##### Derivation process for motion vector predictor candidates

Inputs to this process are

* a luma location ( xC, yC ) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
* a variable nCS specifying the size of the current luma coding block,
* a luma location ( xP, yP ) specifying the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture,
* variables specifying the width and the height of the luma prediction block, nPbW and nPbH,
* the reference index of the current prediction unit partition refIdxLX (with X being 0 or 1),
* a variable partIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are (with N being replaced by A, or B)

* the motion vectors mvLXN of the neighbouring prediction units,
* the availability flags availableFlagLXN of the neighbouring prediction units.



Figure 8‑3 – Spatial motion vector neighbours (informative)

The variable isScaledFlagLX with X being 0 or 1 is set equal to 0.

The motion vector mvLXA and the availability flag availableFlagLXA are derived in the following ordered steps:

1. The sample location (xA0, yA0) is set equal to (xP − 1, yP + nPbH) and the sample location (xA1, yA1) is set equal to (xA0, yA0 − 1).
2. The availability flag availableFlagLXA is set equal to 0 and the both components of mvLXA are set equal to 0.
3. The availability derivation process for a prediction block as specified in subclause 6.4.2 is invoked with the luma location ( xC, yC ), the current luma coding block size nCbS set equal to nCS, the luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, the luma location ( xN, yN ) set equal to ( xA0, yA0 ) and the partition index partIdx as inputs and the output is assigned to the prediction block availability flag availableA0.
4. The availability derivation process for a prediction block as specified in subclause 6.4.2 is invoked with the luma location ( xC, yC ), the current luma coding block size nCbS set equal to nCS, the luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, the luma location ( xN, yN ) set equal to ( xA1, yA1 ) and the partition index partIdx as inputs and the output is assigned to the prediction block availability flag availableA1.
5. When availableA0 or availableA1 is equal to TRUE, the variable isScaledFlagLX is set equal to 1.
6. For ( xAk, yAk ) from ( xA0, yA0 ) to ( xA1, yA1 ), the following applies repeatedly until availableFlagLXA is equal to 1:

* When availableAk is equal to TRUE , PredMode[ xAk][ yAk ] is not equal to MODE\_INTRA and availableFlagLXA is equal to 0, the following applies.
* If, PredFlagLX[ xAk ][ yAk ] is equal to 1 and the reference index refIdxLX[ xAk ][ yAk ] is equal to the reference index of the current prediction unit refIdxLX, availableFlagLXA is set equal to 1 and the following assignments are made.

mvLXA = MvLX[ xAk ][ yAk ] (8‑116)  
refIdxA = RefIdxLX[ xAk ][ yAk ] (8‑117)

* Otherwise, if PredFlagLY[ xAk ][ yAk ] (with Y = !X) is equal to 1 and PicOrderCnt( RefPicListY[ refIdxLY[ xAk ][ yAk ] ] ) is equal to PicOrderCnt( RefPicListX[ refIdxLX ] ), availableFlagLXA is set equal to 1 and the following assignments are made.

mvLXA = MvLY[ xAk ][ yAk ] (8‑118)  
refIdxA = RefIdxLY[ xAk ][ yAk ] (8‑119)

1. When availableFlagLXA is equal to 0, for ( xAk, yAk ) from ( xA0, yA0 ) to ( xA1, yA1 ), the following applies repeatedly until availableFlagLXA is equal to 1:

* When availableAk is equal to TRUE , PredMode[ xAk][ yAk ] is not equal to MODE\_INTRA and availableFlagLXA is equal to 0, the following applies.
* If PredFlagLX[ xAk ][ yAk ] is equal to 1 and LongTermRefPic( currPic, refIdxLX, RefPicListX) is equal to LongTermRefPic( currPic, RefIdxLX [ xAk ][ yAk ], RefPicListX), availableFlagLXA is set equal to 1 and the following assignments are made.

mvLXA = MvLX[ xAk ][ yAk ] (8‑120)  
refIdxA = RefIdxLX[ xAk ][ yAk ] (8‑121)  
refPicListA = RefPicListX (8‑122)

* Otherwise, if PredFlagLY[ xAk ][ yAk ] (with Y = !X) is equal to 1 and LongTermRefPic( currPic, refIdxLX, RefPicListX) is equal to LongTermRefPic( currPic, RefIdxLY[ xAk ][ yAk ], RefPicListY), availableFlagLXA is set equal to 1 and the following assignments are made.

mvLXA = MvLY[ xAk ][ yAk ] (8‑123)  
refIdxA = RefIdxLY[ xAk ][ yAk ] (8‑124)  
refPicListA = RefPicListY (8‑125)

* When availableFlagLXA is equal to 1, and both refPicListA[ refIdxA ] and RefPicListX[ refIdxLX ] are short-term reference pictures, mvLXA is derived as specified below.

tx = ( 16384 + ( Abs( td ) >> 1 ) ) / td (8‑126)

distScaleFactor = Clip3( −4096, 4095, ( tb \* tx + 32 ) >> 6 ) (8‑127)

mvLXA = Clip3( −32768, 32767, Sign2( distScaleFactor \* mvLXA ) \*    
 ( (Abs( distScaleFactor \* mvLXA ) + 127 ) >> 8 ) ) (8‑128)

where td and tb are derived as

td = Clip3( −128, 127, PicOrderCntVal – PicOrderCnt( refPicListA[ refIdxA ] ) ) (8‑129)

tb = Clip3( −128, 127, PicOrderCntVal – PicOrderCnt( RefPicListX[ refIdxLX ] ) ) (8‑130)

The motion vector mvLXB and the availability flag availableFlagLXB are derived in the following ordered steps:

1. Let a set of three sample location (xBk, yBk), with k = 0,1,2, specifies sample locations with xB0 = xP + nPbW, xB1 = xB0− 1, xB2 = xP − 1 and yBk = yP − 1. The set of sample locations ( xBk, yBk ) represent the sample locations immediately to the upper side of the above partition boundary and its extended line.
2. When yP−1 is less than (( yC >> Log2CtbSizeY ) << Log2CtbSizeY), the following applies.

xB0 = (xB0>>3)<<3) + ((xB0>>3)&1)\*7 (8‑131)  
xB1 = (xB1>>3)<<3) + ((xB1>>3)&1)\*7 (8‑132)  
xB2 = (xB2>>3)<<3) + ((xB2>>3)&1)\*7 (8‑133)

1. The availability flag availableFlagLXB is set equal to 0 and the both components of mvLXB are set equal to 0.
2. For ( xBk, yBk ) from ( xB0, yB0 ) to ( xB2, yB2 ) where xB0 = xP + nPbW, xB1 = xB0 − 1, and xB2 =  xP − 1, the following applies repeatedly until availableFlagLXB is equal to 1:

* The availability derivation process for a prediction block as specified in subclause 6.4.2 is invoked with the luma location ( xC, yC ), the current luma coding block size nCbS set equal to nCS, the luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, the luma location ( xN, yN ) set equal to ( xBk, yBk ) and the partition index partIdx as inputs and the output is assigned to the prediction block availability flag availableBk.
* When availableBk is equal to TRUE and availableFlagLXB is equal to 0, the following applies.
* If PredFlagLX[ xBk ][ yBk ] is equal to 1, and the reference index refIdxLX[ xBk ][ yBk ] is equal to the reference index of the current prediction unit refIdxLX, availableFlagLXB is set equal to 1 and the following assignments are made.

mvLXB = MvLX[ xBk ][ yBk ] (8‑134)  
refIdxB = RefIdxLX[ xBk ][ yBk ] (8‑135)

* Otherwise, if PredFlagLY[ xBk ][ yBk ] (with Y = !X) is equal to 1 and PicOrderCnt( RefPicListY[ refIdxLY[ xBk ][ yBk ] ] ) is equal to PicOrderCnt( RefPicListX[ refIdxLX ] ), availableFlagLXB is set equal to 1 and the following assignments are made.

mvLXB = MvLY[ xBk ][ yBk ] (8‑136)  
refIdxB = RefIdxLY[ xBk ][ yBk ] (8‑137)

1. When isScaledFlagLX is equal to 0 and availableFlagLXB is equal to 1, availableFlagLXA is set equal to 1 and the following assignments are made.

mvLXA = mvLXB (8‑138)  
refIdxA = refIdxLXB (8‑139)

1. When isScaledFlagLX is equal to 0, availableFlagLXB is set equal to 0 and for ( xBk, yBk ) from ( xB0, yB0 ) to ( xB2, yB2 ) where xB0 = xP + nPbW, xB1 = xB0 − 1, and xB2 = xP − 1, the following applies repeatedly until availableFlagLXB is equal to 1:

* The availability derivation process for a prediction block as specified in subclause 6.4.2 is invoked with the luma location ( xC, yC ), the current luma coding block size nCbS set equal to nCS, the luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, the luma location ( xN, yN ) set equal to ( xBk, yBk ) and the partition index partIdx as inputs and the output is assigned to the prediction block availability flag availableBk.
* When availableBk is equal to TRUE and availableFlagLXB is equal to 0, the following applies.
* If PredFlagLX[ xBk ][ yBk ] is equal to 1 and LongTermRefPic( currPic, refIdxLX, RefPicListX) is equal to LongTermRefPic( currPic, RefIdxLX [ xBk ][ yBk ], RefPicListX), availableFlagLXB is set equal to 1 and the following assignments are made.

mvLXB = MvLX[ xBk ][ yBk ] (8‑140)  
refIdxB = RefIdxLX[ xBk ][ yBk ] (8‑141)  
refPicListB = RefPicListX (8‑142)

* Otherwise, if PredFlagLY[ xBk ][ yBk ] (with Y = !X) is equal to 1 and LongTermRefPic( currPic, refIdxLX, RefPicListX) is equal to LongTermRefPic( currPic, RefIdxLY[ xBk ][ yBk ], RefPicListY), availableFlagLXB is set equal to 1 and the following assignments are made.

mvLXB = MvLY[ xBk ][ yBk ] (8‑143)  
refIdxB = RefIdxLY[ xBk ][ yBk ] (8‑144)  
refPicListB = RefPicListY (8‑145)

* When availableFlagLXB is equal to 1 and PicOrderCnt( refPicListB[ refIdxB ] ) is not equal to PicOrderCnt( RefPicListX[ refIdxLX ] ) and both refPicListB[ refIdxB ] and RefPicListX[ refIdxLX ] are short-term reference pictures, mvLXB is derived as specified below.

tx = ( 16384 + ( Abs( td ) >> 1 ) ) / td (8‑146)

distScaleFactor = Clip3( −4096, 4095, ( tb \* tx + 32 ) >> 6 ) (8‑147)

mvLXB =Clip3( −32768, 32767, Sign2( distScaleFactor \* mvLXB ) \*   
 ( (Abs( distScaleFactor \* mvLXB ) + 127 ) >> 8 ) ) (8‑148)

where td and tb are derived as

td = Clip3( −128, 127, PicOrderCntVal – PicOrderCnt( refPicListB[ refIdxB ] ) ) (8‑149)

tb = Clip3( −128, 127, PicOrderCntVal – PicOrderCnt( RefPicListX[ refIdxLX ] ) ) (8‑150)