INTER-TEMPORAL VECTOR PREDICTION FOR MOTION ESTIMATION IN SCALABLE VIDEO CODING

Steffen Kamp, Dominic Heyden, and Jens-Rainer Ohm

Institute of Communications Engineering
RWTH Aachen University
52056 Aachen, Germany

ABSTRACT

Motion estimation plays an important role for the coding performance achieved in current video coding schemes. However, the computational burden of motion estimation algorithms is often high, especially if the temporal distance between pictures is large. We describe a predictive motion estimation algorithm that takes advantage of motion correlation found specifically in scalable video coding using open-loop hierarchical B pictures or motion-compensated temporal filtering. The simulation results illustrate that our proposed algorithm significantly reduces the computational complexity of motion estimation while maintaining objective and visual quality almost identical to an exhaustive full search algorithm.

***Index Terms*—** Scalable video coding, motion estimation, hierarchical B pictures

1. INTRODUCTION

Motion compensation (MC) is one among the key factors for the compression performance of most current video coding schemes. However, motion estimation (ME), i. e. determining the motion parameters of a video sequence, is computationally intensive and typically consumes most of the time spent during encoding. Most video-coding standards, such as MPEG-1/2/4 and H.264/AVC [1] employ block based motion compensation where individual pictures are partitioned into rectangular pixel regions and a displaced block from a reference picture is used as prediction for each partition. The encoder only codes the displacement (a motion vector, MV) for each region and a texture residual representing the difference between the original samples and the prediction. In scalable video coding (SVC) using hierarchical B pictures [2] or motion compensated temporal filtering (MCTF) [3], correlation of motion parameters in different temporal layers can be expected. In this paper, we propose to use temporal inter layer MV candidates within a recursive motion estimation scheme [4]. Thus significantly reducing the computational complexity of the motion estimation stage in temporally hierarchical video coding. The temporal inter layer candidates are derived from ME results from earlier temporal layers for which the motion estimation has already been carried out.

The paper is organised as follows. In Section 2 we review the main features of block-based motion estimation and video coding using hierarchical B pictures. The proposed fast motion estimation algorithm is described in Section 3. Finally, simulation results and comparisons are presented in Section 4.

This work was supported by Robert Bosch GmbH.

2. PRELIMINARIES

2.1. Block-Based Motion Estimation

During the ME process an optimal motion vector is typically found by minimising a cost function containing a distortion measure. Perhaps the most common distortion measure is the sum of absolute differences (SAD). For a block of size $M \times N$ located at pixel position $(x, y)^T$ inside the current picture s_t at time index t and a reference block at a displacement of $\boldsymbol{v} = (v_x, v_y)^T$ inside the reference picture $s_{t+\Delta t}$, the SAD is defined as

$$\mathrm{SAD}(v_x, v_y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |s_t(x+m, y+n) - s_{t+\Delta t}(x+m+v_x, y+m+v_y)|. \quad (1)$$

Rate-constrained ME also accounts for the estimated or actual number of bits needed to encode the motion parameters by minimising the Lagrangian cost function

$$J = \mathrm{SAD}(\boldsymbol{v}) + \lambda \cdot R(\boldsymbol{v}), \quad (2)$$

where R is the rate required to encode the motion information. The Langrangian multiplier λ allows for a trade-off between motion vector rate and texture rate and is typically derived from the quantiser settings at the encoder [2].

In general, block based ME can be described as minimising the cost function (2) for a set of motion vector candidates

$$\boldsymbol{S} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_\mathrm{n}\}. \quad (3)$$

Fast motion estimation algorithms can significantly reduce the number of computations while only minimally degrading the compression efficiency. In comparison to a full motion vector search, such fast algorithms typically reduce the set of motion vectors to be tested to sparse search patterns, possibly iteratively applying a pattern search centred around the best candidate ([5] et al.).

In typical video sequences, moving objects often cover image regions that are larger than the maximum MC block size or macroblock size. Therefore, spatially adjacent motion vectors are often highly correlated. This fact is often exploited in video coding systems by coding only the difference between a current motion vector and an associated motion vector predictor (MVP) derived from causal spatially adjacent vectors ([1] et al.). Moreover, correlation among temporally adjacent vectors can also be expected due to only slowly changing content within individual scenes. Many ME methods utilise a MVP as initial vector around which the search algorithm is centred [6, 7] or exclusively use a set of candidates composed of MVPs and vectors derived from the MVPs, relying on a convergence of the estimated MV field [4].

2.2. Hierarchical B Pictures

The fast motion estimation algorithm presented in this paper has been specifically designed for video coding using open-loop hierarchical B pictures or MCTF. A typical prediction structure with hierarchical B pictures is shown in Figure 1. Pictures are denoted as s_t^l with the subscript index t specifying the time index and the superscript index l specifying the temporal layer. The arrows are pointing from the reference pictures used for motion-compensated prediction towards the predicted picture, e. g. s_{2t+1}^{l+1} is bidirectionally predicted from motion compensated pictures s_t^{l+2} and s_{t+1}^{l+2}.

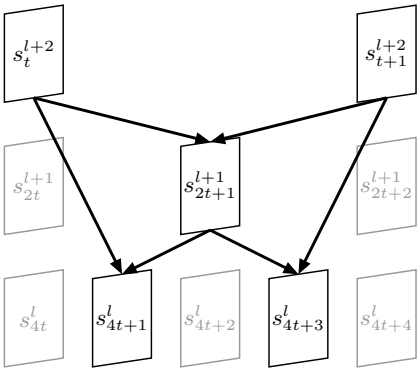

Fig. 1. Hierarchical B picture prediction structure with 3 temporal layers.

While each picture is coded only within one of the temporal layers, finer layers always include all pictures of coarser layers (greyed out pictures) for output to a display, i. e. pictures within a column are identical:

$$s_t^l \equiv s_{2^k t}^{l-k} \quad \forall \quad k \in \mathbb{Z}. \tag{4}$$

For closed-loop coding, decoded pictures are used as prediction references, so each temporal layer l must be coded before its next finer layer $l-1$ can be coded resulting in a coarser-to-finer coding order. In open-loop coding, original pictures are used as prediction references during the encoding process, removing the coding order constraint and allowing for finer-to-coarser encoding. While this increases the number of frame buffers needed during encoding, ME can be performed among pictures with lower temporal distance first. Coarser layers may then exploit inter-layer motion correlation by using already estimated MVs of finer layers for ME initialisation or predictive MV coding [8].

3. PROPOSED MOTION ESTIMATION ALGORITHM

The motion estimation algorithm presented in this paper has been implemented into the Joint Scalable Video Model (JSVM), the reference software used in the standardisation activity of the Joint Video Team (JVT) for a scalable coding extension based on H.264/AVC. The proposed algorithm does not require any normative changes to the decoding process specified in SVC and is only implemented at the encoder side. The unaltered coding in the JSVM is based on 16×16 pixel macroblocks (MB). MC is performed for MBs coded in one of the several possible *inter modes* which allow a partitioning of individual MBs into smaller MC blocks. The available partitionings are 16×16, 16×8, 8×16 and 8×8, where each partition in the 8×8 mode may be further subdivided into 8×4, 4×8 or 4×4 sub-partitions. One or two MVs at quarter-pel precision with accompanying reference picture indices are assigned to each (sub-)partition for uni- or bi-directional MC respectively.

The proposed ME algorithm composes a candidate set $\boldsymbol{S}$ of full-pel accurate MVs for forward and backward prediction of each (sub-)partition (see Figure 2). While the candidates in set $\boldsymbol{S}$ must capture spatial, temporal and inter-layer motion correlation for good coding efficiency, the size of $\boldsymbol{S}$ must also be limited in order to reduce ME complexity. The derivation of the MV candidates requires access to MVs from either the current or previously estimated motion vector fields. In order to simplify calculations and also avoid unavailable MVs (due to unidirectional or intra-coded MBs) we store both the forward and backward MVs of the 16×16 estimation of each MB independent of the actual coding mode used for the MB for further reference. The MV storage is only required at the encoder side, as no changes are made to the JSVM bitstream syntax or semantics. The individual full-pel MV candidates in set $\boldsymbol{S}$ are chosen as follows.

Zero Vector Candidates

Many scenes contain little or no camera or background motion at all, therefore a forward and a backward zero vector $(0,0)^T$ are added to the candidate set.

Spatial Vector Candidates

Up to three candidates per prediction direction are derived from spatially adjacent partitions inside the current picture. First of all we are using the MVP which is also used for differential coding of the current MV and derived using the algorithm specified in [2]. If available, we also include the MVs of the left neighbour and the top-right neighbour partitions taken from the motion vectors used in calculating the MVP. If the top-right neighbour does not exist, it is replaced by the top-left neighbour.

Temporal Vector Candidates

The temporal vector candidates for forward and backward estimation are derived differently due to the availability of previously estimated MVs. Backward MV candidates are derived from inverted forward MVs of the current picture, therefore only MVs from above or to the left of the current MB are used due to causality constraints. We chose the two previously stored 16×16 mode MVs from the MBs to the left and top-right of the current MB as temporal vector candidates. For forward MVs the situation is different, as we can use any of the backward MVs from the already estimated MV field of s_{t-2} as candidates. The selected forward candidates are the inverted MVs from the stored 16×16 field, taken from the right and bottom-left neighbours of the colocated MB.

Temporal Inter-Layer Vector Candidates

The key aspect of our proposed scheme is the usage of a temporal inter-layer candidate (ILC) whose purpose is to improve vector prediction specifically in the context of hierarchical B pictures. With

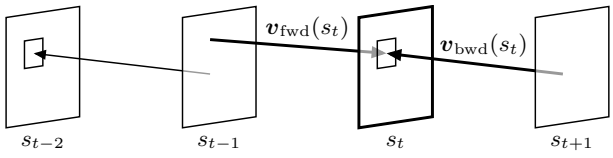

Fig. 2. Motion vectors for forward prediction ($\boldsymbol{v}_{\mathrm{fwd}}$) are referencing pictures from the past, backward MVs ($\boldsymbol{v}_{\mathrm{bwd}}$) from the future.

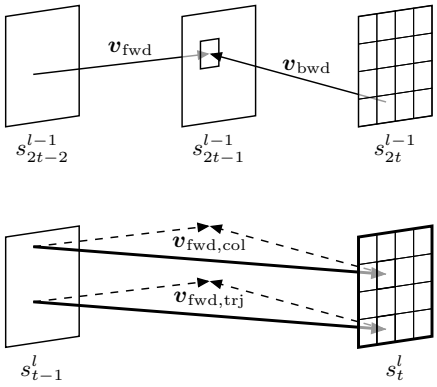

Fig. 3. Derivation of temporal inter-layer vector candidates.

each temporal layer, the temporal distance between motion compensated pictures is doubled and would therefore require an increased MV search range for ME. However, MVs from earlier layers can be combined to predict the motion in later layers. A candidate for temporal layer l is calculated from a stored 16×16 forward and backward MV pair from the previous temporal layer $l-1$:

$$\begin{aligned} \boldsymbol{v}_{\mathrm{fwd}}(s^{l}_{t}) &:= \boldsymbol{v}_{\mathrm{fwd}}(s^{l-1}_{2t-1}) - \boldsymbol{v}_{\mathrm{bwd}}(s^{l-1}_{2t-1}), \\ \boldsymbol{v}_{\mathrm{bwd}}(s^{l}_{t}) &:= \boldsymbol{v}_{\mathrm{bwd}}(s^{l-1}_{2t+1}) - \boldsymbol{v}_{\mathrm{fwd}}(s^{l-1}_{2t+1}). \end{aligned} \tag{5}$$

Two different candidate assignment schemes for the ILC have been examined: (a) assigning candidates to the respective colocated block ($\boldsymbol{v}_{\mathrm{fwd,col}}$) and (b) assigning the candidates to the block following the motion trajectory. (See Figure 3 for an example: the block in s^{l-1}_{2t} maximally overlapping the referenced area of $\boldsymbol{v}_{\mathrm{bwd}}$ is determined, and its colocated block in s^{l}_{t} is assigned $\boldsymbol{v}_{\mathrm{fwd,trj}}$ as candidate.)

It should be noted that although we implement the ILC in a specific motion estimation algorithm, it can also be applied to other motion estimation algorithms, e. g. initialising the centre of the MV search window.

As all described candidates except for the MVP are derived directly from previous motion estimation results, we need a method for adapting to changing motion in the sequence. Therefore, the current set $\boldsymbol{S}$ is extended by adding a vector $\boldsymbol{r}_{\mathrm{i}}$, randomly chosen from the set

$$\boldsymbol{R} = \left\{ \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} -4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 4 \end{pmatrix}, \begin{pmatrix} 0 \\ -4 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix}, \begin{pmatrix} -8 \\ 0 \end{pmatrix} \right\}, \tag{6}$$

to each vector in $\boldsymbol{S}$, resulting in the final candidate set

$$\boldsymbol{S}_{\mathrm{final}} = \{\boldsymbol{v}_1, \ldots \boldsymbol{v}_{\mathrm{n}}, \boldsymbol{v}_1 + \boldsymbol{r}_1, \ldots \boldsymbol{v}_{\mathrm{n}} + \boldsymbol{r}_{\mathrm{n}}\}, \quad \boldsymbol{r}_{\mathrm{i}} \in \boldsymbol{R}. \tag{7}$$

The best MV candidate is determined by minimising the cost function (2) for all unique vectors of the final set. For subsequent full-pel refinement, a pattern search around the best MV candidate is performed using the set

$$\boldsymbol{P} = \left\{ \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}. \tag{8}$$

Finally, sub-pel refinement is done by evaluating the eight surrounding half-pel positions first and further testing the eight quarter-pel positions around the best half-pel candidate.

For the coding mode decision the rate-distortion cost is compared among the two unidirectional modes and the bi-directional mode utilising the two best unidirectional MVs without any further bi-directional refinement.

4. EXPERIMENTAL RESULTS

The proposed algorithm has been integrated in the JVT reference software JSVM and tested using the sequences *Bus*, *Football*, *Foreman* and *Mobile* at CIF resolution and 30 Hz. Results are consistent among the tested sequences and are presented in particular for *Bus* and *Foreman*. Our algorithm performed best for *Bus* and exhibited only slightly less performance for *Football* and *Mobile*. Due to its complex motion, *Foreman* is the most demanding among the tested sequences.

In order to isolate the performance impact of the different motion estimation algorithms, encoding was performed without spatial or quality scalability using fixed QP settings 27, 30, 33, 36, 39, and 42, using open-loop encoder control. *Bus* and *Foreman* were encoded using 5 temporal layers and only the first picture was coded as intra picture. Full-pel motion estimation was performed using the SAD of the Y', C_b and C_r components, sub-pel estimation using the sum of absolute coefficients of the hadamard transformed difference (SATD) of the luma component.

Results are shown for our proposed predictive search without the inter-layer candidate (PS) and the predictive inter-layer search (PILS) including the ILC, and compared to full search motion estimation (FS) using a search range of ± 48 pixels and iterative bi-directional search switched off (bi-directional modes are evaluated using the two estimated unidirectional MVs). We have found that the PSNR and complexity differences between assigning the ILC to the colocated block and the block following the motion trajectory for PILS are negligible. Therefore, only the results for the latter are presented. The reference scheme PS is based on the recursive motion estimation algorithm in [4]. Comparing PILS to PS illustrates the benefit of the proposed temporal ILC. Either of the inter-temporal candidates leads to a significant performance improvement for PILS over the PS algorithm. The average number of examined full-pel vectors per MB for the 16×16 mode (Cand./MB) and the total CPU time of the encoder at QP $=$ 33 are presented in Table 1, results at other QP values and the candidate count ratios for the remaining block partitionings are similar. Timing was averaged over three runs on a Pentium 4 CPU at 3 GHz using binaries compiled using GCC 4 with enabled processor specific optimisations. Compared to FS, the

	Bus, QP $=$ 33			
	Cand./MB	%	Time [s]	%
FS	17171.6	100.00	7681	100.0
PILS	27.6	0.16	157	2.1
PS	26.4	0.15	157	2.1
	Foreman, QP $=$ 33			
	Cand./MB	%	Time [s]	%
FS	17240.0	100.00	15278	100.0
PILS	27.5	0.16	309	2.0
PS	26.3	0.15	305	2.0

Table 1. Candidate count for the 16×16 mode and total encoder time.

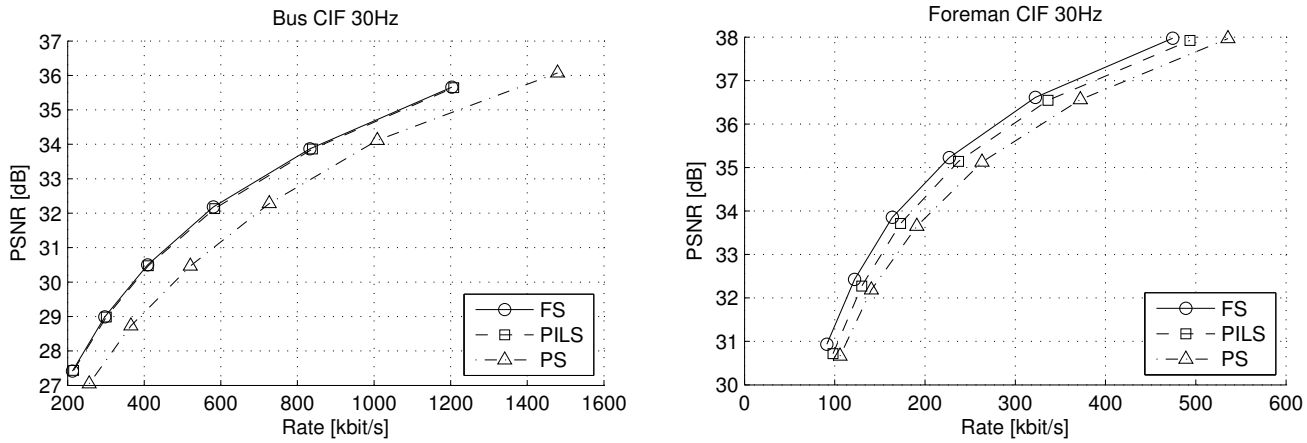

Fig. 4. PSNR comparison for *Bus* and *Foreman* CIF at 30 Hz.

proposed algorithm reduces the number of SAD calculations roughly by a factor of 625 and the total encoder time by a factor of 50. Total encoder speed up is limited due to identical time requirements for sub-pel estimation and time spent in other encoder modules. Rate-distortion curves are shown in Figure 4. Using the Bjøntegaard measurement [9] for PILS in comparison to FS we observe an average PSNR loss of -0.057 dB or a bitrate increased by 1.2 % for *Bus* and an average PSNR loss of -0.336 dB or a bitrate increased by 8.1 % on average for *Foreman*.

5. CONCLUSION

In this paper we have proposed a predictive motion estimation algorithm for scalable video coding using MV prediction specifically suited for the temporal decomposition structure of open-loop hierarchical B pictures and MCTF. It was shown that a temporal inter layer candidate can exploit the motion correlation among temporal layers and lead to a significant reduction of the computational complexity of motion estimation in scalable video coding while maintaining comparable PSNR and visual quality. While the inter layer candidate was employed in a specific motion estimation algorithm it could also be used as search initialisation vector in other motion estimation algorithms.

6. REFERENCES

[1] *ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC: Advanced Video Coding for Generic Audiovisual Services*, version 3: 2005.

[2] Julien Reichel, Heiko Schwarz, and Mathias Wien, "Joint scalable video model JSVM-6," Doc. JVT-S202, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 19th Meeting, Geneva, Switzerland, Apr. 2006.

[3] Jens-Rainer Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559–571, Sept. 1994.

[4] G. de Haan, P. W. A. C. Biezen, H. Huijgen, and O. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[5] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. National Telecommunications Conference*, New Orleans, LA, USA, Nov. 1981, pp. G5.3.1–G5.3.5.

[6] J.C. Tsai, C.H. Hsieh, S.K. Weng, and M.F. Lai, "Block-matching motion estimation using correlation search algorithm," *Signal Processing: Image Communications*, vol. 13, no. 2, pp. 119–133, Aug. 1998.

[7] I. Ismaeil, A. Docef, F. Kossentini, and R.K. Ward, "Efficient motion estimation using spatial and temporal motion vector prediction," in *Proc. IEEE Int. Conference on Image Processing ICIP '99*, Oct. 1999, vol. 1, pp. 70–74.

[8] Deepak S. Turaga, Mihaela van der Schaar, and Béatrice Pesquet-Popescu, "Complexity scalable motion compensated wavelet video encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 982–993, Aug. 2005.

[9] Gisle Bjøntegaard, "Calculation of average PSNR differences between RD curves," in *SG16/Q6 VCEG, 13. Meeting, Document VCEG-M33*, Austin, TX, USA, Apr. 2001, ITU-T VCEG.