

Video Coding with Pixel-Aligned Directional Adaptive Interpolation Filters

¹Dmytro Rusanovskyy, ²Kemal Ugur,

¹Institute of Signal Processing,
Tampere University of Technology,
Tampere, Finland

¹Moncef Gabbouj, ²Jani Lainema

²Nokia Research Centre,
Nokia Corp.,
Tampere, Finland

Abstract— In this paper a novel adaptive interpolation filter structure is proposed to improve the coding efficiency of video coders. Proposed scheme utilizes one dimensional directional adaptive filter for every of the sub-pixel location, whose coefficients are calculated analytically for every frame by minimizing the prediction error energy. The direction of the interpolation filter is different for every sub-pixel position and it is determined based on the alignment of the corresponding sub-pixel with integer pixel samples. Experimental results show that, the proposed method achieves up-to 1.1 dB gain compared to the standard non-adaptive interpolation scheme of H.264/AVC, requiring less number of operations for interpolation. Compared to two-dimensional non-separable adaptive interpolation, proposed scheme has practically the same coding efficiency with approximately 3 times less complexity. Since significant coding efficiency is achieved without increasing the complexity, it is believed that proposed method has important use-cases in mobile multimedia environments where the resources are severely constrained.

I. INTRODUCTION

The state-of-the art video coding standard, H.264/AVC supports use of motion vectors with quarter pixel accuracy for motion prediction. If the motion vector of a block has fractional pixel accuracy, interpolation needs to be performed to obtain the samples at sub-pixel positions. The half-pixel samples are obtained by using a 6-tap FIR filter and quarter-pixel samples are obtained by bi-linear filtering using the two nearest samples at half or integer pixel positions [1]. The interpolation filter of H.264/AVC was designed to minimize the adverse effects of aliasing present in the input image sequence [2]. However, aliasing in a video sequence is not a stationary process, but has a varying characteristic. Adaptive interpolation filters that change the filter coefficients at each frame have been proposed in literature to overcome this non-stationary effect of aliasing and increase the coding efficiency of video codecs [2],[3].

In [3], Vatis *et al.* proposed to use 2D non-separable adaptive interpolation filter (2D-AIF) to reduce the prediction error energy. For each fractional pixel position, this scheme utilizes an independent filter and the coefficients for each filter are

calculated by minimizing the prediction error energy. It was reported that coding gains of up to 1.1 dB is achieved over the standard H.264/AVC especially at high resolution, high quality videos. However, the improvement in the coding efficiency with 2D-AIF scheme comes with the expense of having approximately three times more interpolation complexity compared to the standard H.264/AVC [4]. The reason for this additional complexity is mainly because of the non-separable nature of the filters and the need to perform 32 bit arithmetic. Since interpolation is one of the most complex part of H.264/AVC decoders, a significant increase in its complexity is very problematic, especially for resource constrained devices, such as portable video players or recorders.

In this paper, we propose a novel adaptive interpolation scheme to increase the coding efficiency of video coders, with significantly less complexity than its counterparts. The proposed method uses directional adaptive interpolation filters (DAIF) to compute each sub-pixel sample. The direction of the interpolation filter is different for every sub-pixel position and it is determined based on the alignment of the corresponding sub-pixel with integer pixel samples. Compared to previous schemes that use 2D non-separable adaptive filters, proposed method achieves practically the same coding efficiency with approximately 3 times less complexity. Proposed scheme outperforms the standard H.264/AVC interpolation by up-to 1.1 dB and requires less number of operations for filtering.

This paper is organized as follows; Section 2 describes the proposed filter architecture and its complexity analysis is given in Section 3. Section 4 presents experimental results, and Section 5 concludes the paper.

II. PROPOSED ADAPTIVE FILTER STRUCTURE

Consider Figure 1, where the pixels at integer positions are labeled by upper-case letters {A1,...,F6} within shaded boxes, and lower-case symbols {a,b,...,o} represent sub-pixel positions to be interpolated. Proposed algorithm uses a single directional adaptive filter to interpolate each sub-pixel location.

Let's denote the interpolation filter coefficients used for each sub-pixel Sp , with $h(Sp)$. The direction of these interpolation filters is determined according to the alignment of the corresponding sub-pixel with integer pixel samples. The sub-pixel locations e,o are diagonally aligned with integer pixels in NorthWest-SoutEast direction. The integer samples that are aligned with sub-pixel location in this direction are A1,B2,C3,D4,E5,F6, thus $h(e)$ and $h(o)$ only utilize these integer samples. Similarly, sub-pixel locations g, m are diagonally aligned with integer pixels but in NorthEast-SouthWest direction. Therefore, integer samples A6,B5,C4,D3,E2,F1 are used to interpolate sub-pixel locations g,n. Diagonal pixels-alignment is shown in Fig 1 with red solid arrows. The sub-pixel positions {a,b,c,d,h,l} are either horizontally or vertically aligned with integer image samples. In this case, {C1,C2,...,C6} is utilized for filters $\{h(a), h(b), h(c)\}$ and {A3,B3,...,F3} are used for interpolation of {d,h,l}, shown with green dashed arrows in Fig 1. The sub-pixel positions {j,f,i,k,n} are either aligned with integer-pixel samples in both of the two diagonal directions (position j), or have no integer-pixel samples to be aligned with (positions f,i,k,n). These sub-pixel locations are interpolated using the integer pixels that lie in a diagonal cross structure (A1,B2,C3,D4,E5,F6,F1,E2,D3,C4,B5,A6). The set of integer samples that participate for each sub-pixel position are denoted with $s(Sp) \in \{A1, \dots, F6\}$ and are given as

$$\begin{aligned} s(a) &= s(b) = s(c) = \{C1, C2, C3, C4, C5, C6\}, \\ s(d) &= s(h) = s(l) = \{A3, B3, C3, D3, E3, F3\}, \\ s(e) &= s(o) = \{A1, B2, C3, D4, E5, F6\}, \\ s(m) &= s(g) = \{F1, E2, D3, C4, B5, A6\}, \\ s(j) &= s(f) = s(i) = s(k) = s(n) = \\ &\{A1, B2, C3, D4, E5, F6, F1, E2, D3, C4, B5, A6\}. \end{aligned}$$

A. Calculating Filter Coefficients

The filter coefficients used to interpolate each sub-pixel are calculated for every P- or B- coded frame analytically by minimizing the motion prediction error. First, initial motion vectors with $\frac{1}{4}$ -pixel accuracy are found by performing motion estimation with the standard H.264/AVC interpolation. Using the motion vectors found in this step, a sub-pixel location Sp that fractional motion vector points to for every motion block is found. For each location Sp an independent filter $h(Sp)$ is estimated utilizing the integer pixels of reference and predicted images that are used to obtain the corresponding sub-pixel. Integer samples used to interpolate each sub-pixel location are given as $s(Sp)$ in previous section. Filter-coefficients estimation is done by solving a system of Wiener-Hopf equations constructed independently for each sub-pixel positions using all motion vectors over the coded frame.

In our description, we utilize relative coordinates for image pixels participating in adaptive interpolation, as it is shown in Fig.1. Considering an sample $X_{x,y}$ located at the (x,y) coordinates of current video frame X , its local neighborhood size of 6×6 pixels is shown as $\{A1, \dots, F6\}$, where $X_{x,y}$ is located at the C3 position. The motion vector

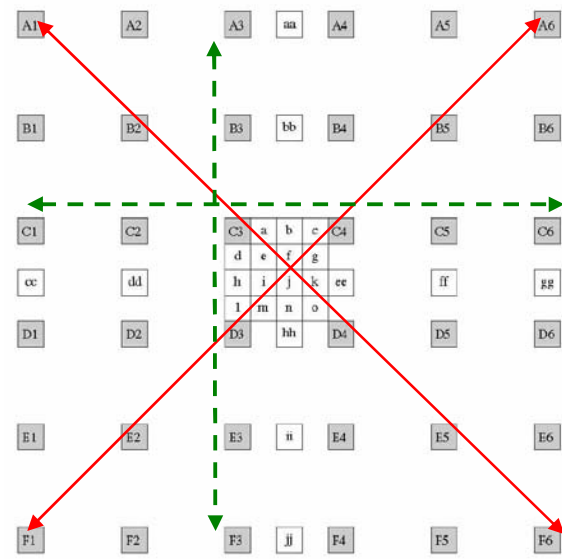


Figure 1. Utilized image samples notation and interpolations filters alignment.

$mv_{x,y} = (dx, dy)$ that is associated with the predicted sample $Y_{x,y}$ points to the location $(x+dx, y+dy)$ in the reference frame Y . Integer part of this motion vector is computed by rounding (dx, dy) towards the negative infinity and it specifies the relative base C3 and its local neighborhood $\{A1, \dots, F6\}$ in reference frame. Thus, a relative coordinate A1 in current frame is corresponding to a sample $X_{x-2, y-2}$ while A1 in the reference frame corresponds to a sample $Y_{x+dx-2, y+dy-2}$. The fractional part of motion vector specifies sub-pixel position $Sp = \{a, b, \dots, o\}$ and thus determines the corresponding adaptive filter $h(Sp)$ computed from these image samples.

Adaptive filter-coefficients for sub-pixel location Sp is given with $h_t(Sp)$, where t is the coefficients index, are analytically computed by solving the system of linear equation [5]. A system for N-tap Wiener filter, where $N=6$ for $\{a, b, c, d, h, l, e, g, m, o\}$ and $N=12$ for $\{f, i, j, k, n\}$ is given in (1):

$$\sum_{i=0}^{N-1} R_{t-i,i}^r(Sp) \cdot h_t(Sp) = R_i^{cr}(Sp), \quad i, t = 0, \dots, N-1 \quad (1)$$

where $R^{cr}(Sp)$ and $R^r(Sp)$ are expected cross-covariance and auto-covariance functions computed independently for each Sp as shown in (2) and (3) from current and reference images.

$$R_t^{cr}(Sp) = E[Y(s_t(Sp)) \cdot X(C3)] \quad , \quad (2)$$

$$R_{t,i}^r(Sp) = E[Y(s_t(Sp)) \cdot Y(s_i(Sp))] \quad , \quad (3)$$

where $s(Sp) \in \{A1, \dots, F6\}$ is the sub-pixel specific set of relative coordinates of integer pixel samples that are used for interpolation. A practical approach to compute expectations (2) and (3) is to average local covariances over all motion vectors which were estimated for currently coded frame.

B. Filter-coefficients symmetry

The filter structure introduced above defines a set of 15 independent filters $\{h(Sp), Sp=\{a,b,\dots,o\}\}$ for each coded frame and is fully described with 120 filter coefficients that should be transmitted to the decoder side. However, transmitting such amount of data at each frame brings a significant overhead and may degrade the coding efficiency. In order to reduce this overhead, we assume the statistical properties of image signal are symmetric [3]. This way, same filter-coefficients are used in case the distance of the corresponding full-pixel positions to the current sub-pixel position are equal. These assumption leads to the following restrictions on the filter coefficients.

Sub-pixel locations $\{a,c,d,l\}$ share the same filter, but horizontally or vertically mirrored, which is defined with 6 coefficients.

$$h_t(a) = h_{-t}(c) = h_t(d) = h_{-t}(l) .$$

Sub-pixel locations $\{b,h\}$ are interpolated with symmetrical 6-tap filter with 3 independent filter coefficients:

$$h_t(b) = h_t(h) .$$

Diagonally aligned sub-pixel locations $\{e,g,m,o\}$ are described with one filter using 6 filter coefficients, but diagonally mirrored:

$$h_t(e) = h_{-t}(o) = h_t(g) = h_{-t}(m) .$$

Sub-pixel location j has the central position, thus it is interpolated with a symmetric filter with only 3 independent filter coefficients:

$$\begin{aligned} h_0(j) &= h_5(j) = h_6(j) = h_{11}(j) , \\ h_1(j) &= h_4(j) = h_7(j) = h_{10}(j) , \\ h_2(j) &= h_3(j) = h_8(j) = h_9(j) . \end{aligned}$$

A single filter with 6 independent filter coefficients is utilized for $\{f,i,k,n\}$ positions. Filters retaliation is following:

$$\begin{aligned} h_0(f) &= h_6(f) = h_5(n) = h_{11}(n) = \\ h_0(i) &= h_{11}(i) = h_5(k) = h_6(k) , \\ h_1(f) &= h_7(f) = h_{10}(n) = h_4(n) = \\ h_1(i) &= h_{10}(i) = h_7(k) = h_4(k) , \\ &\dots \\ h_{11}(f) &= h_5(f) = h_0(n) = h_6(n) = \\ h_6(i) &= h_5(i) = h_0(k) = h_{11}(k) . \end{aligned}$$

Following these derivations, total amount of filter-coefficients to be transmitted to the decoder is reduced to 24.

C. Interpolation process

After adaptive filter-coefficients are calculated, we interpolate reference image samples Y at sub-pixels locations as following:

$$Y(Sp) = \sum_{t=0}^N Y(s_t(Sp)) \cdot h_t(Sp) \quad (4)$$

TABLE I. NUMBER OF ARITHMETIC OPERATIONS REQUIRED TO INTERPOLATE SUB-PIXEL POSITIONS.

Subpel position	H.264/AVC[1]	2D-AIF[3]	DAIF
b,h	10	10	10
a,c,d,l	13	13	13
e,g,m,o	23	58	13
J	32.5	41	16
f,i,k,n	35.5	55	19
Total for all 1/4-pixels	338.5	565	216

Interpolation for $\{f,i,k,n,b,h,j\}$ locations is modified due to a symmetry assumption introduced above, following are exemplary filtering implementation for $\{b,f,j\}$ sub-pixels:

$$\begin{aligned} X(b) &= (C1 + C6) \cdot h_0(b) + (C2 + C5) \cdot h_1(b) + (C3 + C4) \cdot h_3(b) \\ X(f) &= (A1 + A6) \cdot h_0(f) + (B2 + B5) \cdot h_1(f) + (C3 + C4) \cdot h_2(f) \\ &\quad + (D3 + D4) \cdot h_3(f) + (E2 + E5) \cdot h_4(f) + (F1 + F6) \cdot h_5(f) \\ X(j) &= (A1 + A6 + F1 + F6) \cdot h_0(j) + \\ &\quad (B2 + B5 + E2 + E5) \cdot h_1(j) + \\ &\quad (C3 + C4 + D3 + D4) \cdot h_2(j) \end{aligned}$$

III. COMPLEXITY ANALYSIS

We estimated the complexity of the proposed in this paper interpolation (DAIF) assuming the image symmetry and compared it with complexities of 2D-AIF and H.264/AVC interpolations reported in [4]. Our complexity estimation was done as a number of basic arithmetic operations, such sum, multiplication and shift required to interpolate each sub-pixel location, similarly to the method described in [4]. These complexity estimates for each filter are given in the Table I. For every sub-pixel we show the number of operations for H.264/AVC, 2D-AIF and proposed interpolations. We realize that number of arithmetic operations is a simple approximation of the complexity and do not consider computational architecture, e.g. memory bandwidth, use of dual MAC units and so on. However, we believe these estimates are useful for illustration purposes.

Analysis of the Table I shows that proposed DAIF scheme requires about 38% of that number of operations needed for 2D-AIF to interpolate reference image. In comparison to the standard H.264/AVC interpolation, our method requires about 36% less arithmetic operations.

The worst-case complexity is an important parameter that determines the upper-bound of decoding performance. As seen in Table-1, the worst case complexity of the proposed scheme is reduced by 41% compared to H.264/AVC and reduced to one-third of the 2D-AIF scheme.

IV. EXPERIMENTAL RESULTS

Proposed in this paper DAIF scheme was integrated into the official VCEG-KTA software [6]. In order to test performance of the proposed scheme, we simulated rate-distortion curves with VCEG-KTA software following the common test conditions defined in the ITU-T/VCEG group [7] with the baseline profile settings. Encoded bit-streams included side information for adaptive filters such as quantized filter coefficients, thus generated streams were fully decodable.

Simulation were performed with test video sequences listed in [7] with total amount of 300 coded frames for sequences at QCIF and CIF resolutions, and 150 coded frames for video materials at 720p resolution. In addition, we included 4CIF video with total number of 150 coded frames for each sequence. Three interpolation schemes have been compared; those are proposed in this paper (DAIF), the scheme with 2D non-separable adaptive filters (2D-AIF) [3], and H.264/AVC [1], which was utilized for benchmarking. Table 2 presents the simulation results for 2D-AIF and DAIF interpolation schemes in terms of difference in average PSNR (Δ PSNR [7]) compared to the standard H.264/AVC method. In addition to these, exemplary rate-distortion curves are given in Fig. 2 and 3 for “ShuttleStart” and “Crew” test sequences respectively.

As seen from Table II, proposed scheme significantly outperforms in terms of coding efficiency the standard H.264/AVC. The achieved coding gain is up to 1.1dB for high resolution video (see Fig.2 and 3) and 0.6 dB on average for 720p sequences. In comparison to the method with 2D non-separable adaptive interpolation, proposed scheme achieves comparable results, with significantly lower complexity. Compared to 2D non non-separable filtering scheme, proposed scheme algorithm has practically the same coding efficiency (0.02 dB less in average), with significant benefits in complexity.

V. CONCLUSIONS

The recent H.264/AVC video coding standard supports motion compensated prediction with up to quarter-pixel accuracy motion vectors. In this paper we consider a quarter-pixel accuracy motion compensated prediction and propose a novel interpolation scheme using directional adaptive filter for each sub-pixel location. The direction of the interpolation filter is different for every sub-pixel position and it is determined based on the alignment of the corresponding sub-pixel with integer pixel samples. Experimental results show that, the proposed method achieves up-to 1.1 dB gain compared to the standard non-adaptive interpolation scheme of H.264/AVC, requiring less number of operations for interpolation. Compared to two-dimensional non-separable adaptive interpolation, proposed scheme has practically the same coding efficiency with approximately 3 times less complexity.

REFERENCES

- [1] M. Karczewicz, A. Hallapuro; “Interpolation solution with low encoder memory requirements and low decoder complexity”, ITU-T SGI, VCEG-N31, 24-27 Sept. 2001.
- [2] T. Wedi, “Adaptive Interpolation Filter for H.26L”, ITU-T SG16/Q6, doc. VCEG-N28, Santa Barbara, CA, USA, Sep. 2001.
- [3] Y. Vatis, B. Edler, D. T. Nguyen, J. Ostermann, “Motion and Aliasing-Compensated Prediction Using a Two-dimensional Non-Separable Adaptive Wiener Interpolation Filter”, Proc. ICIP 2005, Genova, Italy, September 2005.
- [4] Y.Vatis and J. Ostermann, “Comparison of complexity between two-dimensional non-separable adaptive interpolation filter and standard wiener filter”, ITU-T SGI 6/Q.6 Doc. VCEG-AA11, Nice, France, October 2005.
- [5] Papoulis, Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, 1991.

[6] Reference ITU-T VCEG-KTA Software [Online]. Available: <http://iphome.hhi.de/suehring/tml/download/>

[7] T. K Tan, G. Sullivan and T. Wedi, “Recommended Simulation Common Conditions for Coding Efficiency Experiments,” ITU-T Q.6/SG16, VCEG-AE010, Marrakech, Morocco, January 2000

TABLE II. SIMULATION RESULTS OF THE 2D-AIF AND PROPOSED INTERPOLATION COMPARING TO THE H.264/AVC INTERPOLATION.

Resolution	Sequence	Delta PSNR,dB	
		2D-AIF[3]	DAIF
QCIF	Container	0.33	0.38
	Foreman	0.1	0.11
	Silent	-0.02	0
CIF	Paris	0.05	0.06
	Foreman	0.3	0.27
	Mobile	0.3	0.12
4CIF	Tempete	0.16	0.07
	City	0.45	0.42
	Soccer	0.55	0.49
720p	Ice	0.1	0.1
	Crew	0.4	0.36
	City	0.61	0.57
	BigShips	0.36	0.35
	ShuttleStart	0.75	0.75
	Crew	0.59	0.56

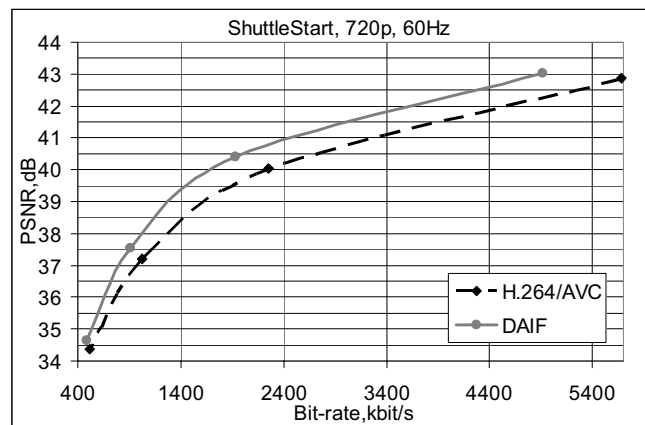


Figure 2. Rate-distortion curves for “ShuttleStart” test sequence.

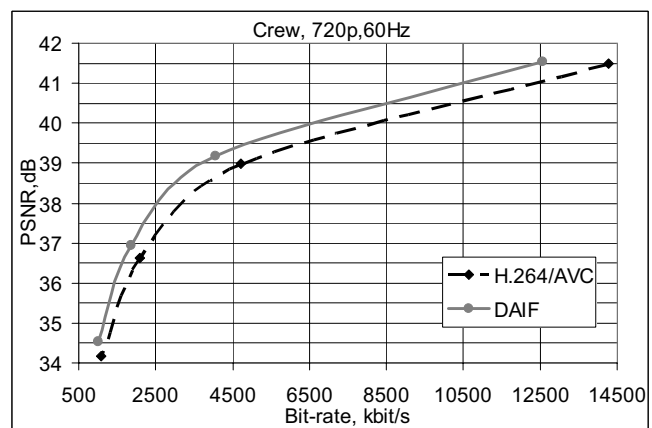


Figure 3. Rate-distortion curves for “Crew” test sequence.