|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  9th Meeting: Sapporo, JP, 3 – 9 July 2014 | Document: JCT3V-I0057 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Clipping Operation for DLT Indexes** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Fabian Jäger Institut für Nachrichtentechnik RWTH Aachen University | Tel: Email: | +49 (0) 241 80 27678 [jaeger@ient.rwth-aachen.de](mailto:jaeger@ient.rwth-aachen.de) |
| *Source:* | RWTH Aachen University | | |

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

# Abstract

This contribution proposes to add a clipping operation when accessing the Depth Lookup Table (DLT) for coding of DC offsets. In B0036 the DLT was proposed as an efficient coding approach for sparse depth maps. In case of transmission errors, predicted DC values and signaled DC offsets may not be aligned anymore and resulting DLT indexes may exceed the defined range of the lookup table. In such cases, the decoder behavior is undefined in the current working draft text. The proposed modification does not have an impact on the performance under CTC.

# Introduction

In 3D video coding, a depth map for each view needs to be encoded besides the conventional video data. These depth maps show different signal characteristics compared to video data as they contain piecewise smooth regions bounded by strong edges. As depth maps are often estimated from texture data or are pre-processed, their histogram might be relatively sparse. As a result, a Depth Lookup Table (DLT) was proposed [1] to exploit the histogram characteristics by only signaling difference indexes of the DLT instead of signaling the residual depth values themselves. By this approach the bit depth of these residual values can be reduced, which consequently results in higher coding efficiency.

The DLT is constructed at the encoder by analyzing the histogram of the original, uncompressed depth map. This DLT is afterwards transmitted to the decoder to allow for the mapping of indexes to actual depth values. In a multiview coding scenario, multiple depth maps may have similar, but still different depth map histograms and in these cases such an updating mechanism is beneficial to the overall coding performance.

# Problem Statement

In the current specification of the DLT, it might happen that due to transmission errors the predicted DC value and the corresponding DLT index is not aligned with the signaled DLT offset index. In such cases, the combined DLT index can exceed the DLT size and the resulting behavior of the decoder is currently not specified. This problem can be avoided by clipping the DLT indexes to the available signal bit depth as this is also the maximum size of the DLT.

# Simulation Results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | video 0 | video 1 | video 2 | video PSNR / video bitrate | video PSNR / total bitrate | synth PSNR / total bitrate | enc time | dec time |
| Balloons | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 101,2% | 100,7% |
| Kendo | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 98,5% | 96,6% |
| Newspaper\_CC | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 103,5% | 99,7% |
| GT\_Fly | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 99,1% | 99,7% |
| Poznan\_Hall2 | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 95,5% | 106,1% |
| Poznan\_Street | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 99,5% | 100,5% |
| Undo\_Dancer | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 97,3% | 103,3% |
| Shark | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 102,4% | 104,0% |
| 1024x768 | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 101,1% | 99,0% |
| 1920x1088 | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 98,8% | 102,7% |
| **average** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **99,6%** | **101,3%** |

# Working Draft Changes

I.8.4.4.2.9 Depth partition value derivation and assignment process

1. …
2. The predicted sample values predSamples[ x ][ y ] are derived as specified in the following:

* For x in the range of 0 to ( nTbS − 1 ), inclusive the following applies:
  + For y in the range of 0 to ( nTbS − 1 ), inclusive the following applies:
    - The variables predDcVal and dcOffset are derived as specified in the following:
      * 1. predDcVal = ( partitionPattern[ x ][ y ] = = partitionPattern[ 0 ][ 0 ] ) ? dcValLT : dcValBR (I‑71)
        2. dcOffset = dcOffsetAvailFlag ? DcOffset[ xTb ][ yTb ][ partitionPattern[ x ][ y ] ] : 0 (I‑72)
    - If DltFlag[ nuh\_layer\_id ] is equal to 0, the following applies:
      * 1. predSamples[ x ][ y ] = predDcVal + dcOffset (I‑73)
    - Otherwise (DltFlag[ nuh\_layer\_id ] is equal to 1), the following applies:
      * 1. predSamples[ x ][ y ] = Idx2DepthValue[ Clip3( 0, ( 1  <<  bitDepth ) − 1, DepthValue2Idx[ predDcVal ] + dcOffset ) ] (I‑74)

Depending on DltFlag[ nuh\_layer\_id ], the variable dcVal is derived as specified in the following:

* If DltFlag[ nuh\_layer\_id ] is equal to 0, the following applies:
  1. DcVal[ 0 ] = dcValLT + ( dcOffsetAvailFlag ? DcOffset[ xTb ][ yTb ][ 0 ] : 0 ) (I‑75)
  2. DcVal[ 1 ] = dcValBR + ( dcOffsetAvailFlag ? DcOffset[ xTb ][ yTb ][ 1 ] : 0 ) (I‑76)
* Otherwise, (DltFlag[ nuh\_layer\_id ] is equal to 1), the following applies:
  1. DcVal[ 0 ] = Idx2DepthValue[ Clip3( 0, ( 1  <<  bitDepth ) − 1, DepthValue2Idx[ dcValLT ] +   
      ( dcOffsetAvailFlag ? DcOffset[ xTb ][ yTb ][ 0 ] : 0 ) ) ] (I‑77)
  2. DcVal[ 1 ] = Idx2DepthValue[ Clip3( 0, ( 1  <<  bitDepth ) − 1, DepthValue2Idx[ dcValBR ] +  
      ( dcOffsetAvailFlag ? DcOffset[ xTb ][ yTb ][ 1 ] : 0 ) ) ] (I‑78)
     + 1. Decoding process for a depth lookup table

This process is only invoked when DltFlag[ nuh\_layer\_id ] is equal to 1.

The list elements Idx2DepthValue[ i ] specifying the depth value of the i-th index in the lookup table with i ranging from 0 to NumDepthValuesInDlt[ nuh\_layer\_id ] − 1, inclusive is derived as follows:

* For i in the range of 0 to NumDepthValuesInDlt[ nuh\_layer\_id ] − 1, inclusive, the elements in Idx2DepthValue are derived as follows:
  + Idx2DepthValue[ i ] is set equal to DltDepthValue[ nuh\_layer\_id ][ i ]
* For i in the range of NumDepthValuesInDlt[ nuh\_layer\_id ] to (1 << BitDepthY) – 1, inclusive, the elements in Idx2DepthValue are set to 0

The list elements DepthValue2Idx[ d ] specifying the index of depth values d in the lookup table with d ranging from 0 to (1 << BitDepthY) − 1, inclusive are derived as specified in the following:

for( d = 0; d < (1 << BitDepthY); d++ ) {  
 idxLower = 0   
 for( iL = 1, foundFlag = 0; iL < NumDepthValuesInDlt[ nuh\_layer\_id ] && !foundFlag; iL++ )   
 if( Idx2DepthValue[ iL ] > d ) {  
 idxLower = iL − 1  
 foundFlag = 1  
 }  
 idxUpper = NumDepthValuesInDlt[ nuh\_layer\_id ] − 1  
 for( iU = NumDepthValuesInDlt[ nuh\_layer\_id ] − 2, foundFlag = 0; iU >= 0 && !foundFlag; iU++ )   
 if ( Idx2DepthValue[ iU ] < d ) {  
 idxUpper = iU + 1  
 foundFlag = 1  
 }  
 if( Abs( d − Idx2DepthValue[ idxLower ] ) < Abs ( d − Idx2DepthValue[ idxUpper ] ) )   
 DepthValue2Idx[ d ] = idxLower  
 else  
 DepthValue2Idx[ d ] = idxUpper

# Conclusion

In this contribution it is proposed to add a clipping operation to the depth value used as an index for the depth lookup table. This clipping operation makes sure that the decoder behavior is well defined in cases of transmission errors where the signaled DC offsets are not synced with the assumed prediction values. The proposed changes do not impact the coding performance under CTC.

# Patent rights declaration

**RWTH Aachen University 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).**

# References

[1] F. Jäger, “3D-CE6.h Results on Simplified Depth Coding with an optional Depth Lookup Table,” Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG, 2nd Meeting, Shanghai, China, JCT3V-B0036, 2012.