|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extension Development**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  2nd Meeting: Shanghai, CN, 13–19 Oct. 2012 | Document: JCT3V-B0189 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On view synthesis prediction for 3DV (Information)** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Information | | |
| *Author(s) or Contact(s):* | Ying Chen  5775 Morehouse Dr San Diego, CA 92121 USA | Tel: Email: | 1-858-845-6589 [cheny@qti.qualcomm.com](mailto:cheny@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This document provides information on view synthesis prediction that has been included and developed as an in-loop coding tool for 3D video coding, especially for 3D-AVC. View synthesis prediction in 3D-AVC originally adopted the concept of Depth-Image-Based Rendering (DIBR) which has been investigated in MPEG 3DV Ad hoc Group (AhG) for 3DV exploratory experiments, which were mainly for the purpose of the MPEG 3DV Call for Proposals (CfP). While the view synthesis solution was used to generate the virtual views, the technology has been integrated into the first version of the ATM software to improve the coding efficiency of non-base views. However, the view synthesis was too expensive for both implementation and decoding complexity. Therefore, there have been continuous efforts on the simplification of the view synthesis prediction. The so-called view synthesis prediction has been gradually made friendlier to video codecs and can be further improved to be aligned with the traditional motion compensation engine in 3D video codecs by utilizing depth information for better prediction of motion vectors.

# Introduction

This section mainly focuses on a rough review of the history of the development of in-loop view synthesis prediction in 3D video coding standards, focusing on 3D-AVC.

## DIBR

Depth-Image-Based Rendering (DIBR) [1] is widely adopted for the generation of a virtual view. DIBR warps a picture from one view to the other view in a different location/perspective. The view that is already available before warping is called reference view while the virtual view to be synthesized is called target view in this document. A simple example of warping one pixel to the location of the other view is shown in Fig. 1.

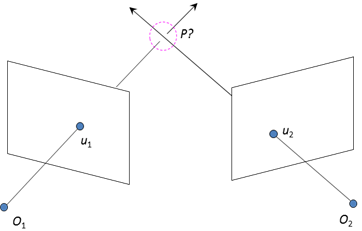


Fig. 1. Pixel re-projection for 3D warping.

Based on whether the depth map is associated with the reference view or target view, the view synthesis can be categorized to forward warping and backward warping.

### Forward warping

In forward warping, each pixel location e.g., denoted as *u1* in the reference view is mapped to a pixel location of *u2* based on the depth value of the pixel location *u1* as well as the camera parameters. After the location *u2* is determined, the pixel values (including up to three color components) in pixel location *u1* of the reference view may be copied (or with possible interpolation) to the pixel location *u2* of the reference view.

Since multiple pixels may be warped to the same location, z-buffering is needed to decide which reference pixel is finally chosen to produce the pixel values. Consequently, there are also pixel locations (in the target view) which are not mapped. Those pixel locations are called holes.

To fill the holes, a hole-filling process is required to go through the picture and fill the holes with pixel values from the neighbors.

### Backward warping

In backward warping, a pixel position *u2* in the target view is mapped back to a pixel location *u1* in the reference view based on the depth value associated with *u2*, and the pixel values of *u1* are copied for the pixel location *u2*.

As long as the picture of the reference view is properly padded, *u1* is always available for any given *u2*. Therefore, z-buffering and hole-filling are not necessarily needed in backward warping.

Note that backward warping assumes that the depth map of the target view is present before view synthesis.

## View synthesis prediction solution in MPEG 3DV

In MPEG 3DV AHG, view synthesis software was originally developed by Tanimoto et. al.. The algorithm is described in [2]. Due to the fact that the 3D video scenarios are mainly for horizontally parallel cameras, Tian et. al. proposed 1D based view synthesis [3], which was later widely used in MPEG 3DV related activities. Various improvements for view synthesis were also proposed in [4], [5] and [6].

## In-loop view synthesis prediction solution in 3D-AVC

In the first version of ATM, 1D based view synthesis in [3] was used for in-loop view synthesis prediction, wherein a synthesized picture is used as an additional reference picture. The important technical aspects of view synthesis prediction are summarized as follows.

* Forward warping: hole-filling is needed.
* 4:4:4 based: the mapping of each pixel position contains a warping of sample value of the pixel position for each of the three color components. The chroma components of the YUV 4:2:0 image are upsampled to form a YUV 4:4:4 image before warping and the synthesized YUV 4:4:4 image is downsampled to YUV 4:2:0.
* ¼-pel accuracy: the texture image and depth image of the reference view are upsampled 4 times horizontally, before view synthesis is performed. After warping, the synthesized high resolution image is downsampled.

As implied above, the original view synthesis software requires significant more computations and implementation of the view synthesis is challenging, especially for hardware solutions. Unlike the major modules in the traditional coding tools, we have relatively less experience on optimized view synthesis design.

To make the original view synthesis prediction less complex and more friendly to implementations, continuous contributions have been adopted into the 3D-AVC. Surprisingly, **all the efforts eventually lead to the (potential) removal of view synthesis prediction.**

### One-loop warping and hole-filing

Typically, the forward warping requires a loop of the pixels in the reference picture for warping, followed by a loop of hole-filling. In [7], one-loop warping and hole-filling was proposed to reduce the complexity as well as the cache miss during the view synthesis process. The reported loss of the method is around 0.3%, with significant decoding time reduction. Note that throughout the whole document, the bitrate gains/losses are calculated based on average PSNR values of the synthesized views and the total bitratems.

### 4:2:0 based warping

To avoid the downsampling/upsampling of the image between YUV 4:2:0 to YUV 4:4:4, it was proposed that the warping can be done based on the YUV 4:2:0 images.

More specifically, in common test condition, depth image has a quarter resolution of the texture, so that practically, one depth pixel is related to mapping a 2x2 block of a luma image [8]. The loss of this method is less than 0.1%.

### Simplified fractional-pel based warping

To keep the synthesized image more accurate while avoiding huge storage of upsampled reference view and target view images as well as the complexity of interpolation, it was proposed that the fractional-pel pixel values were generated in a way that each pixel of the reference image was phase-shifted thus the pixel values are changed prior to warping. The method therefore generates a temporary image which has the same resolution as the reference image. It actually avoids the real interpolation, by skipping generating/interpolating the pixel values for the unnecessary phases [9]. However, this method was only fully adopted into 3D-AVC after a follow-up proposal is made in [10]. The loss of this method is almost negligible (0.01%).

### Block based (2x2) view synthesis prediction

In the last meeting, block based view synthesis prediction [11] was adopted also into the ATM software.

In this method, depth values are used to derive a disparity vector for a block and backward warping is applied for view synthesis.

This method is quite different from the original approach in ATM since it is based on backward warping. However, it is similar to [8] in a way that each warping handles a 2x2 luma block. Since it is backward warping based solution, no hole-filing is needed and the whole view synthesis prediction is done real-time per 2x2 block thus a picture doesn’t need to be really synthesized. The prediction of a 2x2 block is similar to motion compensation for a 4x4 block.

### AVC motion compensation aligned (4x4) view synthesis

Similar to the 2x2 block based view synthesis (B-VSP), a 4x4 block based solution was proposed in [12], this method is purely aligned with the AVC motion compensation engine. The additional part of this method is only (disparity) motion vector prediction.

**Table 1. Coding performance of the B-VSP methods**

|  |  |  |
| --- | --- | --- |
| ***View synthesis simplifications*** | Coding gain VS ATM 4.0 | Compared loss VS [10] |
| Simplified fractional-pel based warping [10] | -0.52% | 0% |
| B-VSP (2x2) [11] | -0.35% | 0.17% |
| B-VSP (4x4) [12] | -0.21% | 0.31% |

**Table 2. Coding loss of the view synthesis simplifications**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1LWH[7] VS ATM 1.0 | 4:2:0 based warping [8] VS [7] | Simplified fractional-pel based warping [10] VS [8] | 4x4 block based[12] VS [10] | ***All vs ATM 1.0*** |
| 0.31% | 0.09% | 0.01% | 0.31% | 0.71% |

Compared to the already simplified pixel-based view synthesis approach, the 4x4 block based view synthesis prediction only introduces a coding loss of 0.3%. The 4x4 block based view synthesis prediction, when compoared bith the 2x4 block based view synthesis, introduces only 0.14% loss.

It can be also roughly estimated that even when the 4x4 block based view synthesis prediction, as proposed in [12] is used, the coding loss compared to the very first version of in-loop view synthesis prediction in ATM 1.0, is roughly smaller than 0.8%, as shown in Table 2.

### Major drawback of in-loop view synthesis prediction

No compatibility to stereoscopic/multiview video: typically in-loop view synthesis prediction requires depth map to be present, unless the depth values for pixels or blocks are estimated from texture. Therefore, it is required that the depth view components have to be present even for the decoding of texture only stereoscopic or multiview representation. The current 3D-AVC is not compatible to stereoscopic/multiview video mainly due to the use of in-loop view synthesis prediction, while the current 3D-HEVC is compatible to stereoscopic/multiview video. In 3D-HEVC, a sub-bitstream containing only texture view components can be extracted and it is decodable by a 3DV/multiview codec.

# Conclusions

In this proposal, the development of in-loop view synthesis prediction in 3D-AVC is briefly reviewed. It was observed that the in-loop view synthesis prediction was greatly simplified with a small loss (less than 0.8%), during the development of 3D-AVC. On the other hand, the current in-loop view synthesis prediction can be regarded as nothing but a method of motion vector prediction.

In 3DV, motion prediction tools are the most efficient ones. For example, inter-view motion prediction brings huge gains for 3D-AVC and also 3D-HEVC. Proposals with very small changes of merge mode in 3D-HEVC or small changes on how skip mode in 3D-AVC easily bring 0.5~0.7% gain. There were also proposals which could be more easily integrated into 3D-AVC or 3D-HEVC and give decent coding gain.

View synthesis prediction in a 3D video codec (e.g. 3D-HEVC) may be just a competitor of motion prediction tools and it is not necessary to include the whole framework of view synthesis into the 3DV codec.

In-loop view synthesis also makes the stereoscopic/multiview compatibility becomes almost impossible.

In hybrid (multi-standard) codecs, especially those that are ignorant to the base layer thus don’t need to define the view synthesis predication as part of the decoder specification, view synthesis prediction is more preferable.

# References

1. C. Fehn, "Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV," Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI, pp. 93–104, San Jose, CA, USA, Jan. 2004.
2. Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, “View generation with 3-D warpiing using depth information for FTV,” Elsevier Signal Processing: Image Communication, vol. 24, pp. 65–72, 2009.
3. D. Tian, P. Pandit, C. Gomila "Simple View Synthesis," MPEG document m15696, ISO/IEC JTC1/SC29/WG11, July 2008.
4. Y. Zhao, L. Yu, D. Tian, "Improved 1D mode in VSRS 3.1," MPEG document m16582, ISO/IEC JTC1/SC29/WG11,Jun, 2009
5. Y. Zhao, D. Fu, L. Yu, and L. Xiong, "Temporal Improvement Method in View Synthesis," MPEG document m16041, ISO/IEC JTC1/SC29/WG11,Feb 2009
6. C. Lee, Y-S. Ho, "Implementation of Boundary Noise Removal for View Synthesis," MPEG document m16064, ISO/IEC JTC1/SC29/WG11, Feb. 2009
7. L. Zhang, Y. Chen, et. al., "Complexity Reduction for View Synthesis Prediction in 3D-AVC," MPEG document m23812, ISO/IEC JTC1/SC29/WG11, Feb. 2012.
8. L. Zhang, Y. Chen, et. al., "3D-CE1.a related: View synthesis based on asymmetric texture and depth resolutions," MPEG document m24914, ISO/IEC JTC1/SC29/WG11, April-May. 2012.
9. S. Shimizu, and S. Kimata, "3D-CE1.a related: Simplification of in-loop view synthesis with fractional-pel disparity," MPEG document m24796, ISO/IEC JTC1/SC29/WG11, April-May. 2012.
10. S. Shimizu, S. Sugimoto and S. Kimata, "CE1.a related results on the modification of in-loop view synthesis," JCT3V-A0023, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Stockholm, SE, 16–20 July 2012.
11. S. Su, D. Rusanovskyy , and M. M. Hannuksela, "3DV-CE1.a: Block-based View Synthesis Prediction for 3DV-ATM," JCT3V-A0107, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Stockholm, SE, 16–20 July 2012.
12. C.-L. Wu, Y.-L. Chang, Y.-P. Tsai, and S. Lei,"3D-CE1.a related: Inter-view Skip mode with sub-partition scheme," JCT3V-A0050, Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Stockholm, SE, 16–20 July 2012.