#### 8.7.2.2 Derivation process of transform block boundary

Inputs to this process are:

– a luma location ( xCb, yCb ) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,

– a luma location ( xB0, yB0 ) specifying the top-left sample of the current luma block relative to the top‑left sample of the current luma coding block,

– a variable log2TrafoSize specifying the size of the current block,

– a variable trafoDepth,

– a variable filterEdgeFlag,

– a two-dimensional (nCbS)x(nCbS) array edgeFlags,

– a variable edgeType specifying whether a vertical (EDGE\_VER) or a horizontal (EDGE\_HOR) edge is filtered.

Output of this process is the modified two-dimensional (nCbS)x(nCbS) array edgeFlags.

Depending on the value of split\_transform\_flag[ xCb + xB0 ][ yCb + yB0 ][ trafoDepth ], the following applies:

– If split\_transform\_flag[ xCb + xB0 ][ yCb + yB0 ][ trafoDepth ] is equal to 1, the following ordered steps apply:

1. The variables xB1 and yB1 are derived as follows:
   * + The variable xB1 is set equal to xB0 + ( 1  <<  ( log2TrafoSize − 1 ) ).
     + The variable yB1 is set equal to yB0 + ( 1  <<  ( log2TrafoSize − 1 ) ).
2. The derivation process of transform block boundary as specified in this subclause is invoked with the luma location ( xCb, yCb ), the luma location ( xB0, yB0 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable filterEdgeFlag, the array edgeFlags, and the variable edgeType as inputs, and the output is the modified version of array edgeFlags.
3. The derivation process of transform block boundary as specified in this subclause is invoked with the luma location ( xCb, yCb ), the luma location ( xB1, yB0 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable filterEdgeFlag, the array edgeFlags, and the variable edgeType as inputs, and the output is the modified version of array edgeFlags.
4. The derivation process of transform block boundary as specified in this subclause is invoked with the luma location ( xCb, yCb ), the luma location ( xB0, yB1 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable filterEdgeFlag, the array edgeFlags, and the variable edgeType as inputs, and the output is the modified version of array edgeFlags.
5. The derivation process of transform block boundary as specified in this subclause is invoked with the luma location ( xCb, yCb ), the luma location ( xB1, yB1 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the variable filterEdgeFlag, the array edgeFlags, and the variable edgeType as inputs, and the output is the modified version of array edgeFlags.

– Otherwise (split\_transform\_flag[ xCb + xB0 ][ yCb + yB0 ][ trafoDepth ] is equal to 0), the following applies:

* If edgeType is equal to EDGE\_VER, the value of edgeFlags[ xB0 ][ yB0 + k ] for k = 0..( 1  <<  log2TrafoSize ) − 1 is derived as follows:
* If xB0 is equal to 0, edgeFlags[ xB0 ][ yB0 + k ] is set equal to filterEdgeFlag.
* Otherwise, edgeFlags[ xB0 ][ yB0 + k ] is set equal to 1.
* Otherwise (edgeType is equal to EDGE\_HOR), the value of edgeFlags[ xB0 + k ][ yB0 ] and edgeFlags[ xB0 + k ][ yB0 + ( ( 1 << log2TrafoSize ) >> 1 ) ] for k = 0..( 1  <<  log2TrafoSize ) − 1 ~~is~~ are derived as follows:
* If yB0 is equal to 0, edgeFlags[ xB0 + k ][ yB0 ] is set equal to filterEdgeFlag.
* Otherwise, edgeFlags[ xB0 + k ][ yB0 ] is set equal to 1.
* If ChromaArrayType is equal to 2, edgeFlags[ xB0 + k ][ yB0 + ( ( 1 << log2TrafoSize ) >> 1 ) ] is set equal to 2.

#### 

#### 8.7.2.4 Derivation process of boundary filtering strength

Inputs to this process are:

– a luma picture sample array recPictureL,

– a luma location ( xCb, yCb ) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,

– a variable log2CbSize specifying the size of the current luma coding block,

– a variable edgeType specifying whether a vertical (EDGE\_VER) or a horizontal (EDGE\_HOR) edge is filtered,

– a two-dimensional (nCbS)x(nCbS) array edgeFlags.

Output of this process is a two-dimensional (nCbS)x(nCbS) array bS specifying the boundary filtering strength.

The variables xDi, yDj, xN, and yN are derived as follows:

* If edgeType is equal to EDGE\_VER, xDi is set equal to ( i  <<  3 ), yDj is set equal to ( j  <<  2 ), xN is set equal to ( 1  <<  ( log2CbSize − 3 ) ) − 1, and yN is set equal to ( 1  <<  ( log2CbSize − 2 ) ) − 1.
* Otherwise (edgeType is equal to EDGE\_HOR), xDi is set equal to ( i  <<  2 ), yDj is set equal to ( j  <<  3 ), xN is set equal to ( 1  <<  ( log2CbSize − 2 ) ) − 1, and yN is set equal to ( 1  <<  ( log2CbSize − 3 ) ) − 1.

For xDi with i = 0..xN and yDj with j = 0..yN, the following applies:

* If edgeFlags[ xDi ][ yDj ] is equal to 0, the variable bS[ xDi ][ yDj ] is set equal to 0.
* Otherwise (edgeFlags[ xDi ][ yDj ] is greater than 0~~equal to 1~~), the following applies:
* The sample values p0 and q0 are derived as follows:
  + - If edgeType is equal to EDGE\_VER, p0 is set equal to recPictureL[ xCb + xDi − 1 ][ yCb + yDj ] and q0 is set equal to recPictureL[ xCb + xDi ][ yCb + yDj ].
    - Otherwise (edgeType is equal to EDGE\_HOR), p0 is set equal to recPictureL[ xCb + xDi ][ yCb + yDj − 1 ] and q0 is set equal to recPictureL[ xCb + xDi ][ yCb + yDj ].
* The variable bS[ xDi ][ yDj ] is derived as follows:
* If the sample p0 or q0 is in the luma coding block of a coding unit coded with intra prediction mode, the following applies.
  + - * If edgeFlags[ xD­i ][ yDj ] is equal to 1, bS[ xDi ][ yDj ] is set equal to 2.
      * Otherwise (edgeFlags[ xD­i ][ yDj ] is equal to 2), the variable bS[ xDi ][ yDj ] is set equal to 3.
    - Otherwise, if the block edge is also a transform block edge and the sample p0 or q0 is in a luma transform block which contains one or more non-zero transform coefficient levels, bS[ xDi ][ yDj ] is set equal to 1.
    - Otherwise, if edgeFlags[ xDi ][ yDj ] is equal to 1 and one or more of the following conditions are true, bS[ xDi ][ yDj ] is set equal to 1:
      * For the prediction of the luma prediction block containing the sample p0 different reference pictures or a different number of motion vectors are used than for the prediction of the luma prediction block containing the sample q0.

NOTE 1 – The determination of whether the reference pictures used for the two luma prediction blocks are the same or different is based only on which pictures are referenced, without regard to whether a prediction is formed using an index into reference picture list 0 or an index into reference picture list 1, and also without regard to whether the index position within a reference picture list is different.

NOTE 2 – The number of motion vectors that are used for the prediction of a luma prediction block with top-left luma sample covering ( xPb, yPb ), is equal to PredFlagL0[ xPb ][ yPb ] + PredFlagL1[ xPb ][ yPb ].

* + - * One motion vector is used to predict the luma prediction block containing the sample p0 and one motion vector is used to predict the luma prediction block containing the sample q0, and the absolute difference between the horizontal or vertical component of the motion vectors used is greater than or equal to 4 in units of quarter luma samples.
      * Two motion vectors and two different reference pictures are used to predict the luma prediction block containing the sample p0, two motion vectors for the same two reference pictures are used to predict the luma prediction block containing the sample q0, and the absolute difference between the horizontal or vertical component of the two motion vectors used in the prediction of the two luma prediction blocks for the same reference picture is greater than or equal to 4 in units of quarter luma samples.
      * Two motion vectors for the same reference picture are used to predict the luma prediction block containing the sample p0, two motion vectors for the same reference picture are used to predict the luma prediction block containing the sample q0, and both of the following conditions are true:
        + The absolute difference between the horizontal or vertical component of list 0 motion vectors used in the prediction of the two luma prediction blocks is greater than or equal to 4 in quarter luma samples, or the absolute difference between the horizontal or vertical component of the list 1 motion vectors used in the prediction of the two luma prediction blocks is greater than or equal to 4 in units of quarter luma samples.
        + The absolute difference between the horizontal or vertical component of list 0 motion vector used in the prediction of the luma prediction block containing the sample p0 and the list 1 motion vector used in the prediction of the luma prediction block containing the sample q0 is greater than or equal to 4 in units of quarter luma samples, or the absolute difference between the horizontal or vertical component of the list 1 motion vector used in the prediction of the luma prediction block containing the sample p0 and list 0 motion vector used in the prediction of the luma prediction block containing the sample q0 is greater than or equal to 4 in units of quarter luma samples.
    - Otherwise, the variable bS[ xDi ][ yDj ] is set equal to 0.

##### 8.7.2.5.2 Horizontal edge filtering process

Inputs to this process are:

– the picture sample array recPictureL, and when ChromaArrayType is not equal to 0, the arrays recPictureCb, and recPictureCr,

– a luma location ( xCb, yCb ) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,

– a variable log2CbSize specifying the size of the current luma coding block,

– an array bS specifying the boundary filtering strength.

Outputs of this process are the modified picture sample array recPictureL, and when ChromaArrayType is not equal to 0, the arrays recPictureCb, and recPictureCr.

The filtering process for edges in the luma coding block of the current coding unit consists of the following ordered steps:

1. The variable nD is set equal to 1  <<  ( log2CbSize − 3 ).
2. For yDm equal to m  <<  3 with m = 0..nD − 1, and xDk equal to k  <<  2 with k = 0..nD \* 2 − 1, the following applies:

* When bS[ xDk ][ yDm ] is greater than 0 and less than 3, the following ordered steps apply:

1. The decision process for luma block edges as specified in subclause 8.7.2.5.3 is invoked with the luma picture sample array recPictureL, the location of the luma coding block ( xCb, yCb ), the luma location of the block ( xDk, yDm ), a variable edgeType set equal to EDGE\_HOR, and the boundary filtering strength bS[ xDk ][ yDm ] as inputs, and the decisions dE, dEp, and dEq, and the variables β and tC as outputs.
2. The filtering process for luma block edges as specified in subclause 8.7.2.5.4 is invoked with the luma picture sample array recPictureL, the location of the luma coding block ( xCb, yCb ), the luma location of the block ( xDk, yDm ), a variable edgeType set equal to EDGE\_HOR, the decisions dEp, dEp, and dEq, and the variables β and tC as inputs, and the modified luma picture sample array recPictureL as output.

When ChromaArrayType is not equal to 0, the following applies.

The filtering process for edges in the chroma coding blocks of current coding unit consists of the following ordered steps:

1. The variable nD is set equal to 1  <<  ( log2CbSize − 3 ).
2. The variable EdgeSpacing is set equal to 8 / SubHeightC.
3. The variable EdgeSections is set equal to nD \* ( 2 / SubWidthC ).
4. For yDm equal to m \* EdgeSpacing with m = 0..nD − 1 and xDk equal to k  <<  2 with k = 0..EdgeSections − 1, the following applies:

* When bS[ xDk \* SubWidthC ][ yDm \* SubHeightC ] is equal to 2 and ( ( ( yCb / SubHeightC + yDm )  >>  3 )  <<  3 ) is equal to yCb / SubHeightC + yDm, the following ordered steps apply:

1. The filtering process for chroma block edges as specified in subclause 8.7.2.5.5 is invoked with the chroma picture sample array recPictureCb, the location of the chroma coding block ( xCb / SubWidthC, yCb / SubHeightC ), the chroma location of the block ( xDk, yDm ), a variable edgeType set equal to EDGE\_HOR, and a variable cQpPicOffset set equal to pps\_cb\_qp\_offset as inputs, and the modified chroma picture sample array recPictureCb as output.

The filtering process for chroma block edges as specified in subclause 8.7.2.5.5 is invoked with the chroma picture sample array recPictureCr, the location of the chroma coding block ( xCb / SubWidthC, yCb / SubHeightC ), the chroma location of the block ( xDk, yDm ), a variable edgeType set equal to EDGE\_HOR, and a variable cQpPicOffset set equal to pps\_cr\_qp\_offset as inputs, and the modified chroma picture sample array recPictureCr as output.