H.8.4.4 Decoding process for intra blocks

H.8.4.4.1 General decoding process for intra blocks

Inputs to this process are:

* a sample location ( xB0, yB0 ) specifying the top-left sample of the current block relative to the top‑left sample of the current picture,
* a variable log2TrafoSize specifying the size of the current transform block,
* a variable trafoDepth specifying the hierarchy depth of the current block relative to the coding unit,
* a variable intraPredMode specifying the intra prediction mode,
* a variable cIdx specifying the chroma component of the current block.

Output of this process is:

* a modified reconstructed picture before deblocking filtering.

The variable splitFlag is derived as follows:

* If cIdx is equal to 0, splitFlag is set equal to split\_transform\_flag[ xB0 ][ yB0 ][ trafoDepth ].
* Otherwise, if all of the following conditions are true, splitFlag is set equal to 1.
  + cIdx is greater than 0
  + split\_transform\_flag[ xB0 << 1 ][ yB0 << 1 ][ trafoDepth ] is equal to 1
  + log2TrafoSize is greater than 2
* Otherwise, splitFlag is set equal to 0.

Depending splitFlag the following applies:

* If splitFlag 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 general decoding process for intra blocks as specified in this subclause is invoked with the location ( xB0, yB0 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.
  3. The general decoding process for intra blocks as specified in this subclause is invoked with the location ( xB1, yB0 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.
  4. The general decoding process for intra blocks as specified in this subclause is invoked with the location ( xB0, yB1 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.
  5. The general decoding process for intra blocks as specified in this subclause is invoked with the location ( xB1, yB1 ), the variable log2TrafoSize set equal to log2TrafoSize − 1, the variable trafoDepth set equal to trafoDepth + 1, the intra prediction mode intraPredMode, and the variable cIdx as the inputs and the output is a modified reconstructed picture before deblocking filtering.
* Otherwise, ( splitFlag is equal to 0 ), the following ordered steps apply:
  1. The variable nT is set equal to 1 << log2TrafoSize
  2. ~~The general intra sample prediction process as specified in subclause G.8.4.4.2.1 is invoked with the location ( xB0, yB0 ), the intra prediction mode intraPredMode, the transform block size nT and the variable cIdx as the inputs and the output is a (nT)x(nT) array predSamples.~~
  3. Depending on sdc\_flag, the following applies:
     + - If the sdc\_flag is equal to 0, following applies:
       - The general intra sample prediction process as specified in subclause 8.4.4.2.1 is invoked with the location ( xB0, yB0 ), the intra prediction mode intraPredMode, the transform block size nT and the variable cIdx as the inputs and the output is a (nT)x(nT) array predSamples.
       - The scaling and transformation process as specified in subclause 8.6.2 is invoked with the location ( xB0, yB0 ), the variable trafoDepth, the variable cIdx, and the transform size trafoSize set equal to nT as the inputs and the output is a (nT)x(nT) array resSamples.
       - The picture reconstruction process prior to in-loop filtering for a colour component as specified in subclause 8.6.5 is invoked with the transform block location ( xB0, yB0 ), the transform block size nT, the variable cIdx, the (nT)x(nT) array predSamples, and the (nT)x(nT) array resSamples as the inputs.
       - Otherwise ( sdc\_flag is equal to 1) the following ordered steps apply:
       - The sdc intra sample prediction process as specified in subclause H.8.4.4.2.14 is invoked with the location ( xB0, yB0 ), the prediction size nT, and the intra prediction mode intraPredMode, as the inputs and the output is the predicted constant partition values dcPred[ p ] with p = 0,1.
       - The depth value reconstruction process as specified in subclause H.8.4.4.3 is invoked with the location ( xB0, yB0 ), the transform size trafoSize set equal to nT, ~~the (nT)x(nT) array predSamples,~~ the predicted constant partition values dcPred[ p ] with p = 0,1, and the intra prediction mode intraPredMode, as the inputs and the output is a (nT)x(nT) array resSamples.
  4. For x in the range of 0 to nT − 1 and y in the range of 0 to nT − 1, the following applies:
     + - When cIdx is equal to 0, ResSamplesL[ xB0 + x ][ yB0 + y ] is set equal to 0.
       - When cIdx is equal to 1, ResSamplesCb[ xB0 + x ][ yB0 + y ] is set equal to 0.
       - When cIdx is equal to 2, ResSamplesCr[ xB0 + x ][ yB0 + y ] is set equal to 0.

H.8.4.4.2.14 Sdc intra sample prediction

Inputs to this process are:

* a luma location ( xB, yB ) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current picture,
* a variable nT specifying the prediction size,
* a variable intraPredMode specifying the prediction mode of the current prediction block,

Output of this process is the predicted constant partition values dcPred[ p ], with p = 0,1.

The nT \* 4 + 1 neighbouring samples p[ x ][ y ] that are constructed samples prior to the deblocking filter process, with x = −1, y = −1.. nT \* 2 − 1 and x = 0.. nT  \* 2 − 1, y = −1, are derived as follows:

– The neighbouring location (xN, yN ) is specified by:

( xN, yN ) = ( xB + x, yB + y ) (H-65)

* The availability derivation process for a block in z-scan order as specified in subclause  6.4.1 is invoked with the current luma location ( xB, yB ) and the neighbouring luma location ( xN, yN ) as inputs, and the output is assigned to availableN.

– Each sample p[ x ][ y ] is derived as follows:

* If one or more of the following conditions are true, the sample p[ x ][ y ] is marked as "not available for intra prediction":
  + The variable availableN is equal to FALSE.
  + CuPredMode[ xN ][ yN ] is not equal to MODE\_INTRA and constrained\_intra\_pred\_flag is equal to 1.
* Otherwise, the sample p[ x ][ y ] is marked as "available for intra prediction" and the sample at the location ( xN, yN ) is assigned to p[ x ][ y ].

When at least one sample p[ x ][ y ] with x = −1, y = −1.. nT  \* 2 − 1 and x = 0.. nT  \* 2 − 1, y = −1 is marked as "not available for intra prediction", the reference sample substitution process for intra sample prediction in subclause 8.4.4.2.2 is invoked with the samples p[ x ][ y ] with x = −1, y = −1.. nT  \* 2 − 1 and x = 0.. nT  \* 2 − 1, y = −1, nT, and cIdx set to 0 as inputs, and the modified samples p[ x ][ y ] with x = −1, y = −1..nT \* 2 − 1 and x = 0.. nT  \* 2 − 1, y = −1 as output.

Depending on the value of intraPredMode, the following ordered steps apply:

1. If intraPredMode is equal to INTRA\_PLANAR, the variable dcPred[ 0 ] is derived as specified in the following:

dcPred[ 0 ] = (H‑68)

1. Otherwise, if intraPredMode is equal to INTRA\_DC, the variable dcPred[ 0 ] is derived as specified in the following:

dcPred[ 0 ] =  (H‑66)

1. Otherwise, (intraPredMode is equal to Intra\_DepthPartition( 35 )), the variables dcPred[ p ] with p=0,1 are derived as specified in the following:
   * + The variable wedgePattern[ x ][ y ] with x, y =0..nT−1, specifying a binary partition pattern is derived as.
       1. wedgePattern = WedgePatternTable[ Log2( nT) ][ wedge\_full\_tab\_idx[ xB ][ yB ] ] (H‑67)
     + The variable log2SubSample is set equal to ( nT < 32 ) 0 : 1.
     + For p being replaced by 0 and 1, the value of dcPred[ p ] is derived as specified in the following:

* The variable sumNeigh specifying the sum of the neighbouring samples depth values of partition p is set equal to 0 and the variable numNeigh specifying the number of the neighbouring samples of partition p is set equal to 0 and the following applies.
  + - * + For x = 0..( nT >> log2SubSample ) − 1, inclusive the following applies:

When wedgePattern [ x << log2SubSample ][ 0 ] is equal to X, the following applies:

sumNeigh += p[ x<< log2SubSample ][ −1 ] (H‑68)  
numNeigh += 1 (H‑69)

* + - * + For y = 0..( nT >> log2SubSample )− 1,inclusive the following applies:

When wedgePattern [ 0 ][ y << log2SubSample ] is equal to X, the following applies:

sumNeigh += p[ −1 ][ y << log2SubSample ] (H‑70)   
numNeigh += 1 (H‑71)

* + - * The variable dcPred[p] specifying the predicted constant partition values for partition p is derived as follows.
      1. dcPred[ p ]= ( numNeigh ! = 0 ) ? ( sumNeigh / numNeigh ) : ( 1 << ( BitDepthY − 1 ) ) (H‑72)

H.8.4.4.3 Depth value reconstruction process

Inputs to this process are:

* a luma location ( xB, yB ) specifying the top-left luma sample of the current block relative to the top-left luma sample of the current picture,
* a variable nT specifying the prediction size
* ~~predicted samples predSamples[ x ][ y ], with x, y =0..nT−1~~
* predicted constant partition values dcPred[ p ], with p = 0,1
* a variable intraPredMode specifying the prediction mode of the current prediction block

Output of this process is:

* reconstructed depth value samples resSamples[ x ][ y ], with x, y = −1..2\*nT−1.

~~The values of the prediction samples predSamples[ x ][ y ], are derived as specified by the following ordered steps.~~

* 1. Depending on intraPredMode the array wedgePattern[ x ][ y ] with x, y =0..nT−1 specifying the binary segmentation pattern is derived as follows.
     + If intraPredMode is equal to Intra\_DepthPartition( 35 ), the following applies.
       - 1. wedgePattern = WedgePatternTable[ Log2( nT) ][ wedge\_full\_tab\_idx[ xB ][ yB ] ]
     + Otherwise ( intraPredMode is not equal to Intra\_DepthPartition( 35 ) ), the following applies.
       - For x, y = 0..nT−1 wedgePattern[ x ][ y ] is set equal to 0.
  2. ~~The variable log2SubSample is set equal to ( nT < 32 ) ? 0 : 1.~~
  3. ~~For p in the range of 0 to 1, inclusive, the variable dcPred[ p ] is derived as specified in the following:~~

~~sumPred = 0  
 numPred = 0.   
 for( x = 0; x < ( nT >> log2SubSample ); x++ ) {   
 x~~~~S~~  ~~= x << log2SubSample    
 for ( y = 0; y < ( nT >> log2SubSample ) ;y++ ) {   
 y~~~~S~~ ~~= y << log2SubSample .  
 if ( p = = wedgePattern[ x~~~~S~~~~][ y~~~~S~~~~] ) {  
 sumPred += predSamples[ x~~~~S~~~~][ y~~~~S~~~~]  
 numPred += 1  
 }  
 }  
 dcPred[ p ] = ( numPred > 0 ) ? ( sumPred / numPred ) : 0~~

~~[ Ed. (GT): Is this averaging necessary for DMM or DC mode? ]~~

* 1. For x, y = 0..nT−1, the reconstructed depth value samples resSamples[ x ][ y ] are derived as specified in the following.
     + If dlt\_flag[ nuh\_layer\_id ] is equal to 0, the following applies:
       - 1. resSamples[ x ][ y ] = dcPred[ wedgePattern[ x ][ y ] ] +   
             SdcResidual[ xB ][ yB ][wedgePattern[ x ][ y ] ]
     + Otherwise ( dlt\_flag[ nuh\_layer\_id ] is equal to 1 ), the following applies.
       - 1. dltIdxPred = DepthValue2Idx[ dcPred[ wedgePattern[ x ][ y ] ] ] (H‑76)  
            dltIdxResi = SdcResidual[ xB ][ yB ][wedgePattern[ x ][ y ] ] (H‑77)  
            resSamples[ x ][ y ] = Idx2DepthValue[ dltIdxPred + dltIdxResi ] (H‑78)