A HEURISTIC SEARCH METHOD OF ADAPTIVE INTERPOLATION FILTERS IN MOTION COMPENSATED PREDICTIVE VIDEO CODING

Huipin Zhang and Frank Bossen

Mobile Media Laboratory, DoCoMo USA Labs., San Jose, CA 95110, USA

ABSTRACT

The paper studies the fractional pixel based motion compensation for video coding where fractional pixels are generated by filtering with adaptive interpolation filters. The design of adaptive filters is formulated as an optimization problem where the prediction error is minimized for each individual frame. A heuristic search method is introduced to obtain an optimized filter with high efficiency and accuracy. Performance of the method is analyzed and examined through simulations. The new filter design method saves up to 75% computation time while retaining coding efficiency equivalent to state-of-the-art methods reported in the literature.

1. INTRODUCTION

The state-of-the-art video codecs are all based on motion compensated prediction with motion vector of fractional pixel resolution. In MPEG-2 the motion vectors can be in half-pixel resolution (or precision); while in MPEG-4 Advanced Simple Profile, the resolution of the motion vectors can be of quarter pixel resolution. The Advanced Video Codec (AVC) [1] allows 1/8-pixel resolution for motion vectors. The purpose of fractional pixel resolution is to get more accurate definition of picture content displacement. The increase of the resolution of motion vectors leads to an increase of the accuracy of the prediction which may result in a coding gain.

Frames in fractional pixel resolutions are generally generated by low-pass filtering the original pixel-precision frame. The underlying reason is that in the prediction process from frame to frame the low and medium frequencies content is typically well suited for prediction whereas high frequencies content tends to be less predictable. Low-pass filters of different lengths and different coefficients have been studied to generate fractional pixels with various resolutions during the standardization of AVC. The generation of fractional pixels with 1/4-pixel resolution is based on a subsequent interpolation scheme in AVC TML-8 [2]. Specifically, as shown in Fig. 1, the low-pass filter

$$f_1 = [1\ -5\ 20\ 20\ -5\ 1]/32 \qquad (1)$$

Email: zhang@docomolabs-usa.com, bossen@docomolabs-usa.com

is used to generate the half-pixels and subsequently the bilinear filter $f_2 = [1\ 1]/2$ is applied to the interpolated image to generate 1/4-pixels.

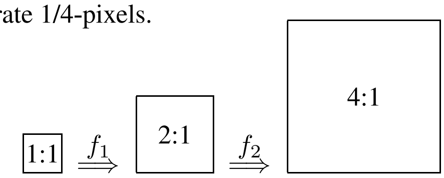

Fig. 1. Subsequent interpolation process for generating fractional pixels of 1/4-pixel resolution.

The focus of this paper is on how to design frame-level adaptive interpolation filters (AIF) f_1 to generate the 1/4-pixels. The essence of the study is to allow the interpolation filter to adapt to the non-stationary statistical properties (e.g. aliasing, motion) of the video signals, thus improving the performance of motion compensation. The paper adopts the same framework as [3, 4] for the design of adaptive interpolation filters of the form:

$$f = [\ a_2\quad a_1\quad a_0\quad a_0\quad a_1\quad a_2\ ]^t,$$

where f has up to six taps and is symmetric. The design of AIF for a given frame can be formulated as an optimization problem of minimizing the overall prediction error $\varepsilon_{\text{pred}}$ for motion compensated macroblocks in a frame. Here the calculation of the prediction error corresponding to an interpolation filter is based on an interpolation of the reference frame with the filter. The prediction error is a function of the interpolation filter and also of the motion vectors. To simplify the problem, the motion search with an initial interpolation filter (such as f_1 in (1)) is performed first and with the obtained motion vectors the prediction error $\varepsilon_{\text{pred}}$ is then assumed to be dependent on the interpolation filter f only. Therefore there are only three parameters in the simplified optimization problem $h_{\text{opt}} = \arg\min_h\ \varepsilon_{\text{pred}}(h)$, where $h = [a_0\ a_1\ a_2]^t$ is the parameter vector, h_{opt} is the optimal parameter vector and the prediction error $\varepsilon_{\text{pred}}(h)$ depends on the parameter vector h only.

The prediction error $\varepsilon_{\text{pred}}$ is defined in this paper as the Sum of Absolute Differences (SAD) between the original pixels and the predicted pixels in a frame. As a function of interpolation filter coefficients, the prediction error

$\varepsilon_{\text{pred}}(h)$ is not a convex function, since the 2-dimensional filter generates a quadratic function of the filter coefficients. There are numerous numerical optimization methods to solve non-convex optimization problems. The variable metric method [5] and the downhill simplex method [5] are employed to design the adaptive interpolation filter in [4]. In this paper a new numerical search method designed specifically for searching optimal interpolation filter is introduced. The heuristic search method is similar to the downhill simplex method searching for a minimum in the search space by a sequence of trials. Trials are successively performed in a direction of improvement until an optimum solution is reached. The method is however quite different from the downhill simplex method in many aspects such as the basic moves, the rules that control the moves, etc. In addition, the new method is gradient-based while the downhill simplex method is not. Experiments have shown that the new search method is up to 75% faster than the numerical optimization methods in [6, 4] while retaining similar coding efficiency, i.e., up to 1.0 dB PSNR improvement over AVC TML-8 for general video sequences.

The paper is organized as follows. The details about the heuristic search method, including formulation, heuristics set design and performance enhancement, are described in Section 2. Experimental results are given in Section 3 and Section 4 concludes the paper.

2. A HEURISTIC SEARCH METHOD

2.1. Filter search over a discrete filter space

Assume that the parameter vector h can be represented in the form

$$h = [\ h_0 \quad h_1 \quad h_2\ ]^t / h_3,$$

where h_i's are all integers. Clearly, the optimal interpolation filter can be obtained by searching a discrete filter space whose precision is signified by the integer h_3. The finer the grid of the search space, the higher the precision of the filter coefficients.

Given a grid search space of filters, an optimal filter with the corresponding precision can be obtained by exclusively examining filters in a search region. The filter which produces the smallest prediction error is the optimal filter. An optimal filter with finer precision may be obtained by searching a filter space with a denser grid, where the expansion of the filter space to a higher precision can be achieved by doubling the filter coefficients. Therefore, a search of an optimal interpolation filter can be performed iteratively, from a search space of a lower precision to higher, until an objective is met. Another view of this procedure is that the optimal filter is progressively quantized to generate optimal filters of various precisions.

2.2. Heuristic search set

The exhaustive search of a search region in a discrete filter space is computationally expensive, since each filter in the region has to be examined by interpolating the reference frame with the filter. A reduction of the amount of computation can be achieved by a reduction of the number of filter candidates examined in the search. Specifically, the search is performed over a dynamic search set rather than a search region. The search set corresponds to a sequence of trials. Trials are successively performed in a direction of improvement until an optimum is reached. The search set is dynamically determined by the trials.

Given an up-to-date minimum $c = [c_0\ c_1\ c_2]^t$ in a successive search, the skew cube centered at c

$$\Omega = \left\{ \begin{array}{l} [h_0\ h_1\ h_2]^t : h_0 \in [c_0 - 1, c_0 + 1], h_1 \in [c_1 - 1, c_1 + 1], \\ h_0 + h_1 + h_2 \in [c_0 + c_1 + c_2 - 1,\ c_0 + c_1 + c_2 + 1] \end{array} \right\},$$

turns out to be an efficient search set. Filters in this search set are located on three parallel planes: $h_0 + h_1 + h_2 = \nabla$, where constant ∇ takes value $c_0 + c_1 + c_2 - 1$, $c_0 + c_1 + c_2$, or $c_0 + c_1 + c_2 + 1$. It is known that the sum of the impulse response of the filter $2(h_0 + h_1 + h_2)$ characterizes the filter gain, thus filters on each of the three planes maintain a same gain. These planes are thus called *equal-gain planes*. It is noted that the designed search set consists of filters that are close to the up-to-date minimum in terms of their performances rather than their geometric positions.

Group 1	Group 2	Group 3
$[c_0 + 1\ c_1\ c_2 - 1]$	$[c_0 + 1\ c_1 + 1\ c_2 - 2]$	$[c_0\ c_1\ c_2 + 1]$
$[c_0 - 1\ c_1\ c_2 + 1]$	$[c_0 + 1\ c_1 - 1\ c_2]$	$[c_0\ c_1\ c_2 - 1]$
$[c_0\ c_1 + 1\ c_2 - 1]$	$[c_0 - 1\ c_1 + 1\ c_2]$	
$[c_0\ c_1 - 1\ c_2 + 1]$	$[c_0 - 1\ c_1 - 1\ c_2 + 2]$	

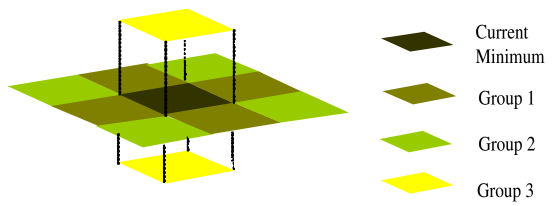

Fig. 2. Design of heuristic search sets: Ten neighbor search method uses Groups 1, 2, and 3; Six neighbor search method uses Groups 1 and 3.

With the designed search set, the search method is still computationally expensive, since there are 27 trials to be performed each time, and 2/3 of the trials are performed with filters having gain different from the up-to-date minimum. Experiments show that an efficient way to reduce the size of the search set is to exclude filters that have different gains from the up-to-date minimum. However, a complete exclusion of these filters may degrade the performance of the searching method, especially in the case when the video signal has a fading transition. So a compromise of the design results in a search set which consists of 10 filters in three groups as shown in Fig. 2. The filters in Group 1 and Group 2 have the same gain as the up-to-date minimum and two filters in Group 3 have respectively a stronger and a weaker gain. Overall this simplified search set is not only efficient but also accurate in the search method. The heuristic search method with 10 neighbors in a search set is call *ten neighbor search method.* It is also interesting to note that the filter set which consists of only 6 filters in Group 1 and Group 3 is an even more time efficient search set. Experiments show that coding efficiency is not degraded by using this smaller search set. This heuristic search method is called *six neighbor search method.*

2.3. Complete procedure

A complete description of the heuristic search method can be briefly summarized as follows.

1. Select an initial filter for the search.
2. Determine a new search set based on the up-to-date minimum filter which may be the initial filter at the beginning of the iterations or a minimum filter from last set of trials.
3. Perform all trials defined in the search set. This involves interpolating the reference image using every trial filter and computing the corresponding motion compensated prediction errors.
4. Control the iterations by examining whether a new minimum is found through the trials. If yes, go to step 1 to continue trials of filters of the same precision, else increase the precision of the current optimal filter by doubling its coefficients and go to step 1 to start a search of filters of a higher precision. The iteration is terminated when a control condition is met.

From the implementation point of view, there are two iterations in the successive trial process. The iterations in the inner loop controls the search region of the filters of a same precision, and the iteration in the outer loop controls the precision of the filters. In the inner loop the search set is dynamically expanded by including new filters described in Groups 1, 2, and 3, and trials are successively performed in a direction of improvement until the optimum solution is reached. In this case, the outer loop increases the precision of the optimal filter by doubling the filter coefficients and form a new search set to start new iterations in the inner loop. The two loops respectively correspond to a deeper search (outer loop) and a wider search (inner loop). The two types of search may also be regarded as two types of move (deeper move and wider move) in a numerical search.

2.4. Performance enhancement

The initial filter should ideally be close to the optimal filter, so that it can be found through a small number of trials. Experiments have shown that the default filter with precision modification $h = [160 \quad -40 \quad 8]/256$ performs very well for general video signals [6]. In addition, for frames following a predicted frame of a video sequence, the optimal filter of the previous frame may be another candidate for the initial filter.

A so-called *restarting mechanism* may be used to enhance the search method. This mechanism works with a threshold that controls the number of wider moves; when the threshold is achieved and the next move is still a wider move, the restarting mechanism is triggered and a new initial filter with a lower precision is set. This can be accomplished by dividing the current optimal filter by a factor of 2 with a possible modification of the third coefficient h_2 to make the new initial filter and the current minimum filter be on the same equal-gain plane. The restarting mechanism not only reduces the dependency of the performance of the search method on the initial filter, but also improves the efficiency of the method.

To terminate iterations in the heuristic search for a filter, it is very beneficial to set a threshold for the difference of the best prediction errors corresponding to the current minimim filter and the previous minimum filter. Whenever the difference is smaller than a threshold, the search is terminated with the current minimum filter. This mechanism is called the *early termination mechanism.* Due to the nonlinearity resulting from the quantization of DCT coefficients and entropy coding, the correlation between the prediction error and bit saving is not perfect. A threshold is thus introduced to avoid superfluous iterations.

During the filter search process, smooth regions may be excluded from the calculation of the prediction error that is being minimized, since the associated prediction error does not change significantly with different interpolation filters. This strategy of improving the efficiency of the heuristic search method is called the *partial prediction error mechanism.* To determine the smoothness of regions, each macroblock (MB) in the current frame is classified as either a smooth MB or a non-smooth MB by considering the activity in the MB.

3. EXPERIMENTAL RESULTS

Experiments have been performed to verify the performance of the heuristic search methods introduced in this paper, including the ten neighbor search method (TNSM) and the six neighbor search method (SNSM). In the experiments two examples are used to compare the performances of the methods and the state-of-the-art downhill simplex search method (DSSM) [6, 4]. The two video sequences in the examples are the 30 Hz CIF size Mobile sequence and the 30 Hz QCIF size Foreman sequence, each containing 300 frames. The experiments are based on AVC TML-8. More specifically, the experimental conditions are the same as TML-8 except the interpolation filter. The conditions are summarized as follows. The software is TML-8 of AVC with subsequent interpolation scheme. The first frame is coded as an I-picture; all remaining ones are as P-pictures. The precision of motion vectors is 1/4-pixel and one reference frame is used for the motion estimation which is not RD-optimized. The initial value for the heuristic search is $h = [160 \quad -40 \quad 8]/256$, or the optimized filter obtained for the previous frame. The restarting mechanism, the early termination mechanism and the partial prediction error mechanism are all enabled.

Method	QP		Bitrate (Kbps)		PSNR_Y (dB)	
	M	F	M	F	M	F
DSSM	16	16	1898.24	151.45	33.88	35.89
TNSM			1909.63	151.84	33.88	35.86
SNSM			1908.96	152.09	33.88	35.86
DSSM	21	20	779.96	84.04	29.69	33.26
TNSM			784.06	84.47	29.67	33.23
SNSM			784.02	83.95	29.68	33.20
DSSM	26	24	327.40	51.69	26.04	30.81
TNSM			328.24	51.44	26.00	30.82
SNSM			330.21	51.46	25.99	30.86
DSSM	31	29	174.03	31.15	22.89	27.68
TNSM			174.17	31.45	22.84	27.86
SNSM			173.86	31.50	22.85	27.79

Table 1. A PSNR/bitrate comparison for three different AIF design methods for Mobile (M) and Foreman (F) sequences.

The compression efficiency and the computational complexity are two issues that are examined in the experiments. Table 1 shows bitrates and corresponding PSNRs for both sequences. The computational complexity of the search method is reflected by the average time used for encoding a frame. This average time, varying with the quantization parameter QP, characterizes the complexity of the video encoder. Fig. 3 shows encoding time savings of the heuristic search method with respect to the downhill simplex method as a function of QP.

It is easily seen from Table 1 that the heuristic search methods introduced in this paper achieve coding efficiency equivalent to the state-of-the-art downhill simplex method, i.e., improving the coding efficiency of up to 1.0 dB over TML-8 for ordinary video sequences [3, 6]. However, compared to the downhill simplex method, the new methods save up to 75% encoding time, thus demonstrating their efficiency.

4. CONCLUSIONS

This paper proposed a heuristic search method of adaptive interpolation filters for motion compensated predictive coding of video signals. The design of the optimal adaptive filter is accomplished by a search over a discrete filter space, where the cost function is the motion prediction error. Since

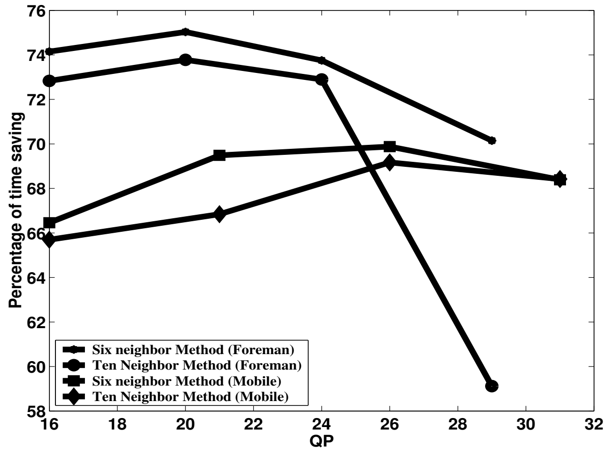

Fig. 3. The encoding time saving of the heuristic search methods with respect to the downhill simplex method is plotted as a function of QP.

the best filter obtained by the search results in a smaller prediction error, the compression performance is generally improved. Experiments showed that for general video sequences up to 1.0 dB PSNR improvement has been achieved with respect to a fixed filter interpolation and a computation time saving of up to 75% with respect to the state-of-the-art methods using adaptive filters.

Acknowledgements: The authors would like to thank T. Wedi for sharing his software.

5. REFERENCES

[1] T. Wiegand, *Joint Committee Draft (CD)*, Joint Video Team of ISO/IEC MPEG and ITU-T VCEG. doc. JVT-C167, Fairfax, USA, May 2002.

[2] G. Bjontegaard and T. Wiegand, *26L Test Model Long Term Number 8 (TML-8)*, ITU-T SG16/Q6. June 28 2001.

[3] T. Wedi, "Adaptive interpolation filter for motion compensated hybrid video coding," in *Picture Coding Symposium (PCS 2001)*, Seoul, Korea, April 2001.

[4] T. Wedi, *New Results on Adaptive Interpolation Filter*, Joint Video Team of ISO/IEC MPEG and ITU-T VCEG. doc. JVT-C059, Fairfax, USA, May 2002.

[5] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C++ - The Art of Scientific Computing*, Cambridge University Press, 2002.

[6] T. Wedi, *Adaptive Interpolation Filter for H.26L*, ITU-T SG16/Q6. doc. VCEG-N28, Santa Barbara, USA, September 2001.