Variable Block-Size Transforms for H.264/AVC

Mathias Wien, *Member, IEEE*

***Abstract*—A concept for variable block-size transform coding is presented. It is called adaptive block-size transforms (ABT) and was proposed for coding of high resolution and interlaced video in the emerging video coding standard H.264/AVC. The basic idea of inter ABT is to align the block size used for transform coding of the prediction error to the block size used for motion compensation. Intra ABT employs variable block-size prediction and transforms for encoding. With ABT, the maximum feasible signal length is exploited for transform coding. Simulation results reveal a performance increase up to 12% overall rate savings and 0.9 dB in peak signal-to-noise ratio.**

***Index Terms*—Block-based hybrid video coding, H264/AVC, integer transforms, variable block-size transforms.**

I. Introduction

THE Joint Video Team (JVT) of ITU-T SG16 WP3 Q.6 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) is currently developing a joint video coding standard. It is referenced as recommendation H.264 for the ITU-T and MPEG-4 Part 10 Advanced Video Coding (AVC) for ISO [1]. The work started off as a design process for a new ITU-T recommendation succeeding H.263 [2] and was merged in late 2001 to a joint project with MPEG. The standardization process is reflected in the Joint Model (JM) software that is used for evaluation of the tools presented to JVT. In this paper, the status of the tools is described as of the Joint Final Committee Draft (JFCD) of the standard [3].

The coder is based on the hybrid coding scheme including sophisticated prediction modes and efficient adaptive entropy coding techniques to achieve improved compression performance. Blocks of size 16×16 down to 4×4 pixels can be employed for motion prediction. The prediction modes are organized in a tree-structured manner, allowing even for flexible combination of different motion compensation block sizes inside a 16×16 pixel macroblock. Main features include also multiple reference frames for inter prediction and a generalized B-frame concept.

For improved intra-compression performance, various prediction methods are specified. The adjacent pixels of the already decoded neighboring blocks are used for intra prediction. A fixed 4×4 block transform is applied to the prediction error signal of inter and intra-prediction modes. For intra macroblocks, an additional transform for 4×4 DC coefficients can be applied.

Manuscript received February 26, 2002; revised May 10, 2003. This work was supported by R. Bosch GmbH, Hildesheim.

The author is with the Institut und Lehrstuhl für Nachrichtentechnik, RWTH Aachen University, Aachen, Germany (e-mail: wien@ient.rwth-aachen.de).

Digital Object Identifier 10.1109/TCSVT.2003.815380

Two entropy coding methods are specified for application in different profiles of H.264/AVC. The Base Line profile and the Extended profile employ context-adaptive variable length codes (CAVLC) for the transform coefficients and universal variable length codes (UVLC) for all other symbols. For high coding efficiency context adaptive binary arithmetic coding (CABAC) is employed in the Main profile. The standard is specified to allow for an implementation using only 16 bit multiplications and 16 bit memory access.

In this paper, the concept of variable block-size transform coding is presented. The scheme is called adaptive block-size transforms (ABT), indicating the adaptation of the transform block size to the block size used for motion compensation. For intra coding, the size of the block transform is adapted to the properties of the intra-prediction signal. Thus, for both inter and intra coding, the maximum feasible signal length can be exploited by the transform. With ABT, the block transform as commonly used in the former image and video coding standards is generalized from a fixed-size transform to a signal-adaptive tool for increased overall coding performance.

The large transforms can provide a better energy compaction and a better preservation of detail in a quantized signal than a small transform does. On the other hand, larger transforms introduce more ringing artifacts caused by quantization than small transforms do. By choosing the transform size according to the signal properties, the tradeoff between energy compaction and preserved detail on the one hand and ringing artifacts on the other can be optimized.

Various proposals for variable block-size coding of images and video are known from the literature. Many of them are based on quadtree structures either for prediction or for transform coding. Quadtree-based variable block-size motion compensation was proposed, e.g., in [4], [5]. A quadtree-based variable block-size discrete cosine transform (DCT) was introduced in [6]. In this paper, the choice of the appropriate block size was based on a mean-based decision rule. However, this approach did not lead to rate/distortion (RD) optimal block tilings [7]. An image compression scheme with variable block-size segmentation was presented in [8], [9]. Here, the amount of detail in the blocks was used for determination of the employed block size. Classified vector quantization (VQ) was employed for high detail 4×4 blocks. A combination of transform coding and VQ was used for nonhigh detail 8×8, 16×16, and 32×32 blocks. Quadtree structures were also used with lapped transforms, e.g., in [10]. An RD optimized quadtree-based variable block-size coder using wavelet packets was proposed in [11]. In [12], a variable block-size transform image coder was presented that is not limited to square blocks. Here, block sizes of 8, 16, or 32 pixels per edge were used. The appropriate block size was selected according to a local activity index of the image scene.

The concept of ABT presented in this paper merges the benefits from variable block-size prediction and variable block-size transform coding.

The organization of the paper is as follows. In Section II, the transformation and quantization used for ABT coding are presented. Section III introduces the application of ABT for inter and intra coding in H.264/AVC. Extensions and modifications of the techniques used in H.264/AVC that are necessary for ABT coding are highlighted. In Section IV, the performance of ABT is analyzed. Finally, conclusions are drawn in Section V.

II. Variable Block Size Transforms

In general, the separable 2-D transform of a 2-D signal can be written as

$$\boldsymbol{C} = \boldsymbol{T}_v \cdot \boldsymbol{B} \cdot \boldsymbol{T}_h^T \tag{1}$$

where $\boldsymbol{B}$ denotes a matrix representing signal block of N pixels and M lines. $\boldsymbol{C}$ is the transform coefficient matrix, whereas $\boldsymbol{T}_v$ and $\boldsymbol{T}_h$ are the $M \times M$ and $N \times N$ transform matrices in vertical and horizontal direction, respectively. The transform matrices comprise a set of orthogonal base functions. $\boldsymbol{T}^T$ is the transposed matrix $\boldsymbol{T}$.

The ABT proposal for H.264/AVC employs transforms for blocks of size 4×4, 4×8, 8×4, and 8×8 pixels. Hence, employing separable vertical and horizontal transforms, a 4×4 and an 8×8 transform need to be specified. Here, the 4×4 transform specified in H.264/AVC is reused. It is a low-dynamic approximation of the 4×4 DCT [13]. In matrix notation, it can be written as

$$\boldsymbol{T}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \quad \boldsymbol{T}_4^I = \begin{pmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{pmatrix}. \tag{2}$$

Both the transform $\boldsymbol{T}_4$ and the inverse transform $\boldsymbol{T}_4^I$ have different norms for the even and the odd base functions. These have to be regarded in the encoding and decoding process. In the standard, the inverse transform $\boldsymbol{T}_4^I$ is specified as a fast butterfly operation using only additions and bit shift operations. For detail on the 4×4 transform and quantization process (see [14]).

The 8×8 transform matrix proposed for application with ABT is a single-norm approximation of the 8×8 DCT matrix as shown in (3), at the bottom of the page. It was originally proposed in [15]. The transform and the inverse transform $\boldsymbol{T}_8^I = \boldsymbol{T}_8^T$ can be implemented in a butterfly structure, using 36 additions, eight bit-shift operations, and ten multiplications. The algorithm is given in Appendix.

A. Quantization and Normalization

Both, the 4×4 as well as the 8×8 transform are not normalized and therefore require normalization. This operation is integrated into the quantization and de-quantization procedure. Since the decoding process is the relevant operation to the standard, the de-quantization is described here.

In the H.264/AVC specification, only the de-quantization values for the six finest QP need to be stored. The remaining values can be generated by scaling the finest de-quantization values with the appropriate power of 2.

Let $c(m,n)$ denote a decoded coefficient at position (m,n) in an $N \times M$ block. Note that m and n indicate the row (line) and column (pixel) respectively. We use mathematical notation in the equations. The size of blocks in a frame is indicated using pixels $\times$ lines. The de-quantization process can be written as

$$w(m,n) = \left[c(m,n) \cdot R_{M,N}^{i_{\mathrm{QP}}}(m,n) \right] \ll \left(\left\lfloor \frac{\mathrm{QP}}{6} \right\rfloor - 2 \right) \tag{4}$$

where $n = 0, \ldots, N-1$, $m = 0, \ldots, M-1$, $i_{\mathrm{QP}} = (\mathrm{QP} \bmod 6)$, and $\lfloor \cdot \rfloor$ denotes rounding with truncation toward zero ("mod" denotes the modulo operation). $R_{M,N}^{i_{\mathrm{QP}}}(m,n)$ is the scaling factor corresponding to the position of the coefficient in the reconstructed block.

The even and odd base functions of $\boldsymbol{T}_4^i$ have different norm. Due to these, the norms of coefficients in 4×4 blocks depend on n, m both being even, both odd, or mixed. For all block sizes, the norms are accounted for in $R_{M,N}^{i_{\mathrm{QP}}}(m,n)$. $\boldsymbol{T}_8^I$ has a single norm for all base functions. For 4×8 or 8×4 blocks, the choice of $R_{M,N}^{i_{\mathrm{QP}}}(m,n)$ depends only on the parity of n or m, respectively. For 8×8 blocks, the value of $R_{M,N}^{i_{\mathrm{QP}}}(m,n)$ only depends on QP. The values for $R_{M,N}^{i_{\mathrm{QP}}}(m,n)$ for all block sizes are given in [3].

The low-dynamic 4×4 transform and quantization allow for an implementation using a dynamic range of 16 bit maximum for all operations. The additional ABT transforms can be implemented using only 16-bit multiplications and 16-bit memory access [16].

III. Inter Coding and Intra Coding

A. Inter Coding

In H.264/AVC, a tree-structured motion-prediction scheme is employed. Each 16×16 macroblock can be motion-compensated using one of four macroblock modes. These modes differ

$$\boldsymbol{T}_8 = \begin{pmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{pmatrix}. \tag{3}$$

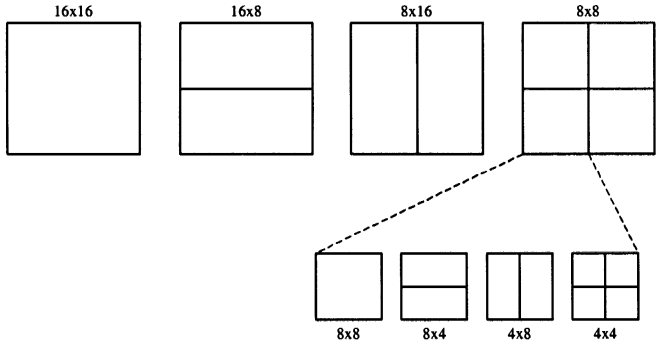


Fig. 1. Block modes available for motion prediction in H.264/AVC. For each subblock, a motion vector is encoded. In the 8×8 mode, each 8×8 block can be individually divided into subblocks of sizes 8×8 down to 4×4.

in the number and size of subblocks with common motion vectors. In 8×8 mode, the subblocks can be further divided into subblocks of sizes 8×8, 8×4, 4×8, or 4×4.

Applying these flexible motion-prediction modes, the resolution of the underlying motion model can be adapted to the content of the encoded video sequence. The block tiling of the motion-prediction modes is depicted in Fig. 1.

The motion vectors are represented in quarter-pel resolution. Multiple reference frames can be exploited for prediction. A generalized concept of B-frames is introduced in H.264/AVC. Besides direct, forward, and backward-prediction modes, bipredictive coding allows employing almost any two frames from the frame store for prediction.

1) ABT Inter Coding: The basic idea of ABT for inter coding is to connect the block size used for transform coding of the prediction error to the block size used for motion compensation. The maximum feasible signal length is exploited for transform coding. As the subblock size is already indicated for motion compensation, no additional side information needs to be encoded for inter ABT.

The maximum transform block size used for ABT is restricted to 8×8 pixels. Larger block sizes are omitted due to increase complexity and dynamic range restrictions. Larger transform blocks also tend to spread more ringing artifacts into the reconstructed signal.

2) Motion Estimation: In the JM, the sum of absolute transformed differences (SATD) is applied for subpel motion estimation. The SATD reflects the application of a transform to the prediction error. It accounts for the amount of prediction error, as well as the cost for transformed representation. The cost can be seen as an estimate for the rate spent for encoding. Hence, the SATD operates as an approximative RD cost criterion for the prediction error that can be used for motion vector search.

For SATD calculation, the 4×4 Hadamard transform $\boldsymbol{H}_4$ is applied to each 4×4 prediction error block. The resulting transformed signal emulates the frequency characteristics of the true transformed block in the subsequent transformation/quantization stage at very low computational cost.

Let $\boldsymbol{D}_{\mathrm{mc}}$ be a motion-compensated prediction error block of size 4×4 pixel. Then

$$\boldsymbol{D}_H = \boldsymbol{H}_4 \cdot \boldsymbol{D}_{\mathrm{mc}} \cdot \boldsymbol{H}_4^T \tag{5}$$

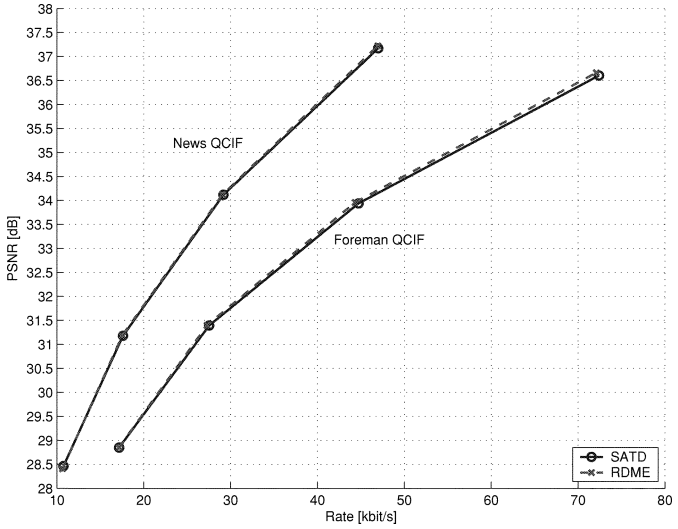


Fig. 2. Comparison of the coding performance for ABT when using the SATD ($\circ$) or the RD cost (RDME, $\times$) for motion estimation. Results are presented for the test sequences FOREMAN and NEWS. The sequences were encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels.

denotes the 2-D transform of the difference block. The SATD of the 4×4 block is then given as

$$\mathrm{SATD} = \sum_{i=0}^{3} \sum_{j=0}^{3} |d_H(i,j)| \tag{6}$$

where $d_H(i,j)$ denotes the coefficient of $\boldsymbol{D}_H$ on position (i,j). For motion estimation, the SATD is computed and accumulated for all 4×4 blocks of the current subblock. The bit rate needed for encoding of the motion vector is weighted by the quantizer dependent weighting factor λ_{ME} (QP) and added to the SATD to constitute the subblock cost criterion. For details on the weighting factor λ_{ME} (QP) see [17].

In the case of ABT, the calculation of the SATD is modified to account for the new transform sizes. Since the Hadamard transform can be defined for all signal lengths that are multiples of 4 [18], the SATD calculation can be extended straightly to the ABT block sizes.

As stated before, the application of the Hadamard transform emulates the transform step in the transformation/quantization stage. The suitability of the Hadamard transform for SATD calculation of blocks larger than 4×4 needs to be verified. The ABT SATD is compared to the performance of a more elaborate RD criterion where the encoded bit rate for the motion vectors and the coefficients is weighted against the sum of the squared reconstructed prediction error. The squared λ_{ME} (QP) was found to give the optimum performance for the RD based motion estimation. Experiments show that the performance of the ABT SATD is very close to the performance of the RD criterion. An example is shown in Fig. 2.

The performance improvement for the single ABT inter coding modes is demonstrated for the test sequence COASTGUARD in Fig. 3. ABT is employed for inter coding only. The RD plots compare the performance for the single prediction block modes when using the 4×4 transform or the corresponding $N \times M$ transform employed with ABT. Since

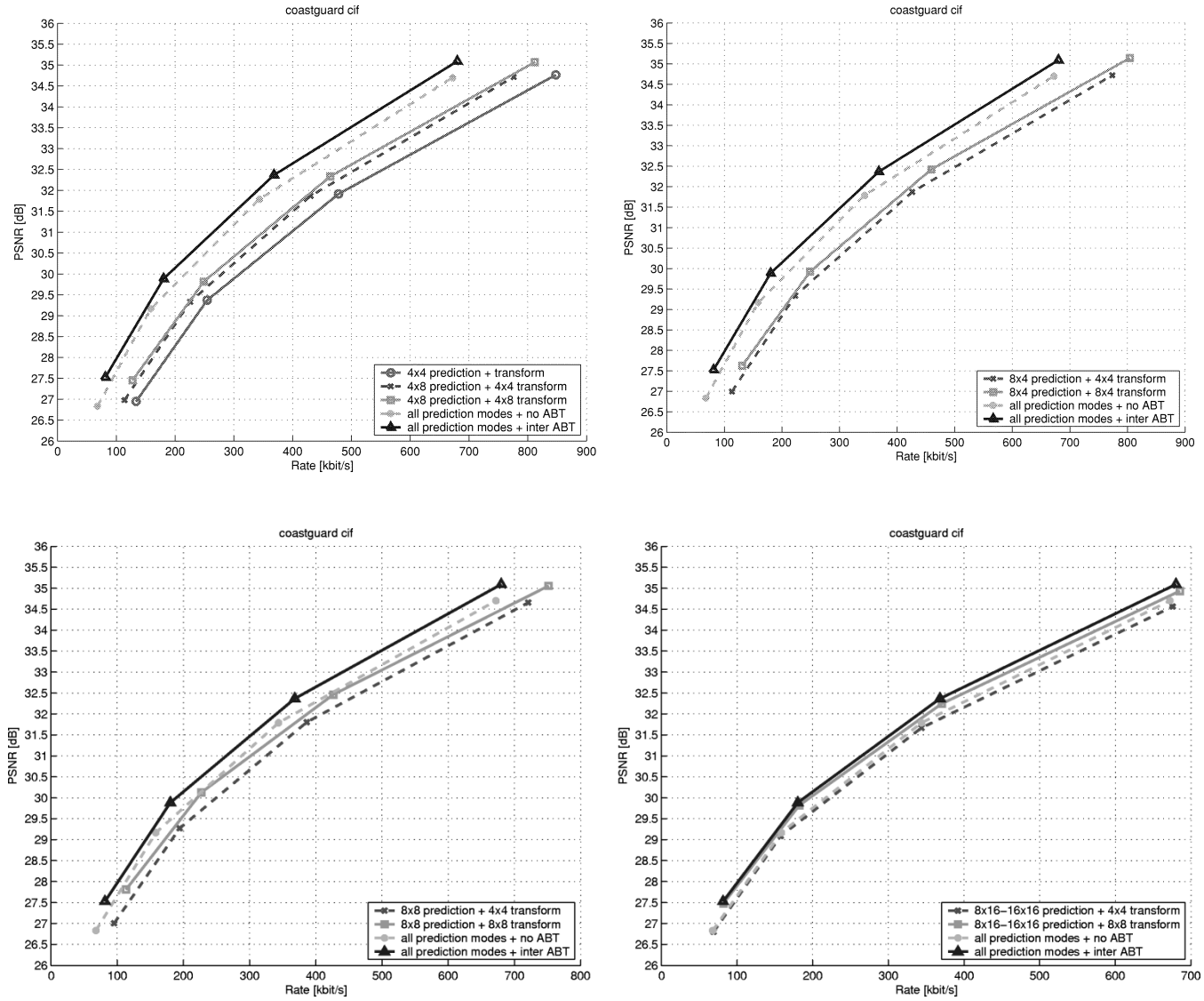


Fig. 3. Comparison of the RD performance with and without ABT for single prediction modes. Results are presented for the test sequence COASTGUARD. The sequence was encoded with constant QP in a IPPP structure at 15 Hz using two reference frames and a search range of 32 pixels. ABT was used only for inter-coded blocks. Intra blocks were encoded without ABT.

the 8×8 transform is used for macroblocks in 8×16, 16×8, and 16×16 mode, these modes were combined in one simulation. The RD performance when using all prediction modes is compared to the performance of the single modes. As can be seen from Fig. 3, the performance gain for ABT increases with the block size used for prediction and transformation. At low rates, the performance when using only the modes larger than 8×8 is comparable to the performance when all modes are used. The benefit of the modes with smaller block sizes becomes more pronounced at higher rates where these expensive modes are chosen more frequently.

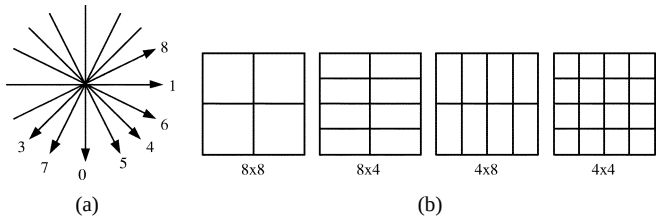


Fig. 4. (a) Intra-prediction directions for non-ABT and ABT coding. (b) Tiling of an intra macroblock for ABT intra coding including the 4×4 tiling for non-ABT coding. For each subblock a separate prediction direction can be employed.

B. Intra Coding

Two basic intra-prediction modes are used in H.264/AVC, comprising 4×4 and 16×16 block-wise prediction from previously decoded blocks. In intra 4×4 prediction mode, each 4×4 block is predicted by the adjacent pixels of the decoded neighboring 4×4 blocks. DC prediction and eight directional prediction modes can be applied to a 4×4 block. The directional prediction modes are shown in Fig. 4. Different prediction modes can be selected for each of the 16 4×4 blocks in a macroblock. The mode information has to be transmitted. To this end, a probability map is generated from the prediction modes of

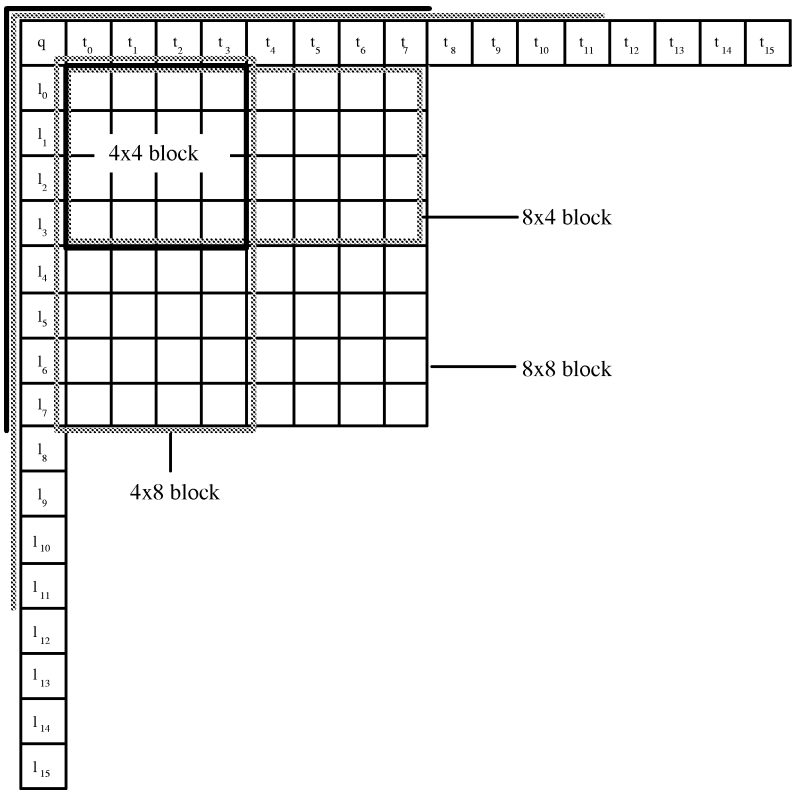


Fig. 5. Pixels t_i, l_i, and q of the top and left neighboring subblocks that may be used for intra prediction of 4×4, 4×8, 8×4, and 8×8 blocks. The maximum number of prediction pixels for an $N \times M$ block is $N + M$ pixels in each direction.

the encoded neighboring blocks. The selected prediction mode is encoded according to its probability in this map.

Smooth macroblocks containing little detail can be predicted more efficiently on a full macroblock basis. This kind of prediction is provided by the intra 16×16 Mode. Here, the current macroblock is predicted by the edge pixels of the adjacent left and top macroblocks. There are four different prediction modes, including DC, horizontal, vertical, and plane prediction.

Blocks in intra 16×16 Mode employ an additional Hadamard transform of the 4×4 DC coefficients after the 4×4 transform of the 16×16 intra prediction error. The two-stage transform decorrelates the signal in the lowest frequencies for the 16×16 block.

1) Intra Coding With Variable Block Sizes: The two-stage transform in the intra 16×16 Mode can be interpreted as a 16×16 transform for the lowest frequencies. Therefore, the choice of two intra modes specified in H.264/AVC can already be seen as a choice between two transform modes. For ABT intra coding, the prediction modes specified for the 4×4 intra-prediction mode are extended to the block sizes used for ABT inter transform coding, see Fig. 4. Hence, the gap between 4×4 intra coding and 16×16 intra coding is closed. With ABT, the encoder can trade energy compaction by larger transforms against the size of the region a single prediction direction is applied to.

An additional codeword has to be transmitted that indicates the block size used for intra prediction. The prediction block size corresponds to the block size used for transform coding. The prediction modes are encoded with probability maps as described for the intra 4×4 Mode. Since the number of subblocks is reduced, the overall overhead information for prediction block size and prediction modes for intra ABT can be smaller than with intra 4×4 Mode.

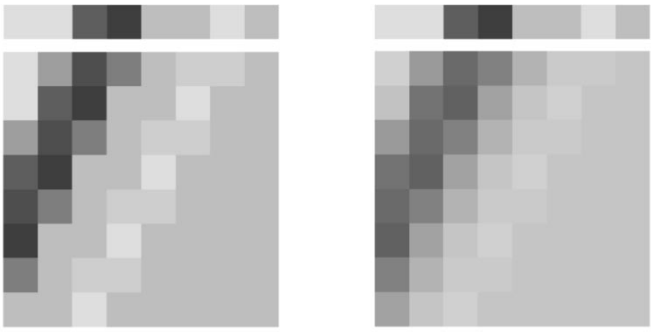
The number of neighboring pixels that are employed for intra prediction varies with the prediction mode. For example, for the horizontal and the vertical prediction mode, exclusively pixels from the left or the top are used, respectively. The reference pixels that may be used for intra-block prediction are shown in Fig. 5. Only pixels of previously decoded subblocks may be used for prediction. If prediction pixel positions required for a prediction mode are not available (e.g., because the corresponding neighboring block has not been not processed yet), the value of the last available pixel position is padded to the unavailable positions.

The maximum number of pixels for directional prediction is used for the *diagonal down/left* prediction (mode 3, see Fig. 4. Here, $N + M$ pixels are used in both the horizontal and vertical direction for one $N \times M$ block.

The prediction from skew directions might introduce visible artifacts into the prediction signal that are unfavorable for an efficient compression. This effect becomes prominent especially for block sizes larger than 4×4 since more pixels are predicted from the same source. Hence, the new ABT intra modes are afflicted. To reduce the artifacts, the pixels used for prediction are filtered with a low-pass filter.

Conceptually, a filter

$$h[n] = \frac{[1 \quad 2 \quad 1]}{4} \tag{7}$$



(a) (b)

Fig. 6. Prediction mode 7 (*vertical/left*) applied to an 8×8 block. The last row of the top neighboring 8×8 block is used for prediction. (a) Without pre-filtering. (b) Prediction from the same pixels with pre-filtering.

is applied to the vector of reference pixels

$$p[n] = [\, l_{\max} \quad \ldots \quad l_0 \quad q \quad t_0 \quad \ldots \quad t_{\max} \,]^T \qquad (8)$$

used for prediction. The reference pixels $p[n]$ are indicated in Fig. 5. The application of the filter sufficiently reduces prediction artifacts for all block sizes. For block sizes larger than 4×4, pre-filtering is applied for horizontal and vertical prediction directions as well. In Fig. 6, the impact of the pre-filtering procedure is demonstrated for *vertical/left* prediction of an 8×8 block.

2) Mode Decision for ABT Intra Coding: The optimum intra ABT mode for a macroblock is selected in a two-stage process. For all possible block sizes, the best prediction mode for each subblocks is determined. Then, the cost for encoding the macroblock with the selected subblock prediction modes is evaluated for all block sizes. The corresponding mode with the optimum overall cost is chosen for encoding the macroblock.

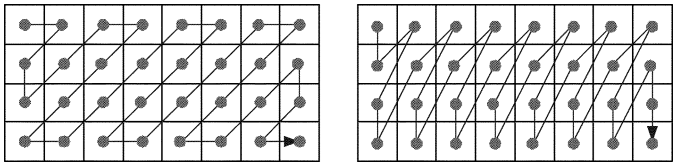
Both decisions can be made based either on the SATD or on a RD cost criterion. The RD criterion weights the rate for the transform coefficients and the corresponding prediction mode against the sum of the squared reconstructed prediction error. The weighting factor used for RD optimized macroblock mode selection in H.264/AVC can be employed here [17].

Examples for the intra mode decision on natural and synthetic test images are presented in Figs. 9 and 11.

C. Deblocking Filter

The application of a 4×4 transform tends to promote blocking artifacts in the reconstructed video, especially at lower bit rates. Therefore, a normative in-loop deblocking filter is employed in H.264/AVC. The reconstructed frames are filtered before application for motion compensation in the prediction loop. The functionality of this filter is essential for the subjective performance of the coder, especially at lower bit rates.

The strength of the filter is determined by the chosen macroblock mode, the existence of nonzero transform coefficients, the applied quantizer, and by the motion vector activity. In general, stronger filtering is applied to intra blocks than inter blocks. A maximum of two pixels on each side of the boundary of two adjacent blocks is affected by applying the deblocking filter on 4×4 blocks. Intra blocks coded with the intra 16×16 mode are filtered only at the macroblock boundaries. Here, a maximum of three boundary pixels may be filtered.



(a) (b)

Fig. 7. (a) Zig-zag scan for frame based coding and (b) alternate scan for field based coding of an 8×4 block with ABT.

The mode of operation of the deblocking filter needs to be adjusted for application to ABT-coded blocks. The strength of the filter is increased relative to the strength for 4×4 blocks according to the sizes of the adjacent blocks. The strength is increased most for two neighboring 8×8 blocks. Deblocking is only performed at transform block boundaries. Hence, the number of filter operations is reduced by ABT coding.

For example, if the 4×4 transform is employed, an 8×8 block has eight 4×4 block boundaries to be filtered. These are four inner boundaries, two on the left and two at the top. The boundaries to the right and at the bottom are encountered for the next blocks. Accordingly, an ABT 8×8 block has only two boundaries to be filtered. Similarly, for 4×8 and 8×4 blocks, the number of filtered boundaries is reduced from four to two.

D. Entropy Coding

H.264/AVC features two entropy coding methods. Variable length coding is supported by all standard compliant decoders. It employs a CAVLC for transform coefficients and a universal VLC for all other symbols. For high coding efficiency, CABAC is employed. Here, nonbinary symbols are binarized, e.g., using a unary code tree. For the separate bins, contexts are generated that are used to adapt the arithmetic coder to the statistics of the encoded symbols. For details, see [19].

Run-length coding is employed for the transform coefficients. Two types of scans can be employed to represent the transform coefficients as (level, run) symbols. The well-known zig-zag scan is employed for frame based encoding. In the case of field-based encoding of interlaced material, an alternate scan is used that accounts for the modified signal statistics in the fields of interlaced video. The alternate scan was originally proposed for ABT in [20]. An example for the zig-zag scan and the alternate scan for an 8×4 block is shown in Fig. 7. A VLC design for ABT transform coefficients was proposed in [21].

E. CABAC Transform Coefficient Encoding

The binarization and context generation for the transform coefficients for 4×8, 8×4,and 8×8 blocks is identical to the method specified for 4×4 blocks. Separate contexts are used for the symbols of each block size.

- For each encoded block, a flag is signaled to indicate whether the block contains coefficients or not.
- A significance map is generated, indicating the position of nonzero coefficients in the scan. For each significant coefficient a "last" flag is encoded. This flag indicates whether the last significant coefficient in the scan is reached.

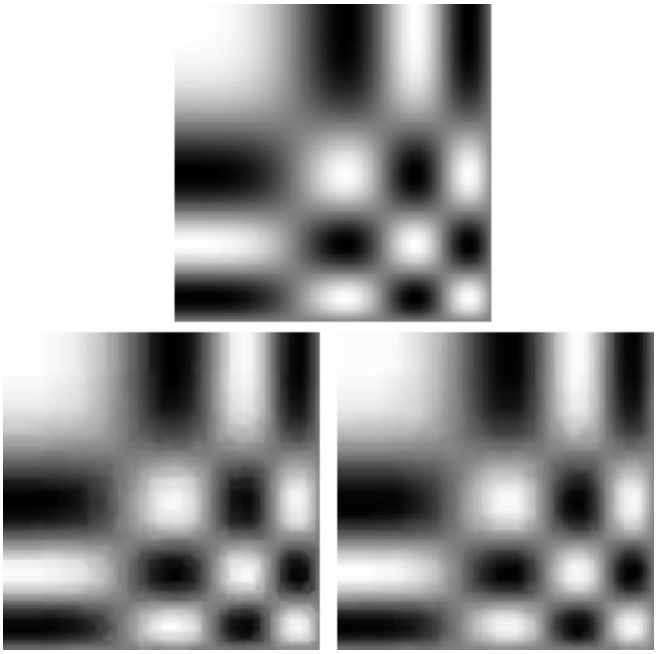


Fig. 8. Subjective comparison: Detail of a chirp image coded with $\mathrm{QP} = 36$. Top: Original. Left: Encoded without ABT (187 720 bit, 30.25 dB). Right: Encoded with ABT (158 304 bit, 32.53 dB).

- The absolute values of the coefficients are binarized using an unary code tree and encoded in reverse order (last coefficient first). The sign of the coefficients is written directly to the bitstream.

IV. Simulation Results

The simulation results given here were generated with the JM 4.2 reference software [22]. ABT is implemented according to [16].

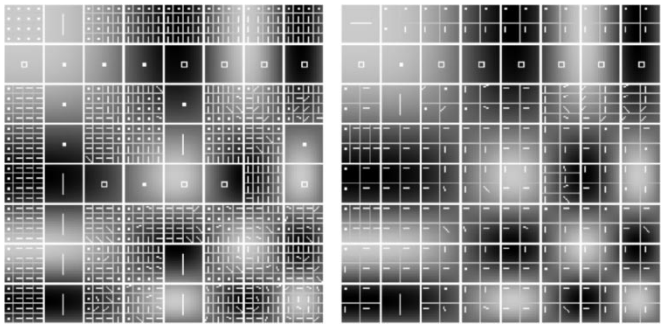
A. Subjective Performance

Two examples for the subjective performance of ABT are given.

Fig. 8 shows an example of the coding performance for a synthetic image. The original gray-level image of size 512×512 contains a separable chirp signal with the frequency increasing from left to right and top to bottom. The image was encoded as a single frame sequence with RD optimization at $\mathrm{QP} = 36$. Fig. 8 shows the top left corner of the chirp image. As can be seen, the ABT-coded image reveals a much more pleasing subjective quality than the non-ABT image does. Also, the rate and PSNR numbers indicate superior objective performance for ABT.

In Fig. 9, the selected intra-prediction modes for ABT and non-ABT coding are shown. For non-ABT, the high number of macroblocks in intra 4×4 Mode with many different prediction directions is very expensive. The larger transform block sizes employed with ABT give an improved compression performance at a convenient subject quality.

In Fig. 10, a detail of first frame of the the HD sequence Shuttle (1280×720 @ 60 Hz) is shown. The sequence was encoded with RD optimization at $\mathrm{QP} = 36$. It can be seen that more detail is preserved by ABT coding. The subjective quality is improved by ABT, even though the PSNR numbers indicate a slight loss.



(a) (b)

Fig. 9. Intra mode selection for the region shown in Fig. 8. (a) Corresponding intra mode selection without ABT. (b) Corresponding intra mode selection wit ABT. Macroblocks are marked white. subblocks of 4×4 or ABT intra modes are marked in light gray. The directional modes are indicated by corresponding lines for both intra 16×16 and subblock modes. DC prediction is indicated by a dot ($\bullet$). Intra 16×16 plane prediction is indicated by an empty square ($\square$).

Fig. 11 shows the intra-prediction modes that are chosen for ABT and non-ABT coding. Without ABT, high detail regions are encoded using the intra 4×4 Mode. In regions with less activity, the intra 16×16 Mode is employed. When ABT is used the number of macroblocks in intra 16×16 Mode is largely reduced. The ABT 8×8 mode is chosen most frequently. In regions with much variation, smaller block sizes are employed. Blocks of size 4×4 are rarely used for intra coding.

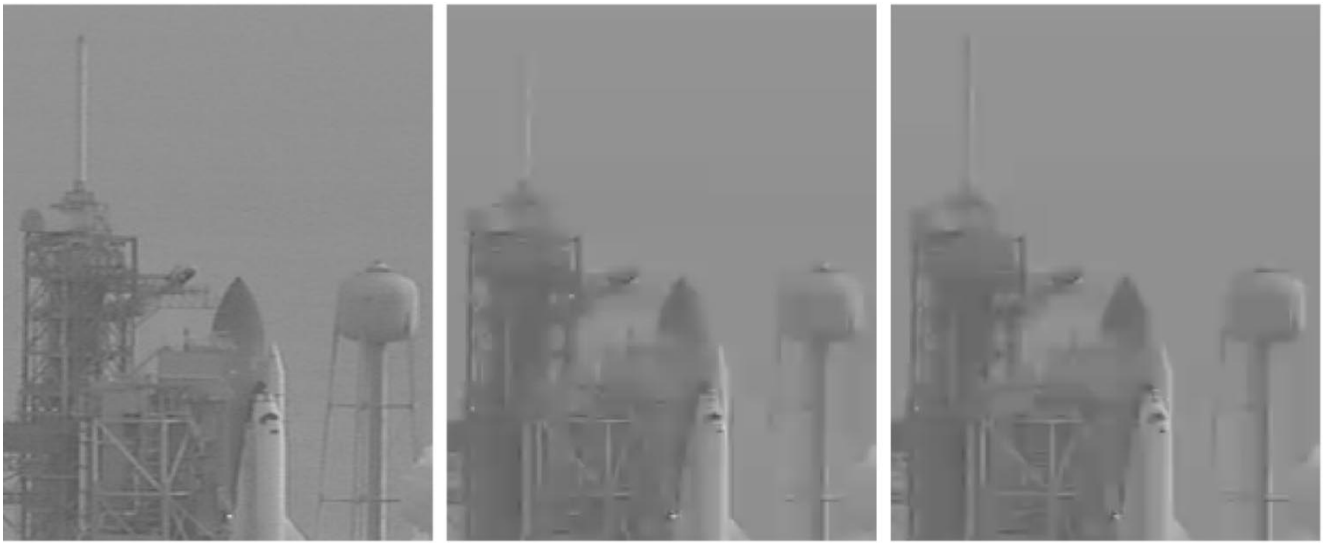
B. RD Performance

The sequences used for the simulations are given in Table I. The sequences RAVEN and KAYAK contain high-definition (HD) video. RAVEN in CIF size was extracted from the full-resolution 1280×720 at 60-Hz sequences by windowing the center of the full-resolution frames and temporal down-sampling by a factor of two. No spatial or temporal filtering was applied. This sequence emulates the characteristics of the full-resolution sequences while saving simulation time. All sequences but KAYAK contain progressive video. KAYAK is an interlaced sequence.

1) Simulation Conditions: In all simulations, RD optimization was switched on. Results for two configurations are presented here. In the first simulation, intra only coding is performed. The inter coding performance was evaluated using an IPPP… and an IBBPBBP… frame structure with a search range of 32 pixels and five reference frames. The sequence KAYAK was encoded in field mode using a search range of 16 pixels and two reference frames. The search range was reduced here to shorten the simulation time. ABT was used for intra and inter macroblocks in the ABT simulations.

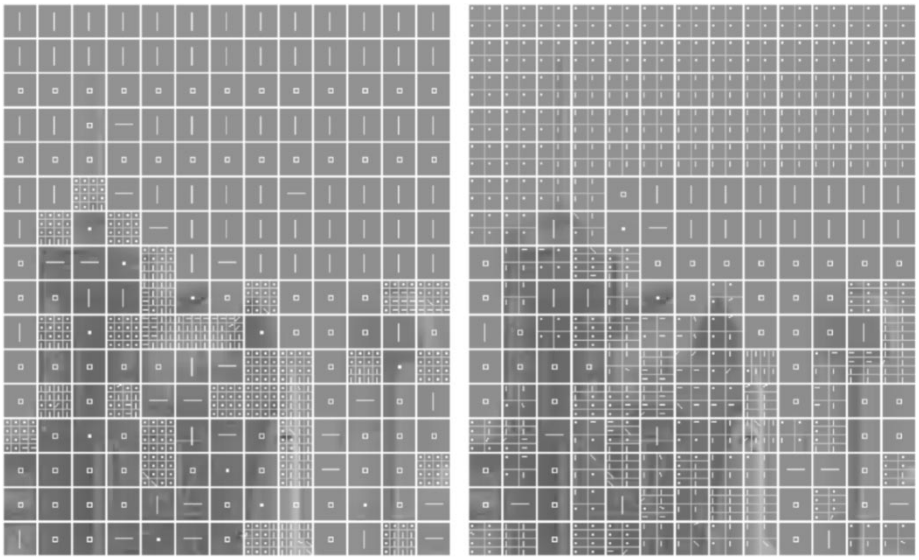
All sequences were encoded using a fixed quantization parameter QP for all frames. The HD sequences were encoded using $\mathrm{QP} = [20, 24, 28, 32]$ for I and P frames. The other sequences were encoded using $\mathrm{QP} = [28, 32, 36, 40]$ for I and P frames. The quantization parameter QP was increased by two for B frames.

2) Results: Table II shows the rate savings and PSNR gains for the simulation conditions given above. Overall, average rate savings of about 8% and PSNR gains of 0.49 dB can be observed for ABT intra coding. The inter gains are lower than the gains achieved for intra coding. This might be due to the re-



(a) (b) (c)

Fig. 10. Subjective comparison: Detail of the first frame of the Shuttle sequence coded with $\mathrm{QP} = 36$. (a) Original. (b) Encoded without ABT (39128 bit, 38.36 dB). (c) Encoded with ABT (38784 bit, 38.25 dB). More detail is preserved with ABT coding (note the flag staff at the top left of the depicted area).



(a) (b)

Fig. 11. Intra mode selection for the region shown in Fig. 10. (a) Encoded without ABT. (b) Encoded with ABT. Macroblocks are marked white. Subblocks of 4×4 or ABT intra modes are marked in light gray. The directional modes are indicated by corresponding lines for both intra 16×16 and subblock modes. DC prediction is indicated by a dot ($\bullet$). Intra 16×16 plane prediction is indicated by an empty square ($\square$).

TABLE I
TEST SEQUENCES

Sequence	Frame Size	Freq.	# Frames
FOREMAN	176×144	10 Hz	100
COASTGUARD	352×288	30 Hz	300
TABLETENNIS	352×288	30 Hz	300
RAVEN	352×288	30 Hz	300
KAYAK	1920×1088	60 Hz	60

TABLE II
OVERALL RATE SAVINGS AND PSNR GAINS FOR ABT CODING

	III		IPPP		IBBP	
Sequence	ΔRate	ΔPSNR	ΔRate	ΔPSNR	ΔRate	ΔPSNR
FOREMAN	-6.89 %	0.43 dB	-5.71 %	0.32 dB	-2.49 %	0.14 dB
COASTG.	-6.20 %	0.33 dB	-8.38 %	0.31 dB	-5.24 %	0.17 dB
TABLET.	-4.33 %	0.23 dB	-4.05 %	0.16 dB	-3.51 %	0.13 dB
RAVEN	-11.65 %	0.58 dB	-8.42 %	0.37 dB	-5.88 %	0.25 dB
KAYAK	-12.35 %	0.90 dB	-9.41 %	0.72 dB	-12.77 %	0.92 dB
Δmean	-8.28 %	0.49 dB	-7.19 %	0.38 dB	-5.98 %	0.32 dB

duced energy and the reduced correlation remaining in the motion predicted signal. Hence, the decorrelation properties of the larger transforms takes less effect. For IPPP, overall average rate savings of about 7% and PSNR gains of 0.38 dB are achieved. When B frames are employed, average rate savings of about 6% and PSNR gains of 0.32 dB are observed for the test set. The delta measurements were performed using the method described in [23]. RD plots for the III and IPPP frame structure are given for the sequences RAVEN and KAYAK in Fig. 12. At high rates, gains of more than 1.0 dB can be observed.

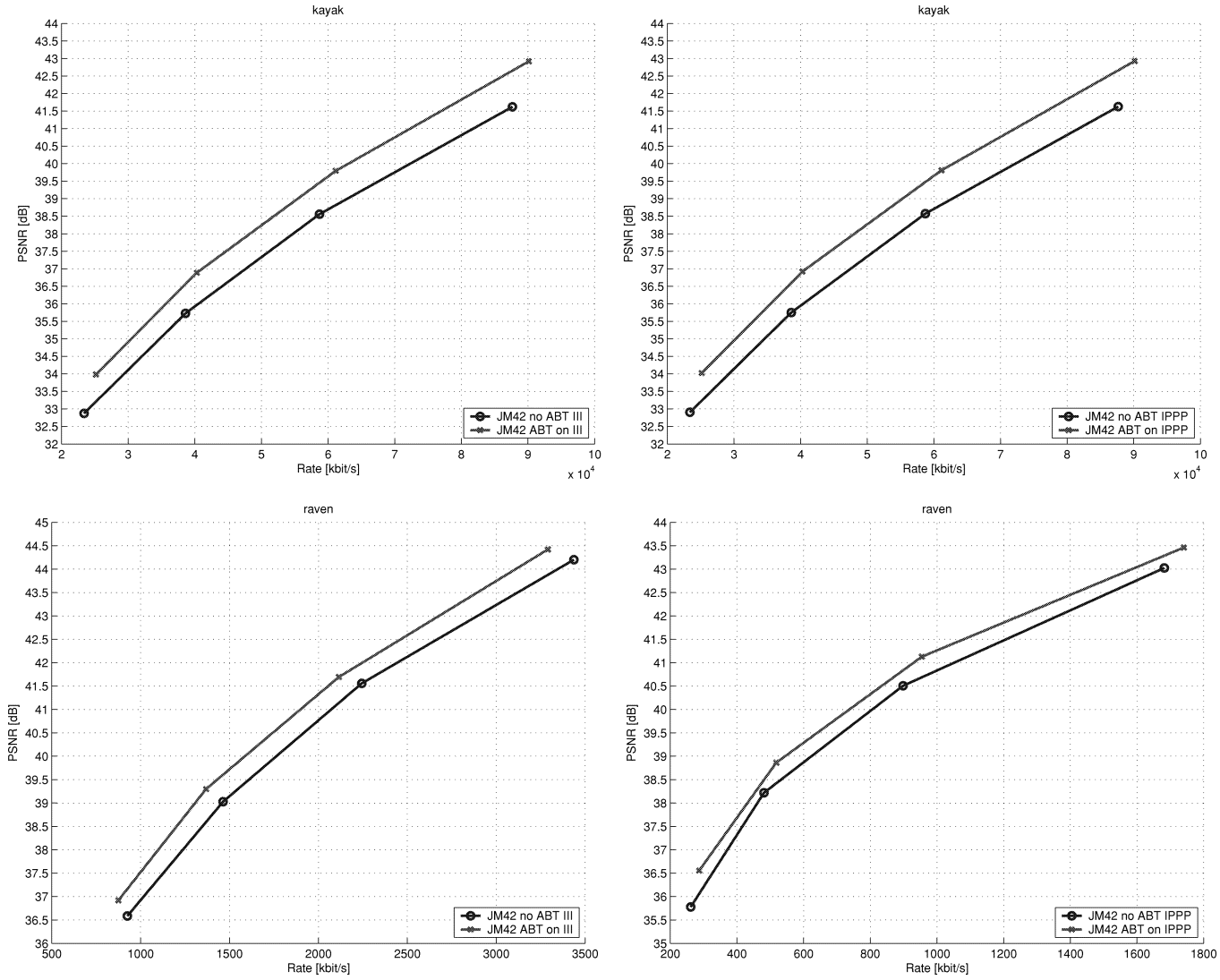


Fig. 12. Simulation results for intra only and inter coding with JM 4.2 without ABT ($\circ$) and with ABT ($\times$) for the sequences RAVEN and KAYAK.

V. CONCLUSION

The concept of ABT for application in H.264/AVC was presented. With ABT, the block size used for transform coding can be aligned to the properties of the encoded signal. Blocks of size 8×8, 8×4, 4×8 and 4×4 can be transformed. For inter coding, no additional overhead information is needed for ABT. For intra coding an additional symbol is transmitted indicating the transform and prediction block size.

The results reveal an increased subjective and objective quality for ABT coding, especially on HD material. Up to 12% rate savings and 0.9 dB increase in PSNR can be observed.

APPENDIX
FAST CALCULATION OF THE 8×8 TRANSFORM

Let $\boldsymbol{c}$ be a coefficient vector of a transformed signal

$$\boldsymbol{c} = [\, c_0 \quad c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5 \quad c_6 \quad c_7 \,]^T \tag{9}$$

and $\boldsymbol{y}$ be the reconstructed signal itself

$$\boldsymbol{y} = \boldsymbol{T}_8^I \cdot \boldsymbol{c}. \tag{10}$$

The transform can be calculated as follows:

$$\begin{aligned}
k_0 &\leftarrow 13(c_0 + c_4); \quad k_4 \leftarrow c_1 + c_7 \\
k_1 &\leftarrow 17c_2 + 7c_6; \quad k_5 \leftarrow c_3 + c_5 \\
k_2 &\leftarrow 13(c_0 - c_4); \quad k_6 \leftarrow c_1 - c_7 \\
k_3 &\leftarrow 7c_2 - 17c_6; \quad k_7 \leftarrow c_3 - c_5 \\
z_0 &\leftarrow k_0 + k_1 \\
z_1 &\leftarrow k_2 + k_3 \\
z_2 &\leftarrow k_2 - k_3 \\
z_3 &\leftarrow k_0 - k_1 \\
z_4 &\leftarrow ((k_5 + c_5) \ll 3) + 3k_4 + k_7 + (c_1 \ll 4) \\
z_5 &\leftarrow ((k_6 + c_1) \ll 3) + 3k_7 - k_4 - (c_5 \ll 4) \\
z_6 &\leftarrow ((k_4 + c_7) \ll 3) - 3k_5 + k_6 - (c_3 \ll 4) \\
z_7 &\leftarrow -((k_7 + c_3) \ll 3) + 3k_6 + k_5 - (c_7 \ll 4) \\
y_0 &\leftarrow z_0 + z_4; \quad y_4 \leftarrow z_3 - z_7 \\
y_1 &\leftarrow z_1 + z_5; \quad y_5 \leftarrow z_2 - z_6 \\
y_2 &\leftarrow z_2 + z_6; \quad y_6 \leftarrow z_1 - z_5 \\
y_3 &\leftarrow z_3 + z_7; \quad y_7 \leftarrow z_0 - z_4.
\end{aligned}$$

$a \ll n$ denotes the bit shift operation $a \cdot 2^n$.

ACKNOWLEDGMENT

The author would like to thank Prof. J. R. Ohm, Aachen University, and U. Benzler, R. Bosch GmbH, as well as A. Dahlhoff and C. Mayer, Aachen University, for many insightful discussions on the topic. The author would also like to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] "Information Technology—Generic Coding of Audio-Visual Objects," ISO/IEC JTC1 IS 14496-2 (MPEG-4), 1998.

[2] Telecommun. Standardization Sector of the ITU, "Video Coding for Low Bitrate Communication (H.263 Version 2)," Sept., 1997.

[3] T. Weigand, "Joint final committee draft (JFCD) of joint video specification," in *Proc. JVT, 4th Meeting*, T. Weigand, Ed., Klagenfurt, Austria, July 2002, ITU-T rec. H.264/ISO/IEC 14 496-10 AVC.

[4] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing ICASSP '91*, Toronto, ON, Canada, May 1991, pp. 2713–2716.

[5] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Trans. Commun.*, vol. 38, pp. 477–486, Apr. 1990.

[6] C.-T. Chen, "Adaptive transform coding via quadtree-based variable block size DCT," in *Proc. IEEE Int. Acoustics, Speech, Signal Processing ICASSP '87*, vol. 3, Glasgow, Scotland, U.K., May 1989, pp. 1854–1857.

[7] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Processing*, vol. 2, pp. 160–175, Feb. 1993.

[8] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Processing*, vol. 40, pp. 2040–2060, Aug. 1992.

[9] ——, "Variable block size image coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing ICASSP '87*, vol. 2, Dallas, TX, Apr. 1987, pp. 1051–1054.

[10] T. J. Klausutis and V. K. Madisetti, "Variable block size adaptive lapped transform-based image coding," in *Proc. IEEE Int. Conf. Image Processing ICIP '97*, vol. 3, Washington, DC, Oct. 1997, pp. 686–689.

[11] K. Ramchandran, Z. Xiong, K. Asai, and M. Vetterli, "Adaptive transforms for image coding using spatially varying wavelet packets," *IEEE Trans. Image Processing*, vol. 5, pp. 1197–1204, July 1996.

[12] I. Dinstein, K. Rose, and A. Heimann, "Variable block-size transform image coder," *IEEE Trans. Commun.*, vol. 38, pp. 2073–2078, Nov. 1990.

[13] H. Malvar, "Low-complexity length-4 transform and quantization with 16-bit arithmetic," in VCEG, 14th Meeting, Santa Barbara, CA, Sept. 2001, Doc. VCEG-N44.

[14] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/avc," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 598–603, July 2003.

[15] G. Bjontegaard, "Addition of 8×8 transform to H.26L," in SG16/Q15, 9. Meeting, Red Bank, NJ, Oct. 1999, Doc. Q15-I-39.

[16] M. Wien, "Clean-up and improved design consistency for ABT," in JVT, 5th Meeting, Geneva, Switzerland, Oct. 2002, Document JVT-E025.

[17] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 688–703, July 2003.

[18] F. E. Douglas and K. R. Ramamohan, *Fast Transforms*. San Diego, CA: Academic, 1982.

[19] D. Marpe, H. Schwarz, and T. Wiegand, "Advanced entropy coding in draft h.264/avc video compression standard," *IEEE Circuits Syst. Video Technol.*, vol. 13, pp. 620–636, July 2003.

[20] W. Limin and B. David, "Alternate coefficient scanning patterns for interlaced ABT coding," in JVT, 3rd Meeting, Fairfax, VA, May 2002, Doc. JVT-C140.

[21] W. Mathias and D. Achim, "16 bit adaptive block size transforms," in JVT, 3rd Meeting, Fairfax, VA, May 2002, Doc. JVT-C107.

[22] K. Sühring, Ed., (2002) JM 4.2 Reference Software. [Online]. Available: ftp://ftp.imtc.org/jvt-experts/reference_software/jm42.zip

[23] G. Bjontegaard, "Calculation of average PSNR differences between rd curves," in SG16/Q6 VCEG, 13. Meeting, Austin, TX, Apr. 2001, Doc. VCEG-M33.

Mathias Wien (S'98–M'03) was born in Bonn, Germany, in 1971. He received the Dipl.-Ing. degree in electrical engineering from Aachen University, Aachen, Germany, in 1997, where he is currently working toward the Dr.-Ing. degree in the Institute of Communications Engineering.

He is an active contributor to the ITU-T VCEG and the Joint Video Team of VCEG and ISO/IEC MPEG where he chaired the AdHoc Group on Additional Transforms and Quantization. His research interests are in the area of image and video processing, with emphasis on space-frequency adaptive and scalable video compression.