|  |  |
| --- | --- |
| **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-D0183 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE6.h related: Simplified DC predictor for depth intra modes** | | |
| *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 simplified DC prediction for depth intra coding methods including DMM, SDC\_DMM1 and Chain Coding in 3D-HEVC. For each of these modes, each of the two partitions is uniquely predicted by a single value, i.e., DC value. To predict the DC value using neighboring reference sample, a DC prediction process is performed which involves a number of additions for large PU sizes and divisions which is not desirable. For many cases, the current DC prediction process can introduce large prediction error which is caused by inaccurate partitioning of neighboring reference samples. To reduce the complexity as well as to improve the accuracy of the current DC prediction process, a simplified DC predictor is proposed in this contribution. It is reported that -0.24% BD-rate gain is achieved using proposed method under common test condition on synthesized views, and -0.14% BD-rate gain is achieved under Intra-only test condition on synthesized views.

# Introduction

In 3D-HEVC, partition-based methods including DMM, SDC\_DMM1 and Chain Coding are applied for depth map intra coding. With partition-based depth map intra coding methods, a depth block is partitioned by two parts, i.e., P0 and P1 as illustrated in Figure 1, and each part is represented by a single DC value. Furthermore, the DC value (DC0 and DC1) for each partition (P0 and P1) can be further compensated by a delta DC/residual value which is explicitly signaled in the bitstream.

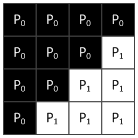
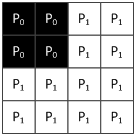
 

Figure 1: Examples of depth block partitioning

## DC prediction in current 3D-HEVC

The DC value of each partition is predicted using all the reconstructed neighboring reference samples. For example, in Figure 2, the samples within an 8x8 depth block are denoted by ci,j, where i, j = 0 ,1, …, 7, and the reconstructed neighboring reference samples are denoted by pi,j, where i = −1, j = −1..7 and i = 0..7, j = −1.

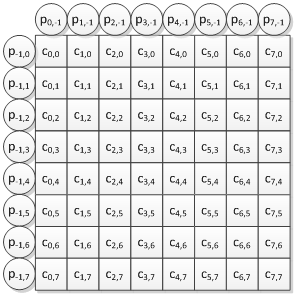


Figure 2: An 8×8 depth block and its neighboring reference pixels

Assume the depth value is using *B*-bit representation, e.g., for the case of *B* being 8, the depth value ranges from 0 to 255. Given the partition pattern which is an *N*×*N* block composed of binary values, i.e., *bPattern*x,y, where x = 0..*N* − 1, y = 0..*N*−1, the below steps are applied to derive the predicted DC value for each partition, i.e., DC0 and DC1,

* **Step 1**: Set DC0 and DC1 as , set variables S0, S1, N0 and N1 equal to zero, set variable nStep = *N*≥32 ? 2 : 1;
* **Step 2**: For i = 0..*N*−1, x += nStep, the following applies:
  + If *bPattern*i,0 equal to 0, set S0 += pi,-1 and N0++;
  + Otherwise, set S1+= pi,-1 and N1 +=1;
* **Step 3**: For i = 0..*N*−1, x += nStep, the following applies:
  + If *bPattern*0,i equal to 0, S0+=p-1,i and N0++;
  + Otherwise, S1+=p-1,i and N1+=1;
* **Step 4**:If *N0* is not zero, set DC0 as ; If *N1* is not zero, set DC1 as .

## Problems of the current design

According to the above DC prediction process, given the partition pattern and reference samples, the predicted DC values are derived, and the prediction block can be generated by setting the samples located in P0 and P1 in the block as DC0 and DC1, respectively.

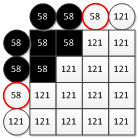
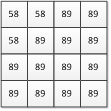
The current DC prediction process described above has the following problems.

### Unnecessary computational complexity

* **A number of addition operations are needed for large block sizes**. For example, for a 16×16 or 32×32 PU, there are 32 additions to calculate the sum of reference samples belonging to each partition, and 32 additions to count the number of reference samples belonging to each partition.
* **Division operation is needed for both encoder and decoder**. There are division operations in step 4 of the above DC derivation process which are complex for video codec and not desirable.

### Inaccurate DC predictors

* **The reference samples may be partitioned inaccurately**. For example, in Figure 3(a), according to the derivation process of predicted DC values described above, two reference samples, highlighted in red, are partitioned inaccurately and the resulting DC0 and DC1 will be 89 and 58. With the derived DC values, the prediction block in Figure 3(b) is generated, and large prediction error is observed from the difference between the prediction block of the original block.

1. (b)

Figure 3: A depth PU partition pattern (a) and the resulting prediction block (b).

Based on the above analysis, the current DC predictor for partition-based depth map intra coding methods is still relatively high complexity and even inaccurate for many cases.

# Proposed solution

The proposed method only uses one or two reference samples to derive predicted DC values for each partition.

As shown in Figure 4, only two samples of the first sample, last sample or middle sample of the top reference sample row or the left reference sample column may be chosen to generate a DC predator under different situations. Therefore, each time, the DC predictor is calculated by only up to two samples.

Assume the partition value of the left-top sample (c0,0) is X, where X= 0 or 1, given the partition pattern *bPattern*x,y, where x = 0..*N* − 1, y = 0..*N*−1, the predicted DC values, i.e., DC0 and DC1, are derived by follows:

Set bT = (*bPattern*0,0 != *bPattern*N-1,0)? 1 : 0; set bL = (*bPattern*0,0 != *bPattern*0,N-1)? 1 : 0

If bT equals bL

* DCX = (p-1,0 + p0,-1)>>1
* DC1-X = bL ? (p-1,N-1 + pN-1,-1)>>1 : 2B-1

Otherwise

* DCX = bL ? p(N-1)>>1,-1 : p-1, (N-1)>>1
* DC1-X = bL ? p-1,N-1 : pN-1,-1

(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

Figure 4: Selection of reference samples for difference partition pattern cases.

# Experimental results

The proposed method is implemented on top of HTM-6.0, and simulations were performed under both “common test condition” [1] and “Intra-only” [2] configurations.

The results are summarized in Table 1 and Table 2. As it is reported, the proposed method achieves small coding gain: -0.24% and -0.14% BD-rate on synthesized views for “CTC” and “Intra-only”, respectively.

Table 1: 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% | -0.15% | 98% | 99% | 96% |
| Kendo | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.02% | 99% | 93% | 95% |
| Newspapercc | 0.0% | -0.2% | 0.1% | 0.0% | -0.1% | -0.24% | 95% | 93% | 93% |
| GhostTownFly | 0.0% | 0.0% | 0.1% | 0.0% | 0.0% | -0.40% | 101% | 96% | 97% |
| PoznanHall2 | 0.0% | -0.2% | -0.4% | -0.1% | -0.2% | -0.21% | 100% | 100% | 103% |
| PoznanStreet | 0.0% | 0.3% | 0.0% | 0.0% | 0.0% | -0.03% | 94% | 90% | 93% |
| UndoDancer | 0.0% | -0.2% | -0.2% | -0.1% | -0.1% | -0.65% | 97% | 95% | 92% |
| 1024x768 | 0.0% | -0.1% | 0.0% | 0.0% | 0.0% | -0.13% | 97% | 95% | 95% |
| 1920x1088 | 0.0% | 0.0% | -0.1% | 0.0% | -0.1% | -0.32% | 98% | 95% | 96% |
| **average** | **0.0%** | **0.0%** | **-0.1%** | **0.0%** | **-0.1%** | **-0.24%** | **98%** | **95%** | **96%** |

Table 2: 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% | -0.09% | 94% | 97% | 97% |
| Kendo | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.11% | 96% | 96% | 95% |
| Newspapercc | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.20% | 97% | 96% | 96% |
| GhostTownFly | 0.0% | 0.0% | 0.0% | 0.0% | -0.2% | -0.40% | 105% | 100% | 101% |
| PoznanHall2 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.09% | 100% | 103% | 99% |
| PoznanStreet | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | -0.05% | 103% | 103% | 106% |
| UndoDancer | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.04% | 100% | 103% | 105% |
| 1024x768 | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.13% | 95% | 96% | 96% |
| 1920x1088 | 0.0% | 0.0% | 0.0% | 0.0% | -0.1% | -0.14% | 102% | 102% | 103% |
| **average** | **0.0%** | **0.0%** | **0.0%** | **0.0%** | **-0.1%** | **-0.14%** | **99%** | **100%** | **100%** |

# References

1. D. Rusanovskyy, K. Müller, A. Vetro, “Common Test Conditions of 3DV Core Experiments,” JCT3V-C1100, Geneva, CH, 17–23 Jan. 2013.
2. 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).**