|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  10th Meeting: Strasbourg, FR, 18–24 Oct. 2014 | Document: JCT3V-J0023 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Simplification of depth prediction in DMM** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Takeshi Tsukuba  Tomohiro Ikai  1-9-2 Nakase, Mihama-ku, Chiba-shi, Chiba 261-8520, JAPAN | Tel: Email: | +81-43-299-8526 [tsukuba.takeshi@sharp.co.jp](mailto:tsukuba.takeshi@sharp.co.jp) |
| *Source:* | SHARP Corporation | | |

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

# Abstract

This contribution proposes a simplified derivation process of predicted DC offsets for DMM by utilizing only partition pattern conditions. Experimental result reportedly shows -0.01 % and -0.01 % BD-rate changes in synthesis in CTC and all-intra configuration respectively.

# Introduction

When Depth Modeling Mode (DMM) is utilized, the pixels in the PU is divided into two segments, R0 and R1, based on a given partition pattern. To derive DC values for each segment, the PU is also classified into five cases based on the partition pattern along with the difference between neighbouring depth values. Then the DC values, dcValLT and dcValBR, are predicted using the reference samples, e.g. P0,-1, P-1,-1, etc…, chosen by the classification as shown in Figure 1.



Figure : Selection of reference samples for each case in HTM12

In the formula, DC predictor dcValLT and dcValBR for R0 and R1 are derived by the partition pattern \*1 along with the difference between neighbouring depth values\*2 as follows:

vertEdgeFlag = ( partitionPattern[ 0 ][ 0 ]  !=  partitionPattern[ nTbS − 1 ][ 0 ] )

horEdgeFlag = ( partitionPattern[ 0 ][ 0 ]  !=  partitionPattern[ 0 ][ nTbS − 1 ] )

(a) Case 1: When (horEdgeFlag, vertEdgeFlag) is (1, 1)

dcValLT = (P0,-1 + P-1,0 )>>1

dcValBR = (PN-1,-1 + P-1,N-1)>>1

(b)(c) Case 2 and Case 3: When (horEdgeFlag, vertEdgeFlag) is (0, 0)

dcValLT = (P0,-1 + P-1,0 )>>1

dcValBR = horAbsDiff > vertAbsDiff \*2 ? P2N-1,-1 :P-1,2N-1

where vertAbsDiff = abs ( P-1,0 – P-1,2N-1 ) and horAbsDiff = abs ( P0,-1, - P2N-1,-1 )

(d) Case 4: When (horEdgeFlag, vertEdgeFlag) is (1, 0)

dcValLT = P(N-1)>>1,-1

dcValBR = P-1,N-1

(e) Case 5: When (horEdgeFlag, vertEdgeFlag) is (0, 1)

dcValLT = P-1,(N-1)>>1

dcValBR = PN-1,-1

**Motivation**

The specified above process is partially quite simple. The values of horEdgeFlag and vertEdgeFlag can be immediately computed by the DMM1 table number or DMM4 pattern calculation. However there is an additional process using the difference between neighbouring depth values. We think the effectiveness of additional process should be questioned because it may complicate the design by bringing about unnecessarily access and computations.

# Proposed Method

It is proposed that

1. Remove depth difference base condition (modification 1)
2. Modify positions to achieve better coding balance under modification 1 (modification 2)

With the proposals we have **only three cases** by using only horEdgeFlag and vertEdgeFlag values.

(a) Case 1: When (horEdgeFlag, vertEdgeFlag) is either (1, 1) or (0, 0)

dcValLT = (P0,-1 + P-1,0 )>>1

dcValBR = (P2N-1,-1 + P-1,2N-1)>>1

(b)Case 2: When (horEdgeFlag, vertEdgeFlag) is (1, 0)

dcValLT = P(N-1)>>1,-1

dcValBR = P-1,N

(c)Case 3: When (horEdgeFlag, vertEdgeFlag) is (0, 1)

dcValLT = P-1,(N-1)>>1

dcValBR = PN,-1



Figure : Selection of reference samples for each case in Proposal

# Text changes

Modification1 is highlighted in yellow.

Modification2 is highlighted in green.

I.8.4.4.2.10 Depth partition value derivation and assignment process

Inputs to this process are:

* the neighbouring samples p[ x ][ y ], with x = −1, y = −1..nTbS \* 2 − 1 and x = 0..nTbS \* 2 − 1, y = −1,
* a binary array partitionPattern[ x ][ y ], with x, y =0..nTbS − 1, specifying a partitioning of the prediction block in a partition 0 and a partition 1,
* a sample location ( xTb, yTb ) specifying the top-left sample of the current block relative to the top‑left sample of the current picture,
* a variable nTbS specifying the transform block size.

Output of this process is:

* the predicted samples predSamples[ x ][ y ], with x, y = 0..nTbS − 1.

The variables vertEdgeFlag and horEdgeFlag are derived as specified in the following:

* 1. vertEdgeFlag = ( partitionPattern[ 0 ][ 0 ]  !=  partitionPattern[ nTbS − 1 ][ 0 ] ) (I‑61)
  2. horEdgeFlag = ( partitionPattern[ 0 ][ 0 ]  !=  partitionPattern[ 0 ][ nTbS − 1 ] ) (I‑62)

The variables dcValBR and dcValLT are derived as specified in the following:

* If vertEdgeFlag is equal to horEdgeFlag, the following applies:
  + The variable dcValBR is derived as follows:

dcValBR = ( ( p[ −1 ][ 2 \* nTbS − 1 ] + p[ 2 \* nTbS − 1 ][ −1 ] ) >> 1 ) (I‑63)

* + - ~~If horEdgeFlag is equal to 1, the following applies:~~
      * 1. ~~dcValBR = ( ( p[ −1 ][ nTbS − 1 ] + p[ nTbS − 1 ][ −1 ] ) >> 1 ) (~~I~~‑63)~~
    - ~~Otherwise (horEdgeFlag is equal to 0), the following applies:~~ 
      * 1. ~~vertAbsDiff = Abs( p[ −1 ][ 0 ] − p[ −1 ][ nTbS \* 2 − 1 ] ) (~~I~~‑64)~~
        2. ~~horAbsDiff = Abs( p[ 0 ][ −1 ] − p[ nTbS \* 2 − 1 ][ −1 ] ) (~~I~~‑65)~~
        3. ~~dcValBR = ( horAbsDiff > vertAbsDiff ) ? p[ nTbS \* 2 − 1 ][ −1 ] : p[ −1 ][ nTbS\*2 − 1 ] ) (~~I~~‑66)~~
  + The variable dcValLT is derived as follows:

dcValLT = ( p[ −1 ][ 0 ] + p[ 0 ][ −1 ] ) >> 1 (I‑67)

* Otherwise (horEdgeFlag is not equal to vertEdgeFlag), the following applies:
  + 1. dcValBR = horEdgeFlag ? p[ −1 ][ nTbS ~~− 1~~ ] : p[ nTbS ~~− 1~~ ][ −1 ] (I‑68)
    2. dcValLT = horEdgeFlag ? p[ ( nTbS − 1 ) >> 1 ][ −1 ] : p[ −1 ][ ( nTbS − 1 ) >> 1 ] (I‑69)

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‑70)
        2. dcOffset = DcOffset[ xTb ][ yTb ][ partitionPattern[ x ][ y ] ] (I‑71)
    - If DltFlag[ nuh\_layer\_id ] is equal to 0, the following applies:
      * 1. predSamples[ x ][ y ] = predDcVal + dcOffset (I‑72)
    - Otherwise (DltFlag[ nuh\_layer\_id ] is equal to 1), the following applies:
      * 1. predSamples[ x ][ y ] = Idx2DepthValue[ Clip1Y( DepthValue2Idx[ predDcVal ] + dcOffset ) ] (I‑73)

# Experimental results

Table 1 and Table 2 show the simulation results of the proposed method. Common test conditions specified in JCT3V-I1100 [2] is used for the evaluation. The simulation results show that the proposed method shows -0.01 % and -0.01 % BD-rate change in synthesis in CTC and all-intra configuration respectively.

Table 1: Result in CTC (TEST1\_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 | 0.00% | -0.02% | -0.03% | -0.01% | -0.03% | -0.05% | 104.7% | 103.0% | 104.2% |
| Kendo | 0.00% | -0.13% | 0.02% | 0.00% | 0.00% | 0.07% | 102.6% | 101.5% | 102.3% |
| Newspaper\_CC | 0.00% | -0.06% | -0.03% | -0.02% | -0.07% | -0.15% | 96.4% | 97.3% | 96.6% |
| GT\_Fly | 0.00% | -0.01% | -0.06% | 0.00% | -0.01% | -0.06% | 100.4% | 100.3% | 99.9% |
| Poznan\_Hall2 | 0.00% | 0.21% | 0.24% | 0.09% | 0.09% | 0.13% | 96.5% | 98.2% | 96.8% |
| Poznan\_Street | 0.00% | -0.05% | 0.03% | 0.00% | -0.01% | -0.02% | 104.6% | 103.5% | 104.3% |
| Undo\_Dancer | 0.00% | -0.11% | -0.06% | -0.02% | -0.01% | -0.05% | 96.0% | 98.0% | 96.6% |
| Shark | 0.00% | 0.15% | 0.01% | 0.03% | 0.04% | 0.07% | 100.2% | 100.9% | 100.1% |
| 1024x768 | 0.00% | -0.07% | -0.01% | -0.01% | -0.03% | -0.04% | 101.3% | 100.6% | 101.0% |
| 1920x1088 | 0.00% | 0.04% | 0.04% | 0.02% | 0.02% | 0.02% | 99.5% | 100.2% | 99.5% |
| **average** | **0.00%** | **0.00%** | **0.02%** | **0.01%** | **0.00%** | **-0.01%** | **100.2%** | **100.3%** | **100.1%** |

Table 2: Result in all-intra configuration (TEST1\_AI)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | -0.01% | 97.8% | 98.0% | 98.2% |
| Kendo | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | -0.01% | 99.8% | 98.9% | 99.8% |
| Newspaper\_CC | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | 0.01% | 102.1% | 101.6% | 101.8% |
| GT\_Fly | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | -0.02% | 99.6% | 99.8% | 100.2% |
| Poznan\_Hall2 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | -0.02% | 96.1% | 97.4% | 96.5% |
| Poznan\_Street | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | -0.02% | 104.2% | 103.4% | 103.3% |
| Undo\_Dancer | 0.00% | 0.00% | 0.00% | 0.00% | 0.02% | 0.00% | 95.8% | 97.3% | 96.3% |
| Shark | 0.00% | 0.00% | 0.00% | 0.00% | 0.03% | 0.02% | 98.2% | 98.5% | 97.8% |
| 1024x768 | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | 0.00% | 99.9% | 99.5% | 99.9% |
| 1920x1088 | 0.00% | 0.00% | 0.00% | 0.00% | 0.01% | -0.01% | 98.8% | 99.3% | 98.8% |
| **average** | **0.00%** | **0.00%** | **0.00%** | **0.00%** | **0.01%** | **-0.01%** | **99.2%** | **99.4%** | **99.2%** |

Additional results on individual modifications:

Results of modification 1 only in CTC shows 0.03 % average loss mainly due to small loss of Poznan Hall and Shark, while the result of modification 2 in CTC shows no gain on average (but 0.03 % average gain in all intra). The combination of modification 1 and 2 shows better results than individual modifications’. Thus it can be said that the combination of 1 and 2 shows better balance in terms of coding efficiency. That means no coding loss (-0.01 % in CTC and -0.01 % in all intra).

Table 3: Result of modification 1 only in CTC (TEST2\_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 | 0.00% | -0.20% | 0.04% | -0.03% | -0.03% | -0.03% | 100.2% | 100.3% | 100.3% |
| Kendo | 0.00% | -0.10% | -0.02% | -0.02% | -0.02% | 0.05% | 104.6% | 103.4% | 104.3% |
| Newspaper\_CC | 0.00% | 0.11% | 0.06% | 0.02% | 0.00% | -0.01% | 94.3% | 95.9% | 94.3% |
| GT\_Fly | 0.00% | -0.08% | -0.09% | -0.01% | 0.01% | 0.00% | 100.7% | 100.4% | 100.2% |
| Poznan\_Hall2 | 0.00% | 0.23% | 0.33% | 0.10% | 0.11% | 0.15% | 100.2% | 100.1% | 100.1% |
| Poznan\_Street | 0.00% | 0.05% | 0.08% | 0.03% | 0.03% | 0.02% | 100.5% | 100.7% | 100.2% |
| Undo\_Dancer | 0.00% | -0.14% | -0.08% | -0.03% | 0.00% | -0.04% | 99.4% | 100.0% | 99.4% |
| Shark | 0.00% | 0.17% | -0.01% | 0.02% | 0.05% | 0.09% | 98.9% | 99.1% | 100.3% |
| 1024x768 | 0.00% | -0.07% | 0.03% | -0.01% | -0.02% | 0.00% | 99.7% | 99.8% | 99.6% |
| 1920x1088 | 0.00% | 0.05% | 0.05% | 0.02% | 0.04% | 0.04% | 99.9% | 100.1% | 100.0% |
| **average** | **0.00%** | **0.00%** | **0.04%** | **0.01%** | **0.02%** | **0.03%** | **99.8%** | **100.0%** | **99.9%** |

Table 4: Results of modification 2 only in CTC (TEST3\_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 | 0.00% | -0.19% | 0.05% | -0.02% | -0.05% | -0.01% | 100.1% | 103.5% | 100.3% |
| Kendo | 0.00% | -0.04% | 0.08% | 0.03% | -0.01% | 0.01% | 100.4% | 102.2% | 101.4% |
| Newspaper\_CC | 0.00% | 0.05% | 0.18% | 0.02% | -0.03% | -0.07% | 100.6% | 103.0% | 100.0% |
| GT\_Fly | 0.00% | -0.01% | -0.12% | -0.01% | -0.01% | -0.05% | 100.2% | 98.6% | 99.9% |
| Poznan\_Hall2 | 0.00% | 0.32% | 0.03% | 0.07% | 0.06% | 0.19% | 100.1% | 99.3% | 100.1% |
| Poznan\_Street | 0.00% | 0.10% | 0.02% | 0.03% | 0.02% | -0.02% | 100.4% | 102.0% | 99.8% |
| Undo\_Dancer | 0.00% | -0.05% | -0.05% | -0.01% | -0.02% | -0.07% | 100.3% | 101.0% | 99.9% |
| Shark | 0.00% | 0.04% | -0.02% | 0.00% | 0.00% | -0.01% | 100.1% | 100.0% | 100.0% |
| 1024x768 | 0.00% | -0.06% | 0.10% | 0.01% | -0.03% | -0.02% | 100.4% | 102.9% | 100.6% |
| 1920x1088 | 0.00% | 0.08% | -0.03% | 0.01% | 0.01% | 0.01% | 100.2% | 100.2% | 99.9% |
| **average** | **0.00%** | **0.03%** | **0.02%** | **0.01%** | **0.00%** | **0.00%** | **100.3%** | **101.2%** | **100.2%** |

# Conclusion

This contribution proposes a simplified derivation process of predicted DC offset. Experimental result reportedly shows -0.01 % and -0.01 % BD-rate changes in synthesis in CTC and all-intra configuration respectively, suggesting no coding loss.

It is recommended to adopt the proposed method into the 3D-HEVC.

# Patent rights declaration(s)

**SHARP Corporation 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. G. Tech, K. Wegner, Y. Chen and S. Yea, “3D-HEVC Draft Text 5,” JCT3V-I1001, July 2014.
2. K. Müller, A. Vetro, “Common test conditions of 3DV core experiments,” JCT3V-I1100, July 2014.
3. 3DV-HTM version 12.0: <https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-12.0/>