The proposed working draft modifications are as follows.

H.8.4.4.2.4 Specification of Intra\_Planar (0) prediction mode

If the sdc\_flag is equal to 1, this process does nothing. Else,

~~T~~the specifications in subclause 8.4.4.2.4 apply.

H.8.4.4.2.5 Specification of Intra\_DC (1) prediction mode

If the sdc\_flag is equal to 1, this process does nothing. Else,

~~T~~the specifications in subclause 8.4.4.2.5 apply.

H.8.4.4.2.7 Specification of Intra\_DepthPartition (35, 36) prediction mode

…

This intra prediction mode is invoked when intraPredMode is equal to 35 or 36.

If the sdc\_flag is equal to 1, this process does nothing. Else, the following process applies.

…

H.8.4.4.2.12 Depth partition value derivation and assignment process

Inputs to this process are:

* neighbouring samples p[ x ][ y ], with x, y = −1..2\*nT−1,
* an binary array partitionPattern[ x ][ y ], with x, y =0..nT−1, specifying a partitioning of the prediction block in a partition 0 and a partition 1.
* a variable nT specifying the transform size,
* a flag dcOffsetAvailFlag, specifying whether DC Offset values are available
* a flag intraChainFlag, specifying whether the current intra prediction mode is equal to Intra\_Chain(43,44)
* the variables dcOffsetP0 and dcOffsetP1, specifying the DC offsets for the block partitions

Output of this process is:

* predicted samples predSamples[ x ][ y ], with x, y =0..nT−1.

The depth partition value derivation process as specified in subclause H.8.4.4.2.12.1 is invoked with the neighbouring samples p[ x ][ y ], the binary pattern wedgePattern [ xB ][ yB ], the transform size nT, the dcOffsetAvailFlag set equal to ( intraPredMode = = Intra\_DepthPartition(36) ), intraChainFlag set equal to 0, and the DC Offsets DcOffsetP0[ xB ][ yB ], and DcOffsetP1[ xB ][ yB ] as inputs and the output is assigned to predVal[ X ] with X = 0, 1.

The predicted sample values predSamples are derived as follows for x = 0..nT−1 and for y = 0..nT−1.

* + - When partitionPattern[ x ][ y ] is equal to X, the following applies.
      * predSamples[ x ][ y ] = predVal[ X ].

~~The variable log2SubSample is set equal to ( !intraChainFlag && ( nT < 32 ) ) 0 : 1.~~

~~For X being replaced by 0 and 1, the values of the prediction samples predSamples[ x][ y ] are derived as specified by the following ordered steps:~~

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

~~sumNeigh +=~~ ~~p[ x<< log2SubSample ][ −1 ] (‑55)  
numNeigh += 1 (‑56)~~

* + - ~~For y = 0..( nT >> log2SubSample )− 1,inclusive the following applies:~~ 
      * ~~When partitionPattern[ 0 ][ y << log2SubSample ] is equal to X, the following applies:~~

~~sumNeigh += p[ −1 ][ y << log2SubSample ] (‑57)   
numNeigh += 1 (‑58)~~

* 1. ~~The variables predDcVal specifying the predicted constant partition values for partition X is derived as follows.~~
     + 1. ~~predDcVal = ( numNeigh ! = 0 ) ? ( sumNeigh / numNeigh ) : ( 1 << ( BitDepth~~~~Y~~ ~~− 1 ) ) (‑59)~~
  2. ~~The variable dcOffset specifying the DC offset is derived as follows.~~
     + ~~If dcOffsetAvailFlag is equal to 1, the following applies. :~~
       - 1. ~~dcOffsetScale = intraChainFlag ? Clip3( 1, ( 1 << BitDepth~~~~Y~~ ~~) − 1, Round( 2~~ ~~(QP’~~~~Y~~ ~~/10)- 2~~ ~~) ) : 1 (‑60)~~
         2. ~~dcOffset = dcOffset \* dcOffsetScale (‑61)~~
     + ~~Otherwise ( dcOffsetAvailFlag is equal to 0), dcOffset is set equal to 0.~~
  3. ~~The predicted sample values predSamples are derived as follows for x = 0..nT−1 and for y = 0..nT−1.~~ 
     + ~~When partitionPattern[ x ][ y ] is equal to X, the following applies.~~
       - 1. ~~predSamples[ x ][ y ] = predDcVal + dcOffset (‑62)~~

**H.8.4.4.2.12.1 Depth partition value derivation process**

Inputs to this process are:

* neighbouring samples p[ x ][ y ], with x, y = −1..2\*nT−1,
* an binary array partitionPattern[ x ][ y ], with x, y =0..nT−1, specifying a partitioning of the prediction block in a partition 0 and a partition 1.
* a variable nT specifying the transform size,
* a flag dcOffsetAvailFlag, specifying whether DC Offset values are available
* a flag intraChainFlag, specifying whether the current intra prediction mode is equal to Intra\_Chain(43,44)
* the variables dcOffsetP0 and dcOffsetP1, specifying the DC offsets for the block partitions

Output of this process is:

* prediction values predVal[X], with X = 0, 1.

The variable log2SubSample is set equal to ( !intraChainFlag && ( nT < 32 ) ) 0 : 1.

For X being replaced by 0 and 1, the values of the prediction samples predSamples[ x][ y ] are derived as specified by the following ordered steps:

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

sumNeigh += p[ x<< log2SubSample ][ −1 ] (‑55)  
numNeigh += 1 (‑56)

* + - For y = 0..( nT >> log2SubSample )− 1,inclusive the following applies:
      * When partitionPattern[ 0 ][ y << log2SubSample ] is equal to X, the following applies:

sumNeigh += p[ −1 ][ y << log2SubSample ] (‑57)   
numNeigh += 1 (‑58)

* 1. The variables predDcVal specifying the predicted constant partition values for partition X is derived as follows.
     + 1. predDcVal = ( numNeigh ! = 0 ) ? ( sumNeigh / numNeigh ) : ( 1 << ( BitDepthY − 1 ) ) (‑59)
  2. The variable dcOffset specifying the DC offset is derived as follows.
     + If dcOffsetAvailFlag is equal to 1, the following applies. :

dcOffsetScale = intraChainFlag ? Clip3( 1, ( 1 << BitDepthY ) − 1, Round( 2 (QP’Y /10)- 2 ) ) : 1 (‑60)

* + - * 1. dcOffset = dcOffset \* dcOffsetScale (‑61)
    - Otherwise ( dcOffsetAvailFlag is equal to 0), dcOffset is set equal to 0.
  1. Calculate the prediction value as
     + - 1. predVal[X ] = predDcVal + dcOffset (‑62)

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~~
* a variable intraPredMode specifying the prediction mode of the current prediction block
* the neighbouring samples Rec[ x ][ y ], with x, y = −1..2\*nT−1

…

1. ~~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~~

If intraPredMode is equal to Intra\_Planar (0) prediction mode,

dcPred[ 0 ] = dcPred[ 1 ] = ( Rec [ nT ][ −1 ] + Rec [ -1 ][ nT ] + 1 ) >> 1;

Else if intraPredMode is equal to Intra\_DC (1) prediction mode,

dcPred[ 0 ] = dcPred[ 1 ] = ( Rec [ 0 ][ −1 ] + Rec [ -1 ][ 0 ] + 1 ) >> 1;

Else if intraPredMode is equal to Intra\_DepthPartition (35) prediction mode,

* 1. The variable wedgePattern[ x ][ y ] with x, y =0..nT−1, specifying a binary partition pattern is derived as.

wedgePattern = WedgePatternTable[ Log2( nT) ][ wedge\_full\_tab\_idx[ xB ][ yB ] ].

* 1. The depth partition value derivation process as specified in subclause .1 is invoked with the neighbouring samples Rec[ x ][ y ], the binary pattern wedgePattern [ xB ][ yB ], the transform size nT, the dcOffsetAvailFlag set equal to false, intraChainFlag set equal to 0, and the DC Offsets DcOffsetP0[ xB ][ yB ], and DcOffsetP1[ xB ][ yB ] as inputs and the output is assigned to dcPred.