|  |  |
| --- | --- |
| **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-I0194 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **BoG report on DBBP (Depth Based Block Partitioning)** | | |
| *Status:* | Input Document to JCT-3V | | |
| *Purpose:* | Report | | |
| *Author(s) or Contact(s):* | Sehoon Yea | Email: | sehoon.yea@lge.com |
| *Source:* | BoG | | |

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

# Summary

The BoG met on 5th of July 2014 to review contributions related to DBBP (Depth Based Block Partitioning). A total of 15 input contributions were presented and discussed during the BoG meeting. These contributions can be categorized as in the following table. Several recommendations were made by BoG regarding the disposition of the input contributions.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | Samsung | Qualcomm | MediaTek | RWTH | Sharp | Hisilicon |
| Partition Derivation | I0077,  I0076 |  | I0097,  I0094 | I0056 | I0107 | I0070,  I0125 |
| Filter Simplification | I0088 |  | I0094 |  | I0109 | I0124 |
| Memory  Requirement | I0078 |  |  |  | I0108 |  |
| Motion/Disparity  Derivation |  | I0054 |  | I0058 |  |  |

# Document Review Details

## Partition Derivation

[JCT3V-I0097](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2180) Removal of the decoder-side DBBP partition derivation process [J.-L. Lin, Y.-W. Chen, X. Zhang, K. Zhang, Y.-W. Huang, S. Lei (MediaTek)]

- Abstract

To signal the depth-based block partitioning (DBBP) mode, a *dbbp\_flag* is signaled at CU level after a partition mode (*part\_mode*) is transmitted. When the DBBP is enabled, the transmitted partition mode will be further replaced by a modified partition mode which is implicitly derived from the corresponding depth block. Since the partition mode is always transmitted to the decoder to indicate the PU partition of current CU even when the DBBP mode is enabled, the partition derivation process for DBBP on the decoder-side in not necessary and could be totally removed. It is proposed to let encoder directly transmit the DBBP partition mode to the decoder through the *part\_mode* syntax and remove the decoder-side DBBP partition derivation process. The experiment results reportedly show no coding loss.

- Notes from BoG discussion

* not necessary to derive the partition mode as it is signaled by encoder
* Two related syntax changes are proposed (Test1/Test2)

: reports no impact on coding performance, WD provided

• BoG identified there are some minor differences in experimental results (e.g. due to the misalignment of WD with the SW regarding initialization values of context model for DBBP flag signaling)

• BoG recommends adoption of the removal of decoder-side partition derivation (e.g. Test1 in I0097 with the proposed WD change, all common in I0056, I0077,I0070,I0097) and adoption of syntax condition of the partition mode as per Test2 (the corresponding WD text is also provided in I0097). Use the WD (harmonized with I0125) and the SW as provided in I0077.

[JCT3V-I0077](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2160) Partition Derivation for DBBP [M. W. Park, J. Y. Lee, Y. Cho, C. Kim (Samsung)]

- Abstract

In the current 3D-HEVC, when the DBBP mode is enabled, the partition mode used for storing the motion parameters needs to be decided between PART\_2NxN and PART\_Nx2N at the decoder side. However, the partition derivation has a relatively high complexity since it needs to derive a threshold and then to generate a 4x4 block-wise segmentation map by using the derived threshold. To reduce the complexity of the partition derivation, two options of simplified methods are proposed. The first option is to use three corner samples in the corresponding depth block at decoder side and to signal the enabling flag only when the partition mode is PART\_2NxN to remove redundant signaling of the flag. The first option reportedly provides 0.07% bit-saving on coded views. And the other option is to completely remove the partition derivation process from the decoder and to signal the enabling flag only when the partition mode is PART\_2NxN and PART\_Nx2N so that received partition mode can be used as is to store the motion parameters. The second option reportedly provides 0.03% bit-saving on coded views.

- Notes from BoG discussion

* essentially the same approach as Test1 & Test2 in I0097

[JCT3V-I0056](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2139) Cleanup of Depth-based Block Partitioning (DBBP) [F. Jäger (RWTH Aachen University)]

- Abstract

At the 6th JCT-3V meeting, a motion/disparity prediction method was introduced in JCT3V-G0106 [1] that uses a depth-derived binary segmentation mask for the derivation of PU partitioning and for merging of two prediction signals. The method is called Depth-based Block Partitioning (DBBP). The segmentation process in JCT3V-G0106 results in two segments, which motion information is coded as a conventional bi-partitioned coding unit with two sets of motion information. At the 7th meeting a minor modification to the partitioning derivation process and the signalling of the DBBP flag was proposed in JCT3V-H0094 [2], which implied some clean-up work to be required. This contribution proposed this clean-up of various aspects of DBBP. The resulting impact on the coding performance is a very minor gain of 0.03% under common test conditions.

- Notes from BoG discussion

* Essentially the same approach as Test1 & Test2 in I0097

[JCT3V-I0107](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2190) Partition decision unification for DBBP and VSP [T. Ikai, T. Tsukuba (Sharp)]

- Abstract

This contribution proposes a unified partition decision for DBBP and VSP. The proposed method utilizes the unified partition method which uses four corner of the target block. The BD-rate change is reported as -0.01 %, -0.01 % and -0.01% in texture, video and synthesis respectively.

- Notes from BoG discussion

* No need to be discussed with the recommended adoption of I0097

[JCT3V-I0070](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2153) Simplification of partition derivation in DBBP [C. Zhu, Y.-H. Zhang (UESTC), Y.-B. Lin, J.-H. Zheng (HiSilicon), J.-J. Song (UESTC)]

- Abstract

In the current 3D-HEVC, the depth-based block partitioning (DBBP) [1] comprises steps of partitioning derivation, segment mask generation, bi-segment compensation, and merging of prediction signals. The partitioning derivation process, with derived virtual depth as input, is relatively high complex, as it needs to perform sub-sample mean value calculation and to search the most matched partition mode. In this contribution, two methods are proposed to simplify the partition derivation for DBBP. Method #1 proposes to use default partition mode in order to remove the existing process of DBBP partition derivation. Method #2 proposes to derive the partition by simply comparing the mask value of two cornor pixels. Experimental results reportedly show the proposed simplifications bring almost no coding efficiency loss.

- Notes from BoG discussion

* Method#1 (default partition mode)

=> propose to use Nx2N only as it shows no coding impact (BD-loss of 0.01%, entirely remove the need for derivation both at encoder & decoder)

=> General comment: Nx2N and 2NxN motion-vector prediction is already in the HEVC version 1

* Method#2 (simplification of derivation)

=> Removes the need for derivation of the anchor method, but employs a condition check to adaptively choose between Nx2N and 2NxN

=> BD-savings of -0.01%

* Proponent prefers Method#1

[JCT3V-I0076](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2159) Simplification of Threshold Derivation for DBBP and DMM4 [M. W. Park, J. Y. Lee, Y. Cho, C. Kim (Samsung)]

- Abstract

In the current 3D-HEVC, DBBP and DMM4 need to generate a binary segmentation mask used for the segmentation based prediction. The binary segmentation mask is generated by using the threshold derived by averaging all the samples in the corresponding depth or texture block. However, accessing all the samples could cause increasing the computation complexity. Therefore, in order to reduce the complexity, this contribution proposes to use four corner samples in the corresponding block to derive the threshold. The proposed method reportedly provides no coding loss with 96.7% encoding time under the common test conditions. For DMM4, the results based on the all intra test conditions reportedly show 0.1% bit-saving for synthesized views.

- Notes from BoG discussion

* Propose to use four-corner samples in calculating the threshold value common in DBBP and DMM4.
* No coding impact. Simplifies threshold calculation.
* BoG recommends adoption of this proposal (I0076)

[JCT3V-I0094](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2177) DBBP simplification based on sub-block level processing [X. Zhang, K. Zhang, J. An, H. Huang, J.-L. Lin, S. Lei (MediaTek)]

- Abstract

In the current 3D-HEVC, depth-based block partitioning (DBBP) is designed to generate a segment mask, combine two predicted blocks and internally filter the combined block all at pixel-level. In this contribution, it is proposed to calculate the mask values at 2x2 sub-block level to reduce complexity of DBBP. Experimental results reportedly show such a simplification brings no coding loss.

- Notes from BoG discussion

* On simplification on mask-generation

=> proposes to use 2x2 subblock to generate the mask (i.e. use one pixel out of 4 in a 2x2 subblock)

=> reduces half of add/cmp operations in filtering, mask generation, and block combination processes with no coding impact. The major benefit, as claimed by the proponent, is block-fetch step can be more regular.

=> it was suggested the numbers regarding overall (MC+DBBP) computational savings be provided

[JCT3V-I0125](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2208) Bug-fixing of depth-based block partitioning signaling [Y. Lin, J. Zheng (HiSilicon), X. Wang, C. Zhu (UESTC)]

- Abstract

In the current 3D-HEVC, the block partition mode of depth-based block partitioning (DBBP) is selected among 2NxN and Nx2N. In the syntax design [1], an encoder signals a DBBP flag to a decoder in a condition that the coding CU block should not be coded with intra mode. However, this condition will lead to the DBBP flag transmission when the block is coded in inter mode with other kinds of partition modes. Moreover, in the corresponding software HTM-11.0 [2], this condition is missed which additionally lead to the DBBP flag transmission when the block is coded in intra mode. In this proposal, a solution has been proposed to fix this bug. The experiment results reportedly show the rate-saving.

- Notes from BoG discussion

* DBBP\_flag is still being signaled when the partition mode is neither of Nx2N or 2NxN (both in SW & WD)
* See notes for the proposal I0097 in Section 1.1
* BoG recommends to adopt I0125

=> Upload a harmonized version of the WD text as a revision of I0077

### Crosschecks for Contributions in *Partition Derivation*

### [JCT3V-I0183](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2271) Crosscheck on Partition Derivation for DBBP (JCT3V-I0077) [S. Shimizu (NTT)] [late]

### [JCT3V-I0186](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2274) Crosscheck on removal of the decoder-side DBBP partition derivation process (JCT3V-I0097) [S. Shimizu (NTT)] [late]

### [JCT3V-I0065](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2148) Cross check of cleanup of DBBP (JCT3V-I0056) [P. Merkle (HHI)]

[JCT3V-I0145](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2228) Cross check of Partition decision unification for DBBP and VSP (JCT3V-I0107) [M. W. Park, C. Kim (Samsung)]

[JCT3V-I0148](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2231) Cross check of Simplification of partition derivation in DBBP (JCT3V-I0070) [M. W. Park, C. Kim (Samsung)] [late]

[JCT3V-I0153](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2236) Cross check of DBBP simplification based on sub-block level processing (JCT3V-I0094) [J. Y. Lee, C. Kim (Samsung)] [late]

[JCT3V-I0138](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2221) Crosscheck for JCT3V-I0076 Simplification of Threshold Derivation for DBBP and DMM4 [H. Huang (MediaTek)] [late]

[**JCT3V-I0163**](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2246) **Cross-check of Bug-fixing of depth-based block partitioning signaling (JCT3V-I0125) [T. Ikai, T. Tsukuba (Sharp)] [late]**

## Filter Simplification

[JCT3V-I0109](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2192) One dimensional DBBP boundary filtering [T. Ikai, T. Tsukuba (Sharp)]

- Abstract

This contribution proposes a simplification for DBBP filtering. The proposal replaces two dimensional horizontal and vertical filtering with one dimensional horizontal or vertical filtering, which decided on the basis of PartMode. It is reported that the BD-rate change is 0.01 %, 0.01 % and 0.01 % in texture, video, and synthesis respectively.

Revision1 added option2, which further simplifies the decision by replacing left/current/right (top/current/bottom) based decision with left/right (top/bottom) based decision. It is reported that the BD-rate change is 0.01 %, 0.01 % and 0.00 % in texture, video, and synthesis respectively.

- Notes from BoG discussion

* 20% reduction of addition, comparison in the filtering operation
* Options2 combines filter simplification from I0088 and I0094 (filtering condition simplification) with one-dimensional filtering simplification
* BoG recommends adoption of option2 of I0109

[JCT3V-I0088](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2171) Simplification of filtering in DBBP and its extension to PU boundary [J. Y. Lee, M. Mishurovskiy, M. W. Park, Y. Cho, C. Kim (Samsung)]

- Abstract

In DBBP, an arbitrarily shaped block partitioning for a collocated texture block is derived from a binary segmentation mask computed by its corresponding depth block. Two partitions are motion-compensated and then merged based on the mask. Finally, both luminance and chrominance components of boundary pixels in the merged block is filtered. In order to simplify the filtering process, this contribution proposes to remove the filtering in the chrominance. Also, the contribution proposes to extend the filtering into PU boundaries in general partitioned modes. The result demonstrates that the proposed method obtains the gain of about -0.3% in dependent views and about -0.1% in overall performance.

- Notes from BoG discussion

* proposes to apply the same one-dimensional filter as in I0109 (used for DBBP) for all PU-boundaries and achieves -0.1%,-0.1%,-0.1% BD-savings for video, total, and synthesis measure cases
* not a DBBP related proposal
* Several experts commented it is not a good idea to use this filter for extended purposes

[JCT3V-I0124](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2207) On DBBP Filtering Simplification [Z. Gu (SCU), J. Zheng (HiSilicon), N. Ling (SCU), P. Zhang (HiSilicon)]

- Abstract

This contribution proposes a simplified filtering process for Depth Based Block Partitioning (DBBP). It reduces the mask checking position from 4 positions to 5 positions and also reduces filtering cases from 4 cases to 2 cases. Experiment result show that this proposal simplified DBBP boundary filtering process without BD-bitrate loss under CTC.

- Notes from BoG discussion

* DBBP uses condition check to determine one out of four two-dimensional filter masks
* proposes to use only one condition check instead of four in the anchor and reduce the number of candidate masks down to two (from four). It also remove the mask checking at the center position as in other three proposals (i.e. I0088, I0094, I0109)
* no coding impact
* Experts seem to prefer the I0109 solution over this.

### Crosschecks for Contributions in *Filter Simplification*

[JCT3V-I0137](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2220) Crosscheck for JCT3V-I0088, Samsung's Simplification of filtering in DBBP and its extension to PU boundary [X. Zhang (MediaTek)] [late]

[JCT3V-I0144](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2227) Cross check of One dimensional DBBP boundary filtering (JCT3V-I0109) [M. W. Park, C. Kim (Samsung)] [late]

[JCT3V-I0189](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2282) Cross check for option2 of one dimensional DBBP boundary filtering (JCT3V-I0109) [H. Huang (MediaTek)] [late]

[JCT3V-I0165](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2248) Crosscheck for JCT3V-I0124, On DBBP Filtering Simplification [X. Zhang, K. Zhang (MediaTek)] [late]

## Memory Requirement

[JCT3V-I0078](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2161) Memory Complexity for DBBP and VSP [M. W. Park, J. Y. Lee, Y. Cho, C. Kim (Samsung)]

(include summary)

Analysis shows that the worst case memory bandwidth of DBBP increases the current HEVC worst case to approx. 128%.

The contribution also includes a proposal how to resolve that – to be discussed in DBBP BoG.

- Abstract

In this contribution, the memory complexity analysis for DBBP and VSP is presented. In the current 3D-HEVC, DBBP uses DoNBDV to derive a binary segmentation map. However, using DoNBDV can cause increasing an external memory bandwidth requirement since it needs to fetch two depth blocks in addition to the motion compensation. Therefore, this contribution proposes two methods to reduce the worst case memory bandwidth of DBBP. The first method is to use NBDV instead of DoNBDV. With this method, the worst case memory bandwidth is reportedly reduced from 120% to 110% with a coding loss of 0.06%. The second method is to allow DBBP only when the current CU size is greater than 8. With this method, the worst case memory bandwidth is reportedly reduced from 120% to 79% without any coding loss (0.00%). Additionally, VSP allows only one depth block fetch to minimize the memory complexity. However, the worst case memory bandwidth for VSP is only 50%. So, it is proposed to allow two depth block fetches for VSP to improve the coding efficiency. With this method, the worst case memory bandwidth is reportedly increased from 50% to 59%, but a coding gain of 0.05% can be achieved.

- Notes from BoG discussion

* proposes two methods
* Method1

=> Use NBDV instead of DoNBDV

=> 128% to 114% with no coding impact

* Method2

=> disallow DBBP for 8x8 CU

=> from 128% to 69% with no coding impact

* BoG recommends to adopt Method 2, which is conceptually orthogonal to the issue of multiple depth-fetch which seems to merit further study

[JCT3V-I0108](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2191) Disparity vector unification for DBBP and VSP [T. Ikai, T. Tsukuba (Sharp)]

- Abstract

This contribution proposes a unification of DMM and VSP for disparity vector (DV). As JCT3V-H0069 reported, DBBP uses DoNBDV for depth access, which requires depth access twice as opposed to other related component such as VSP, which uses NBDV for depth access. In this proposal, we propose to use DoNBDV in DBBP and VSP only when the PU size is larger than 8x8, otherwise we use NBDV. It is asserted that the proposal reduces memory bandwidth complexity in DBBP with unified depth access between VSP and DBBP. It was reported that the gain of proposed method is 0.03 %, 0.03 % and 0.02 % in texture, video and synthesis respectively.

- Notes from BoG discussion

* proposes, for DBBP and VSP, to use DoNBDV for PUs of sizes 16x8, 8x16, 16x16 and NBDV for other sizes
* implies two fetches (possible leading to more cycles) even though the bandwidth requirement is the same as that in Method1 I0078
* A concern was expressed increased number of depth fetch is undesirable, which was discussed and agreed in previous meetings

### Crosschecks for Contributions in *Memory Requirement*

[JCT3V-I0160](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2243) Cross-check of Memory Complexity for DBBP and VSP (JCT3V-I0078) [T. Ikai (Sharp)] [late]

[JCT3V-I0146](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2229) Cross check of Disparity vector unification of DBBP and VSP (JCT3V-I0108) [M. W. Park, C. Kim (Samsung)]

## Motion/Disparity Vector Derivation

[JCT3V-I0054](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2136) Improved DBBP in 3D-HEVC [Y. Chen, Q. Yu, H. Liu (Qualcomm), S. Ma (PKU)]

- Abstract

In the current 3D-HEVC, similar with HEVC inter prediction modes, DBBP (Depth-Based Block Partition) requires computationally intensive motion estimation to find the best motion vector in AMVP (Advanced Motion Vector Prediction) mode, which increases the computational complexity of encoder. This contribution proposes to bypass the motion estimation process and derive disparity vectors for DBBP coded PU directly in AVMP mode. It is reported that proposed simplification can reduce the encoding time by around 10% and bring 0.2% coding gain on dependent views.

- Notes from BoG discussion

* proposes to use depth information (from the baseview) to derive disparity vectors for AMVP mode
* worth noting I0058 also proposes a method to derive motion/disparity information for DBBP via the merge-list construction process

=> In terms of coding gain improvement, I0058 seems more effective. Further study encouraged.

[JCT3V-I0058](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2141) Segment-wise Prediction (SP) Merge Candidate [F. Jäger (RWTH Aachen University), J. Konieczny, G. Cordara (Huawei Technologies)]

- Abstract

At the 6th JCT-3V meeting, a motion/disparity prediction method was introduced in JCT3V-G0106 [1] that uses a depth-derived binary segmentation mask for the derivation of PU partitioning and for merging of two prediction signals. The method is called Depth-based Block Partitioning (DBBP). The segmentation process in JCT3V-G0106 results in two segments, which motion information is coded as a conventional bi-partitioned coding unit with two sets of motion information. This contribution proposed to use the same prediction method as a new Segment-wise Prediction (SP) merge candidate. The corresponding two sets motion information are derived from the already checked merge candidate positions. The proposed method yields 0.12% coding efficiency gain under common test conditions for video PSNR over total bitrate.

- Notes from BoG discussion

* Proposes to introduce an additional merge-candidate which indicates the use of derived motion vectors from spatial and temporal neighbors for DBBP (for 2Nx2N PUs) in addition to the current anchor DBBP MC-prediction process
* shows BD-saving of -0.1%,-0.1%,-0.1%
* BoG encourages further study

### Crosschecks for Contributions in Motion/Disparity Vector Derivation

[JCT3V-I0061](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2144) Cross Check of Improved DBBP in 3D-HEVC (JCT3V-I0054) [F. Jäger (RWTH Aachen University)]

[JCT3V-I0170](http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=2253) Crosscheck of segment-wise Prediction (SP) Merge Candidate (JCT3V-I0058) [H. Liu (Qualcomm)] [late]