# Band Boundary Processing

* Otherwise, , if saoTypeIdx is equal to 5, the following ordered steps apply:

1. If cIdx is equal to 0, the variable bandShift is set equal to BitDepthY − 5
2. Otherwise (cIdx is not equal to 0), bandShift is set equal to BitDepthC − 5.
3. The variable saoLeftClass is set equal to sao\_band\_position[ cIdx ][ rx ][ ry ].
4. The vector bandTable is defined with 32 elements and all elements are initially set to 0. Then, 4 of its elements (indicating the position of bands for which an offset is explicitly transmitted) are modified as follows:

for( i = 0; i < 4; i++ )  
 bandTable[ (i + saoLeftClass) & 0x1F ] = i + 1 (8‑354)

1. The reconstructed picture buffer recSaoPicture is modified as follows.

bandIdx = ( recPicture[ xC + i, yC + j ] >> bandShift )

d = 4

boundaryIdx = (( recPicture[ xC + i, yC + j ] -  (bandIdx <<bandShift ))

* if (boundaryIdx >= -d/2 && boundaryIdx<d/2)

bandIdx2 = (boundaryIdx >= 0) ? bandIdx-1 : bandIdx+1

boundaryIdx += (boundaryIdx >= 0) ? boundaryIdx : -boundaryIdx +1

offset = (saoValueArray[ bandTable[ bandIdx ] ] \* smoothTable[boundaryIdx] + saoValueArray[ bandTable[ bandIdx2 ] ] \*((1<<8)- smoothTable[boundaryIdx]))>>8

where smoothTable[] is a pre-tabulated tables obtained as

for (i = 1; i <= d/2; i++)

smoothTable[i-1] = (i << 8) / (d+1)

* otherwise

offset = saoValueArray[ bandTable[ bandIdx ] ]

recSaoPicture[ xC + i, yC + j ] = recPicture[ xC + i, yC + j ] + offset (8‑355)  
 with i = 0..nS−1 and j = 0..nS−1

– Otherwise (sao\_type\_idx[ cIdx ][ 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 (8‑356)

Table 8‑13 – 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 |

# LCU Boundary Processing

* Otherwise, , if saoTypeIdx is equal to 5, the following ordered steps apply:

1. If cIdx is equal to 0, the variable bandShift is set equal to BitDepthY − 5
2. Otherwise (cIdx is not equal to 0), bandShift is set equal to BitDepthC − 5.
3. The variable saoLeftClass is set equal to sao\_band\_position[ cIdx ][ rx ][ ry ].
4. The vector bandTable is defined with 32 elements and all elements are initially set to 0. Then, 4 of its elements (indicating the position of bands for which an offset is explicitly transmitted) are modified as follows:

for( i = 0; i < 4; i++ )  
 bandTable[ (i + saoLeftClass) & 0x1F ] = i + 1 (8‑354)

1. The reconstructed picture buffer recSaoPicture is modified as follows.

bandIdx = ( recPicture[ xC + i, yC + j ] >> bandShift )

d = 2

if ( (i >= d && i < nS-d) && (j >= d && j < nS-d))

recSaoPicture[ xC + i, yC + j ] = recPicture[ xC + i, yC + j ] + saoValueArray[ bandTable[ bandIdx ] ] (8‑355)  
 with i = 0..nS−1 and j = 0..nS−1

otherwise

if (j < d && !merge\_up)

recSaoPicture[ xC + i, yC + j ] = (saoValueArray[ bandTable[ bandIdx ] ] \* smoothTable[j]

else if (j >= nS-d && !merge\_bottom)

recSaoPicture[ xC + i, yC + j ] = (saoValueArray[ bandTable[ bandIdx ] ] \* smoothTable[j-nS-d]

if (i < d && !merge\_left)

recSaoPicture[ xC + i, yC + j ] = (saoValueArray[ bandTable[ bandIdx ] ] \* smoothTable[i]

else if (i >= nS-d && !merge\_right)

recSaoPicture[ xC + i, yC + j ] = (saoValueArray[ bandTable[ bandIdx ] ] \* smoothTable[i-nS-d]

where smoothTable[] is a pre-tabulated tables obtained as

for (i = 1; i <= d/2; i++)

smoothTable[i-1] = (i << 8) / (d+1)

– Otherwise (sao\_type\_idx[ cIdx ][ 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 (8‑356)

Table 8‑13 – 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 |