#### Sample adaptive offset parameter syntax

|  |  |
| --- | --- |
| sao\_offset\_param( rx, ry, saoDepth , cIdx ) { | Descriptor |
| if( sao\_split\_flag[ cIdx ][ saoDepth ][ rx ][ ry ] ) { |  |
| sao\_offset\_param( 2\*rx + 0, 2\*ry + 0, saoDepth + 1 , cIdx ) |  |
| sao\_offset\_param( 2\*rx + 1, 2\*ry + 0, saoDepth + 1 , cIdx ) |  |
| sao\_offset\_param( 2\*rx + 0, 2\*ry + 1, saoDepth + 1 , cIdx ) |  |
| sao\_offset\_param( 2\*rx + 1, 2\*ry + 1, saoDepth + 1 , cIdx ) |  |
| } else { |  |
| **sao\_type\_idx**[ cIdx ][ saoDepth ][ rx ][ ry ] | ue(v) | ae(v) |
| if( sao\_type\_idx[ cIdx ][ saoDepth ][ rx ][ ry ] ==5) { |  |
| **sao\_first\_band** [ cIdx ][ saoDepth ][ x0 ][ y0 ] | u(v) |
| } |  |
| if( sao\_type\_idx[ cIdx ][ saoDepth ][ rx ][ ry ] != 0 ) |  |
| for( i = 0; i < NumSaoClass[ sao\_type\_idx ]; i++ ) |  |
| **sao\_offset**[ cIdx ][ saoDepth ][ x0 ][ y0 ][ i ] | se(v) | ae(v) |
| } |  |
| } |  |

#### Sample adaptive offset parameter semantics

**sao\_flag\_cb** equal to 1 denotes sample adaptive offset process for Cb shall be applied to the current picture.

**sao\_flag\_cr** equal to 1 denotes sample adaptive offset process for Cr shall be applied to the current picture.

**sao\_split\_flag**[ cIdx ][ saoDepth ][ rx ][ ry ] specifies whether a region is split into four sub regions with half horizontal and vertical number of LCU for the color component cIdx. The array indices rx and ry specify the region index and saoDepth specifies the split depth of the region. When sao\_split\_flag[ cIdx ][ saoDepth ][ rx ][ ry ] is not present, it shall be inferred to be equal to 0.

The maximum allowed depth for sample adaptive offset process SaoMaxDepth is derived as follows:

SaoMaxDepth = Min( 4, Min( Floor( Log2( PicWidthInLCUs ) ), Floor( Log2( PicHeightInLCUs ) ) ) ) (7‑10)

where

PicWidthInLCUs = Ceil( PicWidthInSamplesL ÷ ( 1 << Log2MaxCUSize ) ) (7‑10)  
PicHeightInLCUs = Ceil( PicHeightInSamplesL ÷ ( 1 << Log2MaxCUSize ) ) (7‑10)

[Ed.: (WJ) PicWidthInLCUs and PicHeightInLCUs could be defined elsewhere]

**sao\_type\_idx**[ cIdx ][ saoDepth ][ rx ][ ry ] indicates the offset type for the color component cIdx of the region specified by saoDepth, rx and ry.

**sao\_first\_band**[ cIdx ][ saoDepth ][ rx ][ ry ] indicates the start of the interval when sao\_type\_idx[ cIdx ][ saoDepth ][ rx ][ ry ] is equal to 5 (BO MODE).

**sao\_offset**[ cIdx ][ saoDepth ][ rx ][ ry ][ i ] indicates the offset value of i-th category for the color component cIdx of the region specified by saoDepth, rx and ry.

The variable bitDepth is derived as follows.

* If cIdx is equal to 0, bitDepth is set equal to BitDepthY..
* Otherwise (cIdx is equal to1 or 2), bitDepth is set equal to BitDepthC.

The offset value shall be in the range of [ -( 1 << ( SaoBitsRange – 1) ), ( 1<< ( SaoBitsRange – 1) ) – 1 ] where

SaoBitRange = Min( bitDepth, 10 ) – 4 (7‑10)

An array SaoOffsetVal is specified as

SaoOffsetVal[ cIdx ][ saoDepth ][ rx ][ ry ][ 0 ] = 0 (7‑10)

SaoOffsetVal[ cIdx ][ saoDepth ][ rx ][ ry ][ i + 1 ] =

sao\_offset[ cIdx ][ saoDepth ][ rx ][ ry ][ i ] << ( bitDepth – Min( bitDepth, 10 ) )

with i = 0..NumSaoCategory – 1 (7‑10)

The number of categories, NumSaoClass, is specified in .

Table 7‑8 – Specification of NumSaoClass

|  |  |  |
| --- | --- | --- |
| **sao\_type\_idx[ cIdx ][ saoDepth ][ rx ][ ry ]** | **NumSaoCategory** | **Edge type (informative)** |
| 0 | 0 | Not applied |
| 1 | 4 | 1D 0-degree edge |
| 2 | 4 | 1D 90-degree edge |
| 3 | 4 | 1D 135-degree edge |
| 4 | 4 | 1D 45-degree edge |
| 5 | 4 | Band offset 1/8th of range |

##### 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 sao\_first\_band specifying the start of bands for which offset is transmitted,

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 .
2. A variable edgeIdx is specified as

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 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 [ 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 .

bandTable [ bandIdx ] = (**sao\_band\_start+i), where** ** **. For all other values of** bandIdx bandTable [ bandIdx ] is equal to 0.

– 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 8‑15 – 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 |



\