28th Picture Coding Symposium, PCS2010, December 8-10, 2010, Nagoya, Japan

HIGHLY EFFICIENT VIDEO COMPRESSION USING QUADTREE STRUCTURES AND IMPROVED TECHNIQUES FOR MOTION REPRESENTATION AND ENTROPY CODING

Detlev Marpe, Heiko Schwarz, Sebastian Bosse, Benjamin Bross, Philipp Helle, Tobias Hinz, Heiner Kirchhoffer, Haricharan Lakshman, Tung Nguyen, Simon Oudin, Mischa Siekmann, Karsten Sühring, Martin Winken, and Thomas Wiegand

Image Processing Department, Fraunhofer HHI, Einsteinufer 37, 10587 Berlin, Germany

ABSTRACT

This paper describes a novel video coding scheme that can be considered as a generalization of the block-based hybrid video coding approach of H.264/AVC. While the individual building blocks of our approach are kept simple similarly as in H.264/AVC, the flexibility of the block partitioning for prediction and transform coding has been substantially increased. This is achieved by the use of nested and preconfigurable quadtree structures, such that the block partitioning for temporal and spatial prediction as well as the space-frequency resolution of the corresponding prediction residual can be adapted to the given video signal in a highly flexible way. In addition, techniques for an improved motion representation as well as a novel entropy coding concept are included. The presented video codec was submitted to a Call for Proposals of ITU-T VCEG and ISO/IEC MPEG and was ranked among the five best performing proposals, both in terms of subjective and objective quality.

Index Terms— Video coding; H.265; HEVC

1. INTRODUCTION

Recently, a Call for Proposals (CfP) on video compression technology [1] was issued jointly by ITU-T VCEG and ISO/IEC MPEG in order to identify video coding technology with a substantially higher compression capability than the existing H.264/AVC standard [2]. As a response to this CfP, 27 complete proposals were received and subjectively evaluated [3]. The subjective test results reported in [3] indicate that the best performing proposals achieved in a number of test cases significant improvements over the H.264/AVC High Profile (HP) conforming anchor bitstreams. As a consequence, the Joint Collaborative Team on Video Coding (JCT-VC) initiated the standardization project of High Efficiency Video Coding (HEVC) by specifying a first Test Model under Consideration (TMuC) [4]. This test model is made up of design elements from the best performing proposals including those of [5] which is the topic of this paper.

2. OVERVIEW OF THE VIDEO CODING SCHEME

Our proposed video coding scheme is based on the conventional hybrid approach of using spatial and temporal, *i.e.*, intra and motion-compensated prediction, followed by DCT-based transform coding of the residual and entropy coding of the quantized transform coefficients as well as of other coding parameters. The innovative and distinctive features of our approach are given as follows:

- **Wide-range variable block-size prediction:** The size of the prediction blocks can be adaptively chosen by using a quadtree-based partitioning. Maximum and minimum admissible block size are not fixed, but specified as side information in the bitstream.
- **Nested wide-range variable block-size residual coding:** The block size used for DCT-based coding of the residual is also derived by using quadtree-based partitioning of the corresponding prediction block.
- **Merging of prediction blocks:** In order to reduce the side information required for signaling the prediction parameters, neighboring blocks can be merged into one region, such that the prediction parameters have to be transmitted only once for a whole region.
- **Fractional-sample MOMS interpolation:** Interpolation of fractional-sample positions for motion-compensated prediction is based on a fixed-point implementation of the Maximal Order Minimum Support (MOMS) algorithm.
- **Adaptive in-loop filter:** In addition to the deblocking filter, a separable 2-D Wiener filter is applied within the motion-compensated prediction loop. This filter is adaptively applied to selected regions of the deblocking filter output in horizontal and vertical direction, respectively.
- **Entropy coding**: A novel entropy coding scheme is employed that enables a high degree of parallel processing capability and that can be configured to operate at a complexity level of variable-length coding without any loss in coding efficiency relative to the use of arithmetic codes.

The presentation in this paper will focus on the overall architectural design and the main distinctive features of our proposed video compression scheme, as outlined above. A series of accompanying papers [6] – [9] will be devoted to an in-depth study of individual coding tools. For a more detailed description of our novel video coding scheme along with the presentation of a comprehensive set of coding results the reader is referred to [10].

978-1-4244-7135-5/10/$26.00 ©2010 IEEE

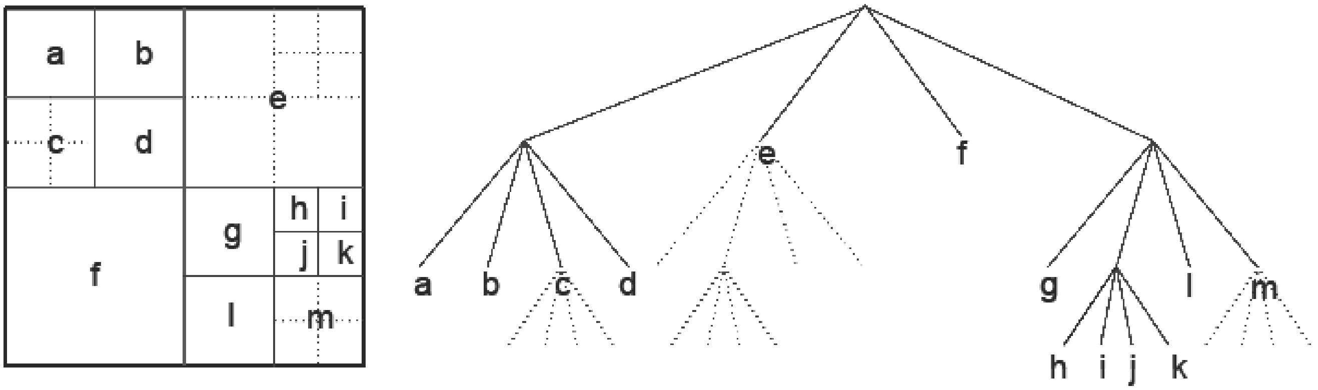

Fig. 1: Example of a nested quadtree structure (right) for dividing a given coding tree block (left; in black) into prediction blocks (solid lines) and transform blocks (dashed lines) of variable size. The order of parsing prediction blocks follows their labeling in alphabetical order.

3. PARTITIONING OF THE INPUT PICTURE FOR PREDICTION AND RESIDUAL CODING

The concept of a macroblock as the basic processing unit in standardized video coding is generalized to what we call a *coding tree block* (CTB) in our approach. A CTB is a square block of $M \times M$ luma samples together with two corresponding blocks of chroma samples, to which a *nested quadtree structure* is attached that indicates how the blocks are further subdivided for the purpose of prediction and residual coding. The edge length M of the corresponding square block of luma samples must be a power of two and is specified as side information in the bitstream. This enables to adapt the maximum prediction block size to the characteristics of the video material. Typically, high resolution video sequences benefit from larger prediction block sizes.

Fig. 1 (left) shows an example of a coding tree block (in black; luma samples only) and how it is subdivided into prediction blocks (solid lines) and transforms blocks (dashed lines). On the right-hand side of the same figure, the corresponding quadtree structure for CTB partitioning is shown. In this example, the quadtree specifying the prediction blocks (solid lines) has four levels, with the root at level 0 corresponding to the full CTB size (maximum prediction block size), and with level 3 corresponding to a block size, *i.e.*, edge length of one eighth of the CTB edge length. Generally, subblocks at level i always have a block edge length of $2^{-i} \cdot M$ with M denoting the edge length of the CTB. The maximum number of levels, and therefore the minimum possible prediction block size, is also specified as side information in the bitstream. Consequently, maximum and minimum possible prediction block size can be freely chosen, depending on the application, the video material, the resolution, etc.

The samples of each prediction block are either intra-predicted, *i.e.*, predicted by using decoded and reconstructed samples of neighboring blocks of the same picture, or they are predicted by using motion-compensated prediction (MCP). In both cases, the corresponding residual signal is further processed by DCT-based coding with variable block sizes. For that, each node of the so-called *prediction quadtree*, as shown in blue in the example of Fig. 1 (right), which corresponds to a prediction block and its related residual signal, can be further split into transform blocks of smaller size than the given prediction block size. In the example of Fig. 1, this is illustrated by the dashed blocks and the corresponding dashed *residual quadtrees*, where each residual signal at a node of the prediction quadtree with a vanishing residual quadtree (or equivalently, with a residual quadtree consisting only of a root node) is considered to be processed by a block transform with dimensions equal to those of the corresponding prediction block.

For the purpose of mode decision or transmission of data associated with each block, all CTBs of a given slice or picture are traversed in raster scan order (left-to-right, top-down), and within each CTB, the subblocks are traversed in depth-first order. Using depth-first traversal has the benefit that both the left neighboring block(s) and the top neighboring block(s) are always encoded/transmitted before the current block. Thus, the data already transmitted for these blocks can be used to facilitate the encoding of the current block such as, *e.g.,* for the purpose of motion vector prediction or context modeling in entropy coding.

4. MOTION REPRESENTATION

As in most hybrid video coding designs, in the presented scheme a translational motion model is used for MCP. Thus, each MCP block is associated with one or two sets of motion parameters, where each set of motion parameters consists of a picture reference index and a motion vector. The prediction signal related to each set of motion parameters is obtained by displacing an area of a previously decoded reference picture selected by the reference index with the displacement being specified by the motion vector. When a prediction block is associated with more than one set of motion parameters, the prediction signal is obtained as a superposition of the motion-compensated prediction signals that are generated using the individual sets of motion parameters.

Both components of a motion vector are represented with the same fractional-sample accuracy. The minimum admissible motion-vector accuracy can be set to units of 2^{-n} the distance between luma samples, with the corresponding parameter $n \geq 0$ being signaled at the slice level. For generation of the coding results, as presented in Sec. 7, the motion-vector accuracy was kept fixed at quarter-sample precision.

4.1. Motion vector prediction

In order to reduce the bit rate required for transmitting motion vectors, we employ a novel concept in which the prediction and coding of the components of a motion vector is interleaved. In a first step, the vertical motion-vector component is predicted using conventional median prediction (as in H.264/AVC) and the difference between the actual vertical component and its prediction is coded. Then, only the motion vectors of the neighborhood for which the absolute difference between their vertical component and the coded vertical component for the current motion vector is minimized are used for the prediction of the horizontal component.

4.2. Fractional-sample interpolation

For motion-compensated prediction, reference blocks belonging to motion vectors with fractional-sample accuracy need to be upsampled accordingly. Instead of using a 6-tap interpolation filter as in H.264/AVC [2], our approach of fractional sample interpolation relies on the so-called *Maximal Order Minimum Support* (MOMS) algorithm [11]. According to this concept, the upsampling process consists of IIR and FIR filtering stages, called *prefiltering* and *interpolation*, respectively. Both the 2-D prefiltering and interpolation kernel are separable and can be implemented in fixed-point arithmetic as a sequence of 1-D horizontal and vertical filtering steps along the rows and columns of the reconstructed picture block, respectively. The number of operations required to perform the prefiltering and interpolation stage depends on the order of the chosen MOMS basis function [6][11]. In our experiments, we considered both 3rd and 5th order MOMS basis functions. For the choice of the 3rd order MOMS, for instance, prefiltering can be realized by a first-order causal and anti-causal filter, while the interpolation stage requires a 4-tap FIR filter. More details on our reference-sample interpolation method can be found in [6].

4.3. Block merging process

For each MCP block, either individual motion parameters have to be transmitted or a so-called *block merging process* is invoked that allows to share the motion parameters with those of a whole neighboring region consisting of a connected set of MCP blocks with a single motion parameter description. The block merging process considers the two directly neighboring blocks of the top-left sample position of the current block as possible merging targets. When this set of merging targets is not empty, *i.e.*, when it contains at least one MCP block, then it is signaled whether the current block is to be merged with one block out of this set and if so, with which of those targets the merging is performed.

5. INTRA PREDICTION, SPATIAL TRANSFORMS, QUANTIZATION, AND IN-LOOP FILTERING

Intra prediction, spatial transforms, quantization, and deblocking in our video coding algorithm are relatively straightforward extensions of the related concepts in H.264/AVC [2]. In the following, we briefly describe how those extensions were obtained and which additional aspects were incorporated in our design.

Similar to intra prediction for 4×4 or 8×8 blocks in H.264/AVC, eight directional intra-prediction modes and one additional DC mode are available. However, in our approach, those intra modes are applied to prediction blocks of all admissible sizes. In addition, an adaptive smoothing operation using the 3rd order binomial filter is applied to the reference samples before computing the prediction signal.

The size of transform blocks for residual coding are specified by the residual quadtrees attached to the node of each prediction quadtree, as described in Sec. 3. The used transform kernel for each block size in the admissible range of 4×4 to 64×64 (for luma) is a separable integer approximation of the 2-D DCT-II (Discrete Cosine Transform type II) of the corresponding block size.

For the quantization of transform coefficients, we have employed a method of rate-distortion optimized quantization (RDOQ) which is similar to that of the JM implementation [12]. As in H.264/AVC, a uniform scalar quantizer with 52 logarithmically increasing step sizes forms the basis of it.

Our proposed video coding scheme also uses the in-loop deblocking filter as specified in H.264/AVC [2]. The filtering operations are extended to larger block sizes, but the derivation of filter strength as well as the transmission of filter parameters is performed exactly as in H.264/AVC.

In addition to the deblocking filter, a separable Wiener filter is applied to selected regions of the output of the deblocking filter. Regions to which the Wiener filter is applied are represented by separate quadtree structures. For more details on that particular tool, the reader is referred to [7].

6. ENTROPY CODING

For entropy coding in our video coding scheme, binarization and context modeling of CABAC in H.264/AVC [2] have been reused along with some modifications and additions for transform blocks of size greater than 8×8. These latter features will be described in more detail in [9].

Actual coding of the binary symbols, however, is based on the novel *probability interval partitioning entropy* (PIPE) coding concept that has been introduced in order to support parallelized implementations of entropy encoding and decoding as well as for decreasing the computational complexity of the entropy decoding process [8]. By partitioning the unit interval into a small number of probability intervals, each binary symbol is assigned to one of the probability intervals depending on its probability estimate, and all binary symbols for a probability interval are coded with a fixed probability by employing a simple variable-to-variable code.

Table 1: Averaged bit-rate savings (BD rate) and BD PSNR values (for luma) obtained for our submission [5] relative to the H.264/AVC HP anchors at the CS 1 test conditions.

Class	Sequence	BD Rate [%]	BD PSNR [dB]
A (2560x1600)	Traffic	-27.92	1.15
	People	-17.92	1.01
B1 (1920x1080)	Kimono	-38.30	1.56
	ParkScene	-24.28	1.00
B2 (1920x1080)	Cactus	-29.22	0.97
	BasketbalIDrive	-35.97	1.29
	BQTerrace	-41.92	0.81
C (832x480)	BasketballDrill	-31.88	1.55
	BQMall	-29.71	1.63
	PartyScene	-28.09	1.24
	RaceHorses	-29.67	1.39
D (416x240)	BasketballPass	-21.97	1.22
	BQSquare	-43.96	2.12
	BlowingBubbles	-23.60	1.14
	RaceHorses	-19.64	1.11
	Average:	**-29.60**	**1.28**

7. CODING CONDITIONS AND RESULTS

In the CfP [1], two sets of coding conditions with different constraints were defined. Due to limited amount of space in this paper, we will restrict the following exposition of coding results to the so-called random access case, also denoted as constraint set 1 (CS 1) in [1]. The complete set of results will be presented in [10].

For CS 1, the structural delay was limited to 8 pictures with random access intervals not exceeding 1.1 sec [1]. According to those constraints, we used for the generation of our submitted bitstreams a hierarchical B picture coding structure with 4 layers and a corresponding intra-frame period. In addition, we configured our encoder to operate with a fixed CTB size of 64×64 (for luma) and a maximum quadtree depth of 4. The internal bit depth for generating the prediction signal for both intra and motion-compensated prediction as well as for generating the reconstructed residual signal was increased to 14 bits, whereas the (maximum of four) reference pictures were stored with 8 bits only.

For motion estimation, rate-constrained coding mode decision, and layer-dependent quantization-parameter (QP) scaling (*i.e.*, QP cascading), our encoder was configured in the same way as the JM encoder for generating the H.264/-AVC HP conforming anchor bitstreams, as specified in [1]. For selection of the nested quadtree partitioning for prediction and residual coding, we applied a top-to-bottom and depth-first decision strategy together with an abort criterion [5][10]. Although known to be suboptimal relative to a full bottom-up tree pruning process, this fast encoder strategy proved to deliver a significant speedup at the expense of only a small loss in rate-distortion (RD) performance [10].

Table 1 shows the average RD gains of our CS 1 related bitstreams, as submitted to the CfP, with the average for each of the test sequences obtained as mean of the Bjøntegaard-delta (BD) PSNR and bit-rate values for the upper four and lower four out of a total of five rate points. Overall, significant objective gains in terms of average 29.6% BD rate savings relative to the H.264/AVC HP anchors have been achieved. In addition, our proposal [5] was among the five best rated proposals in the subjective tests [3].

As a rough indication of computational complexity, the encoding and decoding time has been measured for both our implementation and the JM software using the same hardware platform. By averaging over all rate points of all CS 1 related test sequences, we obtained a factor of about 4 in encoding time and roughly a factor of 3 in decoding time relative to JM version 16.2 and 17.0, respectively [5][10].

8. CONCLUSION

We have presented a candidate design for the next-generation video coding standard. Its most prominent features are given by a flexible block partitioning for prediction and residual coding, a block merging process for efficient region-based motion modeling, a highly efficient fractional-sample interpolation method, and a conceptually new approach to entropy coding.

9. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/16, “Joint Call for Proposals on Video Compression Technology,” WG11 Doc. N11113 and Q6/16 Doc. VCEG-AM91, Kyoto, Jan. 2010.

[2] ITU-T and ISO/IEC, Advanced Video Coding for Generic Audiovisual Services. ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 8 (including SVC): July 2007.

[3] V. Baroncini, J.-R. Ohm, G. J. Sullivan, “Report of Subjective Test Results of Responses to the Joint Call for Proposals on Video Coding Technology for High Efficiency Video Coding (HEVC),” Doc. JCTVC-A204, Dresden, Germ., April 2010.

[4] JCT-VC, “Test Model under Consideration”, Doc. JCTVC-A205, Dresden, Germany, April 2010.

[5] M. Winken et al., “Description of Video Coding Technology Proposal by Fraunhofer HHI,” Doc. JCTVC-A116, April 2010.

[6] H. Lakshman, B. Bross, H. Schwarz, T. Wiegand, “Fractional-Sample Motion Compensation Using Generalized Interpolation”, *Proc. PCS 2010*, to be published.

[7] M. Siekmann, S. Bosse, H. Schwarz, T. Wiegand, “Separable Wiener Filter Based Adaptive In-Loop Filter for Video Coding,” *Proc. PCS 2010*, to be published.

[8] D. Marpe, H. Schwarz, T. Wiegand, “Entropy Coding in Video Compression Using Probability Interval Partitioning,” *Proc. PCS 2010*, to be published.

[9] T. Nguyen, H. Schwarz, H. Kirchhoffer, D. Marpe, T. Wiegand, “Improved Context Modeling for Coding Quantized Transform Coefficients in Video Compression,” *Proc. PCS 2010*, to be published.

[10] D. Marpe, H. Schwarz, S. Bosse, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, T. Nguyen, S. Oudin, M. Siekmann, K. Sühring, M. Winken, T. Wiegand, “Video Compression Using Quadtrees, Leaf Merging, and Novel Techniques for Motion Representation and Entropy Coding,” *IEEE Trans. Circ. and Syst. for Video Techn.,* to be published.

[11] T. Blu, P. Thévenaz, M. Unser, “MOMS: Maximal-Order Interpolation of Minimal Support”, *IEEE Trans. Image Proc*., Vol. 10, No. 7, July 2001.

[12] ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/16, Joint Model (JM) H.264/AVC Reference Software, http://iphome.hhi.de/suehring/tml/.