The proposed working draft modifications are as follows.

H. 7.3.2.1.2 Video parameter set extension 2 syntax

|  |  |
| --- | --- |
| vps\_extension( ) { |  |
| … |  |
| **~~dlt\_flag~~**~~[~~~~layerId ]~~ | ~~u(1)~~ |
| ~~if( dlt\_flag[ layerId ] ) {~~ |  |
| **~~num\_depth\_values\_in\_dlt~~**~~[ layerId ]~~ | ~~ue(v)~~ |
| ~~for ( j = 0; j < num\_depth\_values\_in\_dlt ; j++) {~~ |  |
| **~~dlt\_depth\_value~~**~~[ layerId ][ j ]~~ | ~~ue(v)~~ |
| } |  |

H. 7.3.2.2.1 Sequence parameter set extension syntax

|  |  |
| --- | --- |
| sps\_extension( ) { | **Descriptor** |
| **…** |  |
| if(DepthFlag) |  |
| **dlt\_flag** | u(1) |
| if( dlt\_flag){ |  |
| **diff\_max\_dlt\_value** | u(v) |
| **min\_dlt\_value** | u(v) |
| if(!DLTDeduced){ |  |
| **run\_length\_coding\_flag** | u(1) |
| if(run\_length\_coding\_flag){ |  |
| **min\_run\_length** | u(3) |
| **run\_length\_bits\_minus1** | u(3) |
| for(j = MinDltV+1; j< MaxDltV; j+=Run[j]+1){ |  |
| **run\_length\_diff**[j] | u(v) |
| if(run\_length\_diff[j]==RunBoundary) |  |
| **run\_length\_diff\_rem**[j] | ue(v) |
| } |  |
| } |  |
| else{ |  |
| for(j = MinDltV+1; j< MaxDltV; j++) |  |
| bit\_map\_flag[j] | u(1) |
| **}** |  |
| **}** |  |
| **}** |  |
| **}** |  |

**H.7.4.2.1.2 Video parameter set extension 2 semantics**

**~~dlt\_flag~~**~~[ layerId ] equal to 1 specifies that a depth lookup table is used and that residual DC offset values for PUs with DmmFlag equal to 1 or SdcFlag equal to 1 represent indices of the depth lookup table for layers with nuh\_layer\_id equal to layerId. dlt\_flag[ layerId ] equal to 0 specifies that a depth lookup table is not used and that residual DC offset values for PUs with DmmFlag equal to 1 or SdcFlag equal to 1 represent indices of the depth lookup table for layers with nuh\_layer\_id equal to layerId. When not present, the value of dlt\_flat[ layerId ] shall be inferred to be equal to 0.~~

**~~num\_depth\_values\_in\_dlt~~**~~[ layerId ] specifies the number of different depth values and the number of elements in the depth lookup table for depth view components of the current layer with layer\_id equal to layerId.~~

**~~dlt\_depth\_value[~~**~~layerId~~**~~]~~**~~[ j ] specifies the j-th entry in the depth lookup table for depth view components with layer\_id equal to layerId.~~

H.7.4.2.2 Sequence parameter set RBSP semantics

**dlt\_flag** specifies whether a depth lookup table is used or not. When dlt\_flag is equal to 1, a depth lookup table is used for PUs with DmmFlag equal to 1 or SdcFlag equal to 1. Otherwise, it is not used. When not present, the value of dlt\_flat shall be inferred to be equal to 0.

**diff\_max\_dlt\_value** specifies the difference between the largest and smallest value in the depth lookup table. The number of bits used to represent it is log2(MAX\_DEPTH\_VALUE + 1).

**min\_dlt\_value** specifies the smallest value in the depth lookup table. The number of bits used to represent it is log2(MAX\_DEPTH\_VALUE+1-diff\_max\_dlt\_value). When min\_dlt\_value is not present, it should be inferred to be 0. MinDltV is set equal to min\_dlt\_value. MaxDltV is set equal to min\_dlt\_value+diff\_max\_dlt\_value. Initially, BitMapFlag[i] with i from 0 to MAX\_DEPTH\_VALUE is set equal to 0 and DLTDeduced is set equal to 0. If MaxDltV is greater than MAX\_DEPTH\_VALUE, DLTDeduced is set equal to 1. Otherwise, BitMapFlag [MinDltV] and BitMapFlag[MaxDltV] are set equal to 1. If diff\_max\_dlt\_value is equal to 0 or 1, DLTDeduced is set equal to 1.

**run\_length\_coding\_flag** specifies whether run-length coding is used or not. When run\_length\_coding\_flag is equal to 1, the bit-map is coded by run-length coding. Otherwise, it is coded directly.

**min\_run\_length** specifies the minimum 0-run-length in the DLT bit-map. MinRunLength is set equal to min\_run\_length.

**run\_length\_bits\_minus1** specifies the number of bits used to code run\_length\_diff. RunLengthBits is set equal to run\_length\_bits\_minus1+1.

**run\_length\_diff**[j] specifies the difference between one 0-run-length and MinRunLength. The number of bits used to represent it is RunLengthBits.

**run\_length\_diff\_rem**[j] specifies the remainder of difference between one 0-run-length and MinRunLength. run\_length\_diff\_rem[j] is only present if run\_length\_diff[j] is equal to RunBoundary, where RunBoundary is equal to 2RunLengthBits-1. When run\_length\_diff\_rem[j] is not present, it should be inferred to be 0. Run[j] is set equal to run\_length\_diff[j] +run\_length\_diff\_rem[j]+ MinRunLength. BitMapFlag[j+Run[j]] is set equal to 1. BitMapFlag [i] with i from j to j+Run[j]-1 is set equal to 0, if Run[j] is not equal to 0.

**bit\_map\_flag**[j]specifies the j-th entry in the bit map. BitMapFlag[j] is set equal to bit\_map\_flag[j].

**H.8.3.6 Decoding process for a depth lookup table**

* ~~For i = 0..num\_depth\_values\_in\_dlt –1 the elements in Idx2DepthValue are derived as follows.~~
  + ~~Idx2DepthValue[ i ] is set equal to dlt\_depth\_value[ i ]~~
* The elements in Idx2DepthValue are derived as follows.
* If nuh\_layer\_id >> 1 is not equal to 0
  + for i =0… MAX\_DEPTH\_VALUE-1
    - BitMapFlag[i] = BitMapFlag[i] ^ BitMapFlag’[i], where BitMapFlag’ represents the BitMapFlag in the sequence with nuh\_layer\_id equal to 1.
* Set Idx= 0;
* for i =0… MAX\_DEPTH\_VALUE-1
  + If BitMapFlag[i]==1, then Idx2DepthValue[Idx] = i and Idx++;
* num\_depth\_vaules\_in\_dlt=Idx;