JOINT OPTIMIZATION OF TRANSFORM COEFFICIENTS FOR HIERARCHICAL B PICTURE CODING IN H.264/AVC

Martin Winken, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand

Image Communication Group, Image Processing Department
Fraunhofer Institute for Telecommunications – Heinrich-Hertz-Institut
Einsteinufer 37, 10587 Berlin, Germany, [winken|hschwarz|marpe|wiegand]@hhi.fraunhofer.de

ABSTRACT

Coding of video sequences using hierarchical B pictures in the Joint Scalable Video Model (JSVM) for the scalability amendment of H.264/AVC has the benefit of improved rate distortion efficiency relative to other known temporal decomposition structures, besides providing temporal scalability. In the operational encoder control of the JSVM, the inter-picture dependencies within a hierarchical B picture structure are considered using a heuristic, where pictures that are more frequently used for motion compensation are coded with higher fidelity compared to pictures that are less often used for motion compensation. In this paper, we describe an approach where the dependencies introduced by motion compensation are also considered when selecting transform coefficient values. Our experimental results using an H.264/AVC conforming encoder show bit rate reductions of up to 10 % compared to the quantization method used by JSVM.

Index Terms— Video coding, Rate Distortion optimization

1. INTRODUCTION

H.264/AVC offers increased flexibility compared to any prior video coding standard including coding using hierarchical B pictures as described in [1]. For the operational encoder control, the temporal dependencies within such a structure introduced by motion-compensated prediction (MCP) have to be considered to achieve good rate distortion efficiency. In this paper, we describe, based on the idea in [2], an approach to solve the problem of jointly estimating transform coefficient values for a complete hierarchical B picture prediction structure.

The next section gives a brief overview of hierarchical B pictures. Sec. 3 introduces the problem statement, and Sec. 4 shows how the problem size can be reduced by application of a sliding window approach. A complete description of our optimization algorithm is given in Sec. 5 and in Sec. 6, finally experimental results are presented.

2. HIERARCHICAL B PICTURES

A typical hierarchical prediction structure with 4 dyadic temporal stages is shown in Fig. 1. The pictures denoted as I_0 and P_0 are called *key pictures*. The key pictures build a self-contained subset of the sequence in the sense that for coding of a key picture only other (preceding) key pictures may be used as reference for MCP. A key picture and all pictures that are temporally located between this and the preceding key picture build a *group of pictures* (GOP). The non-key pictures of a GOP are coded as B pictures and use a hierarchical prediction structure as illustrated in Fig. 1. More precisely, for coding of a picture denoted as B_n only other pictures B_m of the same GOP (with $n > m$) or the two enclosing key pictures of the GOP may be used as reference. Thus, the decisions made when coding a picture B_m can only have impact on pictures B_n of the same GOP (with $n > m$). Since the lower the value of m, the more pictures are potentially influenced by this picture B_m, typically a cascading of quantization parameters (QP) is used such that for pictures at the top of the hierarchical prediction structure (I_0, P_0) a smaller quantization step size is used than for those at the bottom (B_3).

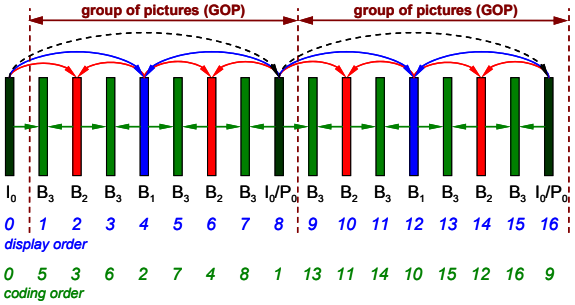

Fig. 1: Hierarchical B picture prediction structure

3. PROBLEM STATEMENT

We use the linear signal model from [2] where the reconstructed sample values are obtained as a linear combination of previously reconstructed samples, the residual samples, and a static predictor. Note, that the sample value clipping and picture deblocking operations of H.264/AVC are neglected in this simplified model. Considering a group of K pictures, each of width W and height H, this can be written as

$$s = Ms + Tc + p$$

with $N = K \times W \times H$, the vectors s, c, and p are $N \times 1$ column vectors with s being the reconstructed sample values, c being the transform coefficient values and p being a static predictor. M and T are $N \times N$ square matrices such that the product $M s$ gives the MCP signal and the product $T c$ gives the residual sample values. The actual values of M depend on the selected macroblock types, reference indices and motion vectors (in the following subsumed as motion parameters), whereas the actual values of T depend on the chosen QP values. Note, that both M and T are highly sparse matrices, since in H.264/AVC even using bidirectional prediction each sample of the MCP signal may depend only on up to 72 samples of the reference pictures, resulting from applying the separable 6-tap filter in both directions in two reference pictures ($6 \times 6 \times 2 = 72$). Thus, each row of M can only have 72 non-zero entries. Similarly, since each 4×4 block is inverse transformed independently, each row of T has exactly 16 non-zero entries. The static predictor p gives the part of the prediction signal which is not contained in $M s$, namely the intra prediction signal and the MCP signal obtained from reference pictures outside the considered group of K pictures.

With fixed, pre-determined motion parameters and therefore fixed M, as well as fixed QP values and therefore fixed T, the problem of selecting optimal transform coefficients can now be stated as in [2]:

$$c_{opt} = \arg\min_c \{D(c) + \lambda R(c)\}, \quad \text{subject to } s = Ms + Tc + p \qquad (1)$$

Here, $D(c)$ gives the distortion between original x und reconstruction s in terms of the sum of squared differences, $R(c)$ gives the bit rate needed for coding the transform coefficients c, and λ is the Langrangian multiplier which determines the trade-off between required bit rate and distortion. Note, that this Langrangian multiplier λ should not to be confused with the one used in the operational rate distortion optimization of the reference encoder software (as described in [3]) since we do not use the real bit rate but a piece-wise linear approximation. Since $R(c)$ is a very intricate function of c, we use, as in [2], the sum of absolute values as an approximation, leading to:

$$D(c) = \|x - s\|_2^2, \qquad R(c) = \|c\|_1$$

Now, the optimization problem can be stated as a (convex) quadratic program with inequality constraints, which enables us to use iterative numerical optimization algorithms to get a real-valued solution vector c_{opt}. For our application of video coding, however, an integer-valued solution is required. Generally, solving the given quadratic program under an integer-constraint on the variables is a hard combinatorial optimization problem. A simple, yet effective heuristic to obtain an integer-valued solution vector c_{opt} by iteratively solving a series of quadratic programs without integer-constraint is given in [2]. Here, in each iteration cycle, a number of variables are fixed to their nearest integer value and in the next iteration the problem with the remaining variables is solved. Since the number of free variables is decreasing with each iteration cycle, the computation time needed to solve the program decreases, too.

3.1. Application to hierarchical B pictures

The described optimization approach is very well suited to hierarchical B picture coding structures, as will be explained below. As shown in Sec. 2, hierarchical B picture structures have very straightforward "tree-like" dependency structures introduced by MCP, since the impact of any B picture is limited to other B pictures in higher temporal stages ("sub-trees") of the same GOP. Thus, considering only non-key pictures and having already determined motion parameters and QP values (e. g., using the method specified in the JSVM reference encoder software), the optimization problem stated in the previous section can now be applied to the whole set of B pictures at once, resulting in optimal transform coefficient values for all the hierarchical B pictures under the given simplifications, without having to neglect the impact on subsequent pictures, as in IPPP… structures, where the impact of a reference picture is potentially unlimited within a sequence of *pictures*.

4. PROBLEM SIZE REDUCTION USING A SLIDING WINDOW APPROACH

The optimization problem as stated above has the major drawback of leading to a very large number of variables. For the case of a GOP size of 16 pictures in QCIF resolution, the problem will consist of $N = 15 \times 176 \times 144 = 380160$ variables. This may be impracticable to solve, especially for even larger resolutions. We therefore apply the described transform coefficient optimization approach using a spatial sliding window. In other words, instead of solving the huge optimization problem covering *all* the

transform coefficients in the complete hierarchical B picture structure, we iteratively solve a series of smaller sized optimization problems, where each only covers a certain area ("optimization window") of the pictures to be optimized. Note, that in our approach this area is the same in all the pictures, and that the areas of subsequent iterations overlap in order to take into account the dependencies across two optimization windows in space and time. The operation of the sliding window is illustrated in Fig. 2, where each square represents one macroblock. The current optimization window is shown in yellow, the macroblocks whose optimized transform coefficients have already been determined are shown in orange, and those macroblocks which are shown in grey will be optimized in a later iteration.

4.1. Selection of the sliding window parameters

The sliding window is characterized by two parameters: the size of the optimization window and the step size by which the window gets shifted after each iteration cycle. Both values should be an integer multiple of the transform size, since the optimization is performed in the transform domain. Further, the window should be large enough, such that almost all the dependencies due to MCP within the hierarchical B picture structure are covered within the window. Generally, the best choice of these parameters depends on the characteristics of the specific sequence. For sequences with fast motion, a larger window size should be chosen than for rather static sequences with nearly no motion. Empirically, we found a window size of 3 macroblocks in each direction with a step size of 2 macroblocks (as shown in Fig. 2) to be an acceptable compromise for QCIF resolution pictures.

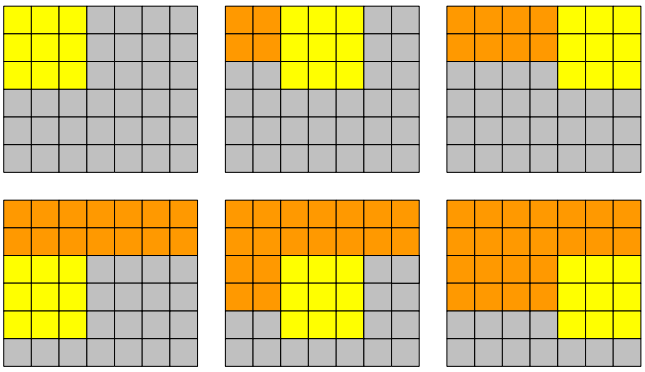

Fig. 2: Sliding window approach (yellow: optimization window, orange: already optimized macroblocks)

5. DESCRIPTION OF THE ALGORITHM

Having stated the basic principles of our approach, we will now describe in more detail how the algorithm proceeds when optimizing a GOP. In the first step, the complete GOP is encoded using the method specified in the JSVM reference encoder software in order to obtain motion parameters and QP values for each macroblock. In the next step, we use this information to generate the matrices M and T and solve the optimization problem for all the non-key pictures of the GOP.

In order to incorporate the changes to a reference picture due to the optimization of the transform coefficient values into the motion parameters of the pictures referencing this picture, we re-estimate in the next step the motion parameters for all pictures but the first one in the GOP. In the next step we again solve the optimization problem for the remaining pictures and repeat this process until reaching the last picture of the GOP.

To summarize, we use the following steps to optimize the transform coefficient levels for the hierarchical B pictures within a GOP of size K. Note that picture 1 is the key picture of the GOP.

1. set $k = 2$
2. Encode picture k and all its dependent pictures using the method specified in the JSVM reference software to obtain motion parameters (and QPs)
3. Solve the optimization problem for the transform coefficient values of picture k and all its dependent pictures using the sliding window approach
4. set $k = k + 1$
5. if $k <= K$, go to step (2)

6. EXPERIMENTAL RESULTS

For our experiments, we used a modified version of the JSVM reference encoder software. We used a GOP size of 16, each key picture but the first one was coded as a P picture. Note, that we did not allow usage of intra coding modes in P and B pictures. The transform coefficients of the key pictures (which build an IPPP… sub-sequence) have been optimized independently from the hierarchical B picture prediction structure using the method described in [2]. Note further, that we introduced weighting factors in the objective function (1), in order to minimize not only the rate distortion (RD) costs of the highest temporal layer (which includes all the pictures) at the cost of a lower temporal layer (which includes only a sub-set of the pictures), but to minimize the average over all temporal layer. More precisely, for our case of a GOP size of 16, the RD costs for pictures B_1 (as in Fig. 1) have been weighted with a factor of 4, the costs for pictures B_2 with a factor of 3, for B_3 with a factor of 2, and for B_1 with a factor of 1. In our experiments, this temporal stage dependent weighting resulted also in an overall improvement of the rate distortion efficiency at lower bit rates which is due to inaccuracies induced by simplifications in our linear signal model.

For numerically solving the sparse quadratic programs occurring in our approach, we used the MOSEK optimization software [5]. The resulting rate distortion plots are shown in Fig. 3. It can be seen that a bit rate reduction of 10% can be achieved by the described optimization method, resulting in a gain of up to 0.7 dB in terms of luma PSNR. Furthermore, it can be noticed that the fluctuation of the PSNR over the sequence is much smaller than using the method of the JSVM reference encoder (see Fig. 5). Comparing the results with and without the described picture weighting factors, it can be stated that without picture weights the fluctuation of the PSNR within the hierarchical B picture structure is the smallest, but this results in inferior rate distortion efficiency for lower temporal stages (see Fig. 4).

7. CONCLUSION

We have presented a new optimization framework for jointly selecting transform coefficient values in hierarchical B picture coding structures. Experimental results have been shown indicating that significant gains can be obtained by this approach. However, as a major issue of the described approach, there is large amount of computational complexity involved. A reduction of this computational complexity is possible if we solve instead of the large optimization problem a series of smaller problems using a sliding window approach.

8. REFERENCES

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF", *Proc. ICME 2006*, Jul. 2006.

[2] B. Schumitsch, H. Schwarz, and T. Wiegand, "Optimization of transform coefficient selection and motion vector estimation considering inter-picture dependencies in hybrid video coding", *Proc. IVCP 2005*, Jan. 2005.

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

[4] M. R. Osborne, B. Presnell, B. A. Turlach, "A new approach to variable selection in least squares problems", *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389-403, 2000.

[5] MOSEK ApS, "The MOSEK optimization tools. Version 4.0 (Revision 50)," http://www.mosek.com.

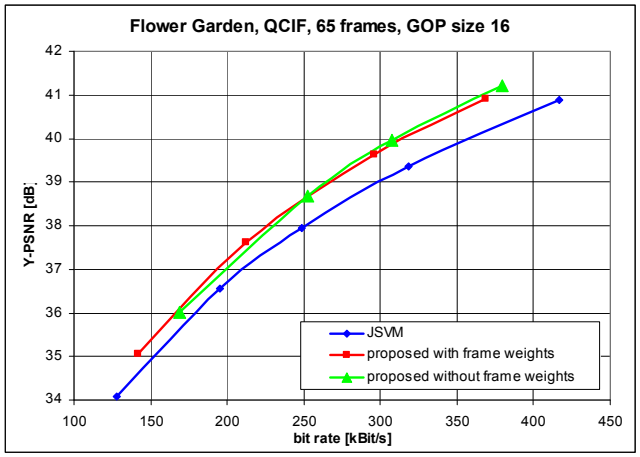

Fig. 3: Comparison of JSVM and described coding method (all pictures for the sequence)

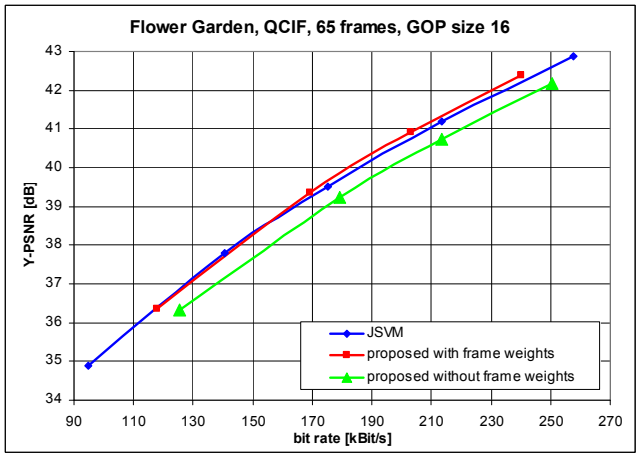

Fig. 4: Comparison of JSVM and described coding method (only for pictures I_0/P_0, B_1, and B_2)

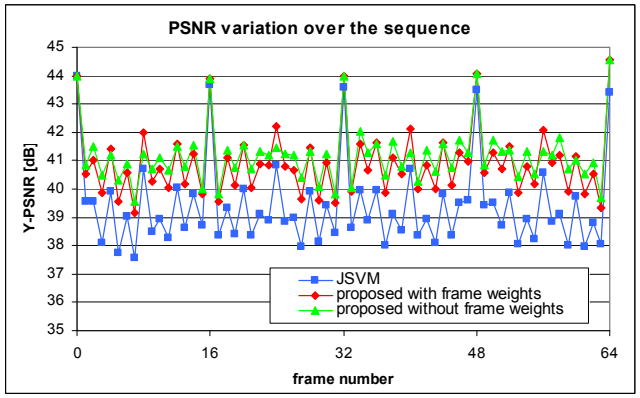

Fig. 5: Comparison of PSNR fluctuations