ADAPTIVE INTERPOLATION FILTER WITH FLEXIBLE SYMMETRY FOR CODING HIGH RESOLUTION HIGH QUALITY VIDEO

Kemal Ugur[1], *Jani Lainema*[1], *Moncef Gabbouj*[2]
[1] Nokia Research Center, Tampere, Finland
[2] Tampere University of Technology, Tampere, Finland

ABSTRACT

In this work, a novel sub-pixel interpolation algorithm is proposed for video coders targeted towards high resolution and high fidelity use cases. Proposed scheme is based on adapting the interpolation filter's symmetry assumptions in a Rate-Distortion-Optimized fashion, taking into account the coding rate and the statistical properties of each image of the video sequence. Experimental results show that, using the proposed algorithm a gain of up-to 1.1 dB is achieved compared to non-adaptive sub-pixel interpolation of H.264/AVC. Compared to other state-of-the-art adaptive sub-pixel interpolation methods, a gain of up-to 0.5 dB is achieved. Proposed scheme outperforms H.264/AVC for all test cases; however, improvement is more significant at high bitrates and at high resolutions. This is especially important for future video coding solutions targeting high fidelity video applications.

Index Terms— Video Coding, Interpolation

1. INTRODUCTION

Motion Compensated Prediction (MCP) is a technique used by all popular video coding standards to exploit the temporal redundancy present in a video sequence. In MCP, each frame of a video sequence is divided into blocks and reference frame is searched for each block to find its best match. The relative position of the best matching block with respect to the original one is called the block's motion vector. Motion Vectors may have either integer or fractional pixel accuracies. If the motion vector has fractional pixel accuracy, decoder needs to perform interpolation to obtain the samples at sub-pixel positions. The state-of-the art video coding standard, H.264/AVC, supports motion vectors with half or quarter pixel accuracy. 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 defined in H.264/AVC was designed to minimize the adverse effects of aliasing present in the input image sequence [2][3]. 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 combat with this non-stationary effect of aliasing. [3] [4]**.**

In [4], Vatis *et. al.* proposed to use 2D non-separable adaptive interpolation filter 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. In order to reduce the number of bits used to code the filter coefficients, it was assumed that statistical properties of the image signal are symmetrical. With this assumption, filter coefficients are assumed to be the same, in case the distance of the corresponding full-pixel to the current sub-pixel is equal. This assumption greatly reduces the number of coefficients that need to be transmitted to the decoder.

In this paper, we claim that symmetry assumption in [4] has two shortcomings. Firstly, the statistical properties of the image sequence are not stationary and fixed symmetry assumption may not hold for every frame of the image sequence. For example, some frames of the video may contain complex scenes that do not hold neither horizontal nor vertical symmetry, whereas some other frames are relatively simpler and both horizontal and vertical symmetry assumption is valid. Secondly, the symmetry assumption in [4] does not provide a good trade-off between the accuracy of the interpolation filter and the overhead for coding the filter coefficients for high resolution, high quality material, and better trade-offs are possible to improve the coding efficiency significantly.

In order to exploit the varying statistical properties of the video sequence, and provide a better trade-off for high quality-high resolution material, we defined several filters with different symmetry assumptions, each one utilizing a different number of coefficients. It is proposed to adapt the filter's symmetry type for each frame of the video sequence in a Rate-Distortion Optimized fashion. Experimental results show that, when compared to H.264/AVC, proposed scheme improves the coding efficiency by up-to 1.1 dB. The improvement is 0.5 dB when compared to the scheme presented in [4]. This paper is organized as follows; Section 2 provides a brief background of 2D adaptive non-separable interpolation filtering. Section 3 presents the details of the proposed scheme. Section 4 presents the experimental results, and Conclusions are presented in Section 5.

2. OVERVIEW OF 2-D NON-SEPERABLE ADAPTIVE INTERPOLATION FILTERING

Consider Figure 1, where the pixels at integer positions are labeled by upper-case letters within shaded boxes, and other symbols represent sub-pixel positions to be interpolated. For each fractional pixel position, an interpolation filter is defined. Let's first assume that interpolation filters do not hold any symmetry. For positions *a,b,c,d,h,n* that are horizontally or vertically aligned with an integer pixel, one-dimensional 6-tap filter is used (Horizontal filter for positions *a,b,c,* and vertical filter for positions *d,h,n*). For other positions, two-dimensional 6x6 tap filter is utilized. Without any symmetry, this scheme results in 6 independent 1-D filters and 9 independent 2-D filters, resulting in 360 coefficients, which is a significant overhead to be transmitted to the decoder.

In order to decrease this overhead, it was assumed that statistical properties of the image signal are symmetric. Thus, the same filter coefficients are used in case the distance of the corresponding full-pixel positions to the current fractional-pixel position is the same. It is also assumed that distance in horizontal and vertical directions are treated equal. In other words, this assumption could be characterized as an interpolation filter with horizontal, vertical and diagonal symmetry, and is denoted as *HVD-Filter*.

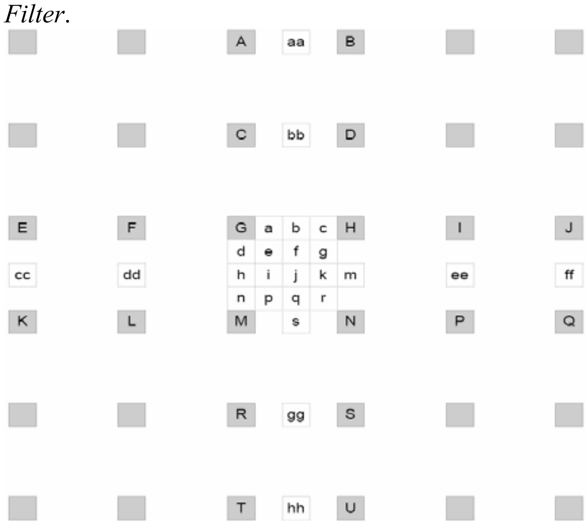

Figure 1 Pixel Positions Used in Interpolation

With the symmetry assumption in place, the number of coefficients needed for defining the interpolation filter is reduced in two ways. First, the number of filters to be transmitted are reduced from 15 to 5 (only filters for positions *a,b,e,f,j* need to be transmitted). Also, a smaller number of coefficients are needed for each filter (for example, only 3 coefficients are needed to define the 1-D 6-tap filter for sub-pixel location *b* as horizontally symmetric filter is being used).

3. PROPOSED INTERPOLATION FILTER WITH FLEXIBLE SYMMETRY

The symmetry assumption described in Section 2.1 provides a good trade-off between the accuracy of the interpolation filter and the overhead used to code the filter coefficients for low and medium resolution material. However, for coding higher resolutions at higher qualities, this tradeoff does not hold. In addition, the statistical properties of the image sequence may not be stationary, and the symmetry assumption for the interpolation filter may not hold for every frame of the video sequence. Some frames of the video may contain complex scenes that neither holds horizontal nor vertical symmetry, whereas some other frames could be relatively simpler and both horizontal and vertical symmetry assumption is valid.

In order to exploit the varying statistical characteristics of the video sequence, and provide a better trade-off for high quality-high resolution material, we defined several filters with different characteristics. Each filter holds a different symmetry type, and is defined with different number of coefficients.

3.1. 2-D Non-Separable Interpolation Filters with Flexible Symmetries

In addition to the *HVD-Filter* as described above, we define four new filters with different combinations of symmetry assumptions. More specifically, following filters are being used in our proposed scheme:

- Filter with Horizontal Symmetry (*HOR-Filter*): Image signal is assumed to have only horizontal, but no vertical or diagonal symmetry.
- Filter with Vertical Symmetry (*VER-Filter*): Image signal is assumed to have only vertical, but no horizontal or diagonal symmetry.
- Filter with Horizontal and Vertical Symmetry (*H+V-Filter*): Image signal is assumed to have horizontal and vertical, but no diagonal symmetry.
- Filter with No Symmetry Assumptions (*FULL-Filter*): Image signal is assumed to have no horizontal, vertical and diagonal symmetry.

Figure 2 illustrates the number of filters used for each one of the above symmetry assumptions. In Figure 2, each square represents a sub-pixel pixel position, and the number inside each square represents the identification number of the interpolation filter used for the corresponding sub-pixel position. Sub-pixel positions with the same identification number share the same interpolation filter (For example, for the *HVD-Filter*, the sub-pixel locations *a,c,d,n* use the same filter that is denoted by ID number **1).** The coefficients of the filter that is shared by more than one sub-pixel positions are transmitted only once. Coefficients are transmitted only for the shaded sub-pixel positions as denoted in Figure 2, whereas for other positions denoted as white squares no coefficients are transmitted.

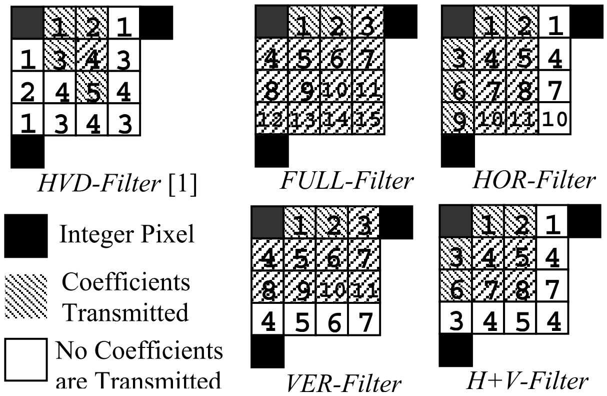

Figure 2 Number of Independent Filters for Different Symmetries

As seen from Figure 2, the number of filters changes for each of different symmetry assumptions. For example, 5 filters are needed for *HVD-Filter*, whereas 11 filters are used for *HOR* and *VER Filters*.

In addition to the number of filters, number of coefficients used for each filter depends on the filter's symmetry assumption. For example, consider *HOR*-Filter where only horizontal symmetry is assumed. In this case, sub-pixel position *b* uses a symmetrical horizontal filter, whereas position *h* uses a 6-tap non-symmetric vertical filter. Similar to *HVD-Filter*, all filter types except the *FULL-Filter* use a 1D filter for sub-pixel positions that are horizontally or vertically aligned with an integer position (sub-pixel positions *a,b,c,d,h,n*). For the *FULL-Filter* these sub-pixel positions do not use a 1D filter, but a 2D filter. It is known that using a 2D filter is not necessary to remove the aliasing effect for positions aligned with the integer pixel. However, using a 2D filter for those positions is useful to combat with the adverse effects of quantization and motion estimation errors that also decrease the prediction quality [3].

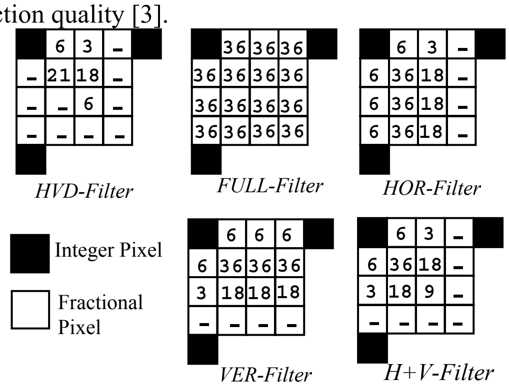

Figure 3 Number of Coefficients for Each Sub-Pixel Position at Different Symmetries

Figure 3 illustrates the number of coefficients used by each sub-pixel position for the filters with different symmetry assumptions. The numbers inside each square represent how many coefficients the interpolation filter uses for the corresponding sub-pixel position. For example, for position *j*, *HVD-Filter* uses 6 coefficients, whereas in *HOR-Filter* and *VER-Filter* it uses 18 coefficients. The total numbers of coefficients are 54, 540, 189, 189 and 99 for the *HVD*, *FULL*, *HOR*, *VER* and *H+V-Filters* respectively.

3.2. Calculating Interpolation Filter's Symmetry Type and Coefficients

In order to find the symmetry type of the interpolation filter and its corresponding coefficients, we make use of the following encoder algorithm that is useful to estimate the upper bound performance of our proposed scheme.

1. Motion Estimation is performed using the standard non-adaptive H.264/AVC interpolation filter, and motion vectors for every macroblock are found.
2. For each symmetry type, the filter coefficients are calculated to minimize the prediction error energy. This step is done by constructing and solving a set of linear equations using the previously reconstructed frame and motion vectors found in Step-1. The number of linear equations is equal to the number of coefficients that need to be transmitted to the decoder for the given symmetry type (For details of constructing the linear equations, reader is referred to [4]).
3. Using the interpolation filter found in Step-2, frame is re-encoded. The cost for using the given filter symmetry type is calculated using Equation 1.

$$J(f) = D(f, \mathbf{m}) + \lambda_{mode} \cdot (R_1 + R_2) \qquad (1)$$

where $\mathbf{m}$ = (mx, my)^T is the new motion vector set for the frame, λ_{mode} being the Lagrange multiplier as defined in [6] and f denoting the interpolation filter's symmetry type. D is the distortion term, and it is defined as the Mean Square Error (MSE) between the original and reconstruction signal. R_1 is the number of bits used to code the residual signal including the header information (such as motion vectors, macroblock modes etc.) and R_2 is the number of bits used to code the filter coefficients. When calculating R_2 it is assumed that filter coefficients are quantized and coded using 10 bits per coefficient.
4. Steps 2&3 are repeated for each candidate symmetry type. Filter symmetry with the minimum cost is chosen, and its corresponding coefficients are signaled to the decoder.

It should be noted that, this encoding scheme requires several encodings per frame, in order to achieve the practical upper-bound performance. However, the number of encodings could be reduced to two per frame, by removing Step-3 and using a modified cost function that does not need the reconstruction for each candidate filter. This approach is conceptually similar to fast mode decision in H.264/AVC software that does not require the reconstructed macroblock [5]. The decoder complexity of our proposed scheme is practically same as in [4], except that *FULL-Filter* has slightly higher complexity because of using 2D filters for positions *a,b,c,d,h,n*.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of our proposed scheme, several sequences at 4CIF (704x576) and 1080p (1920x1080) resolutions are first encoded using the H.264/AVC standard without any adaptive sub-pixel interpolation. Same sequences are then encoded utilizing the adaptive interpolation scheme described in [4] where only *HVD-Filter* is used and then using our proposed scheme that adaptively changes the interpolation filter's symmetry assumptions. Coding tools common in the Baseline profile of H.264/AVC standard are used in all the simulations and we limit the number of reference frames to one. Quantization Parameter (QP) is kept constant for each simulation (no rate control is employed), and QP takes the values between 24 and 32. Figures 4 and 5 present the results for two standard test sequences. For sequence *Soccer*, an average gain of 0.9 dB is achieved compared to H.264/AVC, with gains up-to 1.1 dB at high bitrates. Our scheme outperforms [4] by 0.5 dB at high bitrates and by around 0.25 dB at lower bitrates. Having a larger gain at high bitrates validates our claim that symmetry assumption in [4] does not provide a good trade-off for higher bitrate encoding. In order to see how bitrate affects the choice of filter types, we compare the selection frequency of different filters for two different QP's for the *Soccer* sequence in Figure 6 (For the sake of clarity, only three filters are shown in the figure). It is seen that at low bitrates (with large QP), encoder mostly chooses filters that does not require much overhead for the coefficients. However, with the increasing bitrate more accurate filters utilizing more coefficients are chosen and the coding efficiency is improved further.

5. CONCLUSIONS

In this paper, we presented a novel sub-pixel adaptive interpolation filtering scheme to improve the coding efficiency of a video coder. Proposed scheme is based on adapting the interpolation filter's symmetry at each frame in a Rate-Distortion-Optimized fashion, taking into account the coding rate and the statistical properties of each image. Experimental results show that the proposed scheme outperforms the H.264/AVC standard by up to 1.1 dB. When compared to [4], a gain of up-to 0.5 dB is achieved.
Proposed scheme outperforms H.264/AVC and [4] for all test cases; however, improvement is more significant at high bitrates and at high resolutions. This is especially important for future video coding solutions targeting high fidelity video applications.

6. REFERENCES

[1] Karczewicz, M.; Hallapuro, A.; "Interpolation solution with low encoder memory requirements and low decoder complexity.", VCEG-N31, 24-27 Sept. 2001.

[2] O. Werner, "Drift analysis and drift reduction for multiresolution hybrid video coding", Signal Processing: Image Commun., vol. 8, no. 5, July 1996.

[3] T. Wedi and H. G. Musmann, "Motion- and Aliasing-Compensated Prediction for Hybrid Video Coding", IEEE Trans. on CSVT, vol. 13, No. 7, July 2003.

[4] 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. IEEE ICIP 2005, Genova, Italy, September 2005.

[5] H.264/AVC Reference Software [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[6] Wiegand, T.; Schwarz, H.; Joch, A.; Kossentini, F.; Sullivan, G.J., "Rate-constrained coder control and comparison of video coding standards," IEEE Transactions on CSVT, vol.13, no.7pp. 688- 703, July 2003

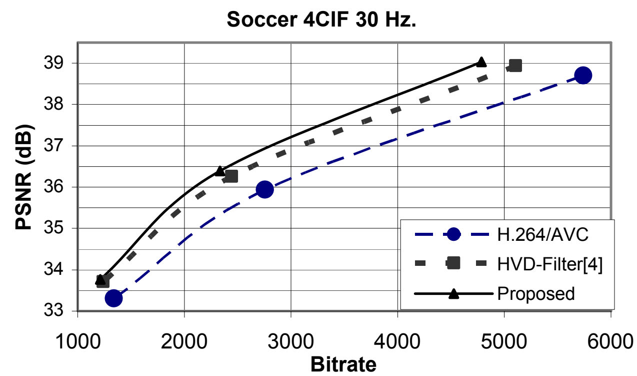

Figure 4 Coding Results for Soccer 4CIF Sequence

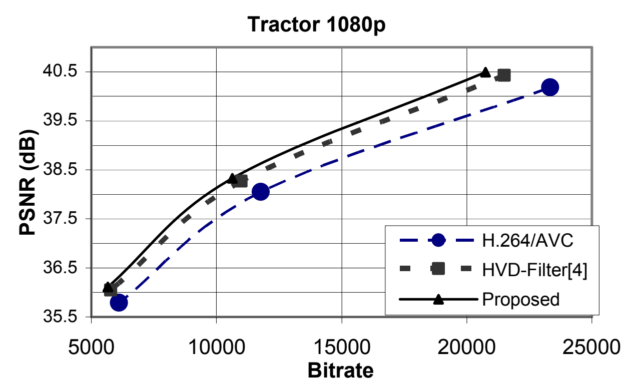

Figure 5 Coding Results for Tractor 1080p Sequence

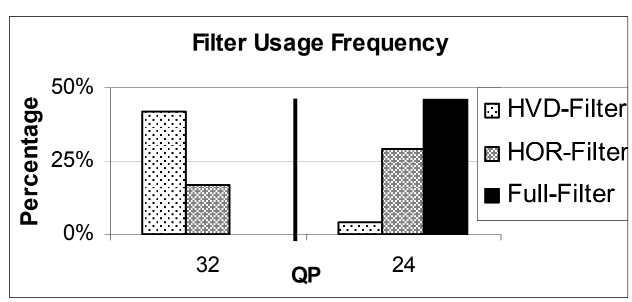

Figure 6 Filter Usage Statistics at Different Bitrates