### Adaptive Loop Filter

For the luma component, one among 25 filters is selected for each 4×4 block, based on the direction and activity of local gradients.

#### Filter shape

Two diamond filter shapes (as shown in Figure 47) are used. The 7×7 diamond shape is applied for luma component and the 5×5 diamond shape is applied for chroma components.

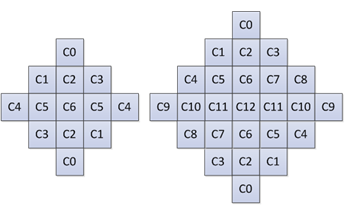


Figure 47 – ALF filter shapes (chroma: 5×5 diamond, luma: 7×7 diamond)

#### Block classification

For luma component, each block is categorized into one out of 25 classes. The classification index *C* is derived based on its directionality and a quantized value of activity , as follows:

(3-56)

To calculate and , gradients of the horizontal, vertical and two diagonal direction are first calculated using 1-D Laplacian:

(3-57)

(3-58)

(3-59)

(3-60)

Where indices and refer to the coordinates of the upper left sample within the block and indicates a reconstructed sample at coordinate .

To reduce the complexity of block classification, the subsampled 1-D Laplacian calculation is applied. As shown in Figure 48, the same subsampled positions are used for gradient calculation of all directions.

|  |  |
| --- | --- |
|  |  |
| (a) Subsampled positions for vertical gradient | (b) Subsampled positions for horizontal gradient |
|  |  |
| (c) Subsampled positions for diagonal gradient | (d) Subsampled positions for diagonal gradient |

**Figure 48 – Subsampled Laplacian calculation**

Then maximum and minimum values of the gradients of horizontal and vertical directions are set as:

, (3-61)

The maximum and minimum values of the gradient of two diagonal directions are set as:

, (3-62)

To derive the value of the directionality , these values are compared against each other and with two thresholds and :

**Step 1**. If both and are true, is set to .

**Step 2**. If , continue from Step 3; otherwise continue from Step 4.

**Step 3**. If , is set to ; otherwise is set to .

**Step 4**. If , is set to ; otherwise is set to .

The activity value is calculated as:

(3-63)

is further quantized to the range of 0 to 4, inclusively, and the quantized value is denoted as .

For chroma components in a picture, no classification method is applied.

#### Geometric transformations of filter coefficients and clipping values

Before filtering each 4×4 luma block, geometric transformations such as rotation or diagonal and vertical flipping are applied to the filter coefficients and to the corresponding filter clipping values depending on gradient values calculated for that block. This is equivalent to applying these transformations to the samples in the filter support region. The idea is to make different blocks to which ALF is applied more similar by aligning their directionality.

Three geometric transformations, including diagonal, vertical flip and rotation are introduced:

Diagonal: (3-64)

Vertical flip: , (3-65)

Rotation: , (3-66)

where is the size of the filter and are coefficients coordinates, such that location is at the upper left corner and location is at the lower right corner. The transformations are applied to the filter coefficients *f* (*k*, *l*) and to the clipping values depending on gradient values calculated for that block. The relationship between the transformation and the four gradients of the four directions are summarized in the following table.

**Table 3‑12 - Mapping of the gradient calculated for one block and the transformations**

|  |  |
| --- | --- |
| Gradient values | Transformation |
| gd2 < gd1 and gh < gv | No transformation |
| gd2 < gd1 and gv < gh | Diagonal |
| gd1 < gd2 and gh < gv | Vertical flip |
| gd1 < gd2 and gv < gh | Rotation |

#### Filtering process

At decoder side, when ALF is enabled for a CTB, each sample within the CU is filtered, resulting in sample value as shown below,

(3-67)

where denotes the decoded filter coefficients, is the clipping function and denotes the decoded clipping parameters. The variable k and l varies between and where *L* denotes the filter length. The clipping function which corresponds to the function The clipping operation introduces non-linearity to make ALF more efficient by reducing the impact of neighbor sample values that are too different with the current sample value.

#### Cross component adaptive loop filter

CC-ALF uses luma sample values to refine each chroma component by applying an adaptive, linear filter to the luma channel and then using the output of this filtering operation for chroma refinement. Figure 49 (a) provides a system level diagram of the CC-ALF process with respect to the SAO, luma ALF and chroma ALF processes.

Filtering in CC-ALF is accomplished by applying a linear, diamond shaped filter (Figure 49 (b)) to the luma channel. One filter is used for each chroma channel, and the operation is expressed as

(3-68)

where is chroma component *i* location being refined is the luma location based on , is filter support area in luma component,

As shown in Figure 51b, the luma filter support is the region collocated with the current chroma sample after accounting for the spatial scaling factor between the luma and chroma planes.

|  |  |
| --- | --- |
|  |  |

(a) (b)

Figure 49 – (a) Placement of CC-ALF with respect to other loop filters (b) Diamond shaped filter

In the VVC reference software, CC-ALF filter coefficients are computed by minimizing the mean square error of each chroma channels with respect to the original chroma content. To achieve this, the VTM algorithm uses a coefficient derivation process similar to the one used for chroma ALF. Specifically, a correlation matrix is derived, and the coefficients are computed using a Cholesky decomposition solver in an attempt to minimize a mean square error metric. In designing the filters, a maximum of 8 CC-ALF filters can be designed and transmitted per picture. The resulting filters are then indicated for each of the two chroma channels on a CTU basis.

Additional characteristics of CC-ALF include:

* The design uses a 3x4 diamond shape with 8 taps.
* Seven filter coefficients are transmitted in the APS.
* Each of the transmitted coefficients has a 6-bit dynamic range and is restricted to power-of-2 values.
* The eighth filter coefficient is derived at the decoder such that the sum of the filter coefficients is equal to 0.
* An APS may be referenced in the slice header.
* CC-ALF filter selection is controlled at CTU-level for each chroma component
* Boundary padding for the horizontal virtual boundaries uses the same memory access pattern as luma ALF.

As an additional feature, the reference encoder can be configured to enable some basic subjective tuning through the configuration file. When enabled, the VTM attenuates the application of CC-ALF in regions that are coded with high QP and are either near mid-grey or contain a large amount of luma high frequencies. Algorithmically, this is accomplished by disabling the application of CC-ALF in CTUs where any of the following conditions are true:

* The slice QP value minus 1 is less than or equal to the base QP value
* The number of chroma samples for which the local contrast is greater than ( 1 << ( bitDepth – 2 ) ) – 1 exceeds the CTU height, where the local contrast is the difference between the maximum and minimum luma sample values within the filter support region.
* More than a quarter of chroma samples are in the range between ( 1 << ( bitDepth – 1 ) ) – 16 and ( 1 << ( bitDepth – 1 ) ) + 16

The motivation for this functionality is to provide some assurance that CC-ALF does not amplify artifacts introduced earlier in the decoding path (This is largely due the fact that the VTM currently does not explicitly optimize for chroma subjective quality). It is anticipated that alternative encoder implementations would either not use this functionality or incorporate alternative strategies suitable for their encoding characteristics.

#### Filter parameters signalling

ALF filter parameters are signalled in Adaptation Parameter Set (APS). In one APS, up to 25 sets of luma filter coefficients and clipping value indexes, and up to eight sets of chroma filter coefficients and clipping value indexes could be signalled. To reduce bits overhead, filter coefficients of different classification for luma component can be merged. In slice header, the indices of the APSs used for the current slice are signaled.

Clipping value indexes, which are decoded from the APS, allow determining clipping values using a table of clipping values for both luma and Chroma components. These clipping values are dependent of the internal bitdepth. More precisely, the clipping values are obtained by the following formula:

AlfClip (3-69)

with B equal to the internal bitdepth, α is a pre-defined constant value equal to 2.35, and N equal to 4 which is the number of allowed clipping values in VVC. The AlfClip is then rounded to the nearest value with the format of power of 2.

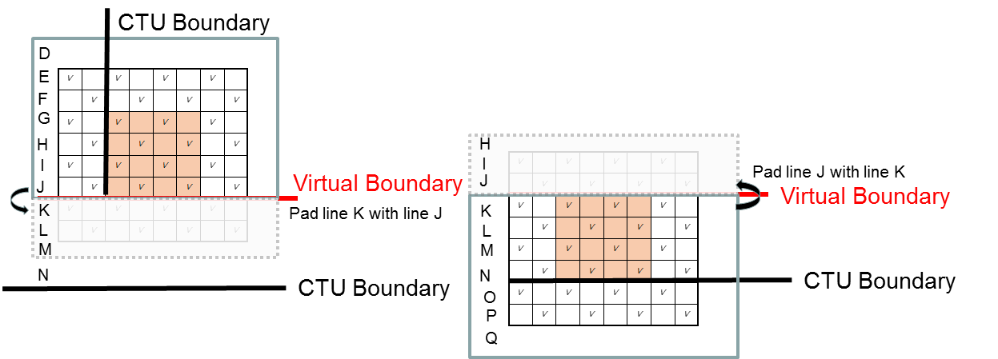
In slice header, up to 7 APS indices can be signaled to specify the luma filter sets that are used for the current slice. The filtering process can be further controlled at CTB level. A flag is always signalled to indicate whether ALF is applied to a luma CTB. A luma CTB can choose a filter set among 16 fixed filter sets and the filter sets from APSs. A filter set index is signaled for a luma CTB to indicate which filter set is applied. The 16 fixed filter sets are pre-defined and hard-coded in both the encoder and the decoder.

For chroma component, an APS index is signaled in slice header to indicate the chroma filter sets being used for the current slice. At CTB level, a filter index is signaled for each chroma CTB if there is more than one chroma filter set in the APS.

The filter coefficients are quantized with norm equal to 128. In order to restrict the multiplication complexity, a bitstream conformance is applied so that the coefficient value of the non-central position shall be in the range of −27 to 27 − 1, inclusive. The central position coefficient is not signalled in the bitstream and is considered as equal to 128.

#### Virtual boundary filtering process for line buffer reduction

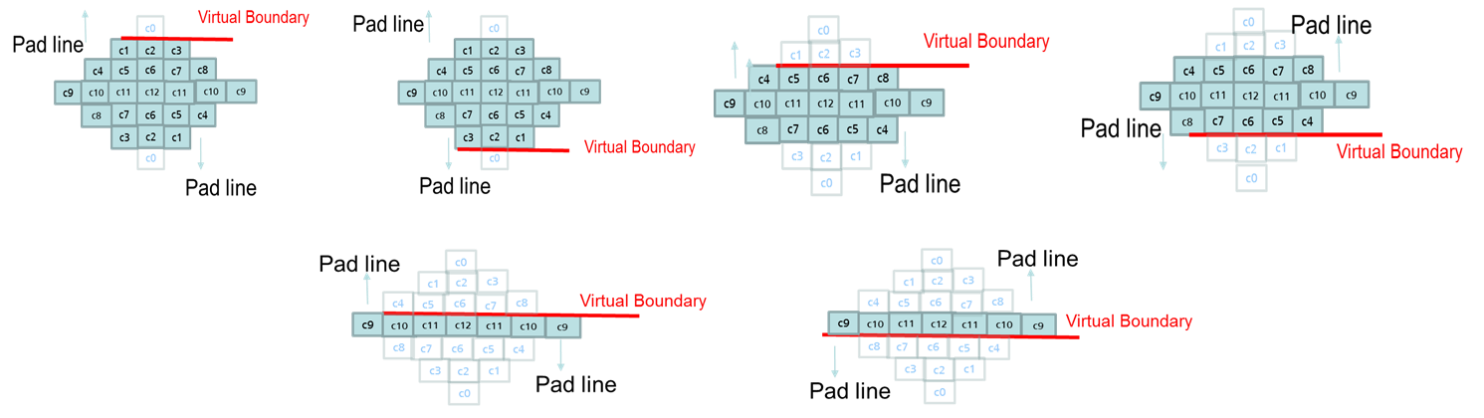
In VVC, to reduce the line buffer requirement of ALF, modified block classification and filtering are employed for the samples near horizontal CTU boundaries. For this purpose, a virtual boundary is defined as a line by shifting the horizontal CTU boundary with “N” samples as shown in Figure 50, with N equal to 4 for the Luma component and 2 for the Chroma component.



**Figure 50 – Modified block classification at virtual boundaries**

Modified block classification is applied for the Luma component as depicted in Figure 50. For the 1D Laplacian gradient calculation of the 4x4 block above the virtual boundary, only the samples above the virtual boundary are used. Similarly for the 1D Laplacian gradient calculation of the 4x4 block below the virtual boundary, only the samples below the virtual boundary are used. The quantization of activity value A is accordingly scaled by taking into account the reduced number of samples used in 1D Laplacian gradient calculation.

For filtering processing, symmetric padding operation at the virtual boundaries are used for both Luma and Chroma components. As shown in Figure 51, when the sample being filtered is located below the virtual boundary, the neighboring samples that are located above the virtual boundary are padded. Meanwhile, the corresponding samples at the other sides are also padded, symmetrically.



**Figure 51 – Modified ALF filtering for Luma component at virtual boundaries**

Different to the symmetric padding method used at horizontal CTU boundaries, simple padding process is applied for slice, tile and subpicture boundaries when filter across the boundaries is disabled. The simple padding process is also applied at picture boundary. The padded samples are used for both classification and filering process. To compensate for the extreme padding when filtering samples just above or below the virtual boundary the filter strength is reduced for those cases for both Luma and Chroma by increasing the right shift in equation 3-61 by 3.