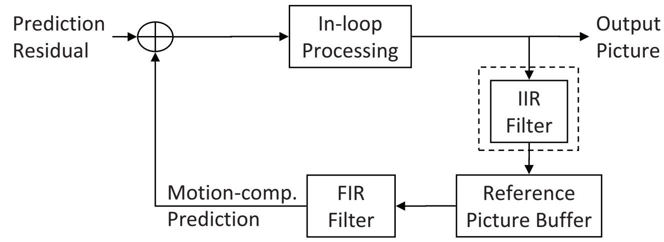

Fig. 1. *Motion-compensated prediction using generalized interpolation. The IIR filtering block in dashed lines constitutes the main difference to the standard techniques.*

are also easy to manipulate, e.g. for computing derivatives, integrals or higher support B-splines. In [7], we used the differentiability property of B-splines to compute gradients for warping motion estimation through the same expansion coefficients that are used for fractional-position sample estimation. The freedom of choosing the basis functions has been exploited in [8], where the authors have optimized the basis function in order to achieve a certain approximation order with the smallest possible support. The resulting basis functions, called MOMS, have been shown to have very good interpolation properties [8].

3.2. IIR filtering of reconstructed picture

The expansion coefficient computation can be done by 1D filtering successively along the rows and columns of the reconstructed image. The IIR filtering can be implemented using a cascade of causal and anti-causal filters.

We demonstrate the coefficient computation with the example of a 3rd order generalized interpolation. Denoting the number of samples as N, the following recursive algorithm [6] can be used to compute the expansion coefficients,

$$p[k] = \quad s[k] + z_1 \cdot p[k-1], \quad k = 1, \cdots, N-1 \tag{3}$$
$$c[k] = \quad z_1 \cdot (c[k+1] - p[k]), \quad k = N-2, \cdots, 0 \tag{4}$$

where, z_1 is the pole of the discrete IIR filter, which takes the value of -0.2679 for the cubic B-spline and -0.3441 for the cubic optimal MOMS basis. When a 5th order basis is chosen, e.g. quintic splines or quintic MOMS, it can be factorized into 2 sets of causal and anti-causal filters of 1st order. Each set of filters has exactly the same structure as in the case of 3rd order, but can differ in the pole values. The boundary values $p[0]$ and $c[N-1]$ are specified [6] so that the procedure of converting image samples to expansion coefficients is reversible.

4. COMPLEXITY ANALYSIS

Compared to the classical interpolation schemes, generalized interpolation introduces an additional step — the IIR filtering to obtain the expansion coefficients. In this section, it is shown that this additional filter can be implemented very efficiently and does not add much to the overall complexity. We analyze the IIR and the FIR parts separately to get an estimate of the overall complexity of the proposed scheme. The number of multiplications (MUL) and additions (ADD) are used as an approximate measure of complexity.

4.1. Expansion coefficient computation using IIR filter

Whenever the reconstructed picture is marked as a reference picture for future MCP, the IIR filtering is performed. As explained in Sec. 3.2, its structure can be fixed for all basis functions for a given order. The number of operations for the 3rd order basis functions, which can be factorized into 1st order causal (Eq. 3) and 1st order anti-causal (Eq. 4) filters, is 4 MUL and 4 ADD per expansion coefficient. The factorization into 2 sets for a 5th order basis results in 8 MUL and 8 ADD per expansion coefficient. The additional complexity for coefficient computation turns out to be especially small in the hierarchical coding structure. In such a structure, some reconstructed pictures are referenced multiple times and some reconstructed pictures are not referenced, hence requiring no filtering.

4.2. FIR interpolation filtering

In our implementation, the FIR interpolation filtering is performed whenever reference values on fractional-sample positions are needed. Therefore, the motion vector accuracy does not affect the complexity. Using the spline and MOMS bases, higher accuracy interpolation, e.g. 1/8th or 1/12th -samples can be performed with the same complexity as 1/4th -sample position. Note that in the classical interpolation schemes, longer filters are needed to produce sharp cut-off for narrow pass-bands like $\pi/8$ or $\pi/12$. The number of FIR operations per sample depends on the order of the basis function. For a 3rd order basis, which is a 4-tap filter, 4 MUL and 3 ADD are required for each filtering direction. In case of 5th order basis, which is a 6-tap filter, 6 MUL and 5 ADD are required for each direction. The FIR structure of multiply-accumulate of samples is well suited for efficient SIMD implementations in hardware/DSP.

4.3. Comparison with classical interpolation

The FIR filter in generalized interpolation also shows symmetries like in classical interpolation for some fractional-sample positions. For a simplified analysis, we ignore the symmetry and compare the worst case complexity for both generalized and classical interpolation. A classical FIR-only interpolation filter needs different number of MUL and ADD for different fractional-sample positions and no filtering is required for integer-sample positions. Assuming an uniform distribution of the 16 positions in a quarter-sample-accurate motion field, we compute the number of filter operations required per sample for an N-tap FIR filter. Six fractional positions require filtering in only one direction and nine fractional positions require filtering in two directions. For these nine positions there is also an initial overhead to compute additional temporary sample values outside block boundary. Hence, the average number of filter taps per sample can be written as,

$$N_{av} = \frac{1}{16}\left\{1 \cdot 0 + 6 \cdot N + 9 \cdot \left(2N + \frac{N(N-1)}{W}\right)\right\} \quad (5)$$

where, W is the block width. The IIR+FIR interpolation with 3rd order basis functions performs 12 MUL and 10 ADD for every sample, independent of the fractional-sample position. For small blocks, e.g. $W = 4$, it can be seen from Eq. 5 that a 6-tap FIR filter ($N = 6$) requires 13.22 MUL and 10.31 ADD on average. Hence, the complexity of the generalized interpolation of 3rd order is comparable to a 6-tap filter FIR-only implementation. For large block sizes ($W \to \infty$), the calculation overhead of additional temporary sample values outside block boundary tends to zero. In this case, a 8-tap FIR filter has a similar complexity (12 MUL, 10.5 ADD on average) as the 3rd order generalized interpolation scheme.

5. SIMULATION SETUP AND RESULTS

The RD performance of the proposed generalized interpolation is evaluated using test sequences defined in the Call for Proposals (CfP) issued by the Joint Collaborative Team on Video Coding (JCT-VC) [9]. Two constraint sets (CS) restricting the coding structures are defined in CfP as follows: a) CS1: structural delay not larger than 8-pictures, b) CS2: no picture reordering in decoder. The software base for our experiments is the Fraunhofer HHI response to the CfP [10]. From each of the test sequences, the first 50 frames are coded. The prediction and transform block sizes range from 4×4 to 64×64 with the constraint that the transform block sizes are always smaller or equal to the prediction block sizes. RD optimization, adaptive loop-filter, deblocking-filter, CABAC, quarter-sample accuracy MCP with 4 reference pictures are used and Intra mode is allowed within B and P pictures. Four QPs are chosen such that the four RD points, for all sequences having the same resolution, are located around the same four bit rates. In both CS1 and CS2 coding structures, the first picture is coded as Intra using the user set QP.

Four realizations of the generalized interpolation concept have been evaluated: cubic splines (3SPL), quintic splines (5SPL), cubic optimal MOMS (3MOMS) and quintic optimal MOMS (5MOMS). The reference uses the 6-tap HPF as interpolation method and the improvements are measured in terms of average bit rate difference over all rate points using the Bjøntegaard Delta Bit Rate (BDBR) metric. The simulation results for CS1 are shown in Tab. 1. Overall, the generalized interpolation methods outperform the 6-tap HPF interpolation. For some sequences like BQSquare there are bit rate savings up to 15% whereas for sequences like ParkScene

Sequence, CS1 Structural delay	Delta bit rate %			
	Cubic Spline	Quintic Spline	Cubic MOMS	Quintic MOMS
BQSquare B	-6.84	-13.67	-12.89	-15.19
BsktballPass B	-0.80	-2.26	-1.92	-2.57
PartyScene B	-1.12	-3.14	-2.53	-3.55
BQMall B	-0.81	-1.98	-1.65	-2.31
BQTerrace B	-1.53	-1.47	-1.56	-1.12
ParkScene B	-0.10	-0.40	-0.23	-0.35
Average	**-1.87**	**-3.82**	**-3.46**	**-4.18**

Table 1. *Bitrate savings using four implementations of the proposed generalized interpolation compared to 6-tap high precision filter with H.264/AVC coefficients. GOP8 coding structure with B pictures on JCT-VC sequences are used under CS1 [9] structural delay constraints.*

Sequence, CS2 Low delay	Delta bit rate %			
	Cubic Spline	Quintic Spline	Cubic MOMS	Quintic MOMS
BQSquare P	-9.98	-18.59	-15.61	-20.37
BQSquare B	-5.38	-16.07	-12.02	-18.43
BsktballPass P	-0.36	-0.89	-0.88	-0.82
BsktballPass B	-1.13	-2.52	-1.67	-2.73
PartyScene P	-1.21	-4.83	-3.12	-5.42
PartyScene B	-0.54	-4.35	-2.89	-5.36
BQMall P	-0.47	-0.44	-0.34	-0.46
BQMall B	-0.67	-1.66	-1.11	-2.08
BQTerrace P	-3.79	-4.52	-4.28	-4.28
BQTerrace B	-2.59	-6.02	-5.37	-6.52
ParkScene P	-0.02	-0.30	0.08	-0.35
ParkScene B	-0.21	-0.84	-0.50	-0.92
Average P	**-2.64**	**-5.10**	**-4.03**	**-5.28**
Average B	**-1.75**	**-5.24**	**-3.93**	**-6.01**

Table 2. *Bitrate savings using GOP4 coding structure with P and B pictures under CS2 [9] low delay constraints.*

there are only small savings. Concerning the different realizations, it can be observed that the MOMS kernel gives higher gains than the spline kernel while higher order basis functions perform better. Only for BQTerrace the cubic basis functions save more bit rate than the quintic ones. Another observation about the two different kernels is that the difference between cubic and quintic MOMS savings is smaller than the difference between cubic and quintic splines bit rate savings.

The results for the low delay CS2 can be found in Tab. 2. We evaluated two GOP4 coding structures: a) prediction with single hypothesis (P), and b) prediction with up to two hypotheses (B). Similar characteristics as in CS1 can be observed in CS2 also. For the sequences on which CS1 has higher savings compared to CS2 P prediction, performance improvements are observed with B prediction. BQMall, e.g., has a bit rate saving of 2.31% on CS1 with 5MOMS and only 0.46% on CS2 with P prediction. Using B prediction results in a 2.08% bit rate saving. This can be traced back to the low pass characteristic of the averaging step in B prediction.

The reported delta bit rates in Tab. 1 and Tab. 2 are for a floating point implementation of generalized interpolation. We have also implemented the IIR and FIR filtering using 16-bit fixed point arithmetic. We observe an average increase of around 0.5% in bit rate with fixed point implementation. The reference software with generalized interpolation can be downloaded from [10].

6. CONCLUSION

The application of generalized interpolation framework to MCP was presented in this paper. The generalization relaxes the zero crossing constraint of basis functions which allows the usage of a new set of functions with desired properties like short support, separability, etc. In our implementation, we used B-splines and MOMS as basis functions. The resulting filters outperform the H.264/AVC 6-tap high precision filter on JCT-VC test sequences with bit rate savings up to 20%.

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. Circuits Syst. Video Technol.*, 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] M. Unser, "Sampling - 50 Years After Shannon", *Proc. IEEE*, vol. 88, no. 4, pp. 569-587, Apr. 2000.

[5] A. Goshtasby, F. Cheng and B.A. Barsky, "B-spline curves and surfaces viewed as digital filters", *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 264-275, 1990.

[6] M. Unser, "Splines: A Perfect Fit for Signal and Image Processing", *IEEE Signal Proc. Magazine*, vol. 16, no. 6, pp. 22-38, Nov. 1999.

[7] H. Lakshman, H. Schwarz, T. Wiegand, "Adaptive Motion Model Selection using a Cubic Spline based Estimation Framework", To appear in *Int. Conf. Image Proc.,* Sep. 2010.

[8] T. Blu, P. Thévenaz, M. Unser, "MOMS: Maximal-Order Interpolation of Minimal Support", *IEEE Trans. on Image Proc.,* vol. 10, no. 7, pp. 1069 – 1080, Jul. 2001.

[9] ITU-T Q6/16, "Joint Call for Proposals on Video Compression Technology", ITU-T VCEG Document *VCEG-AM91*, Kyoto, Japan, Jan. 2010.

[10] M. Winken, S. Boße, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, D. Marpe, S. Oudin, M. Preiß, H. Schwarz, M. Siekmann, K. Sühring, T. Wiegand, "Description of video coding technology proposal by Fraunhofer HHI", *JCT-VC Meeting*, Dresden, Germany, Document *JCTVC-A116*, Proposal, Apr. 2010. `http://ftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A116.zip`