|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  4th Meeting: Incheon, KR, 20–26 Apr. 2013 | Document: JCT3V-D0182 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE6.h related: On signaling of DLT for depth coding** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Xin Zhao Li Zhang Ying Chen Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121 USA | Tel: Email: | +86-10-5776-0696 [xinzhao@qti.qualcomm.com](mailto:xinzhao@qti.qualcomm.com)  +1-858-651-6660 [lizhang@qti.qualcomm.com](mailto:lizhang@qti.qualcomm.com)  +1-858-845-6589 [cheny@qti.qualcomm.com](mailto:cheny@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This contribution proposes a more efficient method to signal depth look-up tables (DLTs) used for depth map intra coding in 3D-HEVC. In the proposed method, both prediction of the values of one DLT table of one depth view and prediction of DLT tables of multiple depth views are enabled to avoid largly increased number of bits requried in Video Parameter Set (VPS). With proposed methods, the total number of DLT bits is reduced to 12% of the original size without any impact on the coded PSNR values.

# Introduction

In 3D-HEVC, DLT is optionally used in simplified depth coding (SDC) method for depth intra coding. When DLT is enabled (a typical scenario of enabling DLT is the number of valid depth values being less than half of the maximum depth value) for SDC, each depth value is mapped to its corresponding index in the list of valid depth maps, and intra prediction is performed using the mapped index values. The residual of index values are transmitted, and the reconstructed index values are then mapped back to the corresponding valid depth values. The advantage of using DLT as described in [1] is the reduced bit depth of the residual index for sequences with reduced depth range.

To construct the look-up tables for mappings between depth values and indexes, all valid depth values are explicitly signaled in VPS as specified in H.7.3.2.1.2 shown below in the current WD [2].

H.7.3.2.1.2 Video parameter set extension 2 syntax

|  |  |
| --- | --- |
| vps\_extension2( ) { | Descriptor |
| … |  |
| **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[ layerId ] ; j++) { |  |
| **dlt\_depth\_value**[ layerId ][ j ] | ue(v) |
| } |  |
| } |  |
| … |  |

As specified in the current WD, when DLT is enabled, i.e., dlt\_flag equals 1, the number of valid depth values and each valid depth value are signalled in VPS using Exp-Golomb codes. As a result, large number of coding bits are needed for signalling DLT in VPS, as shown in Table 1.

Table 1: Number of bits used for signaling DLT in HTM-6.0

|  |  |  |
| --- | --- | --- |
|  |  | DLT Size (bits) |
|  |  |
| Balloons | View 0 | 555 |
| View 1 | 641 |
| View 2 | 617 |
| Kendo | View 0 | 485 |
| View 1 | 637 |
| View 2 | 649 |
| Newspaper | View 0 | 606 |
| View 1 | 621 |
| View 2 | 696 |
| PoznanHall | View 0 | 420 |
| View 1 | 380 |
| View 2 | 381 |
| Average | View 0 | 516.5 |
| View 1 | 569.8 |
| View 2 | 585.8 |
| **All Views** | **557.3** |

Based on our observations, the valid depth values of test sequences used in common test conditions have the below two general features:

1. Delta value between any two consecutive depth values is relatively stable. For example, as shown in Figure 1, for sequence “Poznan\_Hall”, the mapping between index values and depth values is close to linear, which indicates that the delta depth values are close to a constant.



**Figure 1: Mapping between index values (horizontal axis) and depth values (vertical axis)**

1. High correlation between different views. For example, for sequence “Poznan\_Hall”, the valid depth values for different views are:

**View 0 (base view)**:

0, 3, 5, 8, 10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35, 38, 40, 43, 45, 48, 50, 53, 55, 58, 60, 63, 65, 68, 70, 73, 75, 78, 80, 83, 85, 88, 90, 93, 95,

**View 1 (non-base view)**:

3, 5, 8, 10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35, 38, 40, 43, 45, 48, 50, 53, 55, 58, 60, 63, 65, 68, 70, 73, 75, 78, 80, 83, 85, 88,

**View 2 (non-base view)**:

0, 3, 5, 8, 10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35, 38, 40, 43, 45, 48, 50, 53, 55, 58, 60, 63, 65, 68, 70, 73, 75, 78, 80, 83, 85, 88,

and it is observed from this example that the valid depth values of view 1 and view 2 can be fully predicted from view 0 (highlighted part) since part of them are overlapped between view 0 and view 1/2.

Based on the above observations, it is noted that the delta depth values may be signaled instead of explicit depth values since the delta values are relatively stable and small, and the DLT in base-view can also be used to predict the DLT in non-base views.

# Proposed Solution

In the proposed method, the difference of two consecutive depth values are signalled instead of explicitly signalling the depth values. In addition, the prediction of DLT from base-views to non-base views may be applied. Currently, the DLT tables are signalled once for a coded video sequence. If there is a need to signal it in picture level, such parameters can be present in slice header extension of picture parameter set extension associated with the layer components of the base view.

Detailed information of the proposed syntax elements and semantics are described as follows:

## DLT signaling for base view

For coding the DLT of base view, the delta depth values are used instead of explicit depth values, and the detailed proposed signaling method is illustrated in Table 2,

Table 2: Proposed DLT signaling for base view

|  |  |
| --- | --- |
| … | Descriptor |
| **num\_depth\_values\_in\_dlt**[ i ] | u(v) |
| **dlt\_depth\_start\_value**[ i] | u(v) |
| **dlt\_depth\_delta\_equal\_flag**[ i ] | u(1) |
| if( dlt\_depth\_delta\_equal\_flag[ i ] ) |  |
| **dlt\_depth\_detla\_value**[ i ] | u(v) |
| else { |  |
| **max\_diff\_minus1**[ i ] | u(v) |
| for ( j = 1; j < num\_depth\_values\_in\_dlt[ i]; j++) { |  |
| **dlt\_depth\_value\_diff\_minus1**[ i ][ j ] | u(v) |
| } |  |
| } |  |
| … |  |

where the definition of newly added syntax (highlighted in blue) in Table 2 is described as below:

**dlt\_depth\_start\_value**[ i ]specifies the first entry in the DLT for current view

**dlt\_depth\_delta\_equal\_flag**[ i ] indicates whether all the delta values between any two consecutive depth values are same.

**dlt\_depth\_detla\_value**[ i ] indicates the delta value between two consecutive depth values, and it is presented when dlt\_depth\_delta\_equal\_flag is equal to 1.

**max\_diff\_minus1**[ i ] plus 1 equals the largest delta depth value between two consecutive depth values.

**dlt\_depth\_value\_diff\_minus1**[ i ][ j ] plus 1 equals the difference of the j-th entry and the (j-1)-th entry in the DLT for current view, which is signalled with log2(ceil(max\_diff\_minus1[ i ]+1)) bits

As it can be seen from Table 2 and the definition of newly added syntax, if dlt\_depth\_delta\_equal\_flag is 1, the DLT can be fully reconstructed using num\_depth\_values\_in\_dlt, dlt\_depth\_start\_value and dlt\_depth\_detla\_value, otherwise, the DLT can be fully reconstructed using num\_depth\_values\_in\_dlt, dlt\_depth\_start\_value and each dlt\_depth\_value\_diff\_minus1.

## DLT signaling in non-base views

For coding DLT in non-base views, it is proposed to use the DLT of base view to predict the DLT of current non-base view, and the detailed proposed signaling method is illustrated in Table 3 as below,

Table 3: Proposed DLT signaling for non-base view

|  |  |
| --- | --- |
|  | Descriptor |
| if (i != 1) |  |
| **inter\_view\_dlt\_pred\_enable\_flag**[ i ] | u(1) |
| if ( i = = 1 || !inter\_view\_dlt\_pred\_enable\_flag[ i ] ) { |  |
| …… |  |
| } |  |
| else { |  |
| **num\_depth\_values\_in\_dlt**[ i ] | u(v) |
| **depth\_overlap\_idc**[ i ] | u(2) |
| if(depth\_overlap\_idc[ i ] == 3 ) |  |
| **number\_left\_nonoverlap\_depth\_values**[ i ] | u(v) |
| if(num\_depth\_values\_in\_dlt[ i ] > num\_depth\_values\_in\_dlt[ 1 ] ) { |  |
| **max\_diff\_minus1**[ i ] | u(v) |
| for ( j = 0; j < num\_depth\_values\_in\_dlt[ i ] - num\_depth\_values\_in\_dlt[ 1 ]; j++) |  |
| **dlt\_depth\_value\_diff\_minus1**[ i ][ j ] | u(v) |
| } |  |
| } |  |

The definition of newly added syntax (highlighted in green) in Table 3 is described as below:

**inter\_view\_dlt\_pred\_enable\_flag**[ i ] indicates whether the DLT of current non-base view is partially or fully predicted by DLT of base view. If it is 0, the DLT of current non-base view will be signalled using the same scheme for base view described above.

When inter\_view\_dlt\_pred\_enable\_flag is 1, the following newly added syntax will be signaled as

**depth\_overlap\_idc**[ i ] specifies the overlapping status of the depth values of the current view. When not present, depth\_overlap\_idc[ i ] is inferred as 0.

**number\_left\_nonoverlap\_depth\_values**[ i ] specifies the number of non-overlapped depth values at the left side, it is presented when **depth\_overlap\_idc**[ i ] equals 3, otherwise, it is inferred as 0.

When inter\_view\_dlt\_pred\_enable\_flag[ i ] is decoded as 1, the non-overlapped depth values of the DLT of current view compared to base view are derived as follows:

Set numDiff = num\_depth\_values\_in\_dlt[ i ]- num\_depth\_values\_in\_dlt[ 1 ]

Set numLeft = (depth\_overlap\_idc[ i ]==1) ? numDiff : number\_left\_nonoverlap\_depth\_values[ i ]

* if ( depth\_overlap\_idc[ i ]&2)   
  for ( j = num\_depth\_values\_in\_dlt[ 1 ] + number\_left\_nonoverlap\_depth\_values[ i ]; j < num\_depth\_values\_in\_dlt[ i ]; j++)

dlt\_depth\_value[ i ][ j ] = dlt\_depth\_value[ i ][ j-1 ] +

dlt\_depth\_value\_diff\_minus1[ i ][ j-num\_depth\_values\_in\_dlt[ 1 ] ] + 1;

* The following apply:  
  for ( j = numLeft -1; j >= 0; j--)

dlt\_depth\_value[ i ][ j ] = dlt\_depth\_value[ i ][ j+1 ] –

(dlt\_depth\_value\_diff\_minus1[ i ][ numLeft -1-j ] + 1);

The overlapped depth values of the DLT of current view compared to base view are derived as follows:

Set numOverlapValues equal to min (num\_depth\_values\_in\_dlt[ i ], num\_depth\_values\_in\_dlt[ 1 ]).

* When depth\_overlap\_idc[ i ] equal to 1, dlt\_depth\_value[ i ][ j+k ] is set equal to dlt\_depth\_value[ 1 ][ j ], wherein j is from 0 to numOverlapValues-1, inclusive, and k is equal to max(num\_depth\_values\_in\_dlt[ i ] - num\_depth\_values\_in\_dlt[ 1 ], 0).
* When depth\_overlap\_idc[ i ] equal to 2, dlt\_depth\_value[ i ][ j ] is set equal to dlt\_depth\_value[ 1 ][ j+k ], where j is from 0 to numOverlapValues-1, inclusive, and k is equal to max(num\_depth\_values\_in\_dlt[ 1 ] - num\_depth\_values\_in\_dlt[ i ], 0).
* When depth\_overlap\_idc[ i ] equal to 3, the following applies
  + When num\_depth\_values\_in\_dlt[ i ] is larger than num\_depth\_values\_in\_dlt[ 1 ], dlt\_depth\_value[ i ][ j+k ] is set equal to dlt\_depth\_value[ 1 ][ j ], where j is from 0 to numOverlapValues-1, inclusive, and k is equal to number\_left\_nonoverlap\_depth\_values[ i ].
  + When num\_depth\_values\_in\_dlt[ i ] is smaller than num\_depth\_values\_in\_dlt[ 1 ], where j is from 0 to numOverlapValues-1, inclusive, and k is equal to number\_left\_nonoverlap\_depth\_values[ i ].

# Implementation and experiments

## Signaling of DLT in VPS

The current software is not fully aligned with the 3D-HEVC WD text. The DLT table is currently still present in SPS in the software although it is present in VPS in the 3D-HEVC WD text. We first aligned the software with the WD text by changing the parameters from SPS to VPS. Therefore, prediction of the DLTs between views is made possible.

In addition, other syntax elements which were incorrectly present in SPS in the software have been moved to VPS.

## Results

### Size of the DLT tables

The proposed method is implemented on top of HTM-6.0, and the coding bits of DLT using proposed method is compared to the coding bits of DLT in HTM-6.0 under common test condition. As shown in Table 4, with the proposed method, the average number of DLT coding bits for all views is 65.4, which is only 12.0% of the number for anchor (557.3 bits).

Table 4: Comparisons on the number of DLT coding bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | DLT Size (bits) | |  |
| Anchor | Proposed | Proposed/Anchor |
| Balloons | View 0 | 555 | 136 | 24.5% |
| View 1 | 641 | 59 | 9.2% |
| View 2 | 617 | 43 | 7.0% |
| Kendo | View 0 | 485 | 124 | 25.6% |
| View 1 | 637 | 59 | 9.3% |
| View 2 | 649 | 65 | 10.0% |
| Newspaper | View 0 | 606 | 121 | 20.0% |
| View 1 | 621 | 21 | 3.4% |
| View 2 | 696 | 31 | 4.5% |
| PoznanHall | View 0 | 420 | 101 | 24.0% |
| View 1 | 380 | 14 | 3.7% |
| View 2 | 381 | 11 | 2.9% |
| Average | View 0 | 516.5 | 120.5 | 23.5% |
| View 1 | 569.8 | 38.3 | 6.4% |
| View 2 | 585.8 | 37.5 | 6.1% |
| **All Views** | **557.3** | **65.4** | **12.0%** |

### RD performance

The results provided in this sub-section are mainly to confirm that the RD performance of the 3D-HEVC is still normal after the changing of the DLT signaling. It is noted that the proposed method has no impact on the PSNR values, and only the bit-rates are changed.

To test the coding performance using proposed method, simulations were performed under both “common test condition” [3] and “Intra-only” configurations [4].

Table 5: BD rate results for 3-view case under CTC

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | video 0 | video 1 | video 2 | video PSNR / video bitrate | video PSNR / total bitrate | synth PSNR / total bitrate | enc time | dec time | ren time |
| Balloons | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.03% | 97% | 103% | 103% |
| Kendo | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.03% | 101% | 104% | 100% |
| Newspapercc | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.03% | 99% | 103% | 96% |
| GhostTownFly | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 103% | 98% | 100% |
| PoznanHall2 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.04% | 102% | 100% | 100% |
| PoznanStreet | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 93% | 88% | 91% |
| UndoDancer | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 94% | 91% | 89% |
| 1024x768 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.03% | 99% | 103% | 100% |
| 1920x1088 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.01% | 98% | 95% | 95% |
| **average** | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.02% | 98% | 98% | 97% |

Table 6: BD rate results for 3-view case under Intra-only

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | video 0 | video 1 | video 2 | video PSNR / video bitrate | video PSNR / total bitrate | synth PSNR / total bitrate | enc time | dec time | ren time |
| Balloons | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 92% | 92% | 93% |
| Kendo | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 94% | 100% | 97% |
| Newspapercc | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 91% | 92% | 94% |
| GhostTownFly | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 99% | 105% | 102% |
| PoznanHall2 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 97% | 95% | 96% |
| PoznanStreet | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 97% | 97% | 100% |
| UndoDancer | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 98% | 99% | 102% |
| 1024x768 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 92% | 95% | 95% |
| 1920x1088 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.00% | 98% | 99% | 100% |
| **average** | **0.0%** | **0.0%** | **0.0%** | **0.0%** | **0.0%** | **0.00%** | **95%** | **97%** | **98%** |

# References

1. F. Jager, “3D-CE6.h: Simplified depth coding with an optional depth lookup table,” JCT3V-B0036, Shanghai, CN, 13–19 Oct. 2012.
2. G. Tech, K. Wegner, Y. Chen, S.Yea, “3D-HEVC Test Model 3,” JCT3V-C1005, Geneva, CH, 17–23 Jan. 2013.
3. D. Rusanovskyy, K. Müller, A. Vetro, "Common Test Conditions of 3DV Core Experiments," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCT3V-C1100, 3rd Meeting: Geneva, CH, 17–23 Jan. 2013.
4. P. Merkle, “Description of Core Experiment 6 (CE6) on Depth Intra Coding,” JCT3V-C1106, Geneva, CH, 17–23 Jan. 2013.

# Patent rights declaration(s)

**Qualcomm Incorporated 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).**