EFFICIENT CALCULATION OF ADAPTIVE INTERPOLATION FILTER WITH DISTORTION MODELLING

Kemal Ugur[1], *Dmytro Rusanovskyy*[2], *Moncef Gabbouj*[2]
[1]Nokia Research Center, Tampere, Finland
[2]Institute of Signal Processing, Tampere University of Technology, Tampere, Finland

ABSTRACT

A novel method is proposed to calculate the coefficients of adaptive interpolation filter used in hybrid video coders for improving the coding efficiency. The proposed algorithm first selects the motion blocks where the majority of prediction errors result from mismatches in motion estimation and from aliasing present in the signal. This is realized by using a second order distortion model to estimate the effect of quantization on motion prediction error and coding results of the previous frames. Then, the filter coefficients are calculated analytically by minimizing the prediction error of those selected blocks. Experimental results show that the proposed method achieves up-to 0.6 dB gain compared to the standard H.264/AVC. Compared to other methods that calculate the filter coefficients using all motion blocks of the frame, the proposed method has significantly less encoding complexity (83% on average) with practically no penalty on coding efficiency.

***Index Terms*—** Adaptive Interpolation, Video Coding

1. 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 interpolation filters that change the filter coefficients at each frame have been proposed in the literature to combat this non-stationary effect of aliasing [1], [2]. In [1], Vatis *et. al*. proposed a 2D non-separable adaptive interpolation filter to reduce the prediction error energy and improve the coding efficiency of video coders. For each fractional pixel position, this scheme utilizes an independent filter and each filter is calculated analytically by solving a system of linear equations that are constructed by minimizing the prediction error energy. This construction step is the main source of complexity due to the need to compute complex cross-correlation and auto-correlation functions for every fractional pixel sample. After the coefficients of the adaptive filter are found, the reference frame is interpolated with this filter and the frame is encoded.

In this work, we propose a novel algorithm to calculate the coefficients of the adaptive filter with significantly less complexity than its counterparts but still improves the coding efficiency significantly over H.264/AVC. The proposed algorithm first identifies the motion blocks that are not suffering from prediction errors due to aliasing and motion estimation mismatch, marks them as redundant and excludes them in the filter calculation process. This identification process uses a second order distortion model and coding results of previous frames to provide a stable performance over various quantization levels and sequences with different motion characteristics. Using the proposed method, the encoder computes cross-correlation and auto-correlation functions not for every fractional pixel but only for a significantly lower subset of them. As computing the correlation functions is by far the most complex part of the filter calculation process, reduction in its complexity translates in large gains in the overall encoding complexity. Experimental results show that, the proposed method achieves up-to 0.6 dB gain compared to standard H.264/AVC. Compared to other methods that calculate filter coefficients using all the motion blocks of the frame, proposed method has significantly less encoding complexity (83% on average) with practically no penalty on coding efficiency.

This paper is organized as follows; Section 2 describes the 2D adaptive interpolation filter that is used as a basis for our method and presents an analysis of its encoding complexity. Section 3 presents the details of the proposed algorithm. Experimental results are given in Section 4, and Section 5 concludes the paper.

2. NON-SEPARABLE 2D-ADAPTIVE INTERPOLATION FILTER

The adaptive interpolation scheme proposed in [1] is based on 2D non-separable adaptive filters defined independently for each fractional pixel location. For each fractional pixel position, the filter coefficients are calculated analytically by minimizing the prediction error energy. For positions that are horizontally or vertically aligned with an integer pixel, one-dimensional 6-tap filter is used. For other positions, two-dimensional non-separable 6x6 tap filter is utilized.

The filter coefficients are calculated for every frame of the video sequence using the following steps. First, the motion vectors are found by performing motion estimation using the standard non-adaptive filter. In order to find the optimal filter that minimizes the prediction error, a system of linear equations is constructed using the motion vectors found in the previous step (see [1] and [5] for more details). The 2D non-separable filter for the sub-pixel location *SP* is given by $h_{m,n}(SP)$, where *m,n* are the horizontal and vertical coefficient indexes, respectively. The system of linear equations needed to calculate $h_{m,n}(SP)$ is illustrated in (1).

$$\sum_{j=0}^{5}\sum_{i=0}^{5} R^{r}_{m-i,n-j} h_{m,n}(SP) = R^{cr}_{m,n}, \quad m,n = 0,...,5 \tag{1}$$

where R^{cr} and R^{r} are the cross-correlation and auto-correlation functions computed over the reference and the predicted image. Equation-1 is then solved to find the coefficients of the adaptive interpolation filter. The reference frame is interpolated with the new filter and the frame is encoded.

2.1 Complexity Analysis

As seen from the above description, an encoder that supports adaptive interpolation filter needs to perform several additional operations, compared to standard non-adaptive interpolation filtering. These additional operations are, *i)* constructing the system of linear equations (1) using cross correlation and auto-correlation functions, *ii)* solving Equation-1 and *iii)* interpolating using the adaptive filter. Solving Equation-1 is not a complex problem itself, while constructing it is a computationally demanding task. This is mainly because of the need to perform many multiplication and addition operations for each sub-pixel that motion vectors over the entire frame. For example, the computation of the correlation function for a given sub-pixel position requires more operations than interpolating the corresponding sub-pixel.

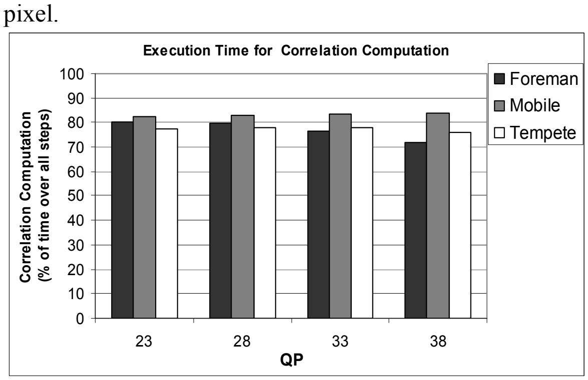

Figure 1 Execution time for computation of correlation functions in percentage of all steps combined

In order to measure how much time it takes for an encoder to construct Equation (1) compared to other operations, we performed several simulations with different sequences and quantization parameters. For every coded frame, we compared the time needed for each of the above steps. The simulation results are illustrated in Figure-2 for three different sequences coded with different quantization parameters. In Figure-2, the time encoder spends on computing the correlation functions is shown in terms of percentage of the three steps combined. As expected, it was found out that computing the correlation functions takes most of the time spent in the filter calculation process (around 75-85%). Therefore, reducing the complexity of constructing Equation-1 has significant importance on reducing the overall encoding complexity.

3. THE PROPOSED ALGORITHM

As it was shown above, the main complexity of calculating the filter coefficients is due to the need of computing correlation functions for every fractional pixel location over the entire frame. The proposed algorithm aims to reduce this complexity by using only a subset of data, and hence reduce the total number of complex correlation calculations. This is achieved by identifying motion blocks that do not suffer from prediction errors due to aliasing and motion estimation mismatch, marking them as redundant. This identification process is done at the first encoding pass, by comparing the prediction error of each block against a threshold value that is dynamically computed for every frame of the video sequence. If a block is marked redundant, fractional pixels within the block are excluded in the filter calculation process. Equation-2 shows the condition to mark a 4x4 block as redundant, whose coordinates are i,j, frame index is N and quantization step is *Q*.

$$SAD_{4x4}(i,j) < D_1(Q) + D_2(N) \tag{2}$$

where $SAD_{4x4}(i,j)$ is the motion prediction error of the 4x4 block given in Sum of Absolute Differences (SAD). It should be noted that the block size of 4x4 is chosen to match the smallest motion block partition supported by the H.264/AVC standard. In Equation-2, there are two terms contributing to the distortion threshold. First term depends only on quantization step-size and it is related to removing the effect of quantization in motion prediction signal, so that proposed algorithm provides reliable results at all quantization step sizes. Second term is related to achieve a stable coding efficiency-complexity trade-off over the entire sequence.. The details of how these two terms are calculated are given in the following subsections.

3.1 Error Threshold due to Quantization

In motion compensated prediction, Motion prediction errors may result from three possible sources, i) quantization of the reference signal, ii) aliasing present in the original signal and iii) errors in displacement estimation [2]. Our goal is to find those blocks in which the majority of the prediction mismatches are due to aliasing and displacement estimation errors, and not due to quantization. For this purpose, an empirical model is developed that relates motion prediction error of blocks to quantization step size using statistical analysis, to estimate the prediction error due to quantization and compute the distortion threshold $D_1(Q)$.

The relationship between prediction error and quantization has been well studied in Rate-Control context, where the goal is to estimate the quantization step size given the target bits and the prediction error. One well known Rate-Control algorithm developed for MPEG-4 utilizes a second-order Rate-Distortion model for this purpose [3]. In the context of this paper, a model is needed that relates the prediction error to quantization step size, without requiring the use of bitrate information. Therefore, we adopt a similar second order model as in [3] but modify it accordingly so that the rate term is not included. The model used is given by:

$$D_1(Q) = a_1.Q^2 + a_2.Q + a_3 \qquad (3)$$

where $D_1(Q)$ is the prediction error due to quantization of the reference signal, Q is the quantization step size and a_1,a_2,a_3 are the model parameters.

The model parameters a_1,a_2,a_3 are calculated empirically using the following statistical analysis. A test sequence is first encoded and the prediction error of every 4x4 block is recorded. According to the resulting prediction errors, a threshold is calculated, which results in *M%* of blocks having less prediction error than the threshold value. This threshold is indicated as $SAD_{4x4}\_Thr(Q,M)$, and it is calculated for many test sequences with different motion characteristics using a wide-range of quantization step sizes. It should be noted that, the percentage *M* controls the trade-off between encoding complexity reduction and coding efficiency. The details of how *M* affects the coding efficiency-complexity tradeoff is shown in the Experimental Results section. After these thresholds are calculated for all the sequences and quantization step-sizes, the model parameters are found by fitting to data, the second order polynomial defined by Equation-3 using least-squares method.

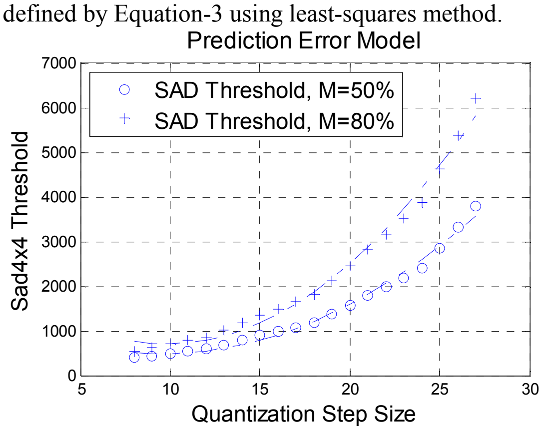

Figure 2. Prediction error due to quantization. Dotted lines indicate the calculated error model.

This process is illustrated using the following example. Let us consider the test sequence Foreman, which is encoded with varying quantization step sizes and the resulting prediction errors of 4x4 blocks are recorded. For each of the quantization step sizes, the threshold values $SAD_{4x4}\_Thr(Q,M)$ are found using the above described process. Figure-2 presents the threshold values plotted against the quantization size for two values of M, 50% and 80%. Circles in Figure-2 indicate the threshold values for M equal to 50% and pluses indicates threshold for M being equal to 80%. After these data are computed, the second order polynomial is found and shown with dotted lines in Figure-2.

3.1 Dynamically Modifying the Error Threshold

The distortion threshold used to determine redundant blocks needs to be dynamically computed for every frame of the sequence. This is because the motion characteristics of a sequence can vary dramatically from one segment to another, and can significantly affect the performance of the proposed scheme. Consider the case where a certain segment of the video has relatively low motion, which results in lower than usual prediction error for the blocks. In this case, most or all of the motion blocks are identified as redundant and excluded in the filter calculation process. This results in significant degradation of the coding efficiency for that segment of the video as the calculated coefficients of the filter will not be optimal. Similarly, if some parts of the video undergo a large steady motion, the prediction error for most of the blocks will be large. This would result in including many blocks in the filter calculation process, even though it is not needed, and complexity could not be reduced efficiently.

In Figure-3 this problem is illustrated for two of the frames from the low and high motion segments of the Foreman sequence. The vertical dashed line indicates the threshold value due to quantization ($D_1(Q)$ in Equation-2), which is calculated empirically as described in Section 3.1. The histogram of the prediction errors of blocks for those frames is also plotted. As seen in Figure-3.a, the prediction error of almost all of the blocks is smaller than the threshold $D_1(Q)$ (%), which means many blocks are marked as redundant. This results in coding efficiency penalty, as the resulting filter coefficients are far from optimal for that frame. Similarly, as seen in Figure 3.b, majority of the blocks have larger error than the threshold value, which results in high interpolation complexity for that frame.

This problem is solved in the proposed algorithm by adding the second term in Equation-2 to provide a stable trade-off between coding efficiency and complexity reduction throughout the entire sequence. $D_2(N)$ is the distortion threshold computed for every frame with frame index *N*, and it is independent from the quantization step size. $D_2(N)$ is computed as follows. After the encoding of a frame *N-1* is finished, the resulting motion prediction error of each 4x4 motion block is recorded and $D_2(N)$ is calculated so that *M*% of the blocks in frame *N-1* has less prediction error than the value of $D_2(N)$. This way, if a high motion segment starts in a video sequence, the value of $D_2(N)$ increases, which in turn increases the final distortion threshold and the same complexity reduction-coding efficiency trade-off is achieved throughout the sequence.

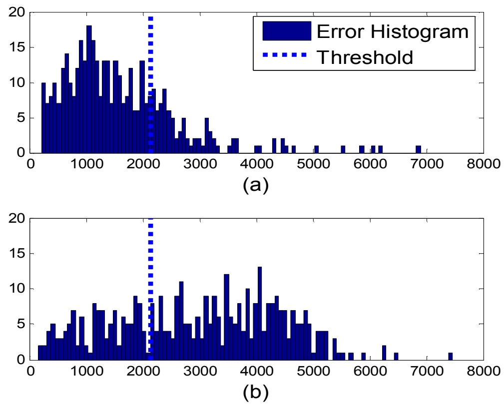

Figure 3. Error Histogram of different frames of Foreman sequence. (a) from low motion segment and (b) from high motion segment.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of our proposed scheme, we first encoded several sequences at different resolutions using standard H.264/AVC without employing any adaptive interpolation filtering and using the using the method [1]. Same sequences are finally encoded using the proposed algorithm with two different values of *M* (50% and 80% respectively) to illustrate how the tradeoff between complexity reduction and coding efficiency could be controlled. The simulations are performed using the Baseline common conditions as detailed in [6]. The encoding complexity and the coding efficiency of the proposed algorithm is compared with [1] in Table 1, where **ΔQ%** indicates the percentage of complexity reduction estimated using the number of fractional pixels used in the filter calculation process, and **Δ**PSNR refers to the coding efficiency difference with respect to [1]. As seen in Table-1, the proposed algorithm achieves practically the same coding efficiency with 83% less encoding complexity on average. It can also be seen that for some sequences, a small gain in coding efficiency is realized. This is because adaptive interpolation filter is optimized to certain areas of the image that suffer the most from aliasing and motion prediction errors. This is also inline with the conclusions drawn in [4], where the interpolation filter is optimized locally inside a frame, using several encoding passes. In Figure-4, the performance of the proposed algorithm is compared against H.264/AVC for one of the test sequences, ShuttleStart. It is seen that the proposed algorithm achieves a coding efficiency gain of up-to 0.6 dB at high bitrates compared to H.264/AVC.

5. CONCLUSIONS

In this work, a novel method is proposed to calculate the coefficients of the adaptive interpolation filter used in hybrid video coders for improving the coding efficiency. The proposed algorithm first identifies the motion-blocks that do not suffer from prediction errors due to aliasing and motion estimation mismatch and excludes them from the filter calculation process. This identification process uses a second order distortion model and coding results of previous frames to provide a stable performance over various quantization levels and sequences with different motion characteristics. It was shown that the proposed method achieves up-to 0.6 dB gain compared to standard H.264/AVC. Compared to other methods that calculate filter coefficients using all the motion blocks of the frame, the proposed method achieves practically the same coding efficiency with significantly less encoding complexity (83% on average).

6. 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", *Proc. ICIP 2005*, Genova, Italy, September 2005.

[2] 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.

[3] Hung-Ju Lee; Tihao Chiang; Ya-Qin Zhang, "Scalable rate control for MPEG-4 video," Circuits and Systems for Video Technology, IEEE Transactions on , vol.10, no.6pp.878-894, Sep 2000

[4] Y. Vatis, J. Ostermann "Locally Adaptive Non-Separable Interpolation Filter for H.264/AVC" *Proc. ICIP 2006*, Atlanta, USA, October 2006

[5] A. Papoulis, Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, 1991.

[6] T.K. Tan, G. Sullivan, T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments", VCEG contribution VCEG-AA10, Nice, Oct. 2005.

Table 1. Simulation Results

	Percentage M			
	50%		80%	
	ΔQ,%	**ΔPSNR**	**ΔQ,%**	**ΔPSNR**
Container	53.13	0.02	93.63	0.08
Foreman	31.93	0	76.19	-0.01
ForemanCIF	46.97	0	84.90	-0.05
Mobile	22.71	0	58.87	0.02
Tempete	29.50	0	83.70	-0.05
ShuttleStart	71.14	0	97.28	-0.02
City	24.41	-0.02	86.40	-0.03
Average	**39.97**	**0.00**	**83.00**	**-0.01**

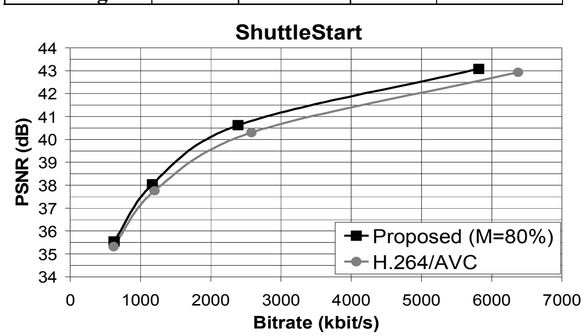

Figure 4. RD performance of proposed algorithm compared with H.264/AVC