|  |  |
| --- | --- |
| **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-I0056 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Cleanup of Depth-based Block Partitioning (DBBP)** | | |
| *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

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.

# Depth-based Block Partitioning Algorithm

The proposed algorithm for using depth information for block partitioning in texture views consists of three sequential steps, which will be described in more detail in the following section.

## Depth Segmentation

In an initial step the collocated depth block of the current coded tree block (CTB) of the texture component is segmented into two arbitrarily shaped segments. As the depth component is coded after the corresponding texture view in the current CTC, a virtual depth map is derived from the base view’s reconstructed depth and shifted by a disparity vector, which is itself derived from the neighboring blocks (by means of DoNBDV).

The segmentation of the (virtual) depth map is performed based on a very simple thresholding mechanism where the threshold is computed from the mean depth value.

Here, defines the width/height of the current texture block and resembles the already coded, corresponding depth map of the reference view’s texture frame.

Afterwards, a binary segmentation mask is generated based on as follows.

The resulting binary segmentation mask defines the shape of the partitioning of the texture block. While motion or disparity compensation in a modern video coder (e.g. in HEVC) is performed on a block-basis, an arbitrarily shaped block partitioning typically requires pixel-based compensation. This concept is applied in pixel-based view-synthesis prediction (VSP), which warps each pixel position based on the corresponding depth value to the position of the particular reference frame. By this fine-grained compensation approach higher order deformations can be approximated. To achieve this amount of precision for the prediction stage of a video coder, pixel-based VSP introduces very high complexity compared to conventional block-based motion/disparity compensation. This is mainly due to the irregular access to the reference buffer and the pixel-wise conversion between depth and disparity.

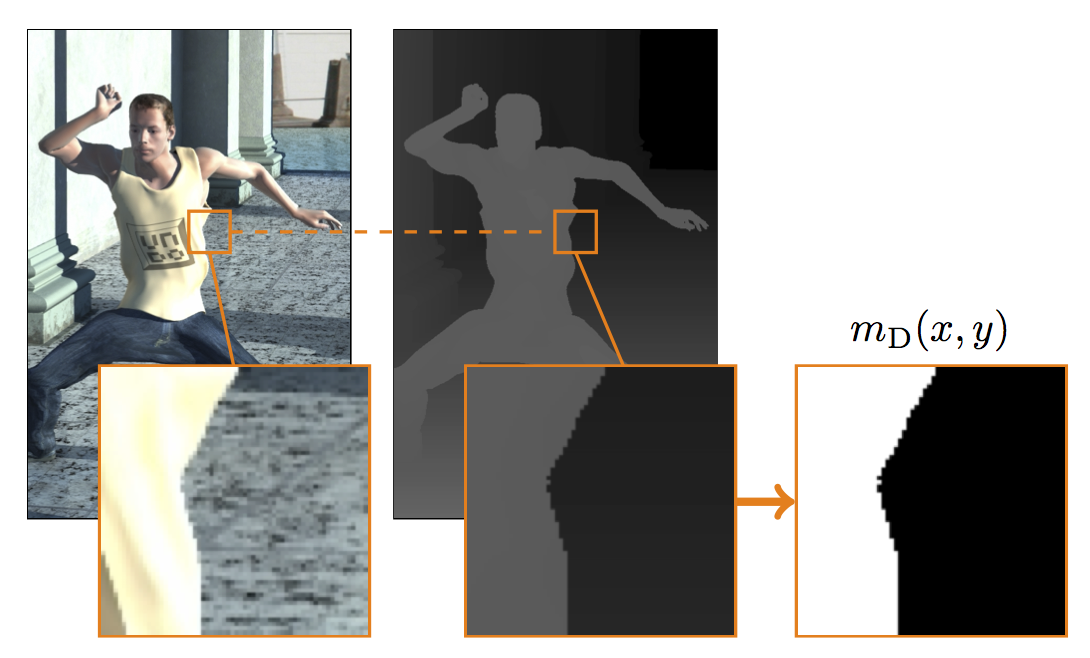


Figure 1: Cropped frame from *Undo\_Dancer* test sequence with magnified component blocks of an example coding unit. The collocated depth block segments the block into foreground and background, which is defined by a binary mask .

To overcome this issue of VSP, the proposed depth-based block-partitioning (DBBP) scheme still uses block-based compensation in the prediction stage, as it is described in the following subsection.

## Block-based Compensation

In the proposed DBBP scheme, the actual motion or disparity compensation is performed on a partitioning, which means that the full CTB is shifted by the coded vector information. This full-size motion/disparity compensation is performed twice, once for each segment, and results in two prediction signals and .

Consequently, two sets of vector information need to be coded for a DBBP block. The assumption behind this approach is that a texture block is typically segmented into foreground and background based on the collocated depth block. These two depth layers can then be compensated independently by their own sets of motion or disparity vectors.

## Merging of Prediction Signals

After having generated two full-size prediction signals and for a DBBP-coded block, the segmentation mask is used to merge these into the final prediction signal for the current texture CTB.

By merging the two prediction signals, shape information from the depth map allows to independently compensate foreground and background objects in the same texture block. At the same time, DBBP does not require pixel-wise motion/disparity compensation. Memory access to the reference buffers is always regular (block-based) for DBBP-coded blocks in contrast to approaches like VSP. Moreover, DBBP always uses full-size blocks for compensation. This is preferable with respect to complexity, because of the higher probability of finding the data in the memory cache.

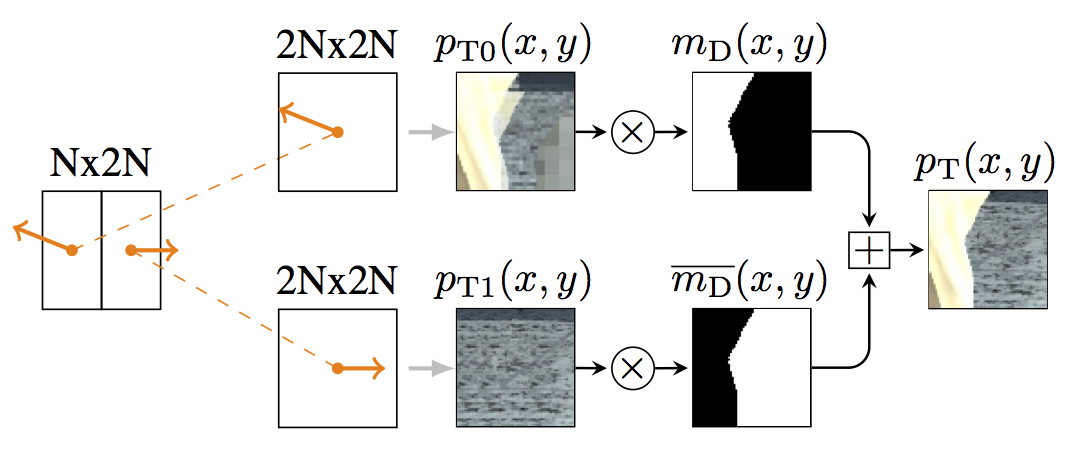


Figure 2: DBBP merging process: For each of the two decoded motion parameters a motion compensation is performed. The resulting prediction signals and are combined using the DBBP mask .

# Proposed Clean-ups

## Removal of Partitioning Derivation Process

JCT3V-H0094 proposed to signal the DBBP flag unconditionally for all partition sizes instead of coding it only for 2NxN, as the original proposal JCT3V-G0106 proposed. In JCT3V-G0106 the true partitioning was derived at the decoder after having decoded the DBBP flag. When sending the DBBP flag unconditionally, the derivation of the partitioning is not required anymore as the true partitioning can be signaled explicitly in the bitstream.

## Signaling of DBBP Flag for non-square Partitioning

Currently, the DBBP flag is signaled unconditionally, but the DBBP mechanism cannot be enabled for square partitioning. Therefore, it is proposed to only signal the DBBP flag for all partition sizes but 2Nx2N and NxN.

## Renaming of Depth-based Block Partitioning (DBBP) to Segment-wise Prediction (SP)

As the proposed removal of the partitioning derivation process reduces the original DBBP mechanism to its segment-wise motion prediction aspect, the initial name is not describing the mechanism anymore. Therefore, it is proposed to rename Depth-based Block Partitioning (DBBP) to Segment-wise Prediction (SP).

# 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 | 0,00% | 0,02% | -0,11% | -0,03% | -0,03% | -0,04% | 106,1% | 98,1% |
| Kendo | 0,00% | -0,08% | -0,11% | -0,05% | -0,09% | -0,02% | 103,4% | 96,1% |
| Newspaper\_CC | 0,00% | -0,05% | -0,12% | -0,05% | -0,04% | -0,04% | 102,9% | 100,1% |
| GT\_Fly | 0,00% | -0,03% | -0,07% | -0,01% | -0,02% | -0,01% | 103,1% | 99,4% |
| Poznan\_Hall2 | 0,00% | -0,19% | -0,36% | -0,08% | -0,06% | -0,04% | 102,9% | 98,5% |
| Poznan\_Street | 0,00% | 0,15% | -0,07% | 0,01% | 0,01% | 0,01% | 99,1% | 98,7% |
| Undo\_Dancer | 0,00% | -0,16% | -0,12% | -0,03% | -0,02% | 0,03% | 98,4% | 99,2% |
| Shark | 0,00% | -0,17% | -0,13% | -0,03% | -0,02% | -0,01% | 101,4% | 97,6% |
| 1024x768 | 0,00% | -0,03% | -0,12% | -0,04% | -0,05% | -0,03% | 104,1% | 98,1% |
| 1920x1088 | 0,00% | -0,08% | -0,15% | -0,03% | -0,02% | 0,00% | 101,0% | 98,7% |
| **average** | **0,00%** | **-0,06%** | **-0,14%** | **-0,03%** | **-0,03%** | **-0,02%** | **102,2%** | **98,5%** |

# Cross Check

The cross check was performed by HHI. They investigated the proposed changes and ran the same simulation configuration to confirm the presented simulation results.

# Conclusion

In this contribution to clean-ups of the Depth-based Block Partitioning algorithm are proposed. Firstly, the defined partitioning derivation process is not required anymore and can be removed. Secondly, the DBBP flag shall only be signaled for non-square partition sizes as the algorithm cannot be applied for square partitions.

# 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, J. Konieczny, and G. Cordara, “CE3: Results on Depth-based Block Partitioning (DBBP),” 7th Meeting, San Jose, USA, 2014.

[2] J. L. Lin, Y. W. Chen, T. D. Chuang, X. Zhang, Y. W. Huang, and S. Lei, “Improvement on the signaling of DBBP,” Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG, Valencia, JCT3V-H0094, Apr. 2014.