|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  5th Meeting: Vienna, AT, 27 July – 02 Aug. 2013 | Document: JCT3V-E0031 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE6: Results on Complexity Reduction for Simplified Depth Coding (SDC)** | | |
| *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 |
| *Source:* | RWTH Aachen University | | |

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

# Abstract

This proposal introduces simplifications to the determination process of the mean prediction values for SDC segments. In the current implementation (HTM 7.0r1), the whole SDC coded block is scanned (sub-sampled by a factor of 2) to compute the average value of each of the two SDC segments before adding the residual DC values.  
In this contribution the DC prediction values are computed depending on the SDC prediction mode. For DC prediction, a single pixel value is used as the SDC prediction value. For DMM prediction, the DMM segment prediction values are reused and for Planar prediction, the four corner values are averaged to get the SDC prediction value.

# SDC Prediction Values

For SDC-coded blocks, a mean prediction value needs to be computed for each SDC segment before the DC offset is added to that prediction value for the final reconstruction step. In HTM 7.0r1 the mean prediction values are computed by averaging all pixels (sub-sampled by a factor of 2) for each SDC segment. This process is relatively memory-intense and can be simplified drastically, as proposed in this contribution.

The proposed computation for SDC prediction DC values is the following (depending on the SDC prediction mode):

|  |  |  |  |
| --- | --- | --- | --- |
| **Prediction Mode** | **Proposed Computation of DC Prediction Value** | **Number of Pixels involved (HTM 7.0r1)** | **Number of Pixels involved (Proposal)** |
| DC | Top-left pixel value of current block | (NxN)/4 | 1 |
| Planar | Mean of four corner pixel values of current block | (NxN)/4 | 4 |
| DMM (Wedgelet) | Four corner pixels associated to each segment | (NxN)/4 | 4 |

# Modifications of Draft Specification Text

**H.8.4.4.3** **Depth value reconstruction process**

**…**

~~3. The variable log2SubSample is set equal to ( nT < 32 ) ? 0 : 1.~~

4. 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\_DC, the following applies.

dcPred[ 0 ] = predSamples[ 0 ][ 0 ]

If intraPredMode is equal to Intra\_Planar, the following applies.

dcPred[ 0 ] = ( predSamples[ 0 ][ 0 ] + predSamples[ 0 ][ nT–1 ] + predSamples[ nT–1 ][ 0 ] + predSamples[ nT–1 ][ nT–1 ] ) >> 2

If intraPredMode is equal to Intra\_DepthPartition( 35 ), the following applies.

for( x = 0; x < nT; x+=(nT–1) ) {  
 for ( y = 0; y < nT; y+=(nT–1) ) {  
 p = wedgePattern[ xS ][ yS ]  
 dcPred[ p ] = predSamples[ x ][ y ]  
 }

# Simulation Results

The simulations were performed according the common test conditions [[1](#Hei11)]. For the All-Intra coding scenario, configuration files from Core Experiment 6 on Simplified Depth Map Coding Tools were used.

The anchor for the simulations was HTM 7.0r1.

## Random Access Coding Configuration (CTC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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% | 0,1% | 0,0% | 0,0% | 0,0% | 0,0% | 99,3% | 99,1% |
| Kendo | 0,0% | 0,1% | 0,0% | 0,0% | 0,0% | 0,0% | 99,6% | 97,6% |
| Newspaper\_CC | 0,0% | 0,0% | -0,1% | 0,0% | 0,0% | -0,1% | 101,2% | 99,3% |
| GT\_Fly | 0,0% | 0,2% | 0,1% | 0,0% | 0,0% | 0,0% | 100,1% | 100,3% |
| Poznan\_Hall2 | 0,0% | 0,1% | 0,2% | 0,1% | 0,0% | 0,0% | 100,8% | 101,8% |
| Poznan\_Street | 0,0% | -0,1% | 0,0% | 0,0% | 0,0% | 0,0% | 98,2% | 97,0% |
| Undo\_Dancer | 0,0% | -0,1% | 0,0% | 0,0% | 0,0% | -0,1% | 99,5% | 99,2% |
| 1024x768 | 0,0% | 0,1% | 0,0% | 0,0% | 0,0% | 0,0% | 100,0% | 98,7% |
| 1920x1088 | 0,0% | 0,0% | 0,1% | 0,0% | 0,0% | 0,0% | 99,7% | 99,6% |
| **average** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **99,8%** | **99,2%** |

## All-Intra Coding Configuration (AI)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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% | 96,2% | 99,2% |
| Kendo | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 99,6% | 99,9% |
| Newspaper\_CC | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 101,0% | 100,4% |
| GT\_Fly | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 102,1% | 98,6% |
| Poznan\_Hall2 | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | -0,1% | 97,2% | 99,5% |
| Poznan\_Street | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 98,7% | 100,1% |
| Undo\_Dancer | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 99,1% | 99,2% |
| 1024x768 | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 98,9% | 99,8% |
| 1920x1088 | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 0,0% | 99,3% | 99,4% |
| **average** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **0,0%** | **99,1%** | **99,6%** |

# Cross Check

The cross check of the proposed modifications to the SDC tool was performed by Intel in JCT3V-E0036. They investigated the source code modifications and ran the simulations for verification of the presented results.

In their investigation they did not find any problems with the source code. Their simulation results perfectly match with those presented in this document.

# Conclusion

In this contribution a complexity reduction for Simplified Depth Coding is proposed. The proposed modifications result in a simplification of the decoding process due to a reduced number of pixel values to be accessed and less additions to be executed. The simulation results show only minimal changes in terms of coding efficiency in both, All-Intra coding configuration and the configuration following the CTC [[2](#Jäg1213)]. At the same time the encoding and decoding time is slightly reduced, but the complexity reduction is clearly visible from the described modifications.

# 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).**