HEVC deblocking filtering and decisions

Andrey Norkin[a], Kenneth Andersson[a], Arild Fuldseth[b], Gisle Bjøntegaard[b]
[a]Ericsson Research, Stockholm, Sweden;
[b]Cisco Systems, Oslo, Norway

ABSTRACT

The emerging High Efficiency Video Coding (HEVC) standard uses a block-based coding scheme, which may cause blocking artifacts, especially at lower bitrates. An adaptive in-loop deblocking filter is used in the standard to reduce visible artifacts at block boundaries. The deblocking filter detects artifacts at the block boundaries and attenuates them by applying a selected filter. This paper will present deblocking decisions and filtering operations that are used in HEVC.

Keywords: deblocking, video coding, video processing, block-based coding, filtering.

1. INTRODUCTION

High efficiency video coding (HEVC) is a new video coding standard currently being developed jointly by ITU-T SG 16 Q.6 Video Coding Experts Group (VCEG) and by ISO/IEC JTC 1/SC 29/WG 11 Moving Picture Experts Group (MPEG) in a framework of a joint collaborative team on video coding (JCT-VC). The first version of HEVC is planned to be finalized in January 2013, and will be followed by development of scalable and 3D extensions in the following years. Similar to the previous video coding standards such as H.264/AVC, HEVC is based on block-based coding scheme, using rectangular blocks for prediction and square blocks for residual transform coding. First, the input picture is split into a number of non-overlapping rectangular blocks that are predicted by means of intra prediction, i.e. prediction from the same picture, or inter-prediction, i.e. prediction from previously decoded pictures. The prediction error is coded with the block transform coding scheme followed by quantization and entropy coding of the quantized transform coefficients.

In HEVC the picture is divided into so-called coding tree blocks or largest coding units (LCU) of size 16 × 16, 32 × 32 or 64 × 64 samples, which are typically coded in the raster-scan order. The coding tree blocks can be further split into coding units (CU) by means of a quad-tree structure thus splitting the picture into non-overlapping square blocks of different size, where the size of the block typically depends on the picture content and the target coding quality. The CU can be further split into prediction units (PU) of rectangular shape and also serves as a root of a transform quad-tree. The leaves of the transform tree form square-shaped transform units (TU) which can be coded by using the transforms of various sizes between 4 × 4 and 32 × 32 samples. The transforms used in the draft HEVC standard are based on integer approximation of DCT except the 4 × 4 integer DST transform used in some cases for coding of 4 × 4 intra-predicted blocks.

As was explained previously, the coded picture is split into non-overlapping blocks of different size that have individual prediction modes and use separate transform blocks for residual coding. This may cause discontinuities at the block boundaries. The discontinuities arise from both coarse quantization of the transform coefficients and from different prediction modes used in prediction coding. For example, neighbor blocks may use different intra prediction modes or different motion vectors for motion-compensated prediction. In motion-compensated prediction, adjacent block in the coded picture may use prediction blocks that do not come from adjacent blocks in the reference picture, thus producing discontinuities at the prediction block boundaries. When the signal on each side of the block boundary does not show much spatial activity, discontinuities between the neighbor prediction and/or transform blocks are perceived subjectively as so-called block artifacts which decrease visual quality of the coded picture. A one-dimensional example of blocking artifact is shown in Figure 1.

In order to decrease block artifacts, the draft HEVC standard uses an in-loop deblocking filter, which is the first of the two in-loop filters currently used in the main profile of HEVC (the second filter being sample adaptive offset (SAO)). Applying the deblocking filter inside the prediction loop makes sure that the blocking artifacts are not transferred inside the larger blocks in the subsequent frames by the motion-compensated prediction. Moreover, by improving the quality of the reference pictures, the deblocking filter improves the compression efficiency and provides some guarantee of the reconstructed video quality.

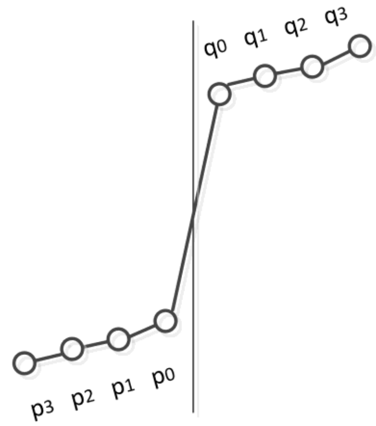


Figure 1 Blocking artifact over the block boundary (one dimension)

The HEVC deblocking filter has shown the ability to improve the subjective quality of the coded video especially at lower bitrates. The complexity of the HEVC deblocking filter is lower than that of the AVC/H.264 deblocking filter thus making it less of a burden for both software and hardware implementations. The HEVC deblocking filtering can also be performed in parallel mode on multiple-processor systems compared to the AVC/H.264 deblocking filter.

This paper describes the deblocking filter and deblocking filtering decisions that are used in the draft HEVC standard. For more details, the reader is referred to [1] and to input contributions to the JCT-VC [4], [5], [6], [7], [8]. Section 2 explains which block boundaries in the coded picture are subject to deblocking filtering. Section 3 explains signal adaptive decisions for deblocking filtering and the corresponding filtering operations. The objective results and subjective quality examples are shown in Section 4, while Section 5 concludes the paper.

2. BLOCK BOUNDARIES FOR DEBLOCKING FILTERING

The deblocking filtering operations are only applied to block boundaries that are likely to have visible blocking artifacts. Therefore, the deblocking filtering process is a combination of deblocking filtering decisions and filtering operations. The deblocking filtering decisions determine whether deblocking filtering should be applied to the block boundary and if deblocking filtering is applied, the strength of the filtering. The filtering operations modify the values of the pixels at the block boundaries chosen by the deblocking filtering decisions.

Not all block boundaries are likely to form visible block artifacts. Unlike the deblocking filtering of AVC/H.264, which is applied on the 4 × 4 luma sample grid, the HEVC luma deblocking is only applied to the block edges that coincide with the 8 × 8 luma sample grid. The HEVC chroma deblocking filtering is only applied to the block boundaries on the 8 × 8 chroma sample grid (see Figure 2). In Figure 2, the boundaries where deblocking can be applied are shown in solid lines. This means that chroma deblocking is only applied to the chroma block boundaries that coincide with the 16 × 16 luma filtering grid (when 4:2:0 chroma subsampling is used). Since the smallest transform block size in HEVC is 4 × 4, this means that not all the block boundaries are filtered. This can theoretically decrease the subjective quality. In case of HEVC, applying the deblocking filtering only on the 8 × 8 sample grid significantly decreases the worst case for the number of operations compared to deblocking filtering applied to all 4 × 4 block boundaries.

Another important aspect of only applying the deblocking filtering on the 8 × 8 sample grid is that HEVC deblocking does not create cross-dependencies over the picture. Since the deblocking filtering operations in HEVC modify at most three pixels on each side of the block boundary and use at most four pixels on each side of the block boundary,, the deblocking filtering operations at one block boundary do not affect the deblocking filtering operations at the next parallel block boundary. Since there are no cross-dependencies, the deblocking filtering in HEVC can be easily performed in parallel on different parts of the reconstructed picture.

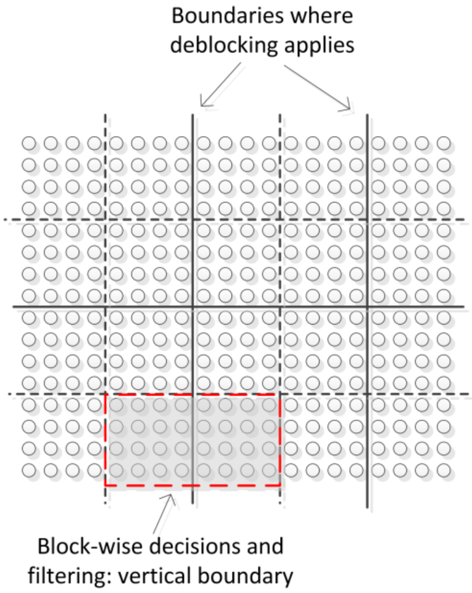


Figure 2 Deblocking filtering is applied on 8 × 8 luma sample grid, deblocking operations do not affect deblocking on parallel boundary

	$p3_0$	$p2_0$	$p1_0$	$p0_0$	$q0_0$	$q1_0$	$q2_0$	$q3_0$	
P	$p3_1$	$p2_1$	$p1_1$	$p0_1$	$q0_1$	$q1_1$	$q2_1$	$q3_1$	**Q**
	$p3_2$	$p2_2$	$p1_2$	$p0_2$	$q0_2$	$q1_2$	$q2_2$	$q3_2$	
	$p3_3$	$p2_3$	$p1_3$	$p0_3$	$q0_3$	$q1_3$	$q2_3$	$q3_3$	

Figure 3 Four-sample long segment of vertical block boundary (confined by red dashed rectangle in Figure 2). Deblocking decisions are based on lines 0 and 3 marked with dashed lines

The deblocking filtering operations in HEVC are also independent across the color components. The chroma deblocking filtering does not use filtering decisions obtained on luma component as in H.264/AVC, thus making it possible to process all three components independently.

2.1 Boundary strength derivation

Not all boundaries on the 8 × 8 sample grid are likely to form block artifacts. Further classification of the block boundaries is used to determine which of the block boundaries are likely to contain artifacts. The deblocking filtering decisions are performed on each four-sample long segment of the block boundary (vertical or horizontal) as shown in Figure 3 [7], [8]. Figure 3 shows the region confined by the red dashed line in Figure 2. First, the following conditions should hold in order for deblocking filtering to be applied to the block boundary. The block boundary shall coincide with the 8 × 8 sample grid and the boundary shall be either a transform unit (TU) or a prediction unit (PU) boundary. This means that the deblocking filtering is not applied to the samples on the 8 × 8 grid, which are located inside TUs or PUs which are larger than 8 × 8 samples.

Each four-sample segment of the block boundary that satisfies the above stated conditions is assigned a boundary strength (*Bs*). The boundary strengths for the luma blocks are determined as show in Table 1. The conditions for boundary strength derivation are similar to those used in AVC/H.264 deblocking filter [3].

The deblocking decision and filtering process is only applied to the luma block boundary if its boundary strength *Bs* is greater than zero. Therefore, deblocking filtering is applied to a block boundary if one of the adjacent blocks is coded with the intra-prediction mode or has at least one non-zero transform coefficient. Deblocking filtering is also applied to a block boundary if the prediction units adjacent to the block boundary refer to different reference pictures or the difference between the motion vectors of the adjacent blocks is greater than one integer pixel.

The deblocking for chroma components is only applied to the block boundaries with Bs equal to two, i.e. only if one of the adjacent blocks is coded with intra-prediction mode.

Table 1 Boundary strength derivation

Conditions	**Bs value**
At least one of adjacent blocks is Intra	2
At least one of adjacent blocks has non-zero coded residual coefficient and boundary is a transform block boundary	1
Absolute difference between corresponding spatial motion vector components of the two blocks is greater or equal to 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

3. SIGNAL ADAPTIVE FILTERING AND DECISIONS

The first part of this section describes the luma deblocking filtering while the chroma deblocking filtering is described in subsection 3.3. In order to decrease the deblocking filter complexity, the deblocking decisions for a block boundary segment of length 4 samples are based on a subset of lines across the block boundary segment. As can be seen in Figure 3, the current version of the draft HEVC standard uses the deblocking decisions based on lines 0 and 3 as was proposed in [7]. Several decisions are made for a block boundary segment: (i) whether the deblocking filter is applied to the block boundary segment, (ii) if the deblocking filter is used, whether a strong or a normal deblocking filter is applied, and (iii) if the normal deblocking filtering mode is used, how many pixels are modified by the deblocking filter.

The decision whether the deblocking filter is applied to the block boundary segment is based on the spatial activity of the signal on each side of the block boundary segment. Due to the property of the human visual system, blocking artifacts are more visible in areas with low spatial activity. In areas with higher spatial activity, blocking artifacts are masked by the texture. Therefore, in order to avoid over-smoothing of the reconstructed video in the areas with high spatial activity, the following condition is checked in order to decide whether the deblocking filter is applied:

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

This condition checks that the signal on each side of the block boundary has a form of an inclined ramp or is flat. One can see that the left-hand side of (1) indicates the deviation of the signal from an inclined or a straight line across the boundary. The left-hand side of (1) is compared with a threshold β, which depends on the quantization parameter (QP). The values of the parameters β have piece-wise linear dependency on QP and are typically derived from a look-up table [1]. If condition (1) holds, the deblocking filter is applied to the corresponding block boundary segment. There are two types of filtering that can be used.

3.1 Strong filtering mode

The following condition checks whether a normal or a strong filtering mode is applied to the four-pixel long segment of the block boundary. The strong filtering is applied if all of the following conditions hold for both row $i = 0$ and $i = 3$ (see Figure 3):

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

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

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

where β is the same threshold as in (1) and t_C is the so-called clipping offset that is used to control the amount of filtering applied to the block boundary. Similar to β, the threshold t_C depends on the QP value and is obtained from a table specified in [1]. One can see from the equations above that the decision to apply the strong filter is done based on the following characteristics of the signal: the variations of the signal from the straight line is below the threshold β/8, which corresponds to approximately four times lower deviation compared to (1). The signal must be flat on both sides of the block boundary, as checked in (3), and the difference between the pixels closest to the block boundary p_0 and q_0 is smaller than a predetermined threshold (which is 2.5 times larger than the maximum modification of the signal allowed at the current value of QP).

The strong filter used in HEVC is similar to that used in AVC/H.264. The strong filter modifies three pixels on each side of the block boundary (unlike normal filtering which modifies at most two pixels from the block boundary). The main difference from AVC/H.264 is the clipping operation proposed in [6]. Since the strong filtering decisions are based on a subset of lines across the block boundary [7], the clipping operation makes sure that making the strong filtering decisions base on a subset of lines does not result in too strong filtering across other lines. The following strong filter operations are applied to pixels, not taking the subsequent clipping operation into account:

$$p_0' = (p_2 + 2p_1 + 2p_0 + 2q_0 + q_1 + 4) >> 3, \quad (5)$$

$$p_1' = (p_2 + p_1 + p_0 + q_0 + 2) >> 2, \quad (6)$$

$$p_2' = (2p_3 + 3p_2 + p_1 + p_0 + q_0 + 4) >> 3, \quad (7)$$

where p_0', p_1', and p_2' are modified values of samples p_0, p_1, and p_2 respectively. The pixels are then clipped to the range of $(p_i - 2t_C, p_i + 2t_C)$. The pixels q_0, q_1, and q_2 are modified similarly.

3.2 Normal filtering mode

If the strong filtering is not applied, i.e. at least one of the conditions (2)−(4) does not hold but condition (1) holds, normal filtering is applied to the block boundary segment. In this case, the following conditions are checked in order to decide how many pixels from the block boundary can be modified by deblocking filtering. By default, the normal filtering modifies only one pixel from each side of the block boundary. In order to determine whether additional pixels on each side of the block boundary are modified by deblocking filtering, the following conditions are checked.

$$| p_{2,0} - 2p_{1,0} + p_{0,0} | + | p_{2,3} - 2p_{1,3} + p_{0,3} | < 3/16\, \beta, \quad (8)$$

$$| q_{2,0} - 2q_{1,0} + q_{0,0} | + | q_{2,3} - 2q_{1,3} + q_{0,3} | < 3/16\, \beta. \quad (9)$$

If condition (8) holds, at most two pixels from the block boundary are modified in block A. If (9) holds, at most two pixels from the block boundary are modified in block B.

The filtering operations applied in the normal filtering mode have the following property: they do not modify the inclined line crossing the block boundary. Since the filtering decisions in (1), (8), and (9) enable application of the normal deblocking filtering to the signal that has a form of the gradient or ramp over a block boundary, it is important that the deblocking filtering does not modify this kind of signal. The following operations are used in normal filtering mode for each of the four lines across the block boundary segment. First, filtered sample values p_0' and q_0' are determined by adding or subtracting a delta value to each of the sample values:

$$p_0' = p_0 + \Delta_0, \quad (10)$$

$$q_0' = q_0 - \Delta_0, \quad (11)$$

where the value of Δ_0 is obtained as in the following:

$$\Delta_0 = \text{Clip3}(-t_C, t_C, (9(q_0 - p_0) - 3(q_1 - p_1) + 8) >> 4), \quad (12)$$

and the Clip3(a,b, x) function means clipping a variable x to the range (a,b), i.e.

$$\text{Clip3}(a,b, x) = \text{Max} (a, \text{Min} (b, x)) \quad (13)$$

Neglecting the clipping operation, the impulse response of this filter is (3 7 9 –3)/16. The offset value, Δ_0, indicates the deviation of the signal at the sides of the block boundary from a perfect ramp. The offset is equal to zero when the signal across the boundary has the form of a perfect ramp.

The deblocking filtering is only applied to the row of samples across the block boundary if the following condition holds:

$$| (9(q_0 - p_0) - 3(q_1 - p_1) + 8) >> 4 | < 10\, t_C. \quad (14)$$

One can see that the left-hand side of the expression reuses the value of Δ_0 before clipping thus reducing the number of operations. This conditions checks whether the difference between the signal on both sides of the block boundary is caused by a blocking artifact or is likely to be a natural edge.

When two pixels from the block boundary are modified by deblocking filtering (i.e. if condition (8) holds), the modified value p_1' is calculated as follows:

$$p_1' = p_1 + \Delta p_1. \quad (15)$$

Likewise, if condition (9) holds, q_1' is calculated as:

$$q_1' = q_1 + \Delta q_1, \quad (16)$$

where the values Δp_1 and Δq_1 are obtained as in the following:

$$\Delta p_1 = \text{Clip3}(-t_C, t_C, ((((p_2 + p_0 + 1) >> 1) - p_1 + \Delta_0) >>1), \quad (17)$$

$$\Delta q_1 = \text{Clip3}(-t_C, t_C, ((((q_2 + q_0 + 1) >> 1) - q_1 - \Delta_0) >>1). \quad (18)$$

The impulse response of the filter that corresponds to modification of the pixel at position p_1 is (8 19 –1 9 –3)/32 if the clipping operation is neglected. One can see that the filtering operation for the second pixels from the block boundary does not modify the signal that show a form of a ramp.

Summarizing the operations of the HEVC deblocking filter in the normal filtering mode, the filtering is applied to the signal, when the signal on the side of the block boundaries is similar to the inclined plane or a flat surface. The filtering operations, in their turn, are designed in a way that they reduce a blocking artifact, when it is likely to be present and avoid modifying the signal, when a blocking artifact is not likely to be present, for example when the signal intensity is gradually changes across the block boundary.

3.3 Chroma deblocking filtering

The HEVC chroma deblocking filter has even lower complexity than the luma deblocking filter. Chroma deblocking is performed when *Bs* is equal to 2, which means that at least one of the adjacent blocks is coded in intra-prediction mode. Therefore, chroma deblocking can be done in parallel to luma deblocking without access to luma deblocking decisions. No further decision is done for chroma deblocking filtering and the following operations are used to obtain the delta value for chroma deblocking filtering:

$$\Delta_c = \text{Clip3}(-t_C, t_C, (((p_0 - q_0) << 2) + p_1 - q_1 + 4) >> 3), \quad (19)$$

The value of Δ_c is used for modifications of chroma samples p_0 and q_0 is a similar way to (10) and (11).

4. OBJECTIVE AND SUBJECTIVE QUALITY

This section presents the results of subjective and objective performance of HEVC deblocking. The HM7.0 (HEVC reference software version 7) is used in the experiments. Table 2 shows objective results of the HEVC with deblocking filtering turned on vs. HEVC with deblocking filtering turned off. The results are provided for Main profile, on all-intra, random-access and low-delay configurations. In random-access configuration, the intra-coded pictures are inserted periodically in order to provide an access point with a period approximately equal to one second. This configuration uses hierarchical-B coding structure and is a typical example of a video coding structure used in a broadcasting scenario. The two low-delay configurations are examples of video coding used in the real-time conversational services. The low-delay configuration only uses one intra-coded frame in the beginning of the coded sequence and inter-prediction can only use pictures preceding the current picture in the temporal order. All-intra configuration represents a coding structure that uses only intra-frame prediction. The simulations use the conditions and the test sequences used in HEVC common test conditions [10]. The objective performance of the HEVC deblocking filter is shown in Table 2. The table shows the increase in BD-rate [9], i.e. the change in the average bitrate at the same PSNR, when the deblocking filter in HM7.0 is turned off. One can see from the results that the average increase in bitrate is in the range of 2.4% to 3.4% on random-access and low-delay configurations. One can also see that the bitrate increase when disabling the deblocking filter is higher for higher resolution sequences (Classes A, B, and E), where the bitrate increase reaches 6.9%.

The subjective improvements due to deblocking filtering are even more significant. Figure 4 shows an example of turning the deblocking filtering on and off for a test sequence "Kristen and Sara", which is a part of the HEVC common test conditions [10]. The sequence is coded with low-delay B configuration at base QP value equal to 37. One can see from the figure that the deblocking filtering improves the subjective quality of the coded picture by removing blocking artifacts of both larger and smaller blocks.

Table 2 Comparison of HM7.0 with deblocking filter turned on vs. HM7.0 with deblocking filter turned off. Positive values of BD-rate indicate increase in bit-rate

	Random Access Main		
	Y	U	V
Class A	3.6%	2.1%	1.7%
Class B	3.3%	2.1%	2.0%
Class C	2.1%	1.4%	1.8%
Class D	1.4%	1.0%	0.9%
Class E			
Class F	1.3%	0.9%	0.9%
Overall	2.4%	1.5%	1.5%

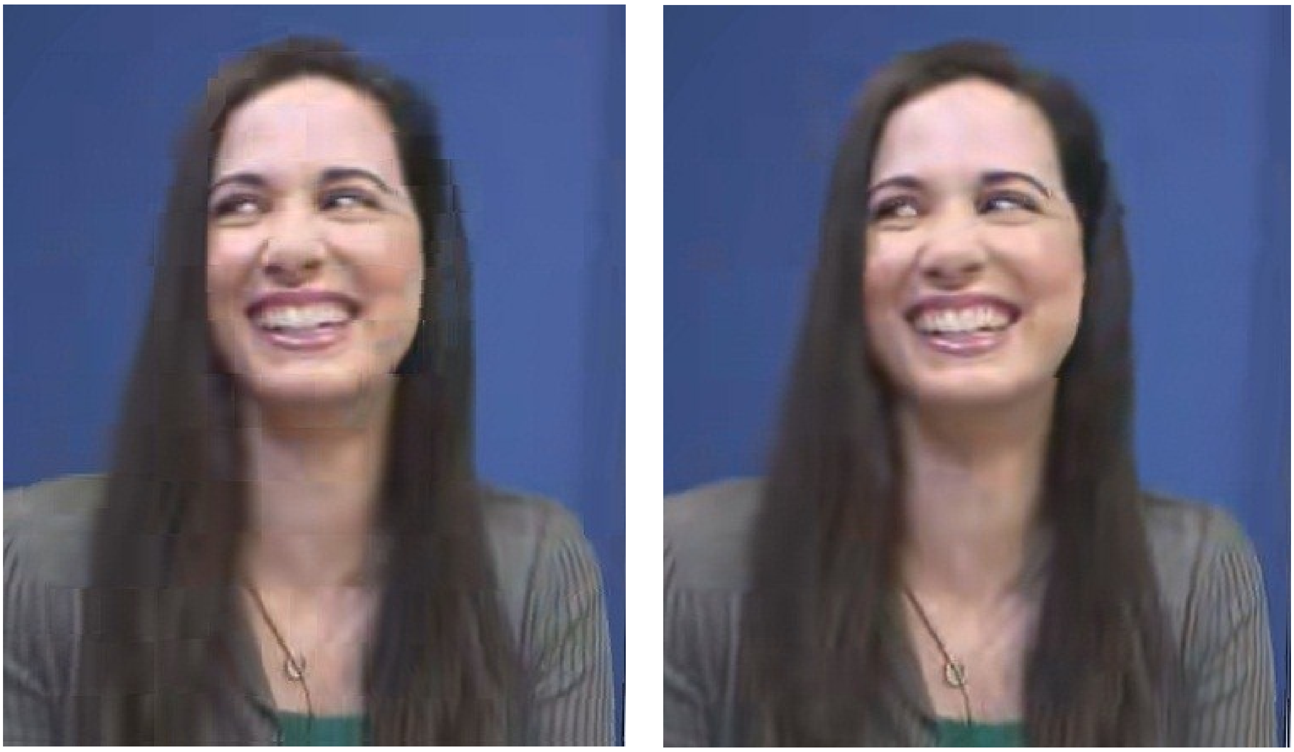
	All Intra Main		
	Y	U	V
Class A	1.9%	2.6%	2.1%
Class B	1.7%	3.0%	3.1%
Class C	0.9%	2.1%	2.6%
Class D	0.7%	2.0%	2.3%
Class E	2.2%	4.7%	5.6%
Class F	0.6%	1.1%	1.1%
Overall	1.3%	2.5%	2.7%

	Low delay B Main		
	Y	U	V
Class A			
Class B	3.4%	1.4%	1.5%
Class C	2.1%	1.4%	1.6%
Class D	1.4%	0.8%	1.6%
Class E	3.8%	3.2%	3.5%
Class F	1.3%	0.8%	0.1%
Overall	2.4%	1.4%	1.6%

	Low delay P Main		
	Y	U	V
Class A			
Class B	5.0%	2.5%	2.2%
Class C	2.6%	1.6%	1.9%
Class D	1.6%	1.4%	1.3%
Class E	6.9%	4.3%	5.9%
Class F	1.6%	0.5%	0.5%
Overall	3.4%	2.0%	2.2%

5. CONCLUSIONS

The deblocking filter is one of the tools included in the main profile of HEVC. The deblocking filter improves both the subjective and objective video quality. Applying the deblocking filter inside the motion compensation loop allows efficient suppression of block artifacts at the block boundaries and avoids propagation of block artifacts inside the blocks during the motion compensation process. The deblocking filter in HEVC does not have cross-picture dependencies and thus can be easily parallelized.



(a) No deblocking (b) Deblocking is on

Figure 4 Subjective results of deblocking filtering on sequence "Kristen and Sara" coded with low-delay configuration at QP 37. (a) Deblocking is turned off. (b) Deblocking is turned on.

REFERENCES

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft ", ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-I1003, Geneva, (May 2012).

[2] ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4) AVC: Advanced Video Coding for Generic Audiovisual Services.

[3] P. List, A. Josh, J. Lainema, G. Bjontegaard, and M. Karczewicz, “Adaptive Loop Filter”, IEEE Trans. Circuits Syst. Video Technol. (vol. 13), 614–619, (July 2003).

[4] K. Ugur, K. R. Andersson, A. Fuldseth, “Video coding technology proposal by Tandberg, Nokia, and Ericsson”, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-A119, Dresden, Germany, (15-23 April 2010).

[5] A. Norkin, K. Andersson, R. Sjöberg, Q. Huang, J. An, X. Guo, S. Lei, "CE12: Ericsson's and MediaTek's deblocking filter", ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-F118, Torino, (July, 2011).

[6] M. Ikeda, 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, San Jose, (Feb., 2012).

[7] M. Narroschke, S. Esenlik, T. Wedi, "CE12 Subtest 1: Results for modified decisions for deblocking", ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-G590, Geneva, (Nov., 2011).

[8] 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, San Jose, (Feb., 2012).

[9] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T-T SG16 Doc. VCEG-M33, (2001).

[10] F. Bossen, "Common test conditions," JCTVC-I1100, Geneva, (May 2012).