|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  24th Meeting: Geneva, CH, 26 May – 1 June 2016 | Document: JCTVC-X0077 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Tile based VR video encoding and decoding schemes** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Information | | |
| *Author(s) or Contact(s):* | Ye-Kui Wang  Hendry Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121, USA | Tel: Email: | +1-908-803-3888 [yekuiw@qti.qualcomm.com](mailto:yekuiw@qti.qualcomm.com) [hendry@qti.qualcomm.com](mailto:hendry@qti.qualcomm.com) [martak@qti.qualcomm.com](mailto:martak@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This document contains the same content as the MPEG input document m38559. It is submitted to the JCT-VC for information as the topic might be of interest to JCT-VC participants.

This document presents a few virtual reality (VR) video encoding and decoding schemes wherein for the highest-resolution representation of the video, only the part that corresponds to the current field of view (FOV) reportedly needs to be transmitted and decoded. The schemes are based on coding of tiles in HEVC and SHVC. Furthermore, signalling of such partial VR video decoding schemes is discussed.

# Introduction

A distinct feature of VR video compared to normal video is that, in VR typically only a subset of the entire video region represented by the video pictures, corresponding to the current field of view (FOV), i.e., the area currently being seen by the user, is displayed, while in normal video applications typically the entire video region is displayed. FOV is sometimes also referred to as viewport. This feature may be utilized to improve the performance of VR video systems, e.g., by using viewport dependent projection mapping or viewport dependent video coding. The performance improvement can be either or both of lower transmission bandwidth and lower decoding complexity compared to conventional VR video systems under the same resolution/quality of the video part presented to the user.

Viewport dependent video coding may also be referred to as viewport dependent partial video decoding, as the key is to enable decoding only partially the entire encoded video region to provide sufficient information for display of the current FOV or viewport.

In the reminder of this document, in Section 2, we firstly present the conventional VR video encoding and decoding method, followed by a few viewport dependent partial VR video encoding and decoding schemes, all based on motion-constrained tiles. Some comparisons of the presented methods are also provided in Section 2. In Section 3, the signalling of the tile based viewport dependent partial VR video encoding and decoding schemes is discussed.

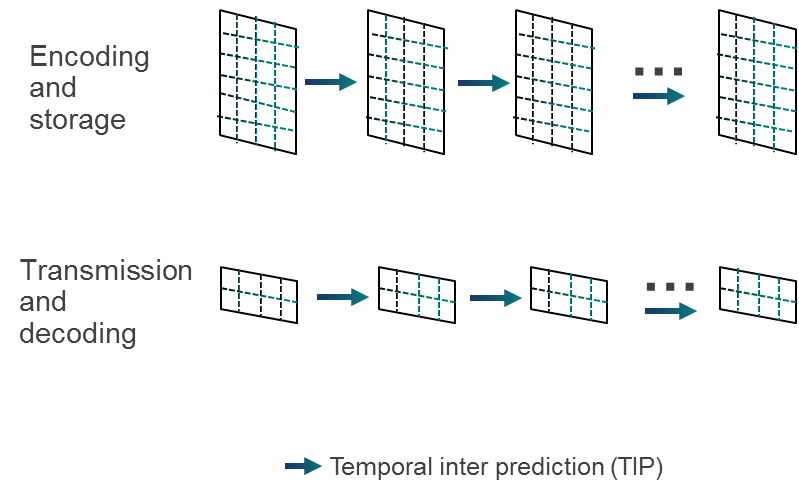
# VR video encoding and decoding schemes

## Conventional VR video encoding and decoding

Conventionally, a VR video is represented with an equi-rectangular or cube-map protection mapping. The video pictures are encoded as a single-layer bitstream using temporal inter prediction (TIP), the entire coded bitstream is stored at a server, if needed, transmitted to the receiver side, fully decoded by the decoder, and the region of the decoded picture corresponding to the current viewport is rendered to the user.

## Simple tile based VR video partial decoding

The VR video pictures can be coded using motion-constrained tiles such that each potential region covering a viewport can be independently decoded from other regions across time. For a particular current viewport, the minimum set of tiles that cover the viewport is sent to the client, decoded, and rendered. This method is referred to as Simple Tile based Partial Decoding (STPD), and is depicted by Figure 1.



**Figure 1 Tile based VR video partial decoding**

A problem of this approach is that when the user turns his or her head quickly to a new viewport that is not covered (entirely or partially) by the currently being sent tiles, nothing in the new area (covered by the new viewport but not the old viewport) can be seen before the tiles covering the new viewport arrive (and the data is sufficiently buffered according to the buffering timeline). Therefore, this method can only work if the network round trip time is extremely low, e.g., at a magnitude of 10 ms, which is not feasible or is at least a big challenge today or in the near future.

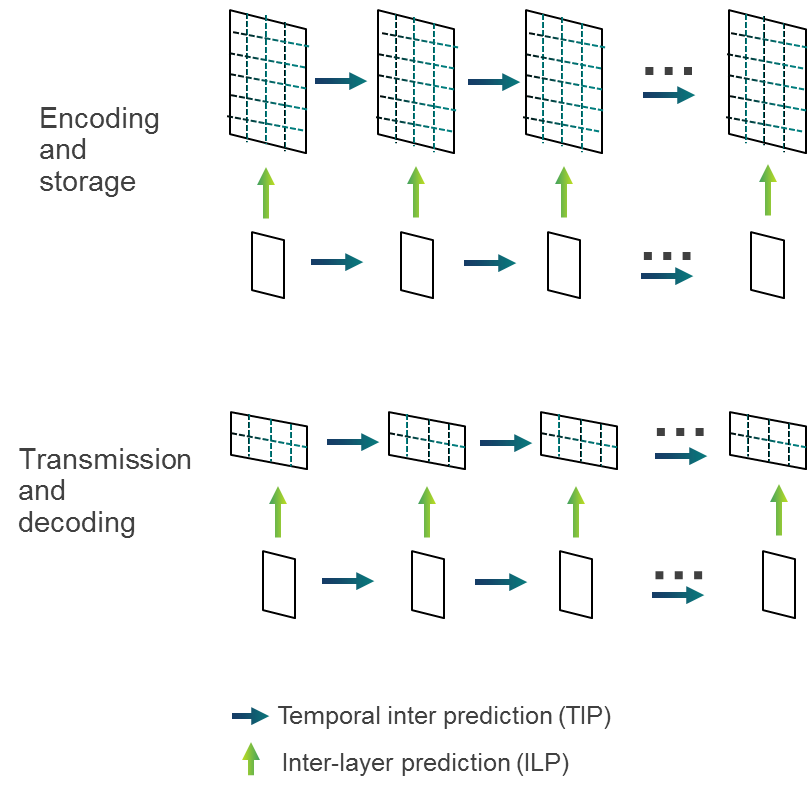
## Scalable coding based VR video partial decoding

Two ScaLable coding based Partial Decoding (SLPD) schemes, referred to as SLPD#1 and SLPD#2, are presented in this section.

In SLDP#1, as depicted by Figure 2, the VR video is scalable-coded using SHVC spatial scalability with multiple resolutions. The bitstream of the lowest resolution video, i.e., the base layer (BL) is always fully sent, such that at any time for any viewport at least the lowest resolution video is available for rendering. The lowest resolution video does not need to be coded using tiles at all, although it would also work if it is coded using tiles or motion-constrained tiles.

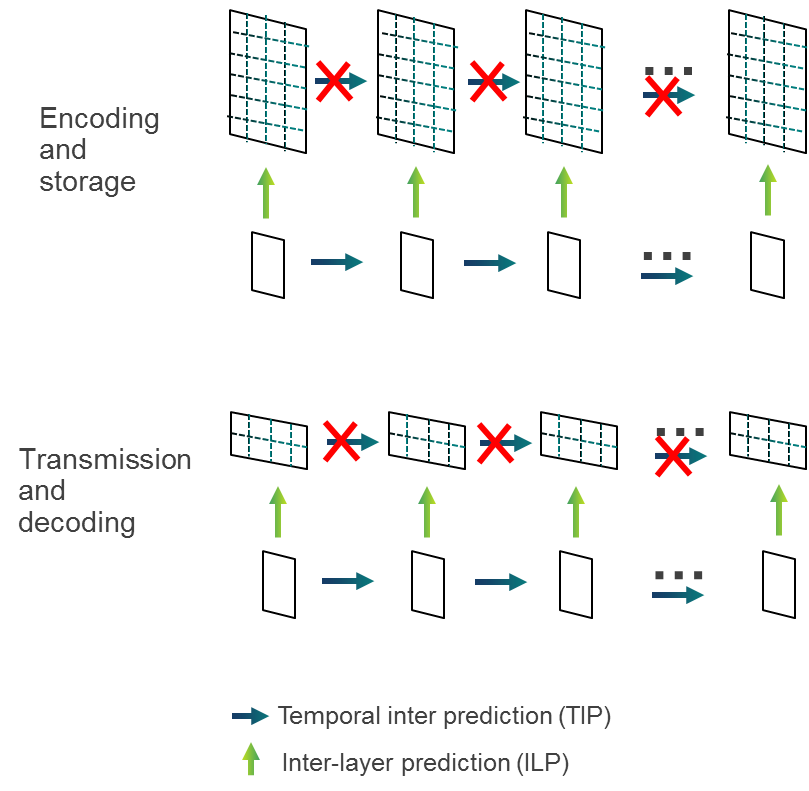
The enhancement layers (ELs) are coded using motion-constrained tiles such that each potential region covering a viewport can be independently decoded from other regions across time, with inter-layer prediction (ILP) enabled. For a particular current viewport, the minimum set of tiles that cover the viewport is sent to the client, decoded, and rendered. From the storage point of view, the full streams of all the resolutions/layers need to be stored.

When more than two layers are coded, tiles covering different viewports can be chosen from different resolutions. For the current viewport, the tiles are chosen from the highest resolution; for viewports neighboring to the current viewport, tiles are chosen from the second highest resolution; and so on.



**Figure 2 First scalable coding based VR video partial decoding**

In SLDP#2, as depicted by Figure 3, the VR video is also scalable-coded using SHVC spatial scalability with multiple resolutions. The BL is coded the same as in SLPD#1, while the ELs are coded similarly as in SLPD#1 but with temporal inter prediction (TIP) disabled.

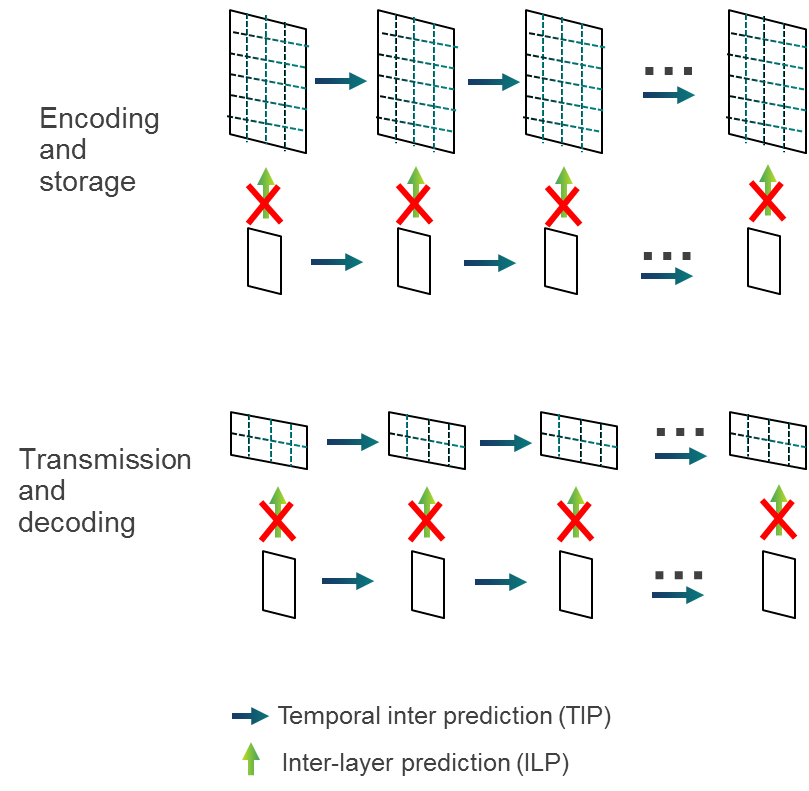


**Figure 3 Second scalable coding based VR video partial decoding**

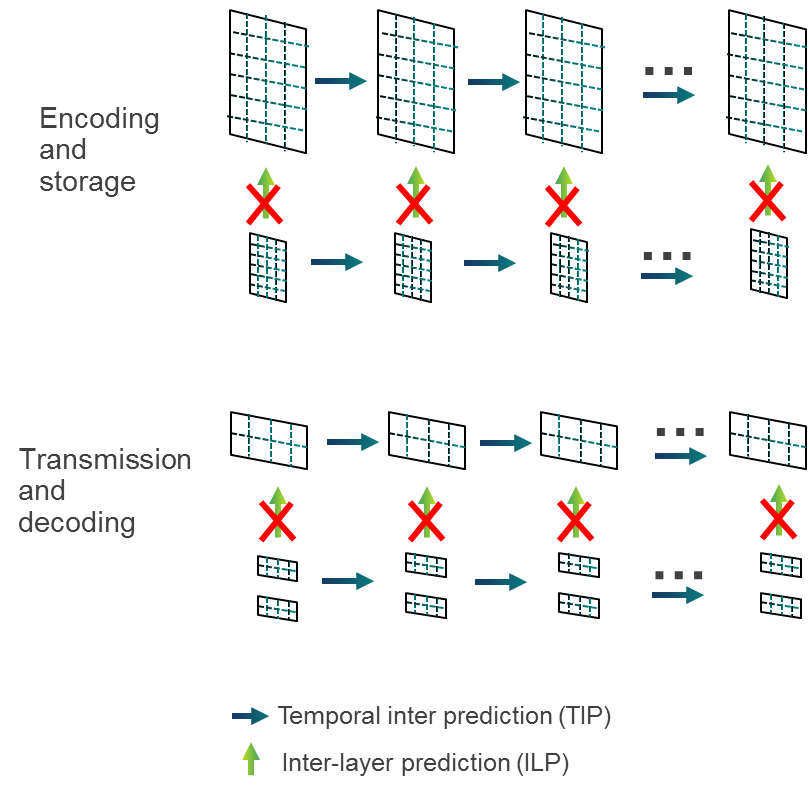
## Simulcast coding based VR video partial decoding

Two SiMulcast coding based Partial Decoding (SMPD) schemes, referred to as SMPD#1 and SMPD#2, are presented in this section.

SMPD#1 is depicted by Figure 4. This method is the same as SLPD#1 except that inter-layer prediction is not used. In other words, except the lowest resolution, all other resolutions are coded the same way as in STPD.



**Figure 4 First simulcast coding based VR video partial decoding**



**Figure 5 Second simulcast coding based VR video partial decoding**

SMPD#2 is depicted by Figure 5. There are two differences between SMPD#2 and SMPD#1:

1. In SMPD#2, the lowest resolution is also coded using motion-constrained tiles same as other resolutions.
2. When SMPD#2 is used, the lowest resolution is also not fully sent, but only the tiles that cover the viewport that is geometrically the most distant from the current viewport and any other viewports not covered by sent tiles from other resolutions.

## Comparison of different VR video coding schemes

Table 1 provides a comparison of the VR video coding schemes described above based on six different aspects.

**Table 1 Comparison of VR video coding schemes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Conventional** | **STPD** | **SLPD#1** | **SLPD#2** | **SMPD#1** | **SMPD#2** |
| TIP | Y | Y | Y | N | Y | Y |
| ILP | n\a | n\a | Y | Y | N | N |
| Need to be tiled | N | Y | N for BL 1 Y for ELs | N for BL Y for ELs | N for BL Y for ELs | Y for both BL and ELs |
| Latency 2 acceptable | Y | N | Y | Y | Y | Y |
| RAP 3 | N | Y | N for BL Y for ELs | N for both BL and ELs | N for BL Y for ELs | Y for both BL and ELs |
| Clean viewport switching 4 | n\a | n\a | Y | Y | Y | N |

1 For simplicity, the lowest resolution and other resolutions are denoted as BL and ELs, respectively, in the table, regardless of scalable or simulcast coding.

2 The latency is the latency between user turning head and (fully) seeing the new viewport.

3 The "RAP" row indicates whether frequent random access points are needed because of the latency between user turning head and (fully) seeing the new viewport (in addition to RAPs required by other purposes).

4 The "Clean viewport switching" row indicates, during viewport switching, whether the sender can immediately stop sending the tiles that cover the old viewport and that are from the highest resolution.

Here are some observations. Due to the latency problem, STPD is not an acceptable option, at least for today and the near future. SLPD#2 seems not acceptable, either, due to the bandwidth issue caused by not using TIP. SMPD#1 is obviously inferior to SLPD#1 because all aspects are the same as SLPD#1 except that ILP is not used and thus would require higher storage and bandwidth costs.

Therefore, we only provide some more comparisons among the three remaining schemes: the conventional method, SLPD#1, and SMPD#2.

Firstly, we present a decoding complexity comparison between the conventional approach and SLPD#1.

The following are assumed for the comparison:

1. Two different spatial resolutions, the lower resolution being half in each direction of the higher resolution
2. At any moment a viewport of 180ox120o (corresponding to 1/3 of the sphere surface) is rendered to the user, thus 1/3 of the EL and the entire BL are transmitted, decoded and rendered for SLPD#1. Note that typically the viewport would be smaller, about 90o~110o in each dimension, which will favor SLPD#1 more against the conventional method.

Based on these assumptions, for the same resolution of the projection-mapped video, the number of pixels to be decoded in SLPD#1 would be 1-(1/3+1/4) = 5/12 = 41.7% less than that for the conventional method. This means significantly lower decoding capability requirement and less power consumption. In other words, for the same decoding capability, the resolution (in terms of number of pixels) that can be handled by SLPD#1 would be 12/7 = 171.4% of that for the conventional approach. This for example would allow using of a video decoder with capability 4096x2160 to handle a VR video resolution of 7680x1920, which is 166.7% of the resolution 4096x2160.

Following the same assumptions above, using SHVC and HEVC, our preliminary results indicate that, compared to the conventional method, SLPD#1 can achieve about 20% to 28% bandwidth saving when the equi-rectangular projection mapping is used, and about 24% to 32% bandwidth saving when the cube-map projection mapping is used.

Secondly, a comparison between SLPD#1 and SMPD#2 is provided below:

1. When SLPD#1 is used, the region in the lowest resolution covering the current viewport needs to be transmitted and decoded. Therefore, for the region covered by the current viewport (but only for this part), SLPD#1 has a traditional scalable-coding-versus-single-layer-coding disadvantage.
2. However, SLPD#1 has higher coding efficient for the entire lowest resolution, because coding of tiles is not needed, and furthermore, random access points (RAPs) in the lowest resolution can be coded less frequently compared to higher resolutions, as RAPs are needed for the higher resolutions for optimization of the latency between user turning head and (fully) seeing the new viewport but not needed for the lowest resolution in SLPD#1.
3. Furthermore, SLPD#1 allows clean (or more efficient) viewport switching. When SLPD#1 is in use, the server or edge server can immediately stop sending the tiles from the highest resolution that cover the old viewport. In SMPD#2, because the tiles of the lowest resolution covering the current viewport is not sent, during viewport switching, the server would have to continue sending the tiles from the highest resolution that cover the old viewport, to be prepared such that the user can at least see something in case he/she turns back to the old viewport (entirely or partially).

# Signalling

The signalling of the tile based viewport dependent partial VR video encoding and decoding schemes on the file format level can use the tiled storage of HEVC and layered HEVC (L-HEVC) in clause 10 of ISO/IEC 14496-15. Preferably, the tile tracks should be used, each carrying one motion-constrained tile, to avoid the need of using lots of byte ranges, at access unit level, for requests of the tiles covering a viewport from a particular DASH representation (corresponding to one track) when the tile region sample group mapping is used.

On the DASH level, for a particular VR video associated with a particular projection mapping and using a particular video coding scheme, each track carrying one motion-constrained tile is encapsulated into one DASH representation, and all the representations of different resolutions and carrying different regions are included in one adaptation set. The mapping of the spatial regions to the representations can be signalled by an adaptation-set-level element. The element can e.g., contain an entry count, followed by a loop of values of {representation ID, region location, and region size}.