High Performance, Low Complexity Video Coding and the Emerging HEVC Standard

Kemal Ugur, Kenneth Andersson, Arild Fuldseth, Gisle Bjøntegaard, Lars Petter Endresen, Jani Lainema, Antti Hallapuro, Justin Ridge, Dmytro Rusanovskyy, Cixun Zhang, Andrey Norkin, Clinton Priddle, Thomas Rusert, Jonatan Samuelsson, Rickard Sjöberg, and Zhuangfei Wu

***Abstract*—This paper describes a low complexity video codec with high coding efficiency. It was proposed to the high efficiency video coding (HEVC) standardization effort of moving picture experts group and video coding experts group, and has been partially adopted into the initial HEVC test model under consideration design. The proposal utilizes a quadtree-based coding structure with support for macroblocks of size 64 × 64, 32 × 32, and 16 × 16 pixels. Entropy coding is performed using a low complexity variable length coding scheme with improved context adaptation compared to the context adaptive variable length coding design in H.264/AVC. The proposal's interpolation and deblocking filter designs improve coding efficiency, yet have low complexity. Finally, intra-picture coding methods have been improved to provide better subjective quality than H.264/AVC. The subjective quality of the proposed codec has been evaluated extensively within the HEVC project, with results indicating that similar visual quality to H.264/AVC High Profile anchors is achieved, measured by mean opinion score, using significantly fewer bits. Coding efficiency improvements are achieved with lower complexity than the H.264/AVC Baseline Profile, particularly suiting the proposal for high resolution, high quality applications in resource-constrained environments.**

***Index Terms*—H.264/AVC, HEVC, standardization, video coding.**

I. Introduction

ISO-IEC/MPEG and ITU-T/VCEG recently formed the joint collaborative team on video coding (JCT-VC). The JCT-VC aims to develop the next-generation video coding standard, called high efficiency video coding (HEVC).

Manuscript received June 30, 2010; revised October 4, 2010; accepted October 19, 2010. Date of publication November 18, 2010; date of current version January 22, 2011. This paper was recommended by Associate Editor T. Wiegand.

K. Ugur, J. Lainema, A. Hallapuro, J. Ridge, and D. Rusanovskyy are with Nokia Corporation, Tampere 33720, Finland (e-mail: kemal.ugur@nokia.com; jani.lainema@nokia.com; antti.hallapuro@nokia.com; justin.ridge@nokia.com; dmytro.rusanovskyy@nokia.com).

K. Andersson, A. Norkin, C. Priddle, T. Rusert, J. Samuelsson, R. Sjöberg, and Z. Wu are with Ericsson Research, Stockholm 16480, Sweden (e-mail: kenneth.andersson@ericsson.com; andrey.norkin@ericsson.com; clinton.priddle@ericsson.com; thomas.rusert@ericsson.com; jonatan.samuelsson@ericsson.com; rickard.sjoberg@ericsson.com; zhuangfei.wu@ericsson.com).

A. Fuldseth, G. Bjøntegaard, and L. P. Endresen are with Tandberg Telecom (Cisco company), Lysaker 1366, Norway (e-mail: arild.fuldseth@tandberg.com; gisle.bjøntegaard@tandberg.com; larspetter.endresen@tandberg.com).

C. Zhang is with Nokia Corporation, Tampere 33720, Finland, and is also with Tampere University of Technology, Tampere 33720, Finland (e-mail: cixun.zhang@tut.fi).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2010.2092613

This paper describes the Tandberg, Ericsson, and Nokia test model (TENTM), a joint proposal to the high-efficiency video coding standardization effort, which has been partially adopted into the test model under consideration (TMuC) as the low complexity operating point [1]. Subjective results for sequences of multiple resolutions between 416 × 240 and 1920 × 1080 show that the proposed codec achieves similar visual quality measured using mean opinion score (MOS) to H.264/AVC High Profile anchors, with around 30% bit rate reduction for low delay experiments, and with around 20% bit rate reduction for random access experiments on average, but with lower complexity than H.264/AVC Baseline Profile [5]. The performance of the proposal was optimized for high resolution use cases; hence, coding efficiency improvements are more visible at higher resolutions, where bit rate reductions at similar MOS reach around 50% and 35% for the low delay and random access experiments, respectively.

This paper is organized as follows. Section II presents the details of the proposed codec, Section III presents a brief complexity analysis, Section IV presents the detailed experimental results, and Section V concludes the paper.

II. Overview of the Proposed Algorithm

The goal of the TENTM proposal was to achieve significantly higher coding efficiency than H.264/AVC at a complexity level not higher than H.264/AVC Baseline Profile. To enable this efficiency–complexity tradeoff, almost all tools of the video codec were improved compared to H.264/AVC. The main aspects of the proposal could be summarized as follows:

1) variable length coding (VLC)-based entropy coding with improved context adaptivity over the H.264/AVC context adaptive variable length coding (CAVLC) design;
2) reduced complexity deblocking filter compared to H.264/AVC deblocking, where a combination of strong and weak filters are utilized;
3) angular intra-picture prediction method using 32 prediction directions;
4) planar coding for representing smooth areas in a visually pleasant manner;
5) interpolation filter using 1-D directional and 2-D separable filter with six taps;

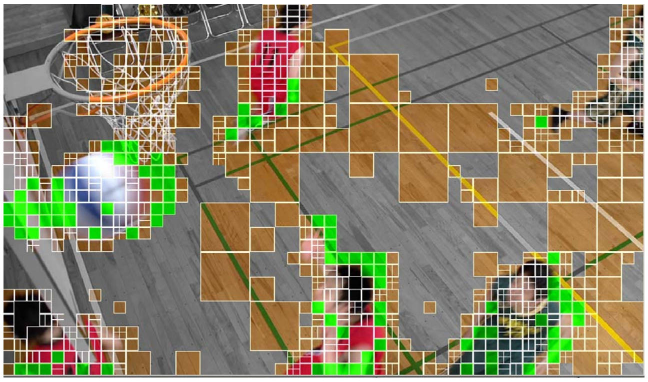


Fig. 1. Illustration of simplified quadtree structure from the *BasketballDrill* sequence frame 71. Gray areas indicate SKIP macroblocks and green blocks indicate macroblocks coded with intra-picture coding methods.

6) quadtree representation of motion with support of 64×64, 32×32, 16×16, 8×16, 16×8, and 8×8 partition sizes.

The detailed description of the proposal is described below.

A. Simple QuadTree Motion and Transform Structure

Coding efficiency can be significantly improved by utilizing macroblock structures with sizes larger than 16×16 pixels, especially at high resolutions [7]. This is due to the ability of large motion and transform blocks to more efficiently exploit the increased spatial correlation that occurs at such high resolutions. At high resolutions, there are more likely to be large homogenous areas that can be efficiently represented by larger block sizes.

The TENTM proposal utilizes large block sizes in the form of a quadtree structure. Large (64×64 and 32×32) macroblocks are employed in addition to traditional 16×16 macroblocks. To limit the encoding complexity, unlike other proposals that also utilized a quadtree structure [3], [4], the TENTM proposal restricts the use of large macroblocks to inter-picture coding modes, and with only one motion vector (the 16×16 macroblocks can be further divided into motion partitions of size 16×8, 8×16, and 8×8). Motion partition sizes smaller than 8×8 are disabled, reducing decoder complexity.

Macroblocks with different sizes can be combined in various ways. The encoder decides on the quadtree structure in a rate–distortion optimized (RDO) fashion. The division of macroblocks from a frame in the *BasketballDrill* test sequence is shown in Fig. 1. Here, the smooth areas are either coded with SKIP macroblocks of size 16×16 pixels (i.e., no information is transmitted for the corresponding macroblocks) or with macroblocks of size 32×32 or 64×64 pixels. As expected, the areas with higher motion and detailed texture are coded using smaller macroblocks.

Transform sizes are represented by a simple quadtree structure. For macroblocks of size 16×16, transforms of size 4×4, 8×8, and 16×16 can be used, with the limitation that the transform size can never be larger than the motion partition size. For large macroblocks of size 32×32 and 64×64, a transform size equal to the size of the macroblock is used. The transforms are separable and use integer-valued basis vectors. In addition, they can be implemented with 16 bit precision after each intermediate stage (horizontal and vertical). For transform sizes larger than 8×8, the proposal utilizes truncated transforms where only the 8×8 low frequency coefficients are calculated. This results in a significant computational complexity saving, and implies that only 4×4 and 8×8 quantization kernels are used. In some cases, this may improve visual quality by avoiding ringing artifacts. This may be explained by more frequent use of small (4×4 and 8×8) transforms for high frequency content (as a result of RDO-based mode decision) while the large transforms are mostly being chosen for smooth areas where ringing is no problem.

The quadtree based prediction representation was common among many proposals to HEVC, and it was also included in the HEVC TMuC. The HEVC TMuC quadtree is more flexible than TENTM, in that it can support more prediction sizes (e.g., 32×16, 64×32), and also support hierarchical coding of the transforms. But it is configurable so that it can operate in the simplified low complexity mode as presented in this paper. The TENTM concept of truncating transforms larger than 8×8 is included in TMuC, in addition to using full transforms.

B. Low Complexity Entropy Coding (LCEC)

H.264/AVC supports two entropy coding methods: CAVLC, which is supported in Baseline Profile due to its low complexity, and the context adaptive binary arithmetic coding, which is an additional option in the Main and High Profiles due to its better coding efficiency. As this proposal targets a low complexity operating point, a low complexity entropy coder is designed based on VLC codes. LCEC is included in HEVC TMuC as the LCEC alternative, and it includes the following features:

1) coding of syntax elements using structured VLC tables;
2) combination of events into single syntax elements;
3) improved context adaptation;
4) improved coding of transform coefficients.

1) *Structured VLC Tables:* LCEC employs fixed-length to variable-length encoding, where each syntax element is converted to a variable-length binary codeword using a structured VLC table. Since different syntax elements have different probability distributions, several VLC tables are defined to closely match a variety of probability distributions. A complete set of structured VLC tables can be found in [5]. One advantage of using structured VLC tables is that the conversion between an integer-valued syntax element and a binary codeword can be computed by a limited number of arithmetic operations rather than storing large tables in memory.

2) *Combination of Parameters:* One limitation of using VLC tables as described above is that each syntax element is encoded with an integer number of bits, resulting in an average bit rate that could be larger than the entropy of that syntax element. This effect is quite significant when encoding binary syntax elements where the probability of 1 (or 0) is much higher than 0.5. In order to reduce the average bit rate,

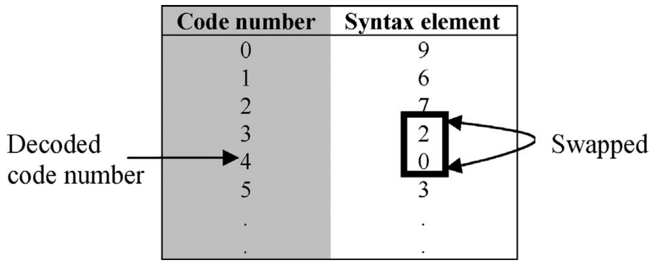


Fig. 2. Illustration of swapping entries in the inverse sorting table.

several parameters are combined into a single syntax element. This increases the granularity of the probability distribution and reduces the difference between the entropy and average bit rate. The details of which parameters are combined into single syntax elements can be found in [5].

3) *Improved Context Adaptation:* Context adaptation in LCEC represents an improvement over the H.264/AVC CAVLC design. Encoding of a parameter (or a combined syntax element) is done in three steps as follows.

a) Convert the parameter value to a table index by using a predefined enumeration scheme.
b) Use the table index to generate a code number through lookup in a sorting table. The purpose of the sorting table is to assign code numbers according to probability so that parameter values with high probability are assigned a code number with a low value.
c) Use the code number to generate a binary codeword by lookup in the predetermined VLC table. The VLC table is designed such that the shortest binary codeword corresponds to the smallest code number.

On the decoder side, the inverse process is used based on an inverse sorting table. The inverse sorting table is adaptively changed based on the occurrence of the symbols. For each code number that is decoded, the corresponding syntax element value is determined by lookup in an inverse sorting table. Next, that entry is swapped in the inverse sorting table with the entry immediately above. This process is illustrated in Fig. 2 where the syntax element values of 0 and 2 are being swapped after decoding a syntax element of value 0. The next time an encoder chooses a syntax element of value 0, the corresponding binary codeword is either shorter or of equal length as before. This mechanism ensures that the value of a frequently occurring syntax element propagates toward the top of the inverse sorting table, corresponding to the most likely value and the shortest binary codeword.

4) *Entropy Coding of Transform Coefficients:* Entropy coding of transform coefficients is performed by an improved version of CAVLC. Assuming a conventional zigzag scan pattern to convert the 2-D transform coefficients into a 1-D array, the proposed scheme has two main features as follows.

a) Backward encoding of the 1-D array from the highest frequency coefficient toward the DC coefficient. This is beneficial since typically, the statistics of the high frequency coefficients are more "stable" than those of the low frequency components (e.g., most high frequency coefficients are quantized to 0 or 1).
b) Typically, the probability of a transform coefficient being quantized to zero increases significantly with frequency. To exploit this, high-frequency transform coefficients are encoded by run-length coding (*run-mode* coding), while the levels of the quantized low-frequency transform coefficients are encoded individually (*level-mode* coding). Switching from *run-mode* to *level-mode* is done adaptively based on the magnitude and position of the transform coefficients.
c) The proposed method is optimized independently for 4×4 and 8×8 transform coefficients. This is different from CAVLC where encoding of 8×8 transform coefficients is based on coefficient interleaving and reuse of the basic CAVLC algorithm which was originally developed for 4×4 transform coefficients.
d) When encoding the highest frequency nonzero coefficient, context adaptation as described above is used.

C. Intra-Picture Prediction

The proposal introduces two techniques to improve the visual quality of decoded video, both of which are included in the HEVC TMuC.

1) *Angular Prediction:* Compared to previous standards, H.264/AVC achieved significant improvement in coding efficiency for intra-picture coding. This is mainly due to spatial prediction, whereby a block can be predicted from its neighbors using up to nine different directions. To represent the directional structures even more accurately, the TENTM proposal extends the set of directional prediction modes for the 8×8 block size. This is done by permitting block prediction in an arbitrary direction by indicating the prediction angle. Assume the prediction block is indicated with a 2-D array *predBlock* and is generated for a block of size $N \times N$ pixels as follows.

a) The angle of prediction is indicated by the displacement of the last row of the prediction block relative to the reference row (the reconstructed row above the prediction block). This displacement is specified with integer pixel accuracy, and indicated with variable $disp\_ref$.
b) For each jth row in the prediction block the displacement relative to reference row is calculated as follows:

$$disp(j) = disp\_ref \cdot ((j+1)/N), \text{ where } j = 0 \ldots N-1. \quad (1)$$

c) For the ith pixel in the prediction row j, $predBlock(i, j)$, its projected position to reference pixels is calculated. The projection can either lay on the upper reference row or the left reference column. If the projection position has fractional accuracy, the prediction is generated using linear interpolation with 1/8 pixel accuracy. Otherwise, the prediction signal is generated by copying the respective reference pixel.

Fig. 3 illustrates this for the case where $disp\_ref$ is given as +1 pixel and the prediction is being generated for the sixth row. The $disp\_ref$ determines the angle used for generating prediction for all the pixels in the block and it is illustrated with arrow. The projections of the pixels fall to the above

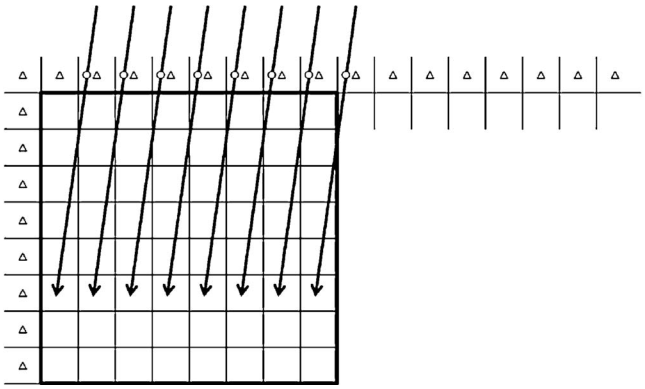


Fig. 3. Example of angular prediction when operating the sixth row of the block with +1 pixel displacement. Triangles indicate the reference pixels and circles indicate the fractional pixels with 1/8 pixel accuracy.

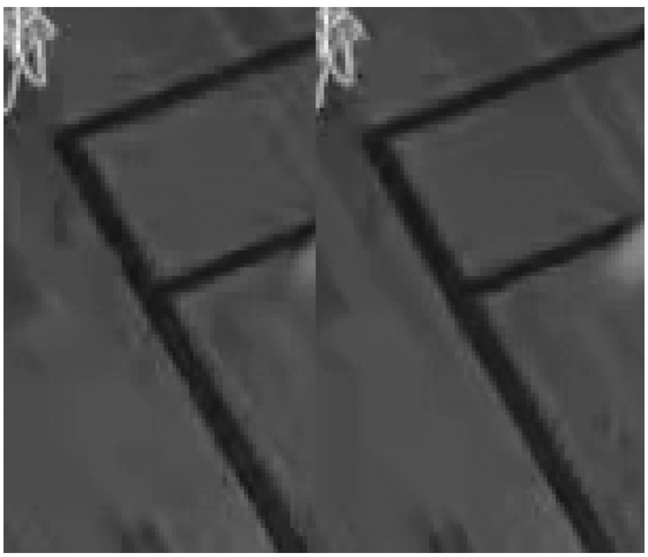


Fig. 4. Example of using angular prediction (right) to improve reconstruction of directional components in reconstructed video (reference on the left using the H.264/AVC prediction directions) with quantization parameter (QP) equal to 22.

reference pixel row with fractional accuracy and are calculated using linear interpolation.

The above procedure is extended so that the left reference column can also used to define the reference displacement variable, *disp_ref*. This way the number of available angles for angular prediction is increased.

The coding efficiency gain provided by this approach over the H.264/AVC directions depends on the sequence and varies between 1.5% and 11.2% for the JCT-VC test set. The average gain over the whole test set was 3.9%. In addition to the objective gains, the method also provides a visual improvement as demonstrated in Fig. 4.

2) *Planar Coding:* The TENTM proposal includes planar coding mode, to enable reconstruction of smooth image segments in a visually pleasing way. The planar mode provides maximal continuity of the image plane at the macroblock borders and is able to follow gradual changes of the pixel values by signaling a planar gradient for each macroblock coded in this mode.

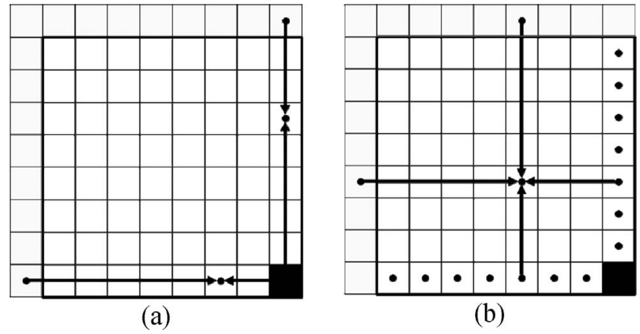


Fig. 5. Illustration of planar coding mode. The uppermost and leftmost pixels belong to the area reconstructed prior to processing the block of pixels surrounded by thick lines. (a) Bottom-right pixel is indicated in the bitstream and the rightmost and bottom sample values are interpolated. (b) Middle sample values are obtained by bilinear interpolation based on the values of the border samples.

When a macroblock is coded in planar mode, its bottom-right sample is signaled in the bitstream. Using this bottom-right sample, the rightmost and the bottom samples in the block are calculated. This is done by linear interpolation between the bottom-right corner sample P_c that is signaled in the bitstream and the closest reference sample P_r in an already-processed neighboring block directly above or to the left from the corner sample. No residual signal is transmitted for a macroblock coded in planar mode.

In more detail, the value for the nth sample in the bottom row or rightmost column of samples is given as follows:

$$P_n = ((S - n) \cdot P_r + n \cdot P_c + S/2)/S) \tag{2}$$

where $n = (1, S)$ and S represents the width and height of a square block of samples. This process is illustrated in Fig. 5(a). Linear interpolation of the border samples is followed by bilinear interpolation of the middle sample values as depicted in Fig. 5(b). During this step, the sample values $P(x, y)$ are obtained by weighted sum of the closest reconstructed reference samples directly above (denoted as $P(x, 0)$), and to the left (denoted as $P(0, y)$), together with the linearly interpolated sample values in the rightmost column and on the bottom row of the block.

As this approach makes the pixel value surface continuous at block boundaries, there is no need to apply traditional deblocking to the edges of planar blocks. Instead, the curvature of the planar surface is reduced by dividing the planar-coded area into piecewise linear sections and reconstructing them using linear interpolation.

The visual improvement using planar mode is more pronounced for smooth regions, where traditional transform coding exhibits visually annoying blocking artifacts. In Fig. 6, the reconstruction of the lady's face is smoother when using planar mode than it is when using discrete cosine transform (DCT) at the same bit rate.

D. Low Complexity Deblocking and Interpolation Filters

1) *Interpolation Filter:* Similar to H.264/AVC, a translational motion model with motion vectors at quarter pixel accuracy is used. The samples at fractional pixel positions are

Fig. 6. Detail of the *Kimono* picture. (a) With DCT coding and finite impulse response deblocking filter. (b) Planar mode at 0.125 b/p.

obtained using two sets of interpolation filters, the directional interpolation filter (DIF) and separable filters (SF).

Directional interpolation is illustrated in Fig. 7. For each sub-pixel position, the filter has a direction determined by the alignment of the sub-pixel position with integer pixel samples. For each of the three horizontal positions and the three vertical positions aligned with full pixel positions, a single 6-tap filter is used. For the nine innermost quarter-pixel positions, two 6-tap filters at +45° and −45° angles are used as follows.

a) Sub-pixel positions e, o are diagonally aligned with integer samples in the northwest-southeast direction. Therefore, the interpolation filter utilizes the integer samples in this direction, which are given as {A1, B2, C3, D4, E5, F6}. In Fig. 7, the integer pixels used in directional interpolation are denoted with circles.
b) Sub-pixel positions g, m are diagonally aligned with integer samples in the northeast-southwest direction. Therefore, the interpolation filter utilizes the integer samples in this direction, which are given as {A6, B5, C4, D3, E2, F1}. In Fig. 7, the integer pixels used in directional interpolation are denoted with squares.
c) Sub-pixel positions f, i, k, n are not aligned with integer pixel samples. To obtain their values, the interpolation filter utilizes the integer samples {A1, A6, B2, B5, C3, C4, D3, D4, E2, E5, F1, F6}, that lie diagonally.

A 12-tap nonseparable filter is used for the central position j. This filter utilizes the integer samples {B3, B4, C2, C3, C4, C5, D2, D3, D4, D5, E3, E4}. This filter is "stronger" than the corresponding directional filter in the sense that the passband is narrower. The motivation for using a strong filter for position j is to have a larger variety of filter responses to choose from during motion vector selection.

Compared to 2-D separable filters, directional filters are sparse and have fewer taps. This can result in poor coding efficiency, especially for highly textured sequences with significant high frequency content. To mitigate this, an additional set of separable filters are used that calculates the interpolated samples by applying a 6-tap filter horizontally and then vertically. For the nine innermost quarter-pixel positions (sub-pixel positions $e, f, g, i, j, k, m, n, o$), the encoder computes the prediction error for both sets of filters (DIF and SF), and chooses the one giving the best rate–distortion performance. A 1-bit flag indicates the filter selection to the decoder. The HEVC TMuC includes directional filters as one of the candidate filter sets, and supports switching between directional and separable filters at slice level.

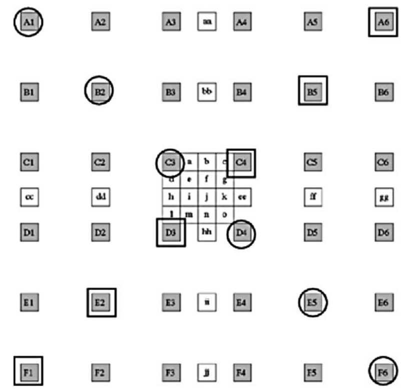


Fig. 7. Directional filter structure to obtain fractional samples.

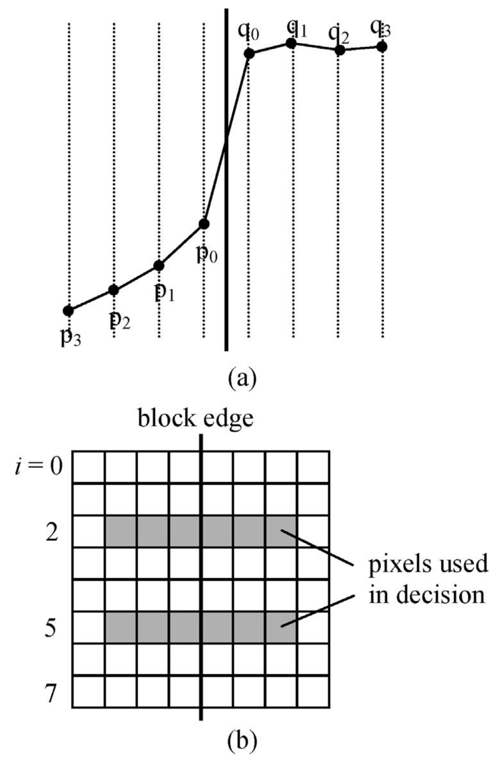


Fig. 8. (a) 1-D visualization of a block edge when the deblocking filter would be turned on. (b) Illustration of filtering decision, only gray pixels are used in decision to filter across edge.

2) *Deblocking Filter:* Similar to H.264/AVC, an in-loop deblocking filter is used to remove blocking artifacts present in the reconstructed video signal. Filtering is potentially performed on all block edges down to the 8 × 8 block level. Transform block edges on the 4 × 4 level are not filtered. This reduces filter complexity without compromising the subjective quality for higher resolutions, considering that the smallest motion prediction partition size in TENTM is 8 × 8.

Filtering of luma block edges is only performed when one of the following conditions is true: at least one of the blocks are coded using intra-picture coding or has nonzero coefficients or the difference between the motion vectors of the blocks is greater than 1 pixel. Further, the decision to filter the edge is done based on the above conditions and a measure d that shows the deviation of the reconstructed signal from straight lines on both sides of the edge as follows:

$$d_i = |p_2 - 2p_1 + p_0| + |q_2 - 2q_1 + q_0| \quad (3)$$

where p_0, p_1, p_2, q_0, q_1, and q_2 refer to pixel values on each side of the edge, as illustrated in Fig. 8(a). To limit the number of operations, the measure d is only calculated for two of the eight rows/columns that are orthogonal to the edge as shown in Fig. 8(b), as opposed to calculating the measure for each edge line separately as done in H.264/AVC deblocking. This is motivated by the fact that the activity between edge lines varies less for high resolution sequences than for lower resolutions. Consequently, significant complexity reduction is achieved with no observable visual degradation at high resolution. The final decision on whether or not to filter the edge is based on comparing the sum of the measures d for each of the two rows/columns with a threshold. Fig. 8(a) shows a typical situation, in which filtering would be turned on.

For each line of pixels across the edge, either a weak or a strong filter is applied. The strong filter is the same filter as the strong filter in H.264/AVC [12]. The weak filter is a low-complexity filter that maintains a straight line across the edge while smoothing the step function. The filtering operations are performed as follows. First, the delta is found as follows:

$$\Delta = (13\cdot(q_0 - p_0) + 4 \cdot (q_1 - p_1) - 5 \cdot (q_2 - p_2))/32. \quad (4)$$

Then Δ is clipped to the interval $(-t_c, t_c)$, where t_c is a parameter that increases with the QP. When one of the blocks is coded using intra-picture coding methods, t_c is calculated from a higher QP value to encourage stronger filtering. Then, pixel values are modified as follows:

$$p_0 = p_0 + \Delta, q_0 = q_0 - \Delta \quad (5)$$

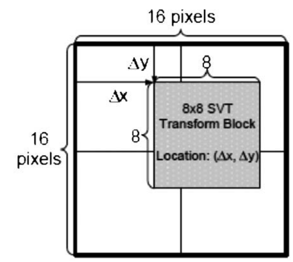
$$p_1 = p_1 + \Delta/2, q_1 = q_1 - \Delta/2. \quad (6)$$

The strong filter is applied over the edges between two smooth flat areas, and is used instead of the weak filtering adaptively based on the pixel values and the value of t_c. Due to its complexity benefits, the proposed deblocking filter is included in the HEVC TMuC.

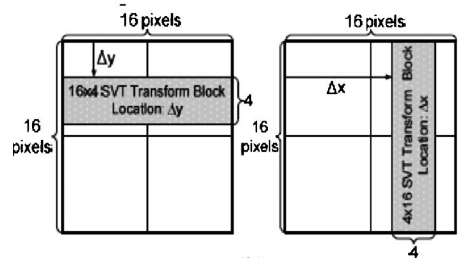
E. Other Features

1) *Spatially Varying Transform (SVT):* The TENTM proposal uses a novel technique called SVT, where the position of the transform block within the macroblock is not fixed but can be varied [11]. The motivations leading to the SVT design are twofold as follows.

a) The block-based transform design in most existing video coding standards does not align the transform blocks with the underlying possible edges in the prediction error.
b) Coding the entire prediction error signal may not be optimal in terms of rate–distortion tradeoff, because the prediction error signal may contain noise which contributes little to quality but is difficult to code.



(a)



(b)

Fig. 9. Illustration of SVT using (a) 8×8 transform and (b) 16×4 and 4×16 transforms. Shaded area illustrate the coded SVT block, and pixels outside the shaded area are set to zero.

The basic idea of SVT is that a single transform block is coded for each macroblock, and its shape and position within the macroblock may vary. This way the prediction error can be better localized and coding efficiency improved. The encoder selects the best transform size and position by searching several candidates using rate–distortion optimization. The encoding complexity of SVT is reduced using certain heuristics, such as limiting the number of available SVT positions and skipping the SVT search if the cost of modes such as SKIP are below a threshold [13]. Fig. 9 illustrates how SVT is implemented in TENTM, where transform blocks of size 16×4, 4×16, and 8×8 are utilized.

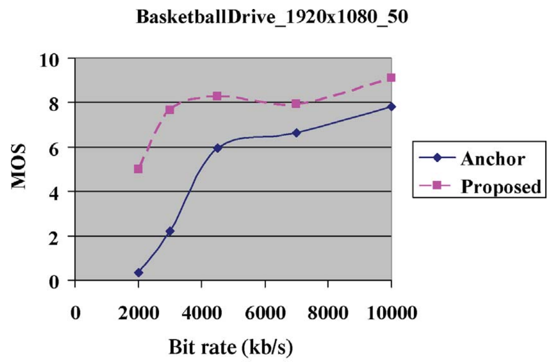
2) *Low Complexity B Pictures:* To reduce the complexity of the B picture decoding process, motion vectors for SKIP and DIRECT modes in B pictures always have integer pixel accuracy. Additionally, the reference frame indices of neighboring blocks are jointly predicted. If neighboring blocks use bi-predictive coding with reference frames A and B, these frames are used for prediction in the current block.

3) *Improved SKIP Mode:* In SKIP mode, the encoder first determines two motion vector candidates for the macroblock, then signals with 1 bit which one of the two predictors are used. If both of the motion vectors have the same magnitude, no signaling is done.

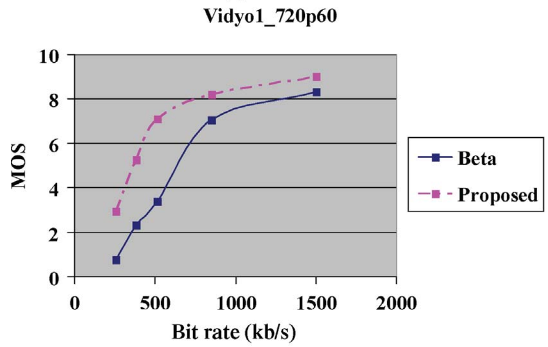
III. Complexity Analysis

As mentioned earlier, the main motivation behind the TENTM proposal was to achieve low complexity operation, and to improve the coding efficiency over H.264/AVC, especially at high resolutions. This section discusses the main factors leading to reduced complexity.

1) Average interpolation complexity is less than that of H.264/AVC due to the 1-D directional interpolation filters.

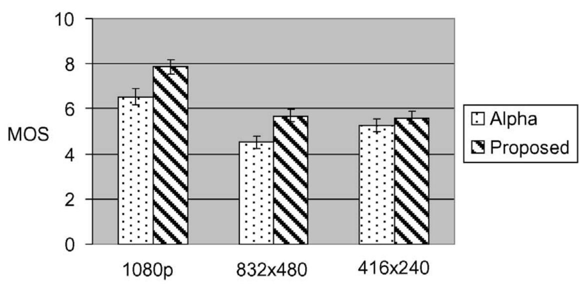


(a)

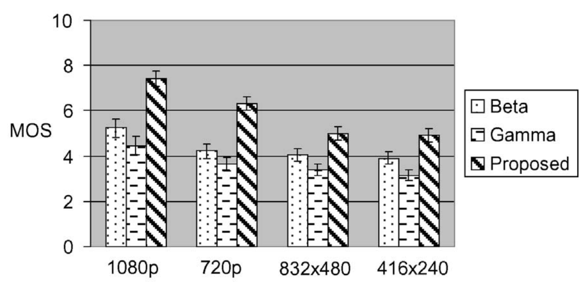


(b)

Fig. 10. MOS–bit rate for (a) Alpha anchor and TENTM for 1080p50 sequence coded in random access and (b) Beta anchor and TENTM for a 720p60 sequence coded in low delay.



(a)



(b)

Fig. 11. Average MOS for different resolutions (a) random access constraint and (b) low delay constraint.

2) Memory bandwidth for motion compensation is lower than H.264/AVC. This is because motion partitions smaller than 8×8 are not used.
3) Interpolation complexity for B pictures is significantly lower in the proposal than in H.264/AVC, as the SKIP and DIRECT modes use integer motion vectors. Using motion vectors with integer pixel accuracy implies that reconstruction of a SKIP/DIRECT block can be done by copying pixels from one location in memory to another location, instead of applying the 6-tap interpolation filter.
4) The deblocking filter has significantly lower computational complexity. This is mainly because a much lower number of block edges are filtered, and the TENTM filter has simpler logic for enabling/disabling the filter on an edge. In addition, filtering can be performed in parallel first for each of the vertical edges and then for each of the horizontal edges, which is not possible for H.264/AVC.
5) VLC (de)coding of coefficients is simpler than H.264/AVC CAVLC coefficient (de)coding. In particular, CAVLC relies on decoding a large number of syntax elements (e.g., *coeff_token, trailing_ones_sign_flag, level_prefix, level_suffix, total_zeros,* and *run_before*). Depending on the value of a particular syntax element, a large number of conditional branches need to be parsed. In TENTM, VLC decoding uses significantly fewer syntax elements, and requires less conditional branches (*run/level/sign* in run mode, and *level/sign* in level mode).

In order to test the encoding and decoding complexity of the proposed algorithm, the encoding and decoding time of the proposal was compared to the H.264/AVC reference software JM17.0 [14]. The simulation results show that encoding using the proposed algorithm is around 25 times faster than JM17.0 encoding, and decoding is 2–3 times faster than that of JM17.0. It should be noted that the significant speedup does not mean the TENTM proposal is 25 times less complex than H.264/AVC; the speedup can be attributed to both low algorithmic complexity of the proposal compared to H.264/AVC, and also to the implementation of the proposal from scratch in clean software with pure C, avoiding many brute-force encoding techniques. The TENTM software was made publically available, so that these complexity claims could be verified by others [5].

IV. Experimental Results

The coding efficiency of the proposal has been evaluated in JCT-VC [2] by performing extensive subjective testing for two different application areas with different constraints: random access constraint (CS1) and low delay constraint (CS2). For both constraint sets, test sequences with resolutions ranging from WQVGA (416×240) to 1080p (1920×1080) were coded at five different bit rates as defined in [6]. In addition, three different anchors were coded using H.264/AVC for the same constraint sets at the same bit rates. Those anchors are as follows.

1) *Alpha Anchor:* High profile using a hierarchical-B prediction structure, satisfying the random access constraint.
2) *Beta Anchor:* High profile using a hierarchical-P prediction structure, satisfying the low delay constraint.
3) *Gamma Anchor:* Baseline profile using an IPPPP prediction structure with low encoder complexity.

The improvements in subjective and objective quality for the five operational rate points for each sequence are estimated as follows. The Bjøntegaard-Delta (BD) measurement [8], that is typically used to measure the bit rate reduction between two rate–distortion curves, is utilized both for peak signal-to-noise ratio (PSNR)-based measurements (BDrate–PSNR) and for measurements using MOS from subjective tests (BDrate–MOS). The use of BDrate–MOS was first presented in [15] where it was shown that the MOS–bit rate curves are similar to the PSNR–bit rate curves for most of the bit rates. One difference is that MOS–bit rate curves tend to saturate at high bit rates. In this paper, the BDrate–MOS measurements have been further improved by using lower order polynomials, order 3, for representation of five operational rate/MOS points (MOS–bit rate curve). This makes the BDrate–MOS measurements more accurate and avoids overfitting to small MOS variations. Example MOS–bit rate curves are shown in Fig. 10, comparing the TENTM performance with H.264/AVC anchors coded in low delay and random access modes. The BDrate–MOS measurement serves as a rough indicator of coding efficiency improvement, as confidence intervals of MOS measurements are not taken into account.

The subjective and objective results, comparing the proposal to Alpha, Beta, and Gamma anchors for five operational points are shown in Tables I and II. Table III shows that the proposed method achieves similar visual quality compared to H.264/AVC High Profile anchors, measured using MOS, with around 30% bit rate reduction for low delay experiments and with around 20% bit rate reduction for random access experiments on average, yet with lower complexity than H.264/AVC Baseline Profile. The proposal was designed to improve the coding efficiency especially at higher resolutions to enable the emerging use cases in resource-constrained devices. Therefore, it is particularly noteworthy that the coding efficiency of the proposal is significantly higher at high-definition (HD) resolutions (720p and 1080p), with around 35% and 50% bit rate reduction for random access and low delay constraints, respectively. In addition, it was noted in the first JCT-VC meeting that "subjectively in the test results overall this proposal did particularly well, when considering its relatively low encoding and decoding complexity" [16].

In Fig. 11, average MOS scores for the anchors and the proposal are shown with 95% confidence intervals for each resolution/class, both for random access and low delay constraints. The proposed codec performs statistically better than the H.264/AVC anchors for all resolutions of CS2 and CS1, except for the smallest resolution (416 × 240) of CS1, where the confidence intervals are overlapping.

It should be noted that the improvement in subjective quality is much more noticeable at higher resolutions. This is due to the fact that the design goal of TENTM proposal was to optimize the quality at high resolutions with as low complexity proposal as possible. Consequently, TENTM omitted many techniques that have high complexity but could improve the coding efficiency at lower resolutions, including the following.

1) *Longer-Tap Filters:* the TENTM directional interpolation filter utilizes 1-D filters with fewer taps than 2-D

TABLE I
SUBJECTIVE PERFORMANCE OF TENTM COMPARED TO H.264/AVC

	Subjective Performance of TENTM (BDrate–MOS, %)		
Sequences	Alpha Anchor	Beta Anchor	Gamma Anchor
Kimono	−20.8	−35.9	−51.1
ParkScene	−32.7	−51.4	−43.7
Cactus	−17.1	−44.9	−53.7
BQTerrace	−41.2	−56.9*	−67.6*
BasketballDrive	−56.6	−64.2	−70.5
Average 1080p	**−33.7**	**−50.7**	**−57.3**
Vidyo1	X	−38.6	−45.8
Vidyo3	X	−34.9	−54.3
Vidyo4	X	−47.8	−54.1
Average 720p	**X**	**−40.4**	**−51.4**
BQMall	−20.4	−21.8	−31.1
PartyScene	6.4	4.1	−26.2
RaceHorses	−36.3	−12.3	−35.6
BasketballDrill	−15.8	−28.3	−44.9
Average 832 × 480	**−16.5**	**−14.6**	**−34.5**
BQSquare	−30.1	−56.4	−82.6
RaceHorses	−11.6	−7.3	−10.1
BasketballPass	−5.1	−19.8	−33.0
BlowingBubbles	8.1	2.0	−31.1
Average 416 × 240	**−9.7**	**−20.4**	**−39.2**
Average all	**−21.0**	**−32.1**	**−46.0**

*The lowest quality of the proposal is better than the highest quality anchor for this sequence. Therefore, to enable a BDrate estimate the highest MOS of the anchor was raised to 0.01 above the lowest MOS of the proposal.

TABLE II
OBJECTIVE PERFORMANCE OF TENTM COMPARED TO H.264/AVC

	Objective Performance of TENTM (BDRate–PSNR, %)		
Sequences	Alpha Anchor	Beta Anchor	Gamma Anchor
Traffic	−13.9	X	X
PeopleOnStreet	−4.0	X	X
Average 2560 × 1600	**−8.9**	**X**	**X**
Kimono	−20.6	−26.8	−44.4
ParkScene	−10.6	−15.5	−36.3
Cactus	−11.1	−18.9	−40.9
BQTerrace	−12.5	−23.8	−52.7
BasketballDrive	−15.8	−22.4	−39.6
Average 1080p	**−14.1**	**−21.5**	**−42.78**
Vidyo1	X	−32.9	−49.4
Vidyo3	X	−26.7	−45.2
Vidyo4	X	−31.4	−51.1
Average 720p	**X**	**−30.3**	**−48.5**
BQMall	−16.2	−15.3	−34.1
PartyScene	−16.7	−21.8	−47.0
RaceHorses	−14.7	−5.4	−17.0
BasketballDrill	−17.7	−12.6	−35.9
Average 832 × 480	**−16.3**	**−13.8**	**−33.5**
BQSquare	−8.9	−8.0	−54.1
RaceHorses	−1.1	3.1	−6.2
BasketballPass	−8.6	−7.7	−21.8
BlowingBubbles	−9.0	−4.5	−33.1
Average 416 × 240	**−6.9**	**−4.3**	**−28.8**
Average all	**−12.1**	**−16.9**	**−38.1**

TABLE III

SPECIFIC EXAMPLES OF SUBJECTIVE PERFORMANCE OF TENTM COMPARED TO H.264/AVC AT SIMILAR MOS

	Alpha Anchor		TENTM		Beta Anchor		TENTM		Gamma Anchor		TENTM	
Sequences	Bit Rate (kb/s)	MOS	Bit Rate (kb/s)	MOS	Bit Rate (kb/s)	MOS	Bit Rate (kb/s)	MOS	Bit Rate (kb/s)	MOS	Bit Rate (kb/s)	MOS
1080p Sequences												
Similar MOS values are achieved at roughly 40–50% less bit rate than H.264/AVC anchors on average												
Kimono	2500	8.17	1600	7.94	2500	5.93	1600	6.22	2500	4.18	1000	4.06
ParkScene	4000	7.41	2500	7.50	4000	6.12	2500	7.50	4000	7.22	2500	7.50
Cactus	7000	8.61	4500	8.94	7000	6.63	3000	7.00	7000	6.35	3000	7.00
BQTerrace	7000	8.15	3000	8.12	7000	7.94	3000	7.94	10000	7.95	3000	7.94
BasketballDrive	7000	7.44	3000	7.83	7000	6.65	3000	7.67	7000	6.03	3000	7.67
720p Sequences												
Similar MOS values are achieved at roughly 40–50% less bit rate than H.264/AVC anchors on average												
Vidyo1	X	X	X	X	850	7.06	512	7.11	850	6.17	512	7.11
Vidyo3	X	X	X	X	850	5.17	512	5.39	850	3.89	384	3.61
Vidyo4	X	X	X	X	850	6.89	384	6.61	850	6.72	384	6.61
832×480 Sequences												
Similar MOS values are achieved at roughly 20–25% less bit rate than H.264/AVC anchors on average												
BQMall	512	3.75	384	3.50	512	3.00	384	2.38	512	1.31	384	2.38
PartyScene	512	2.13	384	1.94	512	2.00	512	2.31	768	2.69	512	2.31
RaceHorses	768	3.25	384	3.31	768	4.63	768	5.50	1200	5.38	768	5.50
BasketballDrill	768	4.06	512	3.50	768	2.63	512	2.31	768	1.73	384	1.81
416×240 Sequences												
Similar MOS values are achieved at roughly 20–25% less bit rate than H.264/AVC anchors on average												
BQSquare	512	3.56	384	4.07	512	1.75	256	1.40	512	0.47	256	1.40
RaceHorses	512	4.53	384	4.27	512	4.33	512	5.60	512	3.50	384	3.33
BasketballPass	512	3.88	512	4.33	512	1.69	384	1.73	512	0.94	384	1.73
BlowingBubbles	512	6.33	512	5.75	512	6.33	512	6.06	512	3.53	384	3.56

separable filters. An interpolation filter with more taps would improve the coding efficiency at lower resolutions, but significantly increase the complexity.

2) *Smaller Motion Partition Sizes:* a 4×4 motion partition would improve the coding efficiency at lower resolutions, but is not included in TENTM as this would increase the memory bandwidth significantly.
3) *Checking Filter Edges for Each Line:* unlike the H.264/AVC deblocking filter, TENTM does not check filter edges for each line. This could impact the subjective quality at lower resolutions.

V. CONCLUSION

This paper presented the joint proposal by Tandberg, Nokia, and Ericsson that was partially adopted into the TMuC by JCT-VC as the low complexity operating point. Subjective testing results showed that the proposal achieved a bit rate reduction of around 20–30% on average when compared to H.264/AVC High Profile. The improvements became more visible at HD resolutions (720p and 1080p), where the proposal required around 35% and 50% fewer bits than H.264/AVC High Profile anchors at random access and low delay experiments, respectively, at the same subjective quality measured using MOS. The coding efficiency improvement was achieved with very low complexity, which makes the proposal especially suitable in resource-constrained scenarios. The HEVC TMuC design included the LCEC, 1-D directional interpolation filters, low complexity deblocking filter, angular prediction, planar coding, and truncated transform techniques from the TENTM proposal. In addition, the TMuC included a very flexible quadtree motion and transform representation, which could be configured to run also in low complexity mode as presented in this paper.

REFERENCES

[1] Joint Collaborative Team: Video Coding, *Test Model Under Consideration* [Online]. JCTVC-A205, Dresden, Germany, Apr. 15–23, 2010. Available: http://wftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A205.zip

[2] Joint Collaborative Team: Video Coding, *Report of Subjective Test Results of Responses to the Joint Call for Proposals (CfP) on Video Coding Technology for High Efficiency Video Coding (HEVC)* [Online]. JCTVC-A204, Dresden, Germany, Apr. 15–23, 2010. Available: http://wftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A204.zip

[3] K. McCann, W.-J. Han, I.-K. Kim, J.-H. Min, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y.-M. Hong, M.-S. Cheon, and N. Shlyakhov, *Video Coding Technology Proposal by Samsung (and BBC)* [Online]. JCTVC-A124, Dresden, Germany, Apr. 15–23, 2010. Available: http://wftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A124.zip

[4] M. Winken, S. Boße, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, D. Marpe, S. Oudin, M. Preiß, H. Schwarz, M. Siekmann, K. Sühring, and T. Wiegand, *Video Coding Technology Proposal by Fraunhofer HHI* [Online]. JCTVC-A116, Dresden, Germany, Apr. 15–23, 2010. Available: http://wftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A116.zip

[5] K. Ugur, K. R. Andersson, and A. Fuldseth, *Video Coding Technology Proposal by Tandberg, Nokia, and Ericsson* [Online]. JCTVC-A119, Dresden, Germany, Apr. 15–23, 2010. Available: http://wftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A119.zip

[6] *Joint Call for Proposals on Video Compression Technology* [Online]. ISO/IEC JTC1/SC29/WG11/N11113, ITU-T Q6/16, document VCEG-AM91, Jan. 2010. Available: http://wftp3.itu.int/av-arch/video-site/1001_Kyo/VCEG-AM91.zip

[7] S. Ma and C.-C. J. Kuo, "High-definition video coding with super-macroblocks," *Proc. SPIE*, vol. 6508, part 1, p. 650816, Jan. 2007.

[8] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves* [Online]. ITU-T SG16 Q.6, document VCEG-M33, Austin, TX, Apr. 2001. Available: http://wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M33.doc

[9] A. Fuldseth, G. Bjøntegaard, D. Rusanovskyy, K. Ugur, and J. Lainema, *Low-Complexity Directional Interpolation Filter* [Online]. ITU-T Q.6/SG16, document VCEG-AI12, Berlin, Germany, Jul. 2008. Available: http://wftp3.itu.int/av-arch/video-site/0807_Ber/VCEG-AI12.zip

[10] D. Rusanovskyy, K. Ugur, A. Hallapuro, J. Lainema, and M. Gabbouj, "Video coding with low complexity directional adaptive interpolation filters," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1239–1243, Aug. 2009.

[11] C. Zhang, K. Ugur, J. Lainema, and M. Gabbouj, "Video coding using variable block-size spatially varying transforms," in *Proc. IEEE ICASSP*, Apr. 2009, pp. 905–908.

[12] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[13] C. Zhang, K. Ugur, J. Lainema, A. Hallapuro, and M. Gabbouj, "Low complexity algorithm for spatially varying transforms," in *Proc. PCS*, May 2009, pp. 1–4.

[14] *H.264/AVC Reference Software* [Online]. Available: http://iphome.hhi.de/suehring/tml/

[15] K. Andersson, R. Sjöberg, and A. Norkin, *BD Measurements Based on MOS* [Online]. ITU-T Q6/SG16, document VCEG-AL23, Geneva, Switzerland, Jul. 2009. Available: http://wftp3.itu.int/av-arch/video-site/0906_LG/VCEG-AL23.zip

[16] Joint Collaborative Team: Video Coding, *Meeting Report of the First Meeting of the Joint Collaborative Team on Video Coding (JCT-VC)* [Online]. JCTVC-A200, Dresden, Germany, Apr. 15–23, 2010. Available: http://wftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A200.zip

Kemal Ugur is currently with Nokia Corporation, Tampere, Finland, researching next-generation video compression algorithms and their efficient implementation on mobile architectures. He also actively participates in various standards and industry bodies (ITU-T, MPEG, 3GPP, DVB, and others) to define next-generation multimedia entertainment and communication standards.

Kenneth Andersson is currently with Ericsson Research, Stockholm, Sweden, researching video compression algorithms and addressing both next-generation compression standards and 3-D video for broadcast, mobile, and conferencing applications. His current research interests focus on efficient and inter-operable use of video compression technologies by driving standardization in MPEG, ITU-T, and 3GPP.

Arild Fuldseth is currently with Tandberg Telecom (now part of Cisco), Lysaker, Norway, researching next-generation video compression algorithms. He is also involved in efficient implementation of existing video compression standards on various platforms for video conferencing equipment.

Gisle Bjøntegaard is currently with Tandberg Telecom (now part of Cisco), Lysaker, Norway, researching next-generation video compression algorithms. He is also involved in efficient implementation of existing video compression standards on various platforms for video conferencing equipment.

Lars Petter Endresen is currently with Tandberg Telecom (now part of Cisco), Lysaker, Norway, researching next-generation video compression algorithms. He is also involved in efficient implementation of existing video compression standards on various platforms for video conferencing equipment.

Jani Lainema is currently with Nokia Corporation, Tampere, Finland, researching next-generation video compression algorithms and their efficient implementation on mobile architectures. He also actively participates in various standards and industry bodies (ITU-T, MPEG, 3GPP, DVB, and others) to define next-generation multimedia entertainment and communication standards.

Antti Hallapuro is currently with Nokia Corporation, Tampere, Finland, researching next-generation video compression algorithms and their efficient implementation on mobile architectures. He also actively participates in various standards and industry bodies (ITU-T, MPEG, 3GPP, DVB, and others) to define next-generation multimedia entertainment and communication standards.

Justin Ridge is currently with Nokia Corporation, Tampere, Finland, researching next-generation video compression algorithms and their efficient implementation on mobile architectures. He also actively participates in various standards and industry bodies (ITU-T, MPEG, 3GPP, DVB, and others) to define next-generation multimedia entertainment and communication standards.

Dmytro Rusanovskyy is currently with Nokia Corporation, Tampere, Finland, researching next-generation video compression algorithms and their efficient implementation on mobile architectures. He also actively participates in various standards and industry bodies (ITU-T, MPEG, 3GPP, DVB, and others) to define next-generation multimedia entertainment and communication standards.

Cixun Zhang is currently with Nokia Corporation, Tampere, Finland, researching next-generation video compression algorithms and their efficient implementation on mobile architectures. He also actively participates in various standards and industry bodies (ITU-T, MPEG, 3GPP, DVB, and others) to define next-generation multimedia entertainment and communication standards.

Andrey Norkin is currently with Ericsson Research, Stockholm, Sweden, researching video compression algorithms and addressing both next-generation compression standards and 3-D video for broadcast, mobile, and conferencing applications. His current research interests focus on efficient and inter-operable use of video compression technologies by driving standardization in MPEG, ITU-T, and 3GPP.

Clinton Priddle is currently with Ericsson Research, Stockholm, Sweden, researching video compression algorithms and addressing both next-generation compression standards and 3-D video for broadcast, mobile, and conferencing applications. His current research interests focus on efficient and inter-operable use of video compression technologies by driving standardization in MPEG, ITU-T, and 3GPP.

Thomas Rusert is currently with Ericsson Research, Stockholm, Sweden, researching video compression algorithms and addressing both next-generation compression standards and 3-D video for broadcast, mobile, and conferencing applications. His current research interests focus on efficient and inter-operable use of video compression technologies by driving standardization in MPEG, ITU-T, and 3GPP.

Jonatan Samuelsson is currently with Ericsson Research, Stockholm, Sweden, researching video compression algorithms and addressing both next-generation compression standards and 3-D video for broadcast, mobile, and conferencing applications. His current research interests focus on efficient and inter-operable use of video compression technologies by driving standardization in MPEG, ITU-T, and 3GPP.

Rickard Sjöberg is currently with Ericsson Research, Stockholm, Sweden, researching video compression algorithms and addressing both next-generation compression standards and 3-D video for broadcast, mobile, and conferencing applications. His current research interests focus on efficient and inter-operable use of video compression technologies by driving standardization in MPEG, ITU-T, and 3GPP.

Zhuangfei Wu is currently with Ericsson Research, Stockholm, Sweden, researching video compression algorithms and addressing both next-generation compression standards and 3-D video for broadcast, mobile, and conferencing applications. His current research interests focus on efficient and interoperable use of video compression technologies by driving standardization in MPEG, ITU-T, and 3GPP.