ADAPTIVE INTERPOLATION WITH FLEXIBLE FILTER STRUCTURES FOR VIDEO CODING

[1,2]*Dmytro Rusanovskyy,* [2]*Kemal Ugur,* [1]*Moncef Gabbouj*

[1]Institute of Signal Processing, Tampere University of Technology, Tampere, Finland
[2]Nokia Research Center, Nokia Inc., Tampere, Finland

ABSTRACT

Two novel algorithms are proposed for improving the coding efficiency of adaptive interpolation schemes for video codecs, without increasing implementation complexity. Proposed algorithms utilize two different filter structures with equal tap length, but with complementary frequency responses. Depending on the content being coded, encoder selects which one of the two filter structure is optimal and signals this information to the decoder. In addition, the symmetry of filters is not pre-defined but is flexible. Encoder selects the optimal filter symmetry depending on the coding rate and the content and signals this information to the decoder. Experimental results show, that proposed improvements bring up to 7% of bit-rate reduction at high bit-rate over conventional adaptive interpolation. When compared to H.264/AVC, average gain over the test set is 11%. Coding efficiency is improved without increasing the complexity, thus proposed algorithms are suitable for mobile multimedia use-cases, where the computational resources are very limited.

Index Terms**—** Adaptive interpolation, video coding, interpolation filter.

1. INTRODUCTION

The standard H.264/AVC codec utilizes motion compensated prediction with motion vectors at fractional (up-to ¼-pixel) resolution. In order to estimate and compensate fractional-pixel displacements, samples at sub-pixel positions need to be interpolated. Interpolation filters of H.264/AVC were designed to minimize the adverse effects of aliasing present in wide range of image sequences. However, aliasing in a video data is not a stationary process, but has a varying characteristic. Adaptive interpolation filters (AIF) that change the filter coefficients at each frame have been proposed in literature to combat this non-stationary effect of aliasing [1-4]. These schemes all provide comparable coding performance over the standard H.264/AVC scheme and they have been adopted by ITU-T VCEG standardization group in the reference KTA software [5].

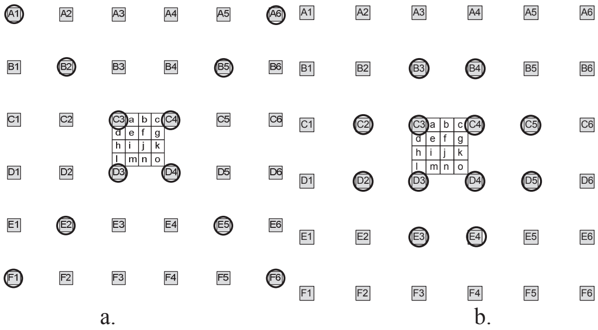

Figure 1. 12-tap filter structures: a. – diagonal cross filter; b. – radial support filter.

In [1], Vatis et. al. proposed a scheme with 36-taps 2D non-separable adaptive interpolation filters, a scheme with separable 36-tap filters was proposed in [2]. Following this, two low complexity adaptive interpolation schemes that use a maximum of 12-tap filters were proposed in [3] and [4]. Directional adaptive interpolation filters (DAIF) [3] uses 12-tap sparse filter structure in a form of diagonal cross, and authors of [4] utilize 12-tap filter structure with a radial support, as shown in Fig. 1. Algorithms [1-4] utilizes filter symmetry to reduce the overhead of transmitting filter coefficients. Same filter-coefficients are used in the case of equal distances between the corresponding full-pixel positions and the current sub-pixel position. This assumption allowed to AIF schemes be applicable for coding of the low resolution sequences, but restricts AIF performance for HD videos at high bitrates.

In this paper, we modified the DAIF scheme [3] with two new techniques allowing larger flexibility for interpolation filters. First tool relaxes the symmetry assumption described above and instead uses an adaptive symmetry that can change depending on the coded bitrate. Symmetry descriptor information is transmitted to the decoder along with the filter coefficients. Decoder reads the symmetry

978-1-4244-5654-3/09/$26.00 ©2009 IEEE

descriptors and performs filter coefficient decoding accordingly. The second tool is switching 12-tap filter structure for five central sub-pixel locations {f,i,j,k,n}. Selection is done among two candidates, which are 12-tap diagonal cross of [3] and 12-tap radial support filter of [4], see Fig. 1 and have complementing frequency responses. Proposed improvements of the DAIF scheme require no complexity increase to interpolation at decoder and bring up-to 7% of bit-rate reduction for HD sequences. Complexity increase at the encoder side is negligible comparing to the original DAIF scheme.

This paper is organized as follows; Section 2 describes the basics of the DAIF scheme. Section 3 presents the details of the proposed techniques. Experimental results are given in Section 4, and Section 5 concludes the paper.

2. DIRECTIONAL ADAPTIVE INTERPOLATION

Consider Fig. 1, where integer pixel locations are labeled by upper-case letters {A1,…,F6} within the shaded boxes, and lower-case symbols {a,b,…,o} represent samples at sub-pixel locations that are to be interpolated. Let's denote an interpolation filter used for each sub-pixel *Sp*, $Sp \in \{a, b, ..., o\}$, with *h(Sp)*. The direction of these filters is determined in the DAIF according to the alignment of the corresponding sub-pixel *Sp* with the integer-pixels sets. We denote these sets as $s(Sp) \subseteq \{A1, ..., F6\}$.

To interpolate locations *{a,b,c}*, *{d,h,l}*, and *{e,g,o,m}*, the DAIF scheme utilizes 1D 6-tap filters aligned horizontally, vertically or diagonally respectively. Five central sub-pixel locations *{f,i,j,k,n}* are interpolated in with a sparse 12-tap filter spanning the integer-pixels that lie on a diagonal cross structure: *s(f|i|j|k|n)={A1,B2,…,F6}* ∪ *{F1,E2,…,A6},* see Fig.1.a.

The sets of integer-pixel locations that contribute to the interpolation at each sub-pixel locations are denoted with $s(Sp)$ and are given explicitly as following.

$$\begin{array}{l} s(a)=s(b)=s(c)=\{C1,C2,C3,C4,C5,C6\}, \\ s(d)=s(h)=s(l)=\{A3,B3,C3,D3,E3,F3\}, \\ s(e)=s(o)=\{A1,B2,C3,D4,E5,F6\}, \\ s(m)=s(g)=\{F1,E2,D3,C4,B5,A6\}, \\ s(j)=s(f)=s(i)=s(k)=s(n)= \\ \{A1,B2,C3,D4,E5,F6,F1,E2,D3,C4,B5,A6\}. \end{array} \quad (1)$$

The DAIF defines a set of 15 independent filters $\{h(Sp), Sp \in \{a, b, ..., o\}\}$ for each coded frame and is fully described with 120 filter coefficients that are computed for each coded frame and transmitted to the decoder. In order to reduce this bit overhead, the original DAIF imposed a full symmetry assumption on the image signal characteristics. The same filter-coefficients are used in the case of equal distances between the corresponding full-pixel positions and the current sub-pixel position. Sub-pixels {a,…,o} were clustered in several groups (defined by symmetry) and a single independent filter is computed for each of these groups. A single 6-tap filter is computed and transmitted for sub-pixel locations *{a,c,d,l}*. Similarly, independent filters were defined for locations *{e,g,o,m}*, *{f,i,k,n}, {c,h}* and *{j}* correspondingly. Due to such arrangement, the total number of filter coefficients to be transmitted in the DAIF scheme was reduced from 120 to 24. Such symmetry made the DAIF be applicable for coding of low resolution sequences, but penalized the coding performance at high bit-rate.

3. THE PROPOSED ALGORITHMS

We propose two new coding tools to improve coding performance of the DAIF scheme. First tool relaxes the symmetry assumption imposed in [3] and instead uses an adaptive symmetry that can change depending on the coded bitrate. The second tool utilizes switching between two different 12-tap filter structures with complementary frequency responses.

3.1. Flexible Filter Symmetry

Adaptive interpolation with flexible symmetry was first proposed in [7]. Authors defined a number of filter sets with different symmetry assumptions. Encoder selects the optimal filter by encoding each frame multiple times for each candidate filter, and selecting the filter giving the smallest Rate-Distortion cost. Such approach limits the number of possible symmetries and drastically increases the encoding complexity, as each frame to be encoded multiple times.

Our proposed tool is more flexible and it allows more variety for filter symmetries. This variety is achieved by signaling a symmetry descriptor for each sub-pixel, which carries the information if the corresponding sub-pixel uses an independent filter or shares the same filter with another sub-pixel. If the same filter is shared with more than one sub-pixel locations, only one filter is transmitted to the decoder.

The symmetry descriptor for each sub-pixel is calculated as follows. First, for each sub-pixel location *Sp* adaptive filter *h(Sp)* is computed without assuming any symmetry. We treat each N-tap length filter as a vector in N-space, and compute the Euclidean distance (2) between coefficients of different sub-pixel positions *Sp1* and *Sp2*.

$$d_{Sp1,Sp2} = \sqrt{\sum_{i=0}^{N-1} \left(h_i(Sp1) - h_i(Sp2)\right)^2} \quad (2)$$

This distance is compared against a threshold that is function of the bit-rate. If the distances between filters are lower than the threshold, those sub-pixel samples share a single filter which is re-computed from merged covariance matrixes of those two sub-pixel locations. Fig. 2, illustrate this process, where filters *h(a)* and *h(c)* to be merged, as their Euclidian distance is lower than the threshold, whereas sub-pixels *d* and *l* will use independent filters.

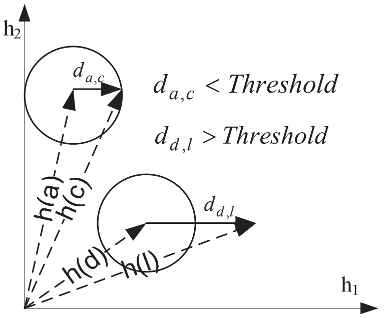

Figure 2. An interpretation of 2-tap length filters diversity analysis. Euclidian distance (solid arrow) between two filters (dashed arrows) is compared against the threshold (circle of allowed).

The threshold value is calculated by running extensive test simulations over a wider set of test video sequences coded at different bit-rates. By doing this, we found an empirical threshold which is modelled a quadratic function of coded frame bit-rate $threshold = \left(nBits / 2 \cdot 10^4\right)^2$.

3.2. Switching Filter Structure

In addition, we propose interpolation of *{f,i,j,k,n}* sub-pixels with switching between two different 12-tap filter structures. Those structures are diagonal-cross and radial filters, their spatial supports are show in Fig. 1.a and Fig. 1.b respectively.

The spatial supports of these 12-tap filter are explicitly defined as

s(Sp)= {A1,B2,C3,D4,E5,F6,F1,E2,D3,C4,B5,A6} (3)

for diagonal cross filter, and as

s(Sp)= {B3,B4,C2,C3,C4,C5,D2,D3,D4,D5,E3,E4 } (4)

for radial support filter.

In order to compare frequency responses of diagonal-cross and radial filters, we display their cut-off frequencies (-3dB level) in Fig. 3 and compared against the standard 2D 6x6-tap H.264/AVC filter. One can see that the 12-tap diagonal-cross filter provides better support for frequencies in horizontal and vertical directions (dash-doted line). Its frequency support in these directions outperforms the support of the radial filter and is close to the cut-off frequency of 36-tap filter of H.264/AVC. However, the diagonal cross filter has “weaker” support (low cut-off frequency) in the diagonal directions, where as 12-tap radial filters, in contrast, performs much better (solid thin line).

Considering this, we propose an interpolation scheme which has an ability to switch between these 12-tap filters and thus has an advantage of efficient use of tap-length against interpolation with a fixed filter structure.

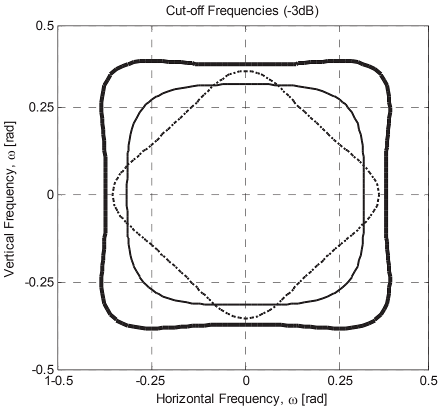

Figure 3. Cut-off frequencies (at the level -3dB) for three *h(j)* filters: thick solid line – 2D 6x6 filter of H.264/AVC, thin solid line – estimate of 12-tap Radial Filter, thin dash-doted line – estimate of Diagonal-Cross Filter.

3.3. Encoder Algorithm

In this section, we describe in detail how the encoder selects the optimal filter symmetry and the filter structure and also we present a brief analysis of the proposed tools on the encoder and decoder complexity. The encoder algorithm is detailed step by step below;

1. Frame is coded with the fixed filter given in H.264/AVC and motion vectors are computed
2. Using the motion vectors and the reference frame, encoder computes independent filters for each sub-pixel {a,…,o}. For sub-pixel locations *{f,i,j,k,n}* filters for two different structures are computed, diagonal-cross and radial.
3. The decision between radial and diagonal cross filter is made based on the motion prediction error. That is, by using the motion vectors obtained at the first step, the error between original and prediction image is computed for both radial and diagonal cross structure. The filter structure that gives the smallest error for each sub-pixel is chosen.
4. The symmetry between each sub-pixel location is calculated as described in Section 3.1, using the bit-rate obtained in the Step-1. According the calculated symmetry, the filter coefficients are re-computed.
5. Input frame is encoded with adaptive interpolation filter, identically as it is done in [3].

Proposed algorithm requires no additional encoding passes compared to the original DAIF scheme, thus encoding complexity increase can be considered negligible. At the decoder side, the complexity is identical to the DAIF design, as the interpolation for each sub-pixel is performed using the same number of taps.

4. EXPERIMENTAL RESULTS

Proposed algorithms were integrated and proposed for adoption into the JMKTA software maintained by ITU-T VCEG standardization group [5]. In order to test performance, we simulated rate-distortion curves with the reference KTA software following the common test conditions defined in the ITU-T/VCEG group [6] with the High Profile settings. Simulation were performed with test sequences listed in [6] with total amount of 150 coded frames for sequences at 720p resolutions, and 60 coded frames for video materials at 1080p resolution.

Three interpolation schemes have been compared; those are proposed in this paper tools (“Proposed”), original algorithm (“Conventional”), and the standard H.264/AVC, which was utilized for benchmarking. Table I presents the simulation results for interpolation schemes in terms of delta bit-rate reduction [6] compared to the standard H.264/AVC. Additionally, Fig. 4 presents simulated rate-distortion curves for “Raven” 720p test sequences.

As seen in Table I proposed algorithms provide an up-to 24.60% bitrate reduction compared to the standard H.264/AVC interpolation, with an average improvement of 10.94%. We then compared the performance of the proposed tools with conventional adaptive interpolation filtering that does not utilize the flexible tools proposed in this paper. As seen in Figure 4, improvement obtained using the proposed scheme is very visible at high bitrates and becomes up-to 7%. This verifies our assumption that having fixed filter symmetry used in previous schemes is not optimal for coding at high bitrates, and using adaptive filter symmetry improves the coding efficiency significantly.

5. CONCLUSIONS

In this paper, we proposed two novel algorithms for adaptive interpolation that allow flexibility in filter structure. First tool replaces the fixed symmetry of adaptive interpolation filters with a flexible structure. Encoder calculates the optimal filter symmetry between independent filters of different sub-pixel locations based on the coded rate and signals this information to the decoder. The second tool is switching the 12-tap filter structure for central sub-pixel locations, between two candidates that have complementary frequency responses.

Experimental results shows, that proposed improvements bring up-to 7% of bit-rate reduction at high bit-rates in comparison to the original DAIF design and require no complexity increase at the decoder side.

11. REFERENCES

[1] Y. Vatis, B. Edler, D. T. Nguyen and J. Ostermann, “Motion and Aliasing-Compensated Prediction Using a Two-dimensional Non-Separable Adaptive Wiener Interpolation Filter”, Proc. ICIP 2005, Genova, Italy, September 2005.

[2] S.Wittman and T. Wedi, “Separable adaptive interpolation filter”, ITU-T SG16/Q6, Doc. C-0219 July, 2007.

[3] D. Rusanovskyy, K. Ugur, J. Lainema, M. Gabbouj,"Video Coding with Pixel-Aligned Directional Adaptive Interpolation Filters," In Proc. ISCAS’08, 2008.

[4] Y. Ye, M. Karczewicz, “Enhanced adaptive interpolation filter”, ITU-T SG16/Q6, doc. C-0464 April, 2008

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

[6] T. K Tan, G. Sullivan and T. Wedi, “Recommended Simulation Common Conditions for Coding Efficiency Experiments,” ITU-T Q.6/SG16, VCEG-AJ010, San-Diego, CA, USA, October 2007

[7] K.Ugur, J.Lainema, M.Gabbouj, "Adaptive Interpolation Filter with Flexible Symmetry for Coding High Resolution High Quality Video", In Proc. Acoustics, Speech and Signal Processing (ICASSP) 2007, Honolulu, HA, USA.

Table I Coding efficiency of the proposed scheme in terms of delta bit-rate [6] with respect to the H.264/AVC interpolation.

Sequence	ΔBit-rate,%	
	Conventional	Proposed
BigShips(720p)	-8.56	-9.37
Night(720p)	-5.79	-6.50
City(720p)	-12.77	-14.41
Crew(720p)	-10.50	-10.67
Raven(720p)	-21.74	-24.60
Jets(720p)	-4.23	-6.49
CrowdRun(1080p)	-1.02	-1.48
ParkJoy(1080p)	-1.34	-2.10
Toys&Call(1080p)	-8.53	-10.14
Sunflower(1080p)	-21.97	-23.64
Average	**-9.65**	**-10.94**

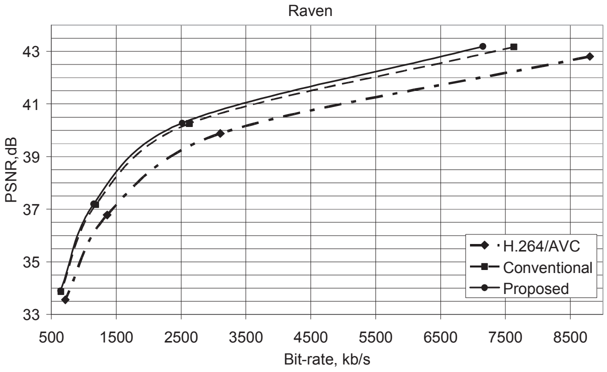

Figure 4. Rate-distortion curves for “Raven” 720p test video sequence at 60 fps.