

# Fast Encoding Algorithms for Video Coding with Adaptive Interpolation Filters

Dmytro Rusanovskyy<sup>1</sup>, Kemal Ugur<sup>2</sup>, Moncef Gabbouj<sup>1</sup>

<sup>1</sup>Tampere University of Technology, Finland

<sup>1</sup> dmytro.rusanovskyy@tut.fi

<sup>3</sup> moncef.gabbouj@tut.fi

<sup>2</sup>Nokia Research Center, Finland

<sup>2</sup> kemal.ugur@nokia.com

**Abstract**—In order to compensate for the temporally changing effect of aliasing and improve the coding efficiency of video coders, adaptive interpolation filtering schemes have been recently proposed. In such schemes, encoder computes the interpolation filter coefficients for each frame and then re-encodes the frame with the new adaptive filter. However, the coding efficiency benefit comes with the expense of increased encoding complexity due to this additional encoding pass. In this paper, we present two novel algorithms to reduce the encoding complexity of adaptive interpolation filtering schemes. First algorithm reduces the complexity of the second encoding pass by using a very lightweight motion estimation algorithm that reuses the data already computed in the first encoding pass. Second algorithm eliminates the second coding pass and re-uses the filter coefficients already computed for previous frames. Experimental results show that the proposed methods achieve between 1.5 to 2 times encoding complexity reduction with practically negligible penalty on coding efficiency.

## I. INTRODUCTION

The interpolation filter defined in H.264/AVC is designed to minimize the adverse effects of aliasing present in the input image sequence. However, aliasing in a video sequence is not a stationary process, but has a varying characteristic. Adaptive Wiener interpolation filters that change the filter coefficients at each frame have been proposed in literature to combat this non-stationary effect of aliasing [1][2][3]. In [1], Vatis et. al. proposed a scheme that uses 2D non-separable adaptive interpolation filter, a scheme using separable filters was proposed in [2] and finally adaptive interpolation using directional filters was proposed in [3]. These schemes provide similar coding gains (up-to 1.1dB) over the standard H.264/AVC scheme and they have been adopted by ITU-T VCEG standardization group in the reference KTA software [8].

In spite of difference in filter structures, these algorithms share the same coding conception, and require two encoding passes for each P- or B-frame as illustrated in Figure 1. In the first coding pass (Encoding 1), the current frame is coded with the reference frame interpolated with static filter. During this pass, motion estimation (ME) and rate distortion optimization (RDO) methods are performed and motion prediction information (motion vectors and macroblock partitions) are found. Using this information, encoder then calculates the coefficients of the adaptive filter, by solving system of linear

equations that minimize the prediction error energy (reader is referred to [1] for more details). Next, frame is re-encoded including ME and RDO using the computed adaptive interpolation filter.

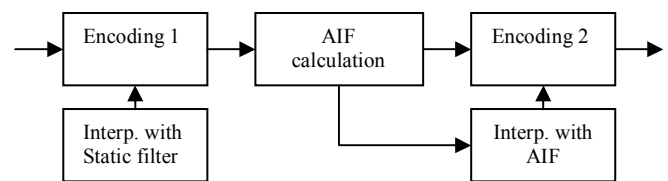


Fig. 1. Flowchart of video encoder with adaptive interpolation

It is clear from Figure 1 that increase in complexity compared to encoding with static filter is due to three additional steps. Those are the AIF calculation step, interpolation of the reference frame with the new filter step and the second encoding pass. In our earlier work, we significantly reduced the complexity of the AIF calculation step [4]. In this work we study simplifying the second encoding pass and propose two novel methods to significantly reduce its complexity.

For the first method, we assume that the motion information obtained by the static filter in the first pass is correlated to the motion information obtained by the adaptive filter in the second pass. This correlation is exploited and the complexity of motion estimation in the second encoding pass is reduced by reusing many of the coding results of the first pass. For the second method, we assume that although aliasing is not stationary, it does not undergo a fast temporal variation. Using this assumption we do not perform the second encoding pass after the adaptive filter is calculated. Instead we use the adaptive filter computed for the current frame to code the subsequent frame. This way, single-pass coding is achieved for the adaptive interpolation filtering schemes.

This paper is organized as follows; Section 2 describes video coding with adaptive Wiener interpolation filtering. Section 3 presents the details of the proposed fast encoding algorithms. Experimental results are given in Section 4, and Section 5 concludes the paper.

## II. VIDEO CODING WITH ADAPTIVE INTERPOLATION FILTERING

Figure 2 shows a generalized block diagram of a hybrid video coding scheme, e.g. H.264/AVC that uses adaptive interpolation filtering. The current image  $X$  is first predicted by motion compensated prediction (MCP) from the already encoded reference frame  $Y$ . The result of the motion compensated prediction is the prediction  $X'$ . The difference between current image  $X$  and prediction  $X'$  is the prediction error  $E'$ . In order to minimize the prediction error  $E'$ , an independent filter for each different sub-pixel position  $Sp$  is calculated (denoted as  $h(Sp)$ ). Filter coefficients are computed by solving the system of Wiener-Hopf equations (1), which are constructed from local signal statistics of  $Y$  and  $X$  using the motion vectors from the first pass. The system of linear equations needed to calculate such N-tap Wiener filter  $h(Sp)$  is given as:

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

where  $R^{cr}$  and  $R^r$  are estimations of local cross-correlation and auto-correlation functions averaged over the reference ( $Y$ ) and the current ( $X$ ) images. Solving (1) provide a filter optimal in terms of minimal prediction error  $E'$ , in other words a Wiener interpolation filter.

After the adaptive filter  $h$  is calculated, reference frame is interpolated with the new filter and the second encoding pass is performed.

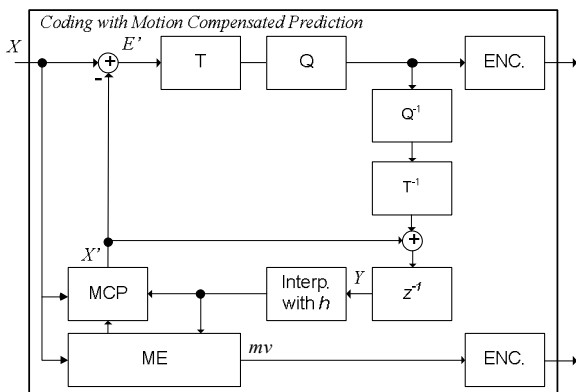


Fig. 2 Flowchart of video encoder with adaptive interpolation

## III. PROPOSED ENCODER ALGORITHMS

As discussed above, AIF schemes have high encoding complexity mainly due to the need to compute the adaptive filter and the additional encoding pass with the new calculated filter. The complexity of the filter calculation process could be reduced by using data from subset of the macroblocks (see [4] for more details). To reduce the complexity of AIF due to the additional encoding pass, two fast encoding algorithms are presented. First algorithm assumes that the motion information obtained by the static filter in the first pass is highly correlated with the motion information obtained by the adaptive filter in the second pass. By exploiting this correlation, very lightweight motion estimation could be

performed in the second encoding pass by reusing many of the coding results of the first pass. Second algorithm assumes that the effect of aliasing has a content depend nature and does not undergo a fast temporal variation. Using this assumption we do not perform the second encoding pass after the adaptive filter is calculated. Instead we use the computed adaptive filter to code the subsequent frame. This way, second pass is completely eliminated and single-pass coding is achieved using adaptive interpolation. Details of the proposed algorithms are given in the next subsections.

### A. Coding with Restricted Second Pass

Our first design is based on the assumption that a full-scale ME and RDO during the second coding pass are unnecessary and they can be restricted to operate over fractional-pixel locations only. To verify our assumption, we performed extensive simulations over VCEG test sequences as defined in [10] and compared the resulting motion vectors from the first and second pass that use static and adaptive filter respectively. The motion vectors resulting from the second pass could differ from the first pass motion vectors in three possible ways. The first case is that the motion vector estimated for a coded block at the second coding pass stays the same as the motion vector selected at the first coding pass. Second case is that the motion vector selected at the second coding pass can be equal to motion vector from the first pass up to integer-pixel accuracy and differ only in the fractional components. Thirdly, second pass results a completely different motion vectors estimated at the first pass either due to an RDO mode change or due to a different ME decision.

The results of this simulation and the probabilities of each one of the three possible cases are shown in Table-1. Analysis of Table I show that on average only 11.7% of the coded blocks retrieve completely new motion vectors at the second coding pass. Whereas, 60% of coded blocks have the same motion vectors and 28% of blocks retrieve motion vectors different in fractional component only. These results show that ME on integer-pixel locations at the second coding pass is unnecessary for 88% of the coded blocks on average. Thus, we can avoid ME on integer-pixel locations at the second coding pass and use motion information from the first coding pass instead, without significant penalty on coding efficiency.

TABLE I  
MOTION VECTOR VARIATIONS BETWEEN THE FIRST AND THE SECOND CODING PASSES

Sequence	No Change of MVs (%)	Fractional Components Change, (%)	Complete Change of MVs, (%)
Container	79.49	19.96	0.55
ForemanQ	55.89	28.63	15.49
Paris	79.73	14.60	5.67
Foreman	47.22	29.26	23.52
Mobile	76.82	16.40	6.78
Tempete	71.17	25.24	3.59
BigShips	48.97	42.47	8.55
City	62.71	24.76	12.52
Crew	34.46	45.47	20.07
Night	49.09	32.18	18.73
ShuttleStart	52.63	33.12	14.25
<b>Average</b>	<b>59.83</b>	<b>28.37</b>	<b>11.7</b>

Using the assumption described above, proposed algorithm is designed as follows.

- 1) *Initial Interpolation*: Interpolate the reference image with the static interpolation filter.
- 2) *First-Pass Encoding*: Encode the input frame, perform ME and RDO.
- 3) *Store Motion Information*: Store motion data (best motion vectors at integer-pixel accuracy) for each of macroblocks' partitions:

$$MD = \{mv\}_S,$$

where  $S = \{16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 4 \times 8, 8 \times 4 \text{ and } 4 \times 4\}$  is a set of all available partitions.

- 4) *Calculate AIF*: Calculate the optimal filter coefficients by minimizing the prediction error using the motion information (optimal modes, partitions and motion vectors) obtained at the first step.

- 5) *Adaptive Interpolation*: Interpolate the reference frame with the computed adaptive filter.

- 6) *Second Encoding Pass*: Avoid ME on integer-pixel locations and use instead motion data from the Step 3. For each macroblock partition we re-use integer-pixel motion vectors preserved in the Step 3 and refine the fractional component of the motion vector. This refinement is illustrated in Figure 3. First, eight half-pixel locations neighboring the integer motion vector is searched, as shown in Fig. 3.a, and then similarly the eight quarter-pixel locations neighboring the best half-pixel location are searched.

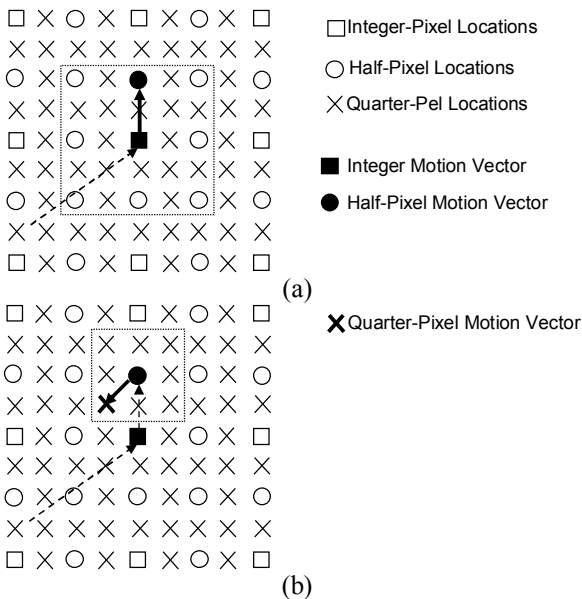


Fig. 3. Refinement of fractional component of motion vectors (a) half-pixel refinement, (b) quarter-pixel refinement

Based on the analysis reported in [5]-[7], the most time-consuming coding modules in a typical H.264/AVC video encoder include the integer pixel Motion Estimation (ME) and Rate-Distortion Optimization (RDO). These modules typically

consume 60% (1 reference frame) to 80% (5 reference frames) of the total encoding time [5], and the proportion can be much higher with larger search range. Whereas quarter-pixel ME is usually estimated as 10% of the total processing time [6]. Based on this data, the encoding complexity of the second encoding pass is significantly reduced (up-to 50-70%) by avoiding the integer-pixel ME.

#### B. Single-Pass Coding with Adaptive Interpolation

First method described above reduces the encoding complexity by performing lightweight motion estimation in the second encoding pass. The second proposed algorithm reduces the encoding complexity even further by completely removing the second pass. This is achieved based on the observation that in majority of the cases content does not change dramatically from frame to frame, thus the effect of aliasing stays almost constant between subsequent frames. Thus, we assume that an adaptive filter computed for frame  $N$ , could accurately compensate the effect of aliasing for frame  $N+1$ . The proposed algorithm could be described as:

- 1) *Initialization 1*: Initialize interpolation filter  $h$  in Fig. 1 with static filter coefficients.
- 2) *Interpolation*: Interpolate the reference frame  $Y$  with  $h$ .
- 3) *Encoding*: Encode current frame  $X$  with reference frame  $Y$ .
- 4) *Calculate AIF*: Calculate the filter coefficients that minimize prediction error for frame  $X$  by using the coding results of previous step.
- 5) *Initialization 2*: Initialize interpolation filter  $h$  with the computed filter, go on to next frame and do Step 2.

## IV. EXPERIMENTAL RESULTS

Performances of the proposed fast encoder algorithms are evaluated by comparing them against Directional Adaptive Interpolation Filtering (DAIF) without any encoder optimizations, and against the static H.264/AVC interpolation filter [9]. The simulations are performed using the Baseline common conditions and IPP prediction structure as detailed in [10] using a single reference frame. The complexity reduction is estimated by measuring the execution times for all mentioned algorithms using ANSI C instructions.

The coding efficiency results of the proposed algorithms are presented in Table I, with respect to static interpolation filter. In addition, exemplary rate-distortion curves are given for "City" and "Crew" in Fig. 4 and 5 respectively. Table III presents the achieved complexity reduction of the proposed algorithms with respect to not using any optimizations at all. *Fast1* and *Fast2* in Tables II and III refer to Restricted Second Pass, and One Pass Coding methods respectively as described in previous section.

As seen from Tables II and III, video coding with adaptive interpolation provide a significant coding gain over the static interpolation. Two pass video coding with DAIF [3] reduces the bit-rate by 11% for high resolution videos but it is also around 2.5 times more complicated than encoding with static filter. The fast encoding algorithm *Fast1* with restricted ME

and RDO provides very close to two-pass DAIF coding performance whereas reducing the execution time delta almost by two times (see Table III Fast1 vs. DAIF). Fast encoding algorithm Fast2 reduces the complexity even further in three times (156% vs. 54%) with higher coding efficiency penalty (-7.76% vs. -4.06% of bit-rate reduction on average). It should be noted, Fast2 is penalized mostly for QCIF and CIF sequences, whereas its performance for 720p sequences is pretty close to two regular two pass encoding scheme.

TABLE II  
CODING EFFICIENCY OF THE PROPOSED ALGORITHMS WITH RESPECT TO H.264/AVC INTERPOLATION FILTER

Resolution	Sequence	Delta Bit-rate, %		
		DAIF[3]	Fast1	Fast2
QCIF	Container	-10.24	-10.02	-3.37
	Foreman	-6.23	-6.34	1.18
CIF	Paris	-0.96	-0.89	1.6
	Foreman	-5.98	-5.81	-1.52
	Mobile	-3.33	-3.32	1.01
	Tempete	-2.19	-2.16	1.05
720p	BigShips	-12.15	-11.95	-10.79
	City	-15.41	-15.26	-13.13
	Crew	-13.32	-12.74	-8.15
	ShuttleStart	-11.4	-10.9	-9.49
	Night	-4.53	-4.35	-3.05
<b>Average</b>		<b>-7.76</b>	<b>-7.61</b>	<b>-4.06</b>

TABLE III  
ENCODING COMPLEXITY OF THE PROPOSED ALGORITHMS WITH RESPECT TO H.264/AVC

Resolution	Sequence	Delta Time, %		
		DAIF[3]	Fast1	Fast2
QCIF	Container	121.52	60.33	22.68
	Foreman	174.37	114.96	63.46
CIF	Paris	126.53	64.65	24.17
	Foreman	173.66	110.12	60.02
	Mobile	152.63	86.12	55.69
	Tempete	167.88	103.61	73.85
720p	BigShips	155.26	84.65	64.95
	City	168.05	99.79	76.69
	Crew	156.10	86.71	48.36
	ShuttleStart	167.19	101.67	56.17
	Night	149.24	84.76	44.54
<b>Average</b>		<b>155.68</b>	<b>90.67</b>	<b>53.69</b>

## V. CONCLUSIONS

In order to compensate for the temporally changing effect of aliasing and improve the coding efficiency of video coders, adaptive interpolation filtering schemes have been recently proposed. In such schemes, encoder computes interpolation filter coefficients for each frame and re-encodes the frame with the new adaptive filter. However, the coding efficiency comes with the expense of increased encoding complexity due to this additional encoding pass. In this paper we presented two fast encoding algorithms for adaptive interpolation filtering schemes. First method performs quarter-pixel refinement of motion vectors in the second coding pass, instead of performing full motion estimation. And secondly, a single-pass encoding scheme is achieved with adaptive interpolation by reusing the filter coefficients already computed for previous frames. Experimental results show that the proposed methods achieve between 1.5 to 2 times encoding complexity reduction with practically negligible penalty on coding efficiency.

## REFERENCES

- [1] Y. Vatis, B. Edler, D. T. Nguyen, J. Ostermann, "Motion and Aliasing-Compensated Prediction Using a Two-dimensional Non-Separable Adaptive Wiener Interpolation Filter," in *Proc. ICIP'05*, 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] K. Ugur, D. Rusanovskyy, M. Gabbouj, "Efficient Calculation of Adaptive Interpolation Filter with Distortion Modeling," in *Proc. ICASP'08*, 2008.
- [5] L. Su, Y. Lub, F. Wub, S. Lib, W. Gao, "Real-time Video Coding under Power Constraint Based on H.264 Codec", SPIE-IS&T/Vol.6508.
- [6] J. Ostermann, J. Bormans, P. List, etc., "Video coding with H.264/AVC: Tools, Performance, and Complexity," *IEEE Circuits and System Magazine*, Vol.4, No.1, pp.7-28, 2004.
- [7] Z. Chen, P. Zhu and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," JVT-F017.doc.
- [8] Reference ITU-T VCEG-KTA Software [Online]. Available: <http://iphome.hhi.de/suehring/tml/download/>
- [9] ITU-T Recommendation H.264 & ISO/IEC 14496-10 (MPEG-4) AVC, "Advanced Video Coding for Generic Audiovisual Services", v.3: 2005.
- [10] 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 2007

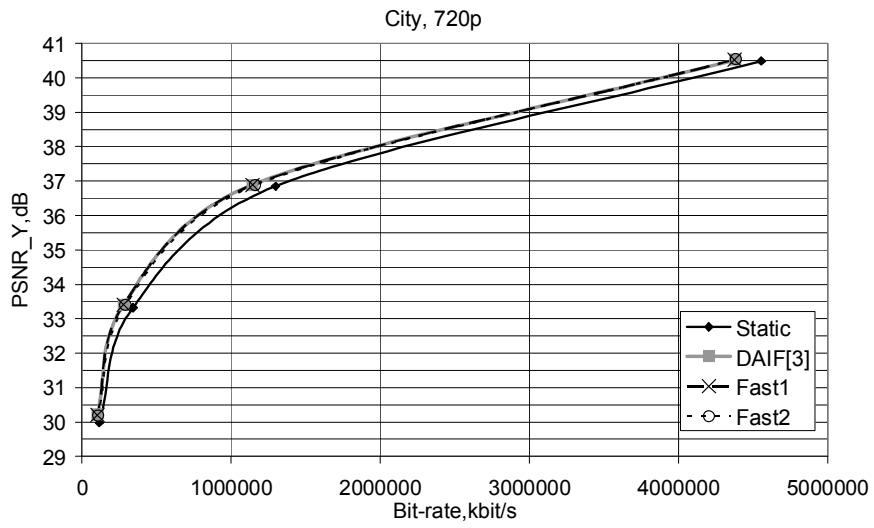


Fig.3. Rate-Distortion Curves for "City", 720p test sequence

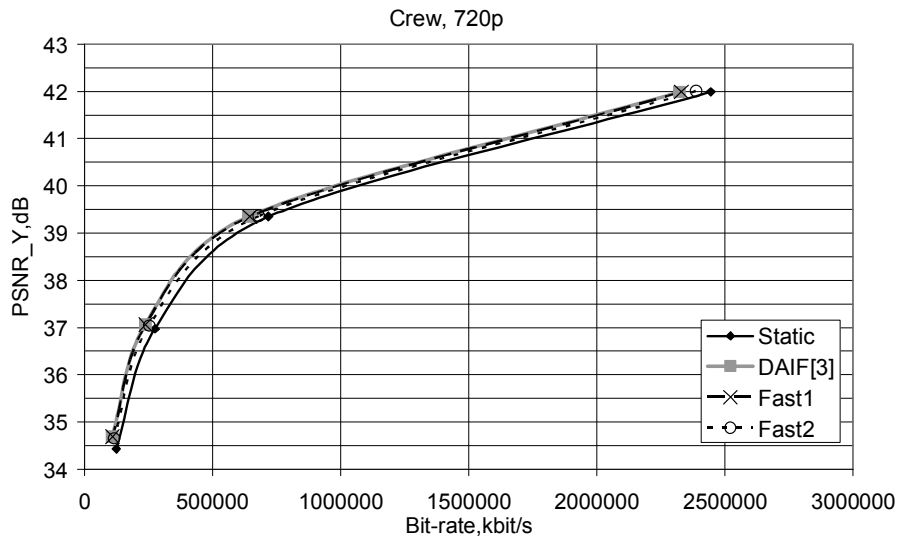


Fig.4. Rate-Distortion Curves for "Crew", 720p test sequence