|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  9th Meeting: Geneva, CH, 27 April – 7 May 2012 | Document: JCTVC-I0115\_r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | Improvement of Implicit Weighted Prediction | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Philippe Bordes, Pierre Andrivon 1, av. De Belle-Fontaine – CS17616 35576 Cesson-Sévigné Cedex - France | Tel: Email: | +33-2-99-27-32-42 [philippe.bordes@technicolor.com](mailto:philippe.bordes@technicolor.com)  [pierre.andrivon@technicolor.com](mailto:pierre.andrivon@technicolor.com) |
| *Source:* | Technicolor | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This document presents an improved implicit weighted prediction (IWP). Weighted prediction (WP) scheme consists in two modes, an explicit WP and an implicit WP. In Explicit WP, the weighting parameters are transmitted explicitly, while in Implicit WP they are derived from the temporal distances of the two reference pictures used for bi-prediction in B-slices.

In the current IWP, the offsets are unforced to zero and it is applicable to bi-prediction and B-slices only. In RA (hierarchical GOP) the weights of closest references are equal to defaults.

The proposed IWP allows to cope with these limitations by interpolating or extrapolating the Explicit WP parameters, in the case Explicit and Implicit WP are combined.

The experimental results in HM software version 6.0 are reported. The BD-rate gain depends on the proportion of frames using Explicit WP and Implicit WP (EP=Explicit Period). For EP=4, the BD-rate gain of the proposed method for black/white-fade sequences is 17% to 34%.

# Introduction: Weighted Prediction recap

The Weighted Prediction (WP) in HEVC allows to compensate for Illumination Change (IC) in Video Sequences between the current slice and the reference frames, in forward and bi-prediction. The principle consists in building the inter prediction signal as a linear combination of the motion compensated signal (predPartLi) as depicted in equations (1) and (2) [1]:

*predPartC[ x, y ] = ( ( predPartL0C[ x, y ] × w0 +1<<(shift-1) ) >> shift ) + o0* (1)

*predPartC[ x, y ] = ( predPartL0C[ x, y ] × w0 + predPartL1C[ x, y ] × w1 + (1<<shift) + (( o0 + o1)<<shift) )*

*>> ( shift + 1 )* (2)

Depending on the type of slice (P\_SLICE or B\_SLICE), one can choose the weighting algorithm:

* Default: the default HEVC uni-prediction or bi-prediction.
* Explicit: the weighting factors are transmitted explicitly in the slice header.
* Implicit: the weighting factors for bi-prediction are derived from the distance of the current POC with the POC of the reference pictures, relatively to the distance between references POC (B\_SLICE only).

The presence of weighting factors and the weighting method to use are indicated in the PPS with **weighted\_pred\_flag** (1 bit)and **weighted\_bipred\_idc** (2 bits) (Table 1). If present, the adaptive weighting factor parameters are transmitted in the slice header for each allowable reference picture. Syntax for coding of the weighting factors is similar to that done in AVC.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_SLICE** | |  | **B\_SLICE** | |
| **weighted\_pred\_flag** | **weighting method** |  | **weighted\_bipred\_idc** | **weighting method** |
| 0 | Default |  | 0 | Default |
| 1 | Explicit |  | 1 | Explicit |
|  |  |  | 2 | Implicit |

Table 1: Interpretation of weighted\_pred\_flag and weighted\_bipred\_idc values.

Explicit: If present, the weighting factor parameters [2] allow deriving the corresponding weights (wi) and offset (oi) values for explicit weighting prediction (equations 1 and 2), both in uni-prediction and bi-prediction. If not present, default weighting is used.

Implicit: The weights (wi) and offset (oi) are determined with the following equation (3):

w0 = 2 – α w1=α where: , oi=0 (3)

In equation (3), *wi* Є [-1.;2.]. Practically, to avoid floating point multiplication, the equation (3) is implemented using only integer multiplications and shift. Implicit weighting method is used in case of linear fade or cross-fade typically.

The gain of using Explicit WP in HM6.0 compared to the anchors is between 16.9% to 32.3% for Fade-to-black sequences, and between 17.8% to 34.2% with Fade-to-white sequences, depending on the configurations (Table 1and Table 2). The Fade sequences are built from the regular HEVC sequences as proposed in [3].

# Implicit Weighted Prediction

## Limitations of existing Implicit WP

The Implicit WP allows to reduce the slice header size and to lightweight the encoder WP weights estimation process in case of linear Fades. The Implicit WP (IWP) has been specially designed in AVC and it provided significant gains with two references for IbbP GOPs (weights=(1/3, 2/3)) and for low delay case where one reference picture is twice as far away in time and in the same direction from the current picture as other reference pictures (weights=(2, -1)), as explained in [4], and as depicted in Figure 1.

(a) (b)

Figure 1: IbbP (a) and low-delay BBB (b) GOPs with two references as used in AVC originally.

However, in the case of HEVC, the test conditions have changed compared to the ones used in JVT. The Random Access (RA) configuration uses Hierarchical GOP. Then, in case of bi-prediction the Implicit WP weights associated with the references frames with the closest POC to current POC, are identical to the default bi-prediction weights (1/2,1/2) as depicted in Figure 2. In case of Low-Delay (LD) configuration, the IWP weights may be negative as depicted in Figure 2. In all the cases, the IWP offsets are zero, favoring Fade-to-black.



Figure 2: Implicit WP weights for RA and LD HEVC configurations.

## Proposal of new Implicit WP

The proposed new Implicit WP (IWP) algorithm (**weighted\_bipred\_idc**=3**)** keeps the same principle as previous IWP: the weights vary linearly with time (POC).

The derivation of the weights and offsets is made by interpolating linearly with time (POC) the explicit or implicit weights and offsets of previously decoded reference pictures. It does not need any additional decoded frame analysis process.

This Implicit WP method allows to make lighter the WP weight signaling and to reduce complexity at encoder side by-passing the WP estimation stage for some frames. It is also compatible with temporal scalability: only high level syntax (WP weights and offsets) of the base layer is used to derive the WP parameters of the temporal enhancement layer.

Algorithm description

A hash table containing the explicit or implicit weights and offsets (*wcj, ocj*) that were used to decode the reference slices is up-dated after decoding every reference slice: a new line *pocCur* containing the set of weights *wcj* and offsets *ocj* (where *j=0,..N*, and *N* is the number of references used by frame *pocCur*) is added, and also a new column with empty cells except the last one with (1;0), as depicted in Figure 3. The information whether the weights are explicit or implicit is also stored. Note some cells may be empty.

The references pictures removed from the DPB are removed from the table by removing the corresponding lines and columns.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **PocS** | **…** | **PocK** | **pocCur** |
| **PocS** | (1,0) |  |  |  |
| **…** |  |  |  |  |
| **PocK** | (wks,oks) |  | (1,0) |  |
| **pocCur** | (wcs,ocs) |  | (wck,ock) | (1,0) |

Figure : hash table containig the explicit or implicit weights and offsets.

To derive *wck* and *ock* corresponding to the reference *k* (kth column), we re-order the n(k) WP parameters (non empty cells) stored in the kth column with the values marked as implicit first, and next those marked as explicit. Then we select the two last elements (*poc0* and *poc1*) of this reordered list, favoring the use of last decoded WP explicit values.

If n(k) is >= 2, then we interpolate the weights for *pocCur* as:

*wck = w0k + (w1k – w0k)\*(poccur - poc0)/(poc1 - poc0)*  (5)

*ock = o0k + (o1k – o0k)\*(poccur - poc0)/(poc1 - poc0)*  (6)

If n(k) =1, and it exists s<k with n(s) >= 2, then the weights are extrapolated as :

*wck = wcs / wks* (7)

*ock = ocs – (wcs / wks )🞨 oks* (8)

The division is implemented with integer values and shift as depicted in Appendix 6.

We also propose to add more flexibility by offering the possibility to choose between using explicit WP or implicit WP for each frame or slice, without duplicating PPS. Then, if **weighted\_bipred\_idc**=3, a flag **wp\_implicit\_flag** is added in the slice header to indicate whether to use Explicit or Implicit algorithm.

# Experimental Results

HM software version 6.0 (HM-6.0) with the common configuration [5] is used in these experiments. Test sequences with black/white-fade are provided by WP AHG [3].

1 and Table 2 show BD-Rate and the relative average encoding/decoding time for explicit WP, implicit WP and the proposed implicit WP for different values of Explicit WP period (EP), compared to HM6.0 anchor, for black-fade sequences and white-fade sequences respectively. The negative value of BD-Rate indicates gain.

It observed that proposed Implicit WP with EP=2 or EP=4 can achieve the gain of 17% to 34% on average for RA, LDB and LDP cases and that gain, decoding and encoding times are almost similar to the explicit WP. Larger EP values bring still significant gains.

**Table 1. BD-Rate[%] and relative encoding/decoding time[%] of explicit WP, implicit WP and proposed Implicit WP compared to HM6.0 anchor for black-fade sequences.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Black-fade case | **Random Access Main** | | | | | **Random Access HE10** | | | | |
| Y | U | V | Enc[%] | Dec[%] | Y | U | V | Enc[%] | Dec[%] |
| **ExplicitWP** | -16.9% | -19.6% | -19.2% | 142% | 98% | -17.2% | -19.6% | -19.4% | 100% | 95% |
| **ImplicitWP** | -0.2% | -0.3% | -0.6% | 175% | 98% | -0.2% | -0.1% | -0.1% | 102% | 97% |
| **Proposed (EP=2)** | -17.5% | -19.4% | -18.8% | 143% | 86% | -17.4% | -19.2% | -18.9% | 102% | 88% |
| **Proposed (EP=4)** | -17.9% | -19.4% | -18.6% | 142% | 105% | -17.7% | -18.4% | -18.2% | 102% | 96% |
| **Proposed (EP=8)** | -17.4% | -19.1% | -17.9% | 143% | 100% | -17.0% | -16.5% | -16.3% | 102% | 95% |
| **Proposed (EP=16)** | -16.7% | -17.2% | -15.6% | 144% | 103% | -16.4% | -13.9% | -14.6% | 102% | 96% |
|  |  |  |  |  |  |  |  |  |  |  |
| Black-fade case | **Low delay B Main** | | | | | **Low delay B HE10** | | | | |
| Y | U | V | Enc[%] | Dec[%] | Y | U | V | Enc[%] | Dec[%] |
| **ExplicitWP** | -32.3% | -40.5% | -39.3% | 141% | 88% | -32.0% | -43.2% | -41.2% | 102% | 82% |
| **ImplicitWP** | 1.4% | 3.1% | 2.2% | 183% | 105% | 1.9% | 2.6% | 2.4% | 102% | 99% |
| **Proposed (EP=2)** | -32.4% | -38.0% | -35.9% | 142% | 75% | -32.1% | -37.8% | -34.9% | 102% | 74% |
| **Proposed (EP=4)** | -31.5% | -37.3% | -32.9% | 143% | 90% | -30.9% | -35.8% | -32.6% | 102% | 81% |
| **Proposed (EP=8)** | -27.6% | -31.7% | -27.2% | 146% | 94% | -27.2% | -29.3% | -24.8% | 102% | 85% |
| **Proposed (EP=16)** | -15.7% | -19.8% | -9.6% | 154% | 101% | -15.8% | -12.8% | -1.7% | 102% | 88% |
|  |  |  |  |  |  |  |  |  |  |  |
| Black-fade case | **Low delay P Main** | | | | | **Low delay P HE10** | | | | |
| Y | U | V | Enc[%] | Dec[%] | Y | U | V | Enc[%] | Dec[%] |
| **ExplicitWP** | -32.0% | -39.5% | -38.3% | 122% | 87% | -31.3% | -42.7% | -41.3% | 103% | 79% |
| **ImplicitWP** | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| **Proposed (EP=2)** | -32.0% | -36.1% | -33.6% | 123% | 73% | -31.4% | -37.1% | -34.2% | 103% | 70% |
| **Proposed (EP=4)** | -30.7% | -35.3% | -31.3% | 127% | 90% | -30.0% | -34.7% | -31.2% | 103% | 84% |
| **Proposed (EP=8)** | -26.6% | -31.7% | -29.0% | 135% | 93% | -25.9% | -30.3% | -28.6% | 103% | 89% |
| **Proposed (EP=16)** | -17.3% | -25.6% | -22.6% | 156% | 105% | -16.6% | -21.5% | -20.8% | 103% | 96% |

**Table 2. BD-Rate[%] and relative encoding/decoding time[%] of explicit WP, implicit WP and proposed Implicit WP compared to HM6.0 anchor for white-fade sequences.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| White-fade case | **Random Access Main** | | | | | **Random Access HE10** | | | | |
| Y | U | V | Enc[%] | Dec[%] | Y | U | V | Enc[%] | Dec[%] |
| **ExplicitWP** | -20.3% | -23.4% | -23.0% | 137% | 98% | -20.5% | -22.9% | -23.1% | 102% | 97% |
| **ImplicitWP** | -0.3% | -0.5% | -0.6% | 182% | 99% | -0.2% | -0.1% | -0.2% | 102% | 105% |
| **Proposed (EP=2)** | -20.5% | -23.0% | -22.5% | 138% | 72% | -20.7% | -22.7% | -23.0% | 102% | 95% |
| **Proposed (EP=4)** | -20.6% | -22.8% | -22.5% | 137% | 81% | -20.8% | -21.7% | -22.0% | 102% | 101% |
| **Proposed (EP=8)** | -19.7% | -22.3% | -21.3% | 140% | 81% | -19.6% | -19.6% | -20.0% | 102% | 96% |
| **Proposed (EP=16)** | -18.4% | -19.9% | -19.0% | 142% | 74% | -18.4% | -16.9% | -17.7% | 102% | 101% |
|  |  |  |  |  |  |  |  |  |  |  |
| White-fade case | **Low delay B Main** | | | | | **Low delay B HE10** | | | | |
| Y | U | V | Enc[%] | Dec[%] | Y | U | V | Enc[%] | Dec[%] |
| **ExplicitWP** | -34.2% | -42.3% | -41.4% | 136% | 88% | -33.6% | -44.3% | -42.8% | 102% | 89% |
| **ImplicitWP** | 1.5% | 2.5% | 2.8% | 187% | 112% | 2.0% | 3.2% | 2.7% | 102% | 105% |
| **Proposed (EP=2)** | -34.2% | -40.3% | -37.0% | 137% | 77% | -33.9% | -39.6% | -37.4% | 102% | 77% |
| **Proposed (EP=4)** | -33.3% | -39.1% | -35.0% | 138% | 90% | -32.7% | -37.7% | -35.4% | 102% | 86% |
| **Proposed (EP=8)** | -29.2% | -33.8% | -29.0% | 142% | 95% | -28.7% | -30.9% | -27.5% | 102% | 88% |
| **Proposed (EP=16)** | -17.3% | -22.1% | -12.3% | 150% | 91% | -17.0% | -13.9% | -4.9% | 102% | 95% |
|  |  |  |  |  |  |  |  |  |  |  |
| White-fade case | **Low delay P Main** | | | | | **Low delay P HE10** | | | | |
| Y | U | V | Enc[%] | Dec[%] | Y | U | V | Enc[%] | Dec[%] |
| **ExplicitWP** | -33.8% | -40.7% | -39.7% | 114% | 83% | -33.1% | -44.2% | -42.4% | 103% | 83% |
| **ImplicitWP** | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| **Proposed (EP=2)** | -33.7% | -37.8% | -34.5% | 116% | 74% | -33.2% | -39.1% | -35.8% | 103% | 73% |
| **Proposed (EP=4)** | -32.0% | -36.2% | -31.8% | 121% | 89% | -31.3% | -35.9% | -32.0% | 103% | 79% |
| **Proposed (EP=8)** | -26.9% | -32.0% | -29.4% | 132% | 96% | -26.3% | -30.8% | -28.4% | 103% | 90% |
| **Proposed (EP=16)** | -17.2% | -25.9% | -22.4% | 158% | 99% | -16.7% | -22.1% | -20.5% | 103% | 98% |

# Conclusion

In this contribution, an improved Implicit WP scheme based on interpolation of Explicit WP parameters is proposed. This method can be applicable to not only bi-prediction in B-slices but also uni-prediction in P/B-slices. It does not need any additional decoded frame analysis process.

This Implicit WP scheme allows to make lighter the WP weight signaling as well as to by-pass the WP estimation stage at the encoder side for some frames. Moreover, it is compatible with temporal scalability: the derivation of WP weights for the enhancement layer depends on the high level syntax (Explicit WP) of the base layer only. This new Implicit WP can be seen as an enhancement of the current Explicit WP mode.

Experimental results show the proposed method can achieve average BD-rate gain of 17% to 34% for RA, LDB and LDP cases, depending on the Explicit WP period.

It’s suggested that this proposal is introduced in the text and the software of HEVC test model (HM).

# References

1. Yan Ye, Eun-Seok Ryu, “Improved Weighted Prediction,” JVT Document, JCTVC-G065, Geneva, November 2011.
2. Akiyuki Tanizawa, Takeshi Chujoh, Tomoo Yamakage, “Redundancy removal of explicit weighted prediction syntax,” JVT Document, JCTVC-G441, Geneva, November 2011.
3. Philippe Bordes, TK. Tan, “JCT-VC AHG report: Weighted Prediction (AHG18),” JVT Document, JCTVC-F018, Torino, July 2011.
4. Jill Boyce, “Changes to Adaptive Reference Picture Weighting,” JVT Document, JVT-E060, Geneva, October 2002.
5. F.Bossen ," Common test conditions and software reference configurations ", Joint Collaborative Team on Video Coding, JCTVC-H1100, San Jose, Feb. 2012.

# Patent rights declaration(s)

**Technicolor may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

# Appendix

## WD text for implicit weighted prediction

#### 7.3.3 Slice header syntax

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| … |  |
| if( slice\_type != I && weighted\_bipred\_idc = = 3) |  |
| **wp\_implicit\_flag** | u(1) |
| if( ( weighted\_pred\_flag && slice\_type = = P) | |  ( weighted\_bipred\_idc = = 1 && slice\_type = = B ) | |  (slice\_type != I && weighted\_bipred\_idc == 3 && !wp\_implicit\_flag) ) |  |
| pred\_weight\_table( ) |  |
| … |  |
| } |  |

#### 7.4.2.2 Picture parameter set RBSP semantics

**weighted\_bipred\_idc** equal to 0 specifies that the default weighted prediction is applied to B slices. weighted\_bipred\_idc equal to 1 specifies that explicit weighted prediction is applied to B slices. weighted\_bipred\_idc equal to 2 specifies that implicit weighted prediction shall be applied to B slices. weighted\_bipred\_idc equal to 3 specifies that implicit weighted prediction mode 2 shall be applied to P and B slices. The value of weighted\_bipred\_idc shall be in the range of 0 to 3, inclusive.

###### 8.5.2.2.3.2 Weighted sample prediction process

Inputs to this process are:

– the same as specified in subclause 8.5.2.2.3.

– variables for weighted prediction logWDC, logWDC, w0C, w1C, o0C, o1C with C being replaced by L for luma samples and, when chroma\_format\_idc is not equal to 0, Cb and Cr for chroma samples.

Outputs of this process are:

– the same as specified in subclause 8.5.2.2.3.

The prediction sample predSamples[ x, y ] with x = 0..(nPSW)−1 and y = 0..(nPSH)−1, with H being replaced by Y for luma samples and by C for chroma samples are derived as follows:

– If the predFlagL0 is equal to 1 and predFlagL1 is equal to 0, the prediction samples are derived as follows:

if( logWDC >= 1 )   
 predSamples[ x, y ] = Clip1H( ( ( predSamplesL0 [ x, y ] \* w0C + 2logWDC − 1 ) >> logWDC ) + o0C ) (8‑215)  
else  
 predSamples[ x, y ] = Clip1H( predSamplesL0 [ x, y ] \* w0C + o0C ) (8‑216)

– Otherwise, if the predFlagL0 is equal to 0 and predFlagL1 is equal to 1, the final predicted sample values predSamples [ x, y ] are derived by

if( logWDC >= 1 )  
 predSamples[ x, y ] = Clip1H( ( ( predSamplesL1 [ x, y ] \* w1C + 2logWDC − 1 ) >> logWDC ) + o1C ) (8‑217)  
else  
 predSamples[ x, y ] = Clip1H( predSamplesL1 [ x, y ] \* w1C + o1C ) (8‑218)

– Otherwise (both predFlagL0 and predFlagL1 are equal to 1), if RefPicOrderCnt( currPic, refIdxL0, L0) is equal to RefPicOrderCnt( currPic, refIdxL1, L1) and mvL0 is equal to mvL1, the final predicted sample values predSamples [ x, y ] are derived by

predSamples[ x, y ] = Clip1H( ( predSamplesL0 [ x, y ] \* ( w0C + w1C ) +   
 ( ( o0C + o1C + 1 ) << logWDC ) ) >> ( logWDC + 1 ) ) (8‑219)

– Otherwise, the final predicted sample values predSamples[ x, y ] are derived by

predSamples[ x, y ] = Clip1H( ( predSamplesL0 [ x, y ] \* w0C + predSamplesL1 [ x, y ] \* w1C +   
 ( ( o0C + o1C + 1 ) << logWDC ) ) >> ( logWDC + 1 ) ) (8‑220)

Where the variables logWDC, o0C, o1C, and w0C, w1C are derived as follows.

– If weighted\_bipred\_idc is equal to 2 in B-slices, implicit mode weighted prediction is used as follows:

logWDc = 5+shift1 (8‑221)

o0C = 0 (8‑222)

o1C = 0 (8‑223)

The variable WeightScaleFactor is derived from the values currPoc, refIdxL0 and refIdxL1 as follows:

tb = Clip3( −128, 127, PicOrderCntVal – PicOrderCnt( RefPicListL0( refIdxL0 ) ) ) (8‑224)

td = Clip3( −128, 127, PicOrderCnt( RefPicListL1( refIdxL1 ) )   
  – PicOrderCnt( RefPicListL0( refIdxL0 ) ) ) (8‑225)

tx = ( 16384 + ( Abs( td ) >> 1 ) ) / td (8‑226)

WeightScaleFactor = Clip3( −1024, 1023, ( tb \* tx + 32 ) >> 6 ) (8‑227)

The variables w0C and w1C are derived as follows.

– If PicOrderCnt( RefPicListL0( refIdxL0 ) ) is equal to PicOrderCnt( RefPicListL1( refIdxL1 ) ) or ( WeightScaleFactor >> 2 ) < −64 or ( WeightScaleFactor >> 2 ) > 128, the following applies.

w0C=32 (8‑228)

w1C=32 (8‑229)

– Otherwise;

w0C = 64 – (WeightScaleFactor >> 2) (8‑230)

w1C = WeightScaleFactor >> 2 (8‑231)

– Otherwise, if weighted\_bipred\_idc is equal to 3 in P-slices or in B-slices, implicit mode 2 weighted prediction is used with refIdxLx to compute wxC , oxC and logWDC as follows:

* If it exists at least 2 elements in hash tables wTab and oTab at column pocRef=PicOrderCnt( RefPicListLx( refIdxLx ) ) then select the 2 elements (poc0 and poc1) corresponding to the more recently decoded pictures (Explicit WP re-ordered first), then:

( w0C ; d0C ) = wTab[ C ][ poc0 ][ pocRef ]

( w1C ; d1C ) = wTab[ C ][ poc1 ][ pocRef ]

( o0C ; d0C ) = oTab[ C ][ poc0 ][ pocRef ]

( w1C ; d1C ) = oTab[ C ][ poc1 ][ pocRef ]

With C=0,1 or 2 for Luma, Cr or Cb respectively.

( wxC ; dxC ) = ( w0C ; d0C ) – (( w1C ; d1C )- ( w0C ; d0C ))\*( (PicOrderCntVal;0) - (poc0;0) ) / ( (poc1;0) - (poc0;0) )

( oxC ; dxC ) = ( o0C ; d0C ) – (( o1C ; d1C )- ( o0C ; d0C ))\*( (PicOrderCntVal;0) - (poc0;0) ) / ( (poc1;0) - (poc0;0) )

* where: (a0;d0) – (a1;d1) = (d0>d1) ? N(a0-a1<<(d0-d1);d0) : N((a0<<(d1-d0))-a1;d1)

(a0;d0) / (a1;d1) = N( (a1==0) ? (a0<<defaultDenom) : ((a0<< defaultDenom)/a1) ;defaultDenom+d0-d1)

(a0;d0) \* (a1;d1) = N( a0\*a1 ;d0+d1)

N(a0;d0) = (a1;d1)

with: shift = d0 – defaultDenom

a1 = (shift>0) ? ( (a0 + (1<<(shift-1))) >> shift ) : ( a0 << (-shift) )

d1 = defaultDenom

* Else if it exists at least one column pocS in hash tables wTab and oTab with at least 2 elements (poc0 and pocRef), then

( w0C ; d0C ) = wTab[ C ][ PicOrderCntVal][ pocS ]

( w1C ; d1C ) = wTab[ C ][ pocRef ][ pocS ]

( o0C ; d0C ) = oTab[ C ][ PicOrderCntVal][ pocS ]

( o1C ; d1C ) = oTab[ C ][ pocRef ][ pocS ]

With C=0,1 or 2 for Luma, Cr or Cb respectively.

( wxC ; dxC ) = ( w0C ; d0C ) / ( w1C ; d1C )

( oxC ; dxC ) = ( o0C ; d0C ) – ( wxC ; dxC )\* ( o1C ; d1C )

defaultDenom = 8

logWDC = defaultDenom +shift1

The process for up-dating the hash tables wTab, oTab of size nbTab x nbTab is called after the marking of reference pictures before decoding stage described in 8.3.5 . All the weighted parameters values in wTab and oTab corresponding to pictures that are not present in the DPB are removed from the wTab and oTab hash tables. Then:

for( j = 0; j <= 3; j++ ) {

wTab[ j ][ PicOrderCntVal ][ PicOrderCntVal ] = (1;defaultDenom)

oTab[ j ][ PicOrderCntVal ][ PicOrderCntVal ] = (0;defaultDenom)

}

If the current slice header has explicit weighted parameters in B-slices and ref\_pic\_list\_combination\_flag is equal to 1 then:

for( i = 0; i <= num\_ref\_idx\_lc\_active\_minus1; i++ ) {

RefPoc = PicOrderCnt( RefPicListLC( i ) )

wTab[0][ PicOrderCntVal ][ RefPoc ] = ( LumaWeightLC[ i ] ; luma\_log2\_weight\_denom+ shift1 )

oTab[0][ PicOrderCntVal ][ RefPoc ] = ( luma\_offset\_lc[ i ] \* ( 1 << ( BitDepthY − 8 ) ) ; 0 )

for( j = 0; j <= 2; j++ ) {

wTab[ j+1][ PicOrderCntVal ][ RefPoc ] = (ChromaWeightLC[ i ][ j] ; ChromaLog2WeightDenom + shift1 )

oTab[ j+1][ PicOrderCntVal ][ RefPoc ] = (ChromaOffsetLC [ i ][ j ] \* ( 1 << ( BitDepthC − 8 ) ) ; 0 )

}

}

If the current slice header has explicit weighted parameters in P-slices or in B-slices and ref\_pic\_list\_combination\_flag is equal to 0 then:

for( i = 0; i <= num\_ref\_idx\_l0\_active\_minus1; i++ ) {

RefPoc = PicOrderCnt( RefPicListL0( i ) )

wTab[0][ PicOrderCntVal ][ RefPoc ] = ( LumaWeightL0[ i ] ; luma\_log2\_weight\_denom+ shift1 )

oTab[0][ PicOrderCntVal ][ RefPoc ] = ( luma\_offset\_l0[ i ] \* ( 1 << ( BitDepthY − 8 ) ) ; 0 )

for( j = 0; j <= 2; j++ ) {

wTab[ j+1][ PicOrderCntVal ][ RefPoc ] = (ChromaWeightL0[ i ][ j] ; ChromaLog2WeightDenom + shift1 )

oTab[ j+1][ PicOrderCntVal ][ RefPoc ] = (ChromaOffsetL0[ i ][ j ] \* ( 1 << ( BitDepthC − 8 ) ) ; 0 )

}

}

If ( current slice is B-slices and ref\_pic\_list\_combination\_flag == 0)

{

for( i = 0; i <= num\_ref\_idx\_l1\_active\_minus1; i++ ) {

RefPoc = PicOrderCnt( RefPicListL1( i ) )

wTab[0][ PicOrderCntVal ][ RefPoc ] = ( LumaWeightL1[ i ] ; luma\_log2\_weight\_denom+ shift1 )

oTab[0][ PicOrderCntVal ][ RefPoc ] = ( luma\_offset\_l1[ i ] \* ( 1 << ( BitDepthY − 8 ) ) ; 0 )

for( j = 0; j <= 2; j++ ) {

wTab[j+1][ PicOrderCntVal ][ RefPoc ] = (ChromaWeightL1[ i ][ j] ; ChromaLog2WeightDenom+shift1 )

oTab[ j+1][ PicOrderCntVal ][ RefPoc ] = (ChromaOffsetL1[ i ][ j ] \* ( 1 << ( BitDepthC − 8 ) ) ; 0 )

}

}

}

– Otherwise, if both weighted\_bipred\_idc and ref\_pic\_list\_combination\_flag are equal to 1 in B slice, combined explicit mode weighted prediction is used as follows:

– If C is equal to L for luma samples,

logWDc = luma\_log2\_weight\_denom+ shift1 (8‑232)

w0C = LumaWeightLC[ RefIdxLCToRefIdxLx[ refIdxL0 ] ] (8‑233)

w1C = LumaWeightLC[ RefIdxLCToRefIdxLx[ refIdxL1 ] ] (8‑234)

o0C = luma\_offset\_lc[ RefIdxLCToRefIdxLx[ refIdxL0 ] ] \* ( 1 << ( BitDepthY − 8 ) ) (8‑235)

o1C = luma\_offset\_lc[ RefIdxLCToRefIdxLx[ refIdxL1 ] ] \* ( 1 << ( BitDepthY − 8 ) ) (8‑236)

– Otherwise (C is equal to Cb or Cr for chroma samples, with iCbCr = 0 for Cb, iCbCr = 1 for Cr),

logWDc = ChromaLog2WeightDenom + shift1 (8‑237)

w0C = ChromaWeightLC[ RefIdxLCToRefIdxLx[ refIdxL0 ] ][ iCbCr ] (8‑238)

w1C = ChromaWeightLC[ RefIdxLCToRefIdxLx[ refIdxL1 ] ][ iCbCr ] (8‑239)

o0C = ChromaOffsetLC[ RefIdxLCToRefIdxLx[ refIdxL0 ] ][ iCbCr ] \* ( 1 << ( BitDepthC − 8 ) ) (8‑240)

o1C = ChromaOffsetLC[ RefIdxLCToRefIdxLx[ refIdxL1 ] ][ iCbCr ] \* ( 1 << ( BitDepthC − 8 ) ) (8‑241)

– Otherwise (weighted\_pred\_flag is equal to 1 in P slice or ref\_pic\_list\_combination\_flag is equal to 0 in B-slice) explicit mode weighted prediction is used as follows:

– If C is equal to L for luma samples,

logWDc = luma\_log2\_weight\_denom+ shift1 (8‑242)

w0C = LumaWeightL0[refIdxL0] (8‑243)

w1C = LumaWeightL1[refIdxL1] (8‑244)

o0C = luma\_offset\_l0[refIdxL0] \* ( 1 << ( BitDepthY − 8 ) ) (8‑245)

o1C = luma\_offset\_l1[refIdxL1] \* ( 1 << ( BitDepthY − 8 ) ) (8‑246)

– Otherwise (C is equal to Cb or Cr for chroma samples, with iCbCr = 0 for Cb, iCbCr = 1 for Cr),

logWDc = ChromaLog2WeightDenom + shift1 (8‑247)

w0C = ChromaWeightL0[refIdxL0][ iCbCr ] (8‑248)

w1C = ChromaWeightL1[refIdxL1][ iCbCr ] (8‑249)

o0C = ChromaOffsetL0[refIdxL0][ iCbCr ] \* ( 1 << ( BitDepthC − 8 ) ) (8‑250)

o1C = ChromaOffsetL1[refIdxL1][ iCbCr ] \* ( 1 << ( BitDepthC − 8 ) ) (8‑251)