Bilateral Loop Filter in Combination with SAO

Jacob Ström, Per Wennersten, Jack Enhorn, Du Liu, Kenneth Andersson and Rickard Sjöberg
Ericsson Research, Sweden
Email: jacob.strom@ericsson.com, per.wennersten@ericsson.com, jack.enhorn@ericsson.com,
du.liu@ericsson.com, kenneth.r.andersson@ericsson.com, rickard.sjoberg@ericsson.com

***Abstract*—This paper describes a bilateral filter that is being proposed as a coding tool for the Versatile Video Codec (VVC). The filter acts as a loop filter in parallel with the sample-adaptive offset (SAO) filter. Both the proposed filter and SAO act on the same input samples, each filter produces an offset, and these offsets are then added to the input sample to produce an output sample that, after clipping, goes to the next stage. The method has been implemented and tested according to the common test conditions in VVC test model version 5.0. For the all-intra configuration, we report a BD rate figure of -0.4% with an encoder run time increase of 6% and a decoder run time increase of 4%. For the random access configuration, the BD rate figure is -0.5% with an encoder run time increase of 2% and a decoder run time increase of 2%.**

***Index Terms*—video, compression, bilateral filter, loop filter**

I. Background

In VVC, prediction residuals are converted to a transform domain and are then quantized. This quantization can lead to ringing artifacts around edges. There are several ways to mitigate this ringing. Perhaps the most important one is to use an adaptive block structure where a transform block containing a strong edge can be divided into two or more smaller blocks where the edge is not as pronounced. However, even with the advanced block partitioning used in VVC, not all ringing can be avoided this way. Another approach is therefore to filter the data. The VMV 9 video codec uses a deringing post-processing filter that classifies samples as edge or non-edge, and employs different filters based on the classification [2]. C. Y. Tsai et al. propose an adaptive loop filter (ALF), which determines suitable filter coefficients that are signaled to the decoder and used to catch and fix artifacts [3]. Subsequently decoded images can predict from the filtered images, making it a loop filter rather than a post-processing filter. Sample adaptive offset (SAO) is another loop filter which addresses ringing. SAO was introduced in [4] and is included in HEVC [5]. It classifies samples using a 3×3 neighborhood and signals an offset for every class to the decoder. The offset is added to the classified samples in the decoder.

Our work is inspired by that in [1], where decompressed JPEG images are filtered by applying shifted versions of the transform to the image and averaging the result. One way to interpret this work is to say that the quantization artifacts will be different in every shifted version, and hence will be averaged away, whereas the signal will stay the same. In [1] the same JPEG settings are used during the filtering as are used for decompression—hence heavily compressed images will automatically be filtered more strongly. This makes sense since higher compression will give stronger artifacts.

Bilateral filtering for video coding using look-up tables is introduced in [6]. Filtering is applied on the reconstructed block, i.e., right after the prediction and the residual have been added together. This means that the next intra block will be able to predict from filtered samples. This method is included into version 6 of the Joint Exploration Model, an experimental software developed by JVET as a precursor to VVC [7]. However, predicting from filtered samples means that the bilateral filter is situated in the critical path of hardware implementations. This can make it difficult to decode large resolutions at a sufficient frame rate. Versions where the look-up table is replaced by a piece-wise linear function are presented in [19] and [20]. Filtering of the reconstructed block in the Hadamard transform domain is proposed in [8], but all these approaches suffer from being in the critical path.

Moving the bilateral filtering to the loop filter stage, after the entire image has been reconstructed, is proposed in [9] as a way to avoid the critical path. A similarly placed Hadamard filter is also proposed in [10]. While not being able to predict from filtered samples harms the coding efficiency of both of these methods, this is compensated for by the fact that it is now possible to access neighboring samples to the right and below the filtered block. However, the current draft of VVC [18] already contains three loop filter stages (deblocking, SAO and ALF) and adding another loop filter stage has its drawbacks: To increase throughput, decoders typically start applying one loop filter stage (such as ALF) before all samples from the previous stage (such as SAO) are filtered. If the ALF stage is faster than the SAO stage, it will be starved of data and will have to stall. Introducing yet another loop filter stage makes this situation worse, and is thus undesirable.

II. Detailed Background

This paper builds on the contributions in [11], [6] and [13], so we will go through these in some more detail. First introduced by Tomasi and Manduchi [11], bilateral filtering is a technique to make the filter weights decrease not only with the distance between the samples but also with increasing difference in intensity. This way, over-smoothing of edges can be ameliorated. A weight is defined as

$$w(\Delta x, \Delta y, \Delta I) = e^{-\frac{\Delta x^2+\Delta y^2}{2\sigma_d^2} - \frac{\Delta I^2}{2\sigma_r^2}}, \quad (1)$$

where Δx and Δy is the distance in the x- and y-dimension and ΔI is the difference in intensity between the samples.

In [6], the spatial filtering strength σ_d is determined by the block size, with smaller blocks being filtered more strongly, and the intensity filtering strength σ_r is determined by the quantization parameter (QP), with stronger filtering being used for higher QPs. Only the four closest samples are used, so the filtered sample intensity I_F can be calculated as

$$I_F = I_C + \frac{w_A \Delta I_A + w_B \Delta I_B + w_L \Delta I_L + w_R \Delta I_R}{w_C + w_A + w_B + w_L + w_R}, \quad (2)$$

where I_C denotes the intensity of the center sample, $\Delta I_A = I_A - I_C$ the intensity difference between the center sample and the sample above, and ΔI_B, ΔI_L and ΔI_R denote the intensity differences between the center sample and that of the samples below, to the left and to the right respectively.

Here $w_C = 1$, since $\Delta x = \Delta y = \Delta I = 0$ for the center sample, but every other weight in (2) will depend on the intensity difference. This means that the denominator will change for every sample and a per-sample division is inevitable. In [6] this division is implemented using a multiplication and a 576-byte-long look-up table (LUT). The division is instead avoided altogether in [13] by changing the weights in the denominator from w to w', where

$$w'(\Delta x, \Delta y) = e^{-\frac{\Delta x^2 + \Delta y^2}{2\sigma_d^2}}. \quad (3)$$

This means that the denominator becomes $1 + 4e^{-\frac{1}{2\sigma_d^2}}$ which is constant over a block. Equation (2) is further simplified in [13] by noting that w_A can be written as $w_A = e^{-\frac{1}{2\sigma_d^2}} w''_A$, where $w''_A = e^{-\frac{\Delta I_A^2}{2\sigma_r^2}}$, and the same goes for w_B, w_L and w_R. This yields

$$I_F = I_C + \frac{e^{-\frac{1}{2\sigma_d^2}} (w''_A \Delta I_A + w''_B \Delta I_B + w''_L \Delta I_L + w''_R \Delta I_R)}{1 + 4e^{-\frac{1}{2\sigma_d^2}}} \quad (4)$$

which can be written as

$$I_F = I_C + d(\sigma_d)(w''_A \Delta I_A + w''_B \Delta I_B + w''_L \Delta I_L + w''_R \Delta I_R), \quad (5)$$

with $d(\sigma_d) = e^{-\frac{1}{2\sigma_d^2}} / (1 + 4e^{-\frac{1}{2\sigma_d^2}})$, which is constant over the block. The division can therefore be replaced by a multiplication with $d(\sigma_d)$ without the need for a division look-up table. The simplification makes the denominator larger, and hence reduces the filter strength. This can be compensated for by modifying the strength parameters σ_d and σ_r, although the filters are not equivalent even after compensation.

III. Proposed Method

This paper proposes a number of modifications to the bilateral filter described in the previous section. Some of these modifications have been described in standardization contributions [9] [14] but have not been published academically until here. The first contribution is a simplification that avoids multiplications. The second contribution is an increase of the filter size from four surrounding samples to eight. The last contribution is a way to combine the filter with SAO so that it can be done in parallel with SAO without the data starvation problems discussed in Section I.

A. Multiplication Simplification

Studying the expression for w'', we see that it depends on the two variables ΔI and σ_r

$$w''(\Delta I, \sigma_r) = e^{-\frac{\Delta I^2}{2\sigma_r^2}}, \quad (6)$$

and it can therefore be tabulated with a two-dimensional LUT. However, as is seen from (5), the expression is only used when multiplied by ΔI. We can therefore instead tabulate this expression directly. Denoting it $\mu_{\sigma_r}(\Delta I)$,

$$\mu_{\sigma_r}(\Delta I) = w''(\Delta I, \sigma_r)\Delta I = e^{-\frac{\Delta I_A^2}{2\sigma_r^2}} \Delta I, \quad (7)$$

Equation (4) can now be simplified to

$$I_F = I_C + d(\sigma_d) \left[\mu(\Delta I_A) + \mu(\Delta I_B) + \mu(\Delta I_L) + \mu(\Delta I_R) \right] \quad (8)$$

where we have used the shorthand $\mu(\Delta I)$ for $\mu_{\sigma_r}(\Delta I)$. It is therefore possible to avoid any multiplications between the weight and the intensity difference when filtering a sample.

To remove the remaining multiplication, we limit the number of different values of $d(\sigma_d)$ used. We use the weakest filtering $d(\sigma_{dw})$ as a baseline, and use three strengths $d(\sigma_{dw})$, $2d(\sigma_{dw})$, and $3d(\sigma_{dw})$. Hence the factor $d(\sigma_{dw})$ can be baked in to the LUT and (8) can be rewritten as

$$I_F = I_C + c[\mu'_{\Delta I_A} + \mu'_{\Delta I_B} + \mu'_{\Delta I_L} + \mu'_{\Delta I_R}], \quad (9)$$

where $\mu'_{\Delta I_A} = \mu_{\sigma_r}(\Delta I_A) d(\sigma_{dw})$ is called a modifier value, and where c is a either 1, 2 or 3. If we denote the square-bracketed expression in Equation 9 by m_{sum} we can calculate the right term $c * m_{sum}$ as

$$c_v = k_1 \,\&\, (m_{sum} \ll 1) + k_2 \,\&\, m_{sum}, \quad (10)$$

where k_1 is the most significant bit of c, k_2 is the least significant bit, $\ll$ denotes leftwards bit shift and & denotes a logical AND operation. It is therefore possible to implement this multiplication with a single addition. Storing the look-up table values of $\mu'_{\Delta I_A}$ with five fractional bits of precision means that the final filtered value can be obtained using

$$I_F = I_C + ((c_v + 16) \gg 5). \quad (11)$$

B. Increasing the Size of the Filter

Using only the nearest four neighbors restricts the size of ringing features that the filter can correct. Therefore we propose to also include the diagonal neighbors into the filter kernel. Using (1) we can calculate the attenuation a we should give a diagonal sample weight compared to a neighboring sample weight with the same intensity difference:

$$a = \frac{w(1, 1, \Delta I)}{w(1, 0, \Delta I)} = \frac{e^{-\frac{1+1}{2\sigma_d^2}} e^{-\frac{\Delta I^2}{2\sigma_r^2}}}{e^{-\frac{1+0}{2\sigma_d^2}} e^{-\frac{\Delta I^2}{2\sigma_r^2}}} = e^{-\frac{1}{2\sigma_d^2}}, \quad (12)$$

which we approximate with 0.5. This only holds true when $\sigma_d = 0.849$, but given that this is close to the range used ($\sigma_d \in [0.52, 0.82]$ in [6]) and it is much simpler to calculate, it is a reasonable trade-off. Hence the same LUT can be used for diagonal values given that the value is right-shifted after look-up.

C. Combination with SAO

The left diagram in Fig. 1 shows the structure proposed in [9], where the bilateral filter is its own loopfilter stage. In

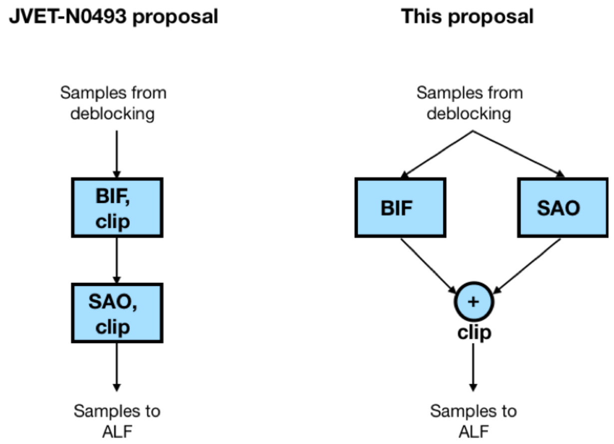

Fig. 1. Left: Bilateral filter as its own loopfilter stage. Right: Bilateral filter combined with SAO.

this configuration, SAO may have to wait for the BIF stage to finish filtering and clipping before processing can commence. In contrast, we propose that both the bilateral filter and SAO operate on the same input samples, and that they are then combined and clipped.

In detail, for every sample, the bilateral filter produces a difference value $\Delta I_F = I_F - I_C$ using the samples coming from the deblocking stage as input. In parallel, the SAO process produces an offset value ΔI_{SAO}, also using the deblocking stage as input. The final combined output is

$$I_{OUT} = \mathrm{clip}(I_C + \Delta I_F + \Delta I_{SAO}), \tag{13}$$

where $\mathrm{clip}(\cdot)$ makes sure the output is in the range $[0, 1023]$.

This arrangement brings great flexibility to the implementation. A hardware implementation may select to synchronize the processing of the bilateral filter and SAO on a per sample basis, whereas for a software implementation it may be more efficient to process them in sequence.

D. Detailed Explanation of Filtering

This section will describe in detail how a block is filtered. The decoding procedure follows that of VVC draft 5 [18] up to the point of deblocking. After deblocking, each coding tree unit (CTU) is processed transform unit by transform unit. Intra blocks are bilaterally filtered if the QP is larger than 17. Inter blocks have the additional constrains that the CBF flag must be 1 (which means that transform coefficients are present) and that $\min(width, height) < 32$. If these constraints are not met, ΔI_F is set to zero.

For blocks that should be filtered, the appropriate look-up table row is selected based on the QP according to Table I. Note that even though the table is two-dimensional, only a one-dimensional LUT-row is used per block, since the QP cannot change within a transform unit (TU). Also, the maximum

TABLE I
QP DETERMINES LUT ROW USED

QP range	LUTrow
18 to 23	[0, 4, 4, 4, 3, 2, 1 ,2, 1, 1, 1, 1, 0, 1, 1, -1]
24 to 28	[0, 8, 11, 11, 7, 5, 5, 4 , 5, 4, 4, 2, 2 , 2, 2, -2]
29 to 33	[0, 9, 16, 19, 22, 22, 20, 15, 12, 12, 11, 9, 9, 7, 8, -3]
34 to 38	[0, 12, 21, 28, 33, 36, 40, 40, 40, 36, 29, 22, 19 ,17, 15, -3]
≥ 39	[0, 17, 23, 33, 37, 41, 44, 44, 45, 44, 42, 27, 22, 17, 15, -3]

number of items per row is 16, which means that efficient SIMD code can be utilized for the per-sample look-up. Since the values are tabulated, they do not have to strictly obey the bilateral filtering equations. The values in Table I are based on these equations, but have been optimized for BD-rate using a sequence not part of the test set (CrowdRun720p).

Next, the contributions from the neighbors are calculated. First the intensity difference is calculated

$$\Delta I_R = (|I_R - I_C| + 4) \gg 3, \tag{14}$$

where $|\cdot|$ denotes the absolute value, and where the downshift by 3 is needed to make the table rows no larger than 16 items. Here we have assumed 10-bit input; if the input is n bits, we will add 2^{n-8} and shift $n-7$ steps. The resulting value can still be larger than 15, so its absolute value is clipped

$$sI_R = \min(15, \Delta I_R). \tag{15}$$

The modifier value $\mu'_{\Delta I_R}$ is now calculated as

$$\mu'_{\Delta I_R} = \begin{cases} \mathrm{LUTrow}\,[sI_R]\,, & \mathrm{if} I_R - I_C \geq 0 \\ -\mathrm{LUTrow}\,[sI_R]\,, & \mathrm{otherwise.} \end{cases} \tag{16}$$

Contributions from diagonal samples are right-shifted,

$$\mu'_{\Delta I_{BL}} = \begin{cases} \mathrm{LUTrow}\,[sI_{BL}] \gg 1, & \mathrm{if} I_{BL} - I_C \geq 0 \\ -(\mathrm{LUTrow}\,[sI_{BL}] \gg 1), & \mathrm{otherwise.} \end{cases} \tag{17}$$

We now create m_{sum} by adding together all the contributions,

$$m_{sum} = \mu'_{\Delta I_A} + \mu'_{\Delta I_B} + \mu'_{\Delta I_L} + \mu'_{\Delta I_R} + \tag{18}$$

$$\mu'_{\Delta I_{AL}} + \mu'_{\Delta I_{AR}} + \mu'_{\Delta I_{BL}} + \mu'_{\Delta I_{BR}}, \tag{19}$$

where $\mu'_{\Delta I_{AL}}$ is the contribution from the diagonal sample above and to the left, $\mu'_{\Delta I_{AR}}$ above and to the right etc. Note that $\mu'_{\Delta I_L}$ is equal to $-\mu'_{\Delta I_R}$ for the sample immediately to the left. Hence this value does not need to be calculated but can instead be reused. In a similar manner, $\mu'_{\Delta I_A}$, $\mu'_{\Delta I_{AL}}$ and $\mu'_{\Delta I_{AR}}$ can be obtained from previously calculated variables, and only four modifier values need to be calculated are per sample: $\mu'_{\Delta I_B}$, $\mu'_{\Delta I_R}$, $\mu'_{\Delta I_{BL}}$ and $\mu'_{\Delta I_{BR}}$.

The m_{sum} value shall now be multiplied with 1, 2 or 3 depending upon the minimum block dimension $D = \min(width, height)$. This is described in Tab. II. A software implementation, for which multiplications are cheap, may simply multiply m_{sum} with the c-value to get the correction value: $c_v = c * m_{sum}$. However, for a hardware implementation, it

TABLE II
BLOCK SIZE DETERMINES C-VALUE USED

Block type	$D < 4$	$D = 4$	$4 < D < 16$	$D \geq 16$
Intra	2	3	2	1
Inter	2	2	2	1

may be less expensive to instead use (10), which only uses one addition. Finally, the bilateral difference value is calculated as

$$\Delta I_F = (c_v + 16) \gg 5, \tag{20}$$

and this value can be combined with the SAO offset using (13) to obtain the combined filtered sample. For n-bit data, we add $\lfloor 2^{14-n} \rfloor$ and shift 2^{15-n}.

IV. ENCODING

The proposed method is implemented on top of the test model software VTM 5.0 [17], which encodes in two stages: The first stage decides which block partitioning to use, which prediction and transform to use etc., and it results in an image that has not been loop-filtered. In the second stage this image undergoes loop-filtering (deblocking, SAO and ALF).

During the first stage, the VTM 5.0 encoder reconstructs each block by inversely transforming the quantized coefficients, adding the prediction and clipping. The distortion is then calculated between the original samples and the reconstructed samples, and a decision is based on this distortion and the coding cost. In this paper, we insert a step of bilateral filtering after the clipping, so that the distortion is calculated on a bilaterally filtered block. A limitation of this approach is that many encoders evaluate most distortion calculations in the transform domain, and it may be considered costly to go back to the spatial domain to do bilateral filtering. However, it helps the encoder make a good decision. As an example, a block that contains ringing may be a better choice than a smooth block, since the bilateral filter will later remove some of the ringing. By including bilateral filtering in the rate distortion optimization (RDO), it is possible for the encoder to correctly select the block with ringing over the smooth block. Samples to the right and below the block are not available at this stage, so mirroring is used for these samples.

During the second stage, the parameters for SAO must be estimated. This is done by first running the bilateral filter on the output of the deblocking filter, and then the SAO estimation on that output. If the SAO estimation were instead done directly on the output of the deblocking filter, it could happen that both SAO and the bilateral filter would compensate for the same error, resulting in an overcompensation. Finally, SAO and the bilateral filter are applied to the output of the deblocked data as shown in the right diagram of Fig. 1.

V. RESULTS

Table III shows the BD-rate results and run-time increases for the proposed method. The anchor is VTM 5.0, the proposed method is implemented on top of VTM 5.0 and the common test conditions specified in [15] are used. A subjective test was carried out at the 15th JVET meeting between the proposed method and the anchor [16]. In this test, the proposed method was found to be better in one of the six sequences (ArenaOfValor, LDB) with non-overlapping confidence intervals, and no sequence was found to be worse. An example from this sequence can be found in Fig. 2. The proposed encoder will make different mode choices compared to the anchor encoder, which affects the result. With a rather small objective gain of around -0.5% it was not a given that the method would result in subjective gains. That one sequence was found to give non-overlapping confidence intervals can be seen as a confirmation that the method can ideed provide subjective gains.

TABLE III
PERFORMANCE OF PROPOSED ALGORITHM

	BRD Change per Class %								Runtime %	
	A1	***A2***	***B***	***C***	***E***	***ave***	***D***	***F***	***enc***	***dec***
AI	-0.29	-0.38	-0.33	-0.55	-0.45	-0.40	-0.49	-0.61	106	104
RA	-0.37	-0.52	-0.48	-0.62		-0.50	-0.57	-0.62	102	102
LDB			-0.22	-0.46	-0.27	-0.31	-0.65	-0.60	103	104

Fig. 2. Top left: Part of original frame 41. Top right: Original zoom-in. Bottom left: Anchor. Bottom right: Using the proposed bilateral filtering.

VI. COMPLEXITY ANALYSIS AND CONCLUSION

This section includes a complexity analysis that should be relevant for hardware implementations. As is done in core experiments in JVET, we calculate the number of adds, shifts and checks, where a check is a min- max- or abs operation. Table IV shows the number of such operations for calculating the modifier value of a sample below or to the right. Rounding additions are calculated separately in brackets, since these can be implemented with a carry bit in the nearby addition or check. No adder is needed to negate the LUT value, since it will be used in an addition later and it is possible to invert all the bits and set the carry to 1. Table IV is also valid for diagonal samples, except that one more shift is needed. Table V describes the total number of operations per sample. In total 13 additions and 5 rounding additions are needed per sample, as well as 8 shifts, 8 checks and 4 look-ups. Table VI show the bit widths of the inputs of the additions marked with $\oplus$ or $\ominus$. Only non-rounding additions

TABLE IV
ARITHMETIC OPERATIONS PER MODIFIER VALUE

Operation	add	shift	check	LUT
$\Delta I_R = (\lvert I_R - I_C \rvert + 4) \gg 3$	1 (1)	1	1	
$sI_R = \min(15, \Delta I_R)$			1	
$\mu'_{\Delta I_R} = \begin{cases} \mathrm{LUT}\,[sI_R] & \text{if} I_R - I_C \geq 0 \\ -\mathrm{LUT}\,[sI_R] & \text{otherwise} \end{cases}$				1
in total	1 (1)	1	2	1

TABLE V
ARITHMETIC OPERATIONS PER SAMPLE

Variable calculated	add	shift	check
$\mu'_{\Delta I_R}$ and $\mu'_{\Delta I_B}$	2 (2)	2	4
$\mu'_{\Delta I_{BL}}$ and $\mu'_{\Delta I_{BR}}$	2 (2)	4	4
m_{sum} (Equation (19))	7		
$c_v = k_1 \,\&\, (m_{sum} \ll 1) + k_2 \,\&\, m_{sum}$	1	1	
$\Delta I_F = (c_v + 16) \gg 5$	(1)	1	
$I_{OUT} = \mathrm{clip}(I_C + \Delta I_F + \Delta I_{SAO})$	1		
in total	13 (5)	8	8

are counted. There is one 11-bit adder, four 10-bit adders, one 9-bit adder, one 8-bit adder, three 7-bit adders and three 6-bit adders. The LUT values can be stored in 6 bits, which means that $16 \times 6 \times 5 = 480$ bits or 60 bytes are needed. Normally 7 bits would be required for signed values, but the sign bit can be uniquely recovered by ANDing the two next most significant bits, and thus does not need to be stored.

TABLE VI
BIT WIDTH FOR ADDITIONS

Operation	bits	#
$\Delta I_R = (\lvert I_R \ominus I_C \rvert + 4) \gg 3$	10	4
$m_{sum} =$ $\mu'_A + \mu'_B + \mu'_L + \mu'_R + \mu'_{AL} \oplus \mu'_{AR} + \mu'_{BL} \oplus \mu'_{BR}$	6	2
$\mu'_A \oplus \mu'_B + \mu'_L \oplus \mu'_R + \mu'_{AL} + \mu'_{AR} \oplus \mu'_{BL} + \mu'_{BR}$	7	3
$\mu'_A + \mu'_B \oplus \mu'_L + \mu'_R + \mu'_{AL} + \mu'_{AR} + \mu'_{BL} + \mu'_{BR}$	8	1
$\mu'_A + \mu'_B + \mu'_L + \mu'_R \oplus \mu'_{AL} + \mu'_{AR} + \mu'_{BL} + \mu'_{BR}$	9	1
$c_v = k_1 \,\&\, (m_{sum} \ll 1) \oplus k_2 \,\&\, m_{sum}$	11	1
$I_{OUT} = \mathrm{clip}(I_C + \Delta I_F \oplus \Delta I_{SAO})$	6	1

In conclusion, this paper describes the current status of the bilateral loop filter work in VVC and its combination with SAO. Simplificaitons of the filter as well as complexity are discussed. The results show improvements in both BD-rate and subjective quality over VTM 5.0, with a slight increase in encoder and decoder run time.

REFERENCES

[1] A. Nosratinia, "Enhancement of jpeg-compressed images by re-application of jpeg," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 27, pp. 69–79, Feb. 2001.

[2] S. Srinivasan, P. J. Hsu, T. Holcomb, K. Mukerjee, S. L. Regunathan, B. Lin, J. Liang, M.-C. Lee, and J. Ribas-Corbera, "Windows media video 9: overview and applications," *Signal Processing: Image Communication*, vol. 19, no. 9, pp. 851 – 875, 2004.

[3] C. Y. Tsai, C. Y. Chen, T. Yamakage, I. S. Chong, Y. W. Huang, C. M. Fu, T. Itoh, T. Watanabe, T. Chujoh, M. Karczewicz, and S. M. Lei, "Adaptive loop filtering for video coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 934–945, Dec 2013.

[4] C. M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S.-M. Lei, J.-H. Park, and W.-J. Han, "Sample adaptive offset in the hevc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1755–1764, Dec. 2012.

[5] *ISO/IEC 23008-2: High Efficiency Video Coding - Recommendation H.265 (11/88)*, ITU-T Std., 2013.

[6] P. Wennersten, J. Ström, Y. Wang, K. Andersson, R. Sjöberg and J. Enhorn, "Bilateral Filtering for Video Coding," in *Proceedings of the IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Saint Petersburg, FL, USA, Dec. 2017.

[7] J. Ström, K. Andersson, P. Wennersten, M. Pettersson, J. Enhorn and R. Sjöberg "EE2-JVET related: Division-free bilateral filter," in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 6th Meeting*, Input document JVET-F0096, Hobart, AU, 31 Mar 7 Apr 2017

[8] V. Stepin, S. Ikonin, R. Chernyak, J. Chen, "CE2 related: Hadamard Transform Domain Filter," in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 11th Meeting*, Input document JVET-K0068, Ljubljana, SI, 10-18 Jul 2018

[9] J. Ström, P. Wennersten, J. Enhorn, and R. Sjöberg "CE1-related: Multiplication-free bilateral loop filter" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting*, Input document JVET-N0493, Geneva, CH, 19-27 Mar 2019

[10] S. Ikonin, V. Stepin, A. Karabutov, J. Chen, "CE1-related: Hadamard transform domain in-loop filter also using bottom/right reconstructed samples" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting*, Input document JVET-N0479, Geneva, CH, 19-27 Mar 2019

[11] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision*, Bombay, India, Jan. 1998, pp. 839–846.

[12] G. Van der Auwera, et al., "Advanced Hybrid Video Codec Based on Flexible Block Partitioning," *IEEE Transactions on Circuits and Systems for Video Technology*, Special Issue on Video Compression with Capability beyond HEVC, 2019.

[13] Y.-W. Chen et al. "Description of SDR, HDR and 360 video coding technology proposal by Qualcomm and Technicolor low and high complexity versions" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting*, Input document JVET-N0493, San Diego, US, 10-20 Apr 2018

[14] J. Ström, P. Wennersten, J. Enhorn, D. Liu, K. Andersson and R. Sjöberg "Combined bilateral/SAO loop filter" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting*, Input document JVET-O0548, Gothenburg, SE, 3-12 Jul 2019

[15] F. Bossen, J. Boyce, K. Suehring, X. Li and V. Seregin "JVET common test conditions and software reference configurations for SDR video" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting*, Ouptut document JVET-N1010, Geneva, CH, 19-27 Mar 2019

[16] M. Wien "Core Experiment Viewing Test Procedure and Results" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting*, Input document JVET-O1118, Gothenburg, SE, 3-12 Jul 2019

[17] F. Bossen, X. Li, K. Suehring and A. Norkin "AHG report: Test model software development (AHG3)" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting*, Input document JVET-O0003, Gothenburg, SE, 3-12 Jul 2019

[18] B. Bross, J. Chen, J. Boyce, S.-H. Kim, S. Liu, Y. Ye, "JVET AHG report: Draft text and test model algorithm description editing (AHG2)" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting*, Input document JVET-O0002, Gothenburg, SE, 3-12 Jul 2019

[19] J. Ström, P. Wennersten, J. Enhorn, D. Liu, K. Andersson and R. Sjöberg "CE14: Reduced latency, LUT-free bilateral filter" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting*, Input document JVET-L0172, Macao, CN, 3-12 Oct 2018

[20] D. Rusanovskyy, N. Shlyakhov, T. Hsieh and M. Karczewicz "CE14: Test on in-loop bilateral filter from JVET-J0021/JVET-K0384 with parametrization (CE14.2)" in *Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting*, Input document JVET-L0406, Macao, CN, 3-12 Oct 2018