HEVC Deblocking Filter

Andrey Norkin, Gisle Bjøntegaard, Arild Fuldseth, Matthias Narroschke, Masaru Ikeda, Kenneth Andersson, Minhua Zhou, and Geert Van der Auwera

***Abstract*—This paper describes the in-loop deblocking filter used in the upcoming High Efficiency Video Coding (HEVC) standard to reduce visible artifacts at block boundaries. The deblocking filter performs detection of the artifacts at the coded block boundaries and attenuates them by applying a selected filter. Compared to the H.264/AVC deblocking filter, the HEVC deblocking filter has lower computational complexity and better parallel processing capabilities while still achieving significant reduction of the visual artifacts.**

***Index Terms*—Block-based coding, deblocking, video coding, video filtering, video processing.**

I. Introduction

HIGH EFFICIENCY Video Coding (HEVC) [1] is a new video coding standard currently being developed jointly by ITU-T SG 16 Q.6, also known as the Video Coding Experts Group (VCEG), and by ISO/IEC JTC 1/SC 29/WG 11, also known as the Moving Picture Experts Group (MPEG) in the joint collaborative team on video coding (JCT-VC). The first version of the HEVC standard is planned to be finalized in January 2013, while the development of the scalable and 3-D extensions of HEVC is expected in the following years. Similar to the previous video coding standards, such as H.264/AVC, the upcoming HEVC standard is based on a hybrid coding scheme using block-based prediction and transform coding. First, the input signal is split into rectangular blocks that can be predicted from previously decoded data either by motion-compensated prediction [3] or intra prediction. The resulting prediction error is coded by applying block transforms based on an integer approximation of the discrete cosine transform, which is followed by the quantization and coding of the transform coefficients. While H.264/AVC [2] divides a picture into fixed size macroblocks of 16×16 samples, HEVC divides a picture into coding tree units (CTU) of 16×16, 32×32 or 64×64 samples. The coding tree units can be further divided into smaller blocks using a quadtree structure; such a block, called a coding unit (CU), can further be split into prediction units (PUs) and is also a root for the transform quadtree. Each of the child nodes of the transform quadtree defines a transform unit (TU). The size of the transforms used in the prediction error coding can vary from 4×4 to 32×32 samples, thus allowing transforms larger than in H.264/AVC, which uses 4×4 and 8×8 transforms. As the optimal size of the above-mentioned blocks depends typically on the picture content, the reconstructed picture is composed of blocks of various sizes, each block being coded using an individual prediction mode and the prediction error transform.

In a coding scheme that uses block-based prediction and transform coding, discontinuities can occur in the reconstructed signal at the block boundaries. Visible discontinuities at the block boundaries are known as blocking artifacts. A major source of blocking artifacts is the block-transform coding of the prediction error followed by coarse quantization. Moreover, in a motion-compensated prediction process, predictions for adjacent blocks in the current picture might not come from adjacent blocks in the previously coded pictures, which creates discontinuities at the block boundaries of the prediction signal. Similarly, when applying intra prediction, the prediction process of adjacent blocks might be different causing discontinuities at the block boundaries of the prediction signal.

Two approaches to reduce blocking artifacts are post-filtering and in-loop filtering. Post-filtering is not specified by the video coding standard and can be performed, e.g., in the display buffer. The implementer has a freedom to design an algorithm driven by application-specific requirements. In-loop filters operate within the encoding and decoding loops. Therefore, they need to be normative to avoid drift between the encoder and decoder.

The HEVC draft standard defines two in-loop filters that can be applied sequentially to the reconstructed picture. The first one is the deblocking filter and the second one is the sample adaptive offset filter (SAO) that are currently included into the main profile. This paper describes the first of these two in-loop filters, the deblocking filter. Depending on the configuration, SAO can be applied to the output of the deblocking filtering process.

Manuscript received April 15, 2012; revised July 19, 2012; accepted August 20, 2012. Date of publication October 5, 2012; date of current version January 8, 2013. This paper was recommended by Associate Editor B. Pesquet-Popescu.

A. Norkin and K. Andersson are with Ericsson Research, Stockholm 164 89, Sweden (e-mail: andrey.norkin@ericsson.com; kenneth.r.andersson@ericsson.com).

G. Bjøntegaard and A. Fuldseth are with Cisco Systems Norway, Lysaker 1366, Norway (e-mail: arilfuld@cisco.com; gbjonteg@cisco.com).

M. Narroschke is with the Panasonic Research and Development Center, Langen 63255, Germany (e-mail: matthias.narroschke@eu.panasonic.com).

M. Ikeda is with the Technology Development Group, Sony Corporation, Tokyo 141-8610, Japan (e-mail: masaru.ikeda@jp.sony.com).

M. Zhou is with the Systems and Applications Research and Development Center, Texas Instruments, Inc., Dallas, TX 75243 USA (e-mail: zhou@ti.com).

G. Van der Auwera is with Qualcomm Technologies, Inc., San Diego, CA 92121 USA (e-mail: geertv@qualcomm.com).

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.2012.2223053

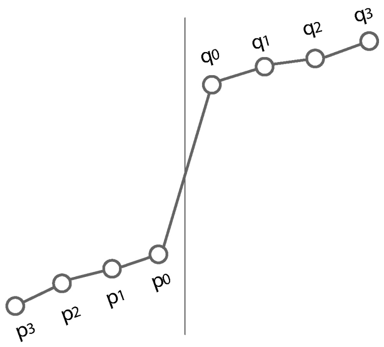


Fig. 1. 1-D example of block boundary with blocking artifact.

The deblocking filter in HEVC has been designed to improve the subjective quality while reducing the complexity. The latter consideration is important since the deblocking filter of the H.264/AVC standard [2], [4] constitutes a significant part of the decoder complexity. As a result, the HEVC deblocking filter is less complex as compared to the H.264/AVC deblocking filter, while still having the capability to improve the subjective and objective quality.

Another aspect that received significant attention in the HEVC deblocking filter design is its suitability for parallel processing. Deblocking in HEVC has been designed in a way to prevent spatial dependences across the picture, which, together with other design features, enables easy parallelization on multiple cores.

In the following sections, an overview of the HEVC deblocking filter design is provided. For more details, the reader is referred to [1], and to the corresponding input contributions to the JCT-VC. The initial deblocking filter design was adopted from [5]. The filtering decisions and operations, as described in Sections II and III, mainly result from the adoption of the contributions in [6] and [7]. For sequence and picture-level adaptivity (see Section IV) the main adopted contribution is [12]. The parallel processing capabilities, as described in Section V, mainly result from adoption of [8]–[10].

II. Filtering Decisions

A. Block Boundaries for Deblocking

As mentioned above, independent coding of blocks creates discontinuities at block boundaries. An example of a block boundary with a blocking artifact is shown in Fig. 1. Blocking artifacts can easily be noticed by the human visual system when the signal on both sides of the block boundary is relatively smooth, but are more difficult to notice when the signal shows high variation. Furthermore, if the original signal across the block boundary is subjected to higher variations, then it is difficult to say whether changes in the reconstructed signal across the block boundary are caused by coding or belong to the original signal.

The main difficulty when designing a deblocking filter is to decide whether or not to filter a particular block boundary, and to decide on the filtering strength to be applied. Excessive filtering may lead to unnecessary smoothing of the picture details, whereas lack of filtering may leave blocking artifacts

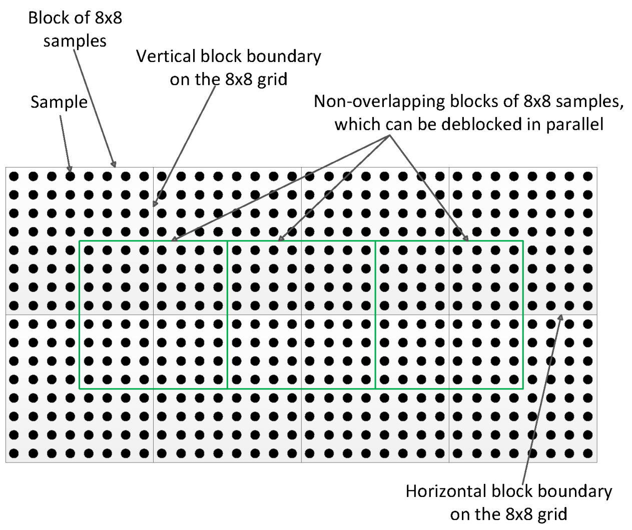

Block of 8x8 samples
Vertical block boundary on the 8x8 grid
Sample
Non-overlapping blocks of 8x8 samples, which can be deblocked in parallel
Horizontal block boundary on the 8x8 grid

Fig. 2. Illustration of picture samples and horizontal and vertical block boundaries on the 8×8 grid, and the nonoverlapping blocks of the 8×8 samples, which can be deblocked in parallel.

that would reduce the subjective quality. Deciding whether to filter a block boundary should, therefore, depend on the characteristics of the reconstructed pixel values on both sides of that block boundary, and on the coded parameters indicating whether it is likely that a blocking artifact has been created by the coding process.

Filtering decisions that are elaborated in the following subsections are made separately for each boundary of four-sample length that lies on the grid dividing the picture into blocks of 8×8 samples. Block boundaries on the 8×8 grid are illustrated in Fig. 2. Only boundaries on the 8×8 grid, which were either prediction unit or transform unit boundaries, are subjected to deblocking.

Deblocking is, therefore, performed on a four-sample part of a block boundary when all of the following three criteria are true: 1) the block boundary is a prediction unit or transform unit boundary; 2) the boundary strength is greater than zero; and 3) variation of signal on both sides of a block boundary is below a specified threshold (see Fig. 4). When certain additional conditions (Section II-D) hold, a strong filter is applied on the block edge instead of the normal deblocking filter.

B. Boundary Strength (Bs) and Edge-Level Adaptivity

Boundary strength (Bs) is calculated for boundaries that are either prediction unit boundaries or transform unit boundaries. The boundary strength can take one of the three possible values: 0, 1, and 2. The definition of the Bs is shown in Table I.

For the luma component, only block boundaries with Bs values equal to one or two are filtered. This implies that there is typically no filtering within the static areas. This helps avoid multiple subsequent filtering of the same areas where pixels are copied from one picture to another with a residual equal to zero, which can cause oversmoothing. The difference in filtering operations between Bs equal to one and Bs equal to two is described in Section III-D.

In the case of the chroma components, only boundaries with Bs equal to two are filtered. This implies that only those block

TABLE I
DEFINITION OF BS VALUES FOR THE BOUNDARY BETWEEN TWO NEIGHBORING LUMA BLOCKS

Conditions	*Bs*
At least one of the blocks is Intra	2
At least one of the blocks has non-zero coded residual coefficient and boundary is a transform boundary	1
Absolute differences between corresponding spatial motion vector components of the two blocks are >= 1 in units of integer pixels	1
Motion-compensated prediction for the two blocks refers to different reference pictures or the number of motion vectors is different for the two blocks	1
Otherwise	0

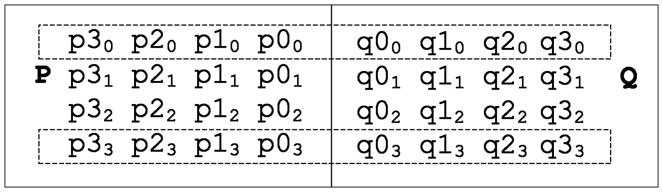


Fig. 3. Four-pixel long vertical block boundary formed by the adjacent blocks P and Q. Deblocking decisions are based on lines marked with the dashed line (lines 0 and 3).

boundaries are filtered where at least one of the two adjacent blocks is intra predicted.

C. Local Adaptivity and Filtering Decisions

If Bs is greater than zero, additional conditions are checked for luma block edges to determine whether the deblocking filtering should be applied to the block boundary or not.

As we can see from Fig. 1, a blocking artifact is characterized by low spatial activity on both sides of the block boundary, whereas there is discontinuity at the block boundary. Therefore, for each block boundary of four-sample length on the 8×8 sample grid that satisfies the conditions described above, the following condition is checked to decide whether the deblocking filtering is applied (see Fig. 3):

$$| p_{2,0} - 2p_{1,0} + p_{0,0}| + |p_{2,3} - 2p_{1,3} + p_{0,3}| + \\ |q_{2,0} - 2q_{1,0} + q_{0,0}| + |q_{2,3} - 2q_{1,3} + q_{0,3}| > \beta \quad (1)$$

where threshold β depends on the quantization parameter QP that is used to adjust the quantization step for quantizing the prediction error coefficients [5]. The threshold is derived from a table that has a piecewise linear dependence with values of QP, as described in Section IV. Equation (1) evaluates how much signal on both sides of the block boundary deviates from a straight line (a constant level signal or a ramp). Only the first and fourth lines in a block boundary of length 4 are evaluated to reduce complexity. The example in Fig. 3 and equations in this and the following sections only consider the case of a vertical block boundary for the sake of brevity. The example can easily be extended to deblocking of horizontal block boundaries by rotating the figure by 90° in the clockwise direction and changing row and column subscript indices in the equations.

For block boundaries with an associate Bs greater than zero, and for which (1) holds, deblocking filtering is performed. There are two deblocking filtering modes in HEVC, namely, a normal filtering mode and strong filtering mode. For each block boundary of four samples in length, the deblocking filter switches between the normal and the strong filtering mode based on the local signal characteristics.

D. Decisions Between Normal and Strong Deblocking

Whether to apply strong or normal deblocking is also determined based on the first and the fourth lines across the block boundary of four samples (see Fig. 3). The following expressions using information from lines $i = 0$ and $i = 3$ are evaluated to make a decision between the normal and the strong filtering [5], [9]:

$$|p_{2,i} - 2p_{1,i} + p_{0,i}| + |q_{2,i} - 2q_{1,i} + q_{0,i}| < \beta/8 \quad (2)$$

$$|p_{3,i} - p_{0,i}| + |q_{0,i} - q_{3,i}| < \beta/8 \quad (3)$$

$$|p_{0,i} - q_{0,i}| < 2.5t_C. \quad (4)$$

The threshold parameter t_C depends on QP and is defined by a table (see Section IV for details). If (2), (3), and (4) hold for both lines 0 and 3, the strong filtering is applied to the block boundary. Otherwise, normal filtering is applied. Condition (2) checks that there is a low spatial activity on the side of block boundary [similar to (1) but using a lower threshold], condition (3) checks that the signal on the sides of the block boundary is flat, and condition (4) checks that the difference in intensities of samples on two sides of the block boundary does not exceed the threshold, which is a multiple of the clipping value t_C(QP) dependent on QP (see Section V). The sequence of deblocking filtering decisions described above is summarized in Fig. 4.

E. Deblocking Decisions in Normal Filtering Mode

Normal filtering has two modes differing in the number of pixels being modified on each side of the block boundary. One of the two modes is selected for each boundary based on the following two conditions [6]:

$$|p_{2,0} - 2p_{1,0} + p_{0,0}| + |p_{2,3} - 2p_{1,3} + p_{0,3}| < 3/16\beta \quad (5)$$

$$|q_{2,0} - 2q_{1,0} + q_{0,0}| + |q_{2,3} - 2q_{1,3} + q_{0,3}| < 3/16\beta. \quad (6)$$

If (5) is true, the two nearest pixels to the block boundary can be modified in block P. Otherwise, only the nearest pixel in block P can be modified. Similarly, if (6) holds, the two nearest pixels to the block boundary can be modified in block Q. Otherwise, only the nearest pixel can be modified. The thresholds used in (5) and (6) are also dependent on quantization parameter QP since they are multiples of threshold β. The values of the thresholds used in (5) and (6) are less than the value of the threshold in (1), but greater than the value of the threshold in (3). This means that the longer (stronger) filtering is allowed for the block boundaries that have lower spatial activity on the sides of the boundaries.

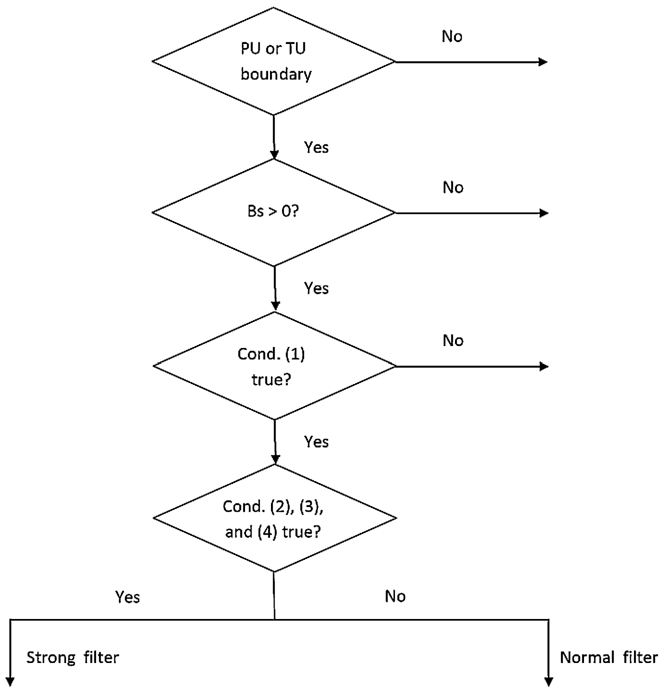


Fig. 4. Decisions for each segment of block boundary of four samples in length lying on 8×8 block boundary. PU: prediction unit. TU: transform unit.

III. Filtering Operations

A. *Normal Filtering Operations*

When a picture contains an inclined surface (or linear ramp signal) that crosses a block boundary, the filter will be active. In these cases, the normal deblocking filter operations should not modify the signal.

In the normal filtering mode for a segment of four lines (see Fig. 3), filtering operations are applied for each line. In the following, the second indices of pixels, indicating the line number, are omitted for brevity.

The filtered pixel values p_0' and q_0' are calculated for each line across the block boundary as follows:

$$p_0' = p_0 + \Delta_0 \tag{7}$$

$$q_0' = q_0 - \Delta_0 \tag{8}$$

where the value of Δ_0 is obtained by clipping δ_0

$$\delta_0 = (9(q_0 - p_0) - 3(q_1 - p_1) + 8) >> 4. \tag{9}$$

The clipping operation is described in Section III-D. Neglecting the clipping operation, the impulse response of this filter is (3 7 9 −3)/16. The offset value δ_0 corresponds to the deviation of the signal at the sides of the block boundary from a perfect ramp. The offset is zero if the signal across the block boundary forms a ramp.

Furthermore, the deblocking filtering is applied to the row or column of samples across the block boundary, if and only if the following expression holds:

$$|\delta_0| < 10t_C. \tag{10}$$

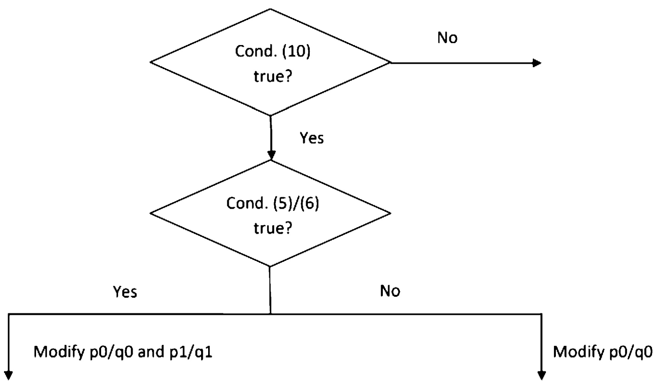


Fig. 5. Decisions for normal filter that are applied to each line of four-sample segment.

If this condition does not hold, it is likely that the change of the signal on both sides of the block boundary is caused by a natural edge and not by a blocking artifact.

If (5) is true, the modified value p_1' in each line across the block boundary is obtained by

$$p_1' = p_1 + \Delta_{p1}. \tag{11}$$

Similarly, if (6) is true, then q_1' is calculated as

$$q_1' = q_1 + \Delta_{q1} \tag{12}$$

where the offset values Δ_{p1} and Δ_{q1} are obtained by clipping the corresponding δ_{p1} and δ_{q1} values, which are calculated as

$$\delta_{p1} = (((p_2 + p_0 + 1) >> 1) - p_1 + \Delta_0) >> 1 \tag{13}$$

$$\delta_{q1} = (((q_2 + q_0 + 1) >> 1) - q_1 - \Delta_0) >> 1. \tag{14}$$

Neglecting the clipping operation, the impulse response of the filter that corresponds to modification of the pixel at position p_1 is (8 19 −1 9 −3)/32.

The sequence of filtering decisions for each line of pixels in the normal filtering mode is summarized in Fig. 5.

B. *Strong Filtering Operations*

The strong filter affects more pixels on each side of the block boundary. Modifications of three pixels on each side of the block boundary are similar to strong filtering in H.264/AVC [4]. The offset values Δ_{0s}, Δ_{1s}, and Δ_{2s} are added to pixels p_0, p_1, and p_2, respectively, after clipping of the following δ_{0s}, δ_{1s}, and δ_{2s} values:

$$\delta_{0s} = (p_2 + 2p_1 - 6p_0 + 2q_0 + q_1 + 4) >> 3 \tag{15}$$

$$\delta_{1s} = (p_2 - 3p_1 + p_0 + q_0 + 2) >> 2 \tag{16}$$

$$\delta_{2s} = (2p_3 - 5p_2 + p_1 + p_0 + q_0 + 4) >> 3. \tag{17}$$

The offset values for modification of pixels q_0, q_1, and q_2 are calculated by exchanging q and p in (15), (16), and (17). Impulse responses of the filters that correspond to modification of pixels p_0, p_1, and p_2 are (1 2 2 2 1)/8, (1 1 1 1)/4, and (2 3 1 1 1)/8, respectively, if the clipping operation is neglected.

C. Chroma Deblocking

As mentioned previously, chroma deblocking is only performed when Bs is equal to two. In this case, no further deblocking decisions are done. Only pixels p_0 and q_0 are modified as in (7) and (8). The deblocking is performed with the Δ_c value, which is obtained by clipping the following δ_c offset value:

$$\delta_c = (((p_0 - q_0) << 2) + p_1 - q_1 + 4) >> 3 \quad (18)$$

which corresponds to filtering by the filter with the impulse response of (1 4 4 –1)/8.

D. Clipping

To prevent excessive blurriness, deblocking filtering is done on a signal after QP-dependent clipping. Clipping is applied to the δ values after their calculation and before modification of the pixel values. The Δ values used in filtering are obtained by clipping the δ values to the range $-c$ to c as in (19). Clipping provides more adaptivity to deblocking filtering. The clipping is applied by performing the following operations:

$$\Delta = \mathrm{Min}(\mathrm{Max}(-c, \delta), c) \quad (19)$$

where the value of c is equal to $t_C(n)$ for p_0 and q_0, and $t_C(n)/2$ for p_1 and q_1 in the case of normal filtering. In the case of strong filtering, c is set equal to $2t_C(n)$. Variable n is equal to QP when both blocks adjacent to the boundary are inter predicted and QP+2, if one of the blocks is intra predicted (Bs = 2).

The dependence of the parameter t_C on QP is illustrated in Fig. 7. The blocking artifacts strength is generally greater for intra predicted blocks. Therefore, larger modifications of pixel values are allowed for intra-blocks than those for inter-blocks by using the clipping value t_C(QP + 2) for block boundaries with Bs equal to 2.

The filtered pixel values p'_0, q'_0, p'_1 and q'_1 for normal filtering and p_0' and q_0' for chroma deblocking are also clipped to stay in the range defined by the bit depth N

$$p'' = \mathrm{Min}(\mathrm{Max}(0, p'), 2^N - 1). \quad (20)$$

IV. Sequence and Picture Level Adaptivity

Since different video sequences have different characteristics, deblocking strength can be adjusted on a sequence and even on a picture basis.

As mentioned earlier, the main sources of blocking artifacts are block transforms and quantization. Therefore, blocking artifact severity depends, to a large extent, on the quantization parameter QP. Therefore, in the deblocking filtering decisions, the QP value is taken into account. Thresholds β and t_C depend on the average QP value of two neighboring blocks with common block edge [13] and are typically stored in corresponding tables. The dependence of these parameters on QP is shown in Figs. 6 and 7.

The parameter β controls what edges are filtered, controls the selection between the normal and strong filter, and controls how many pixels from the block boundary are modified in the normal filtering operation. One can observe that the value of β increases with QP. Therefore, deblocking is enabled more frequently at high QP values compared to low QP values, high QP values correspond to coarse, and low QP values correspond to fine quantization. One can also see that the deblocking operation is effectively disabled for low QP values by setting one or both of β and t_C to zero.

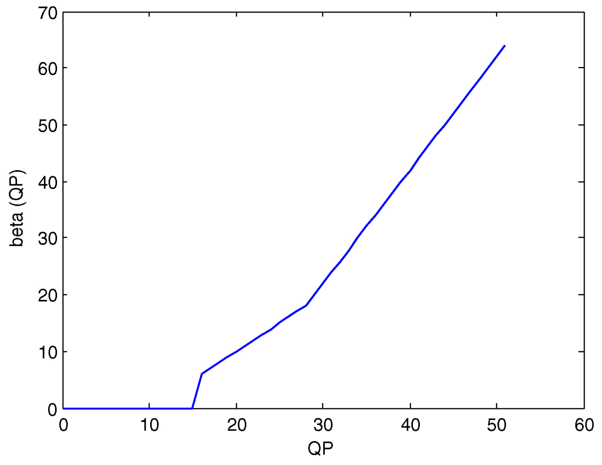


Fig. 6. Dependence of β on QP.

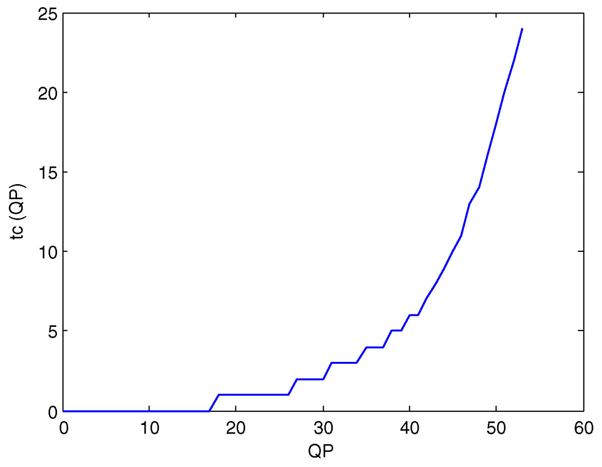


Fig. 7. Dependence of t_C on QP.

The parameter t_C controls the selection between the normal and strong filter and determines the maximum absolute value of modifications that are allowed for the pixel values for a certain QP for both normal and strong filtering operations. This helps adaptively limit the amount of blurriness introduced by the deblocking filtering.

The deblocking parameters t_C and β provide adaptivity according to the QP and prediction type. However, different sequences or parts of the same sequence may have different characteristics. It may be important for content providers to change the amount of deblocking filtering on the sequence or even on a slice or picture basis. Therefore, deblocking adjustment parameters can be sent in the slice header or picture parameters set (PPS) to control the amount of deblocking filtering applied. The corresponding parameters are tc_offset_div2 and beta_offset_div2 [12]. These parameters specify the offsets (divided by two) that are added to the QP value before determining the β and t_C values. The parameter beta_offset_div2 adjusts the number of pixels to which the deblocking filtering is applied, whereas parameter tc_offset_div2 adjusts the amount of filtering that can be applied to those pixels, as well as detection of natural edges.

V. Computational Complexity and Parallelism

Compared to H.264/AVC, the complexity of the deblocking filter has been significantly reduced in HEVC due to several factors that are described in this section. Performing deblocking on a grid of 8×8 samples as opposed to a grid of 4×4 samples in H.264/AVC reduces the number of deblocking operations by a factor of two. Deblocking of the chroma component in the 4:2:0 format is also performed on the grid of 8×8 samples. Furthermore, the chroma blocks are filtered only in cases when one of the adjacent blocks is intra predicted. This decreases the amount of chroma filtering further for inter-coded slices. Filtering on an 8×8 sample grid may potentially lead to reduction in subjective quality. However, since the number of 4×4 blocks in the picture for HEVC is generally lower than that for H.264/AVC and 4×4 blocks in HEVC are usually used in the areas with higher temporal or spatial activity, applying filtering on an 8×8 sample grid is a tradeoff between computational complexity and subjective quality.

Another source of complexity reduction in HEVC deblocking is related to the transform and prediction unit size. In H.264/AVC, the largest transform size is 8×8, whereas the largest prediction unit size is 16×16 samples, i.e., a macroblock. However, in HEVC the largest transform size is 32×32 and the largest prediction unit size is 64×64 samples. This additionally reduces the average amount of operations (although not necessarily for the worst case) since deblocking is never performed inside these large blocks.

Deblocking in HEVC has been designed to prevent spatial dependences of the deblocking process across the picture. There is no overlap between the filtering operations for one block edge, which can modify at most three pixels from the block edge, and the filtering decisions for the neighboring parallel block edge, which involves at most four pixels from the block edge. Therefore, any vertical block edge in the picture can be deblocked in parallel to any other vertical edge. The same holds for horizontal edges. Note, however, that sample values modified by deblocking of vertical block boundaries are used as the input for deblocking of horizontal block boundaries.

For CTU-based processing, the deblocking in HEVC can be performed on an 8×8 block basis. A picture can be divided into nonoverlapping blocks of 8×8 samples (see Fig. 2). Each of those blocks contains all data required for its deblocking. Consequently, deblocking can be performed independently for each of those blocks of 8×8 samples. This makes the HEVC deblocking easily parallelizable for any degree of parallelism by simply replicating the same 8×8 deblocking logic.

The order of filtering of vertical and horizontal edges in HEVC is also different from that in H.264/AVC. In H.264/AVC, deblocking is performed on a macroblock basis. However, the deblocking in HEVC is first applied to all vertical edges and then to all horizontal edges in the picture. Consequently, the order of vertical and horizontal filtering for each of the 8×8 blocks, as shown in Fig. 1, is exactly the same irrespective of the block position. Moreover, the order of filtering the block boundaries does not change with different orders of CTU decoding, which reduces hardware complexity.

As HEVC deblocking is independent for each 8×8 block, an encoder or decoder has the option of deblocking inner blocks of a slice or a tile [11] only and leaving the slice or tile boundary blocks out of the deblocking process in the first pass. In the second pass, an encoder or decoder can go back and perform deblocking along the slice or tile boundaries as a patch. Such an option basically breaks in-loop filter (deblocking and SAO) dependence across the slice or tile boundaries and is very useful for parallel processing on multicore platforms when the in-loop filters are enabled across slice or tile boundaries. By taking advantage of this property, each core can process a portion of a picture in parallel by skipping the in-loop filtering for the slice or tile boundary blocks. After the entire picture is processed, a separate core can load the slice or tile boundary blocks back and conduct a patch for in-loop filters along the slice or tile boundaries to complete in-loop filtering for the picture. Therefore, there is no need for de-coupling the entire in-loop filtering process from the rest of the coding process, that significantly improves the throughput and greatly reduces memory bandwidth requirements for multi-core based HEVC implementations. This is not possible with the H.264/AVC deblocking filter design, in which the deblocking has to be decoupled if multiple slices are processed in parallel and deblocking across slice boundaries is enabled.

Another advantage of the highly parallelizable HEVC deblocking filter is that it provides enough cycle margins to enable a combination of the deblocking filter and SAO in the same building block in hardware implementations. In a typical architecture, the HEVC deblocking filter only consumes from 84 to 88 cycles per 16×16 block, which is less than half of the typical 200 cycles per 16×16 block cycle budget (for a 1080p@120 f/s video running at 250 MHz clock rate) [14]. Combining the deblocking filter and SAO in the same building block is beneficial in terms of hardware area cost, since SAO and deblocking can share the same memory interface, in contrast to having separate building blocks and memory interfaces for SAO and deblocking.

Since deblocking in HEVC is computationally less intensive and more parallelizable than in H.264/AVC, it can be said that the HEVC deblocking is much less of a bottleneck when implementing a video decoder. The deblocking in HEVC is a better tradeoff among coding efficiency (i.e., subjective and objective quality), throughput, and implementation complexity when compared to the H.264/AVC design.

VI. Results

This section demonstrates the objective and subjective impact of deblocking filtering. Tables II–V show the BD-rate resulting from disabling the deblocking filtering for various configurations used in the HEVC standardization [16]. These configurations are all-intra where only intra prediction is used,

TABLE II
AVERAGE BIT RATE INCREASE AT THE SAME QUALITY BY DISABLING THE DEBLOCKING FILTER FOR THE ALL-INTRA CONFIGURATION

	All Intra Main		
	Y	U	V
Class A	1.9%	4.2%	3.7%
Class B	1.7%	4.5%	5.1%
Class C	0.9%	3.7%	4.3%
Class D	0.7%	3.0%	3.4%
Class E	2.1%	7.4%	8.8%
Class F	0.6%	1.9%	1.8%
Overall	1.3%	4.0%	4.4%

TABLE III
AVERAGE BIT RATE INCREASE AT THE SAME QUALITY BY DISABLING THE DEBLOCKING FILTER FOR THE RANDOM-ACCESS CONFIGURATION

	Random Access Main		
	Y	U	V
Class A	3.6%	2.1%	1.9%
Class B	3.2%	1.9%	1.9%
Class C	2.1%	1.5%	1.9%
Class D	1.5%	1.1%	1.2%
Class F	1.2%	0.9%	0.9%
Overall	2.6%	1.6%	1.7%

TABLE IV
AVERAGE BIT RATE INCREASE AT THE SAME QUALITY BY DISABLING THE DEBLOCKING FILTER FOR THE LOW-DELAY CONFIGURATION

	Low delay B Main		
	Y	U	V
Class B	3.3%	1.3%	1.6%
Class C	2.1%	1.5%	1.5%
Class D	1.3%	0.8%	1.6%
Class E	3.8%	5.9%	7.3%
Class F	1.3%	0.4%	0.0%
Overall	2.4%	1.8%	2.1%

random-access that uses intra pictures over certain time intervals and hierarchical-B coding structure, and two low-delay configurations that have only one intra picture, and motion-compensated prediction uses only temporally preceding pictures. The low-delay P configuration does not use bidirectional motion-compensated prediction. The BD rate is used in the HEVC standardization as a measure for the average bit rate reduction at the same mean squared error [15]. As a positive number in the tables indicates an increased bit rate at the same quality, the HEVC deblocking filter leads to an average bit rate reduction of 1.3%–3.3% at the same quality, dependent on the configuration. For certain sequences, more than 6% bit rate reduction is achieved. Figs. 8 and 9 compare the visual quality of coded sequences when the deblocking is turned on with the configuration with the deblocking turned off. Fig. 8 shows a cropped part of a frame from the *Basketball Drive* sequence (1080p@50 f/s) coded in random access configuration at QP 32, where the deblocking filtering was applied and the frame where the deblocking was turned off. Fig. 9 provides a comparison for a sequence *Kristen and Sara* (720p@60 f/s) coded in low-delay B configuration at QP 37. It can be seen that the deblocking filter effectively attenuates the blocking artifacts. The HEVC reference software HM6.0 was used in all experiments.

TABLE V
AVERAGE BIT RATE INCREASE AT THE SAME QUALITY BY DISABLING THE DEBLOCKING FILTER FOR THE LOW-DELAY P-FRAME CONFIGURATION

	Low Delay P Main		
	Y	U	V
Class B	4.9%	2.5%	2.7%
Class C	2.6%	1.5%	2.1%
Class D	1.6%	1.4%	0.8%
Class E	6.2%	7.8%	9.0%
Class F	1.7%	1.0%	0.4%
Overall	3.3%	2.5%	2.7%

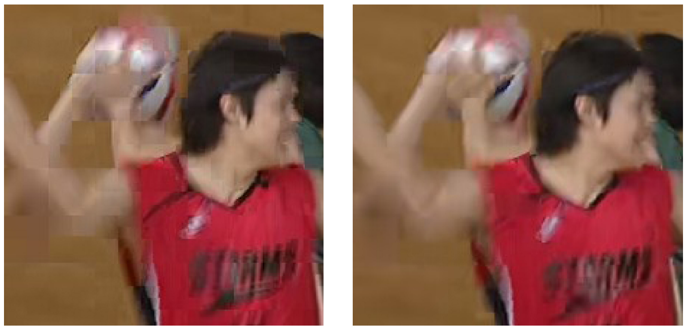


Fig. 8. *Basketball Drive* sequence coded in random access configuration at QP32. (a) Deblocking turned off. (b) Deblocking turned on.

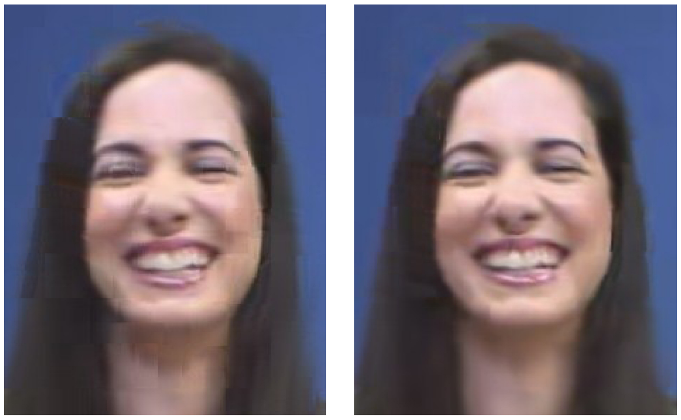


Fig. 9. *Kristen and Sara* sequence coded in low-delay B configuration at QP37. (a) Deblocking turned off. (b) Deblocking turned on.

VII. CONCLUSION

The deblocking filter in the upcoming HEVC standard improves both the subjective and objective quality of the coded video sequences, while being less computationally expensive than the deblocking filter in H.264/AVC. The decrease in computational complexity is achieved by reconsidering a number of tools. The HEVC deblocking filtering operations can also be easily performed in parallel on multiple processors, which is important for coding and decoding higher resolution video sequences.

REFERENCES

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 8*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVTC-J1003, Joint Collaborative Team on Video Coding (JCTVC), Stockholm, Sweden, Jul. 2012.

[2] ITU-T and ISO/IEC JCT 1, *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), May 2003 (and subsequent editions).

[3] T. Wedi and H. G. Musmann, "Motion and aliasing compensated prediction for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 577–586, Jul. 2003.

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

[5] K. Ugur, K. R. Andersson, and A. Fuldseth, *Video Coding Technology Proposal by Tandberg, Nokia, and Ericsson*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-A119, Joint Collaborative Team on Video Coding (JCTVC), Dresden, Germany, Apr. 2010.

[6] A. Norkin, K. Andersson, R. Sjöberg, Q. Huang, J. An, X. Guo, and S. Lei, *CE12: Ericsson's and MediaTek's Deblocking Filter*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-F118, Joint Collaborative Team on Video Coding (JCTVC), Turin, Italy, Jul. 2011.

[7] M. Ikeda and T. Suzuki, *Non-CE10: Introduction of Strong Filter Clipping in Deblocking Filter*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-H0275, Joint Collaborative Team on Video Coding (JCTVC), San Jose, CA, Feb. 2012.

[8] M. Ikeda, J. Tanaka, and T. Suzuki, *CE12 Subset2: Parallel Deblocking Filter*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-E181, Joint Collaborative Team on Video Coding (JCTVC), Geneva, Switzerland, Mar. 2011.

[9] M. Narroschke, S. Esenlik, and T. Wedi, *CE12 Subtest 1: Results for Modified Decisions for Deblocking*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-G590, Joint Collaborative Team on Video Coding (JCTVC), Geneva, Switzerland, Nov. 2011.

[10] A. Norkin, *CE10.3: Deblocking Filter Simplifications: Bs Computation and Strong Filtering Decision*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-H0473, Joint Collaborative Team on Video Coding (JCTVC), San Jose, CA, Feb. 2012.

[11] A. Fuldseth, M. Horowitz, S. Xu, A. Segall, and M. Zhou, *Tiles*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-F335, Joint Collaborative Team on Video Coding (JCTVC), Turin, Italy, Jul. 2011.

[12] T. Yamakage, S. Asaka, T. Chujoh, M. Karczewicz, and I. S. Chong, *CE12: Deblocking Filter Parameter Adjustment in Slice Level*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-G174, Joint Collaborative Team on Video Coding (JCTVC), Geneva, Switzerland, Nov. 2011.

[13] G. Van der Auwera, X. Wang, M. Karczewicz, M. Narroschke, A. Kotra, and T. Wedi (Panasonic), *Support of Varying QP in Deblocking*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-G1031, Joint Collaborative Team on Video Coding (JCTVC), Geneva, Switzerland, Nov. 2011.

[14] M. Zhou, O. Sezer, and V. Sze, *CE12 Subset 2: Test Results and Architectural Study on De-Blocking Filter Without Parallel on/off Filter Decision*, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-G088, Joint Collaborative Team on Video Coding (JCTVC), Geneva, Switzerland, Nov. 2011.

[15] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, ITU-T-T SG16 document VCEG-M33, Joint Collaborative Team on Video Coding (JCTVC), 2001.

[16] F. Bossen, *Common Test Conditions*, JCTVC-H1100, Joint Collaborative Team on Video Coding (JCTVC), San Jose, CA, 2012.

Andrey Norkin received the M.Sc. degree in computer engineering from Ural State Technical University, Yekaterinburg, Russia, in 2001, and the Doctor of Science degree in signal processing from the Tampere University of Technology, Tampere, Finland, in 2007.

From 2002 to 2007, he was a Researcher with the Institute of Signal Processing, Tampere University of Technology. He is currently a Senior Researcher and Work Package Leader with Ericsson Research, Stockholm, Sweden. He has been an active participant in the ITU-T/ISO/IEC Joint Collaborative Team for Video Coding and a Coordinator of the Core Experiment on deblocking filtering in HEVC standardization. His current research interests include video compression, 3-D video, error-resilient coding, and image processing.

Gisle Bjøntegaard received the Dr. Philos degree in physics from the University of Oslo, Oslo, Norway, in 1974.

From 1974 to 1996, he was a Senior Scientist with Telenor Research and Development, Oslo. His areas of work included radio link network design, reflector antenna design and construction, and digital signal processing. From 1980 to 1996, he was mainly involved in the development of digital video compression methods. He has contributed actively in development of the ITU video standards H.261, H.262, H,263, H.264, and ISO/IEC MPEG2 and MPEG4. From 1996 to 2002, he was a Group Manager with Telenor Broadband Services, Oslo, engaged in design of point-to-point satellite communication and development of a digital satellite TV platform. He has produced numerous contributions toward the development of the ITU-T standards H.263 and H.264. Since 2002, he has been a Principal Scientist with Tandberg Telecom, Lysaker, Norway, working with video coding development and implementation. Since 2006, he has worked on further improvement of digital video coding and is currently taking part in the definition of HEVC, developed jointly between ITU and ISO. In 2010, Tandberg Telecom was acquired by Cisco Systems and he was promoted to a Cisco Fellow and is presently with Cisco Systems, Norway.

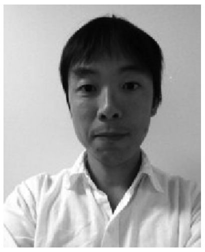
Arild Fuldseth received the B.Sc. degree from the Norwegian Institute of Technology, Trondheim, Norway, in 1988, and the M.Sc. and Ph.D. degrees in signal processing from North Carolina State University, Raleigh, and the Norwegian University of Science and Technology, Trondheim, in 1989 and 1997, respectively.

From 1989 to 1994, he was a Research Scientist with SINTEF, Trondheim. From 1997 to 2002, he was a Manager with the Signal Processing Group, Fast Search and Transfer, Oslo, Norway. Since 2002, he has been with Tandberg Telecom, Oslo (now part of Cisco Systems, Oslo), where he is currently a Principal Engineer working on video compression technology.

Matthias Narroschke received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the University of Hannover, Hannover, Germany, in 2001 and 2008, respectively.

From 2001 to 2007, he was a Research Engineer and Teaching Assistant with the Institut für Informationsverarbeitung, University of Hannover. In 2003, he became the Ober-Ingenieur. Since 2007, he has been with the Panasonic Research and Development Center, Langen, Germany, where he is currently a Principal Engineer. In 2008, he became a Guest Lecturer with the University of Hannover. He has several patents pending in the area of video coding, mostly in cooperation with Panasonic. His current research interests include video coding standardization activities and the development of future video coding schemes.

Dr. Narroschke is an active contributor to the Motion Picture Experts Group of ISO/IEC SC29, and to the video coding experts group of ITU. He was the recipient of the Robert-Bosch-Prize for the Best Dipl.-Ing. Degree in Electrical Engineering in 2001.



Masaru Ikeda received the B.S. and M.S. degrees in electrical and communication engineering from Tohoku University, Sendai, Japan, in 1999 and 2001, respectively.

From 2001 to 2008, he was a Research Engineer with Sony Research Center and worked on computer vision. He is currently with the Technology Development Group, Sony Corporation, Tokyo, Japan. His current research interests include video compression, 3-D video compression, and image processing.

Kenneth Andersson received the M.Sc. degree in computer science and engineering from Luleå University, Luleå, Sweden, in 1995, and the Ph.D. degree from Linköping University, Linköping, Sweden, in 2003.

He has been with Ericsson Research, Stockholm, Sweden, since 1994, where he has worked on speech coding and is currently a Senior Researcher working on video coding. His current research interests include image and video signal processing and video coding.

Minhua Zhou received the B.E. degree in electronic engineering and the M.E. degree in communication and electronic systems from Shanghai Jiao Tong University, Shanghai, China, in 1987 and 1990, respectively, and the Ph.D. degree in electronic engineering from Technical University, Braunschweig, Germany, in 1997.

From 1993 to 1998, he was a Researcher with the Heinrich-Hertz Institute, Berlin, Germany. He is currently a Research Manager of video coding technology with the Systems and Applications Research and Development Center, Texas Instruments, Inc., Dallas. His current research interests include video compression, video pre- and postprocessing, end-to-end video quality, joint algorithm and architecture optimization, and 3-D video.

Dr. Zhou was the recipient of the Rudolf-Urtel Prize in 1997 from the German Society for Film and Television Technologies in recognition of his Ph.D. thesis work on Optimization of MPEG-2 Video Encoding.

Geert Van der Auwera received the Ph.D. degree in electrical engineering from Arizona State University, Tempe, in 2007, and the Belgian MSEE degree from Vrije Universiteit Brussel, Brussels, Belgium, in 1997.

He is currently with Qualcomm Technologies, Inc., San Diego, CA, where he is actively contributing to the JCT-VC standardization effort on high-efficiency video coding. Until January 2011, he was with Samsung Electronics, Irvine, CA. Until December 2004, he was a Scientific Advisor with IWT-Flanders, Institute for the Promotion of Innovation by Science and Technology, Flanders, Belgium. In 2000, he joined IWT-Flanders after researching wavelet video coding at IMEC's Electronics and Information Processing Department, Brussels. His current research interest include video coding, video traffic and quality characterization, and video streaming mechanisms and protocols.

Dr. Van der Auwera received the Barco and IBM Prizes in 1998 for his thesis on motion estimation in the wavelet domain from the Fund for Scientific Research of Flanders, Belgium.