##### 8.6.2.1.1 Modification process for luma and chroma samples

Inputs to this process are:

– a variable cIdx specifying chroma component index,

– a sample position ( xC, yC ),

– a pair of variables ( rx, ry ) specifying the region index,

– a variable saoDepth specifying the split depth of the region,

– an array saoValueArray specifying offset values,

– a block size nS.

Output of this process is a modified picture buffer for the chroma component cIdx.

Let recSaoPicture represents the processed sample array of the current picture of chroma component cIdx and saoTypeIdx is set equal to sao\_type\_idx[ cIdx ][ saoDepth ][ rx ][ ry ].

Variable bitDepth is set equal to BitDepthY if cIdx is equal to 0, otherwise, set equal to BitDepthC.

Depending on the value of saoTypeIdx, the following applies:

– If saoTypeIdx is equal to one of the values of 1, 2, 3 or 4, the following ordered steps apply:

1. Arrays hPos[2] and vPos[2] are specified in Table 8‑16.
2. A variable edgeIdx is specified as

If loop\_filter\_across\_slice\_flag is equal to 0,

If recPicture[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] with both k = 0 and k =1 is out of slice boundary.

If saoTypeIdx equals to 3

If recPicture[ xC + i, yC + j - 1 ] inside the slice boundary

recPicture’[ xC + i -1, yC + j - 1 ] = recPicture[ xC + i, yC + j - 1 ]

otherwise,

recPicture’[ xC + i +1, yC + j + 1 ] = recPicture[ xC + i, yC + j + 1 ]

If saoTypeIdx equals to 4

If recPicture[ xC + i, yC + j - 1 ] inside the slice boundary

recPicture’[ xC + i + 1, yC + j - 1 ] = recPicture[ xC + i, yC + j - 1 ]

otherwise,

recPicture’[ xC + i - 1, yC + j + 1 ] = recPicture[ xC + i, yC + j + 1 ]

otherwise, if recPicture[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] with k = 0 is out of slice boundary.

recPicture’[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] = recPicture[ xC + i , yC + j ]

otherwise, if recPicture[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] with k =1 is out of slice boundary.

recPicture’[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] = recPicture[ xC + i , yC + j ]

Otherwise ,

recPicture’[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] = recPicture[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] with k = 0..1

edgeIdx = 2 + ∑k( Sign( recPicture[ xC + i, yC + j ] –

recPicture’[ xC + i + hPos[ k ], yC + j + vPos[ k ] ] ) ) with k = 0..1 (8‑463)

1. The reconstructed picture buffer is modified as

recSaoPicture[ xC + i, yC + j ] = recPicture[ xC + i, yC + j ] + saoValueArray[ edgeTable[ edgeIdx ] ] (8‑463)

with i = 0..nS-1 and j = 0..nS-1 where edgeTable[5] = { 1, 2, 0, 3, 4}.

– Otherwise, if saoTypeIdx is equal to one of the values of 5 or 6, the following ordered steps applies:

1. A variable bandShift is set equal to BitDepthY – 5 if cIdx is equal to 0, otherwise, set equal to BitDepthC – 5.
2. The reconstructed picture buffer is modified as

recSaoPicture[ xC + i, yC + j ] = recPicture[ xC + i, yC + j ] +

saoValueArray[ bandTable[ saoTypeIdx – 5 ][ bandIdx ] ] (8‑463)

with i = 0..nS-1 and j = 0..nS-1 where bandIdx is set equal to ( recPicture[ xC + i, yC + j ] >> bandShift ) and bandTable is specified in .

– Otherwise (sao\_type\_idx[ cIdx ][ saoDepth ][ rx ][ ry ] is equal to 0), the following applies:

recSaoPicture[ xC + i, yC + j ] = recPicture[ xC + i, yC + j ] with i = 0..nS-1 and j = 0..nS-1 (8‑463)

[Ed. (WJ): copy operation is necessary to use recSaoPicture later.]

Table 8‑16 – Specification of hPos[2] and vPos[2] according to the type of sample adaptive offset process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sao\_type\_idx[ cIdx ][ saoDepth ][ rx ][ ry ] | 1 | 2 | 3 | 4 |
| hPos[0] | -1 | 0 | -1 | 1 |
| hPos[1] | 1 | 0 | 1 | -1 |
| vPos[0] | 0 | -1 | -1 | 1 |
| vPos[1] | 0 | 1 | 1 | -1 |

Table 8‑17 – Specification of array bandTable according to bandIdx

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| bandIdx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| bandTable[0][bandIdx] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| bandTable[1][bandIdx] | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| bandIdx | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| bandTable[0][bandIdx] | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| bandTable[1][bandIdx] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |