INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2020/m51638

Jan 2020, Brussels, BE

Source **Sungkyunkwan University(SKKU)**

Title **CE1-related: On Viewport Tile Selector Implementation in TMIV for Viewport-dependent Rendering**

Authors **Jong-Beom Jeong, Soonbin Lee, Eun-Seok Ryu**

1 Introduction

After 128th MPEG meeting, test model for immersive video (TMIV) version 3.0 [1] has been proposed to MPEG-Immersive (MPEG-I) standard. TMIV 3.0 selects basic views and additional views from the input source views and removes the redundancy between basic views and additional views in MIV mode. However, TMIV 3.0 has the limitation that sends all pixels of the basic views. As a user watches only a viewport of the 360 video, transmitting the entire 360 video is not mandatory.

This contribution presents a description of SKKU's implementation and the following experiment on view tile selector (VTS) in TMIV for 3DoF+/6DoF 360 video viewport-dependent rendering based on HEVC/VVC tiles. VTS gets the camera parameter and depth map as inputs, and prints the viewport area tiles for each source view. Compared to the existing TMIV, the proposed method selects tiles from the source views which contains the user's viewport area to extract the tiles from each basic view bitstream. Thus, in viewport-dependent TMIV, the server does not need to transmit all basic views but needs encoded bitstreams using motion-constrained tile set (MCTS) [2] and tile bitstream extractor which is already available on HEVC test model (HM). Figure 1 shows the system architecture of TMIV with the proposed VTS. As shown in the figure, VTS can be applied to both MIV mode and MIV view mode.

Here are advantages of the proposed VTS method.

Advantages:

- Compatible with existing MCTS-encoder and TMIV.
- Significantly improved objective and subjective quality.
- Significant bandwidth saving with selective viewport tiles streaming.

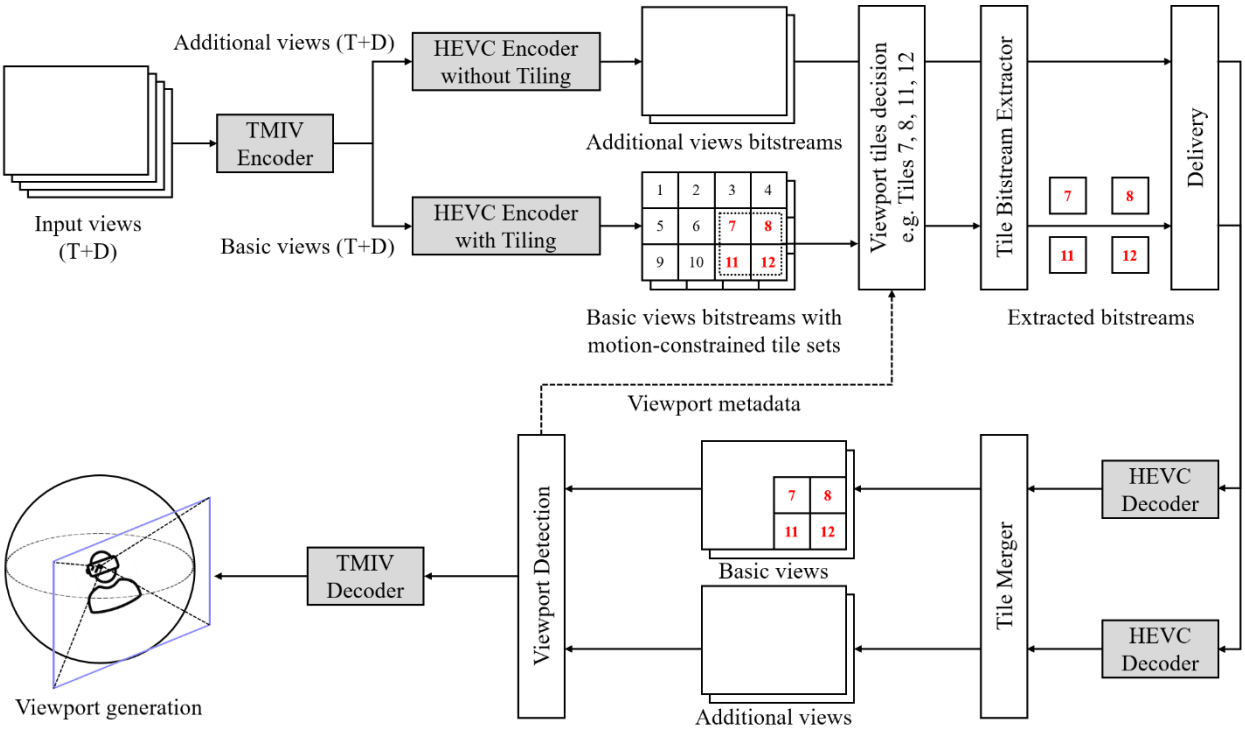

Fig. 1. System architecture of the prposed viewport-dependent VTS based on TMIV.

2 Description of the method

2.1 Viewport-dependent single 360 video tile selection

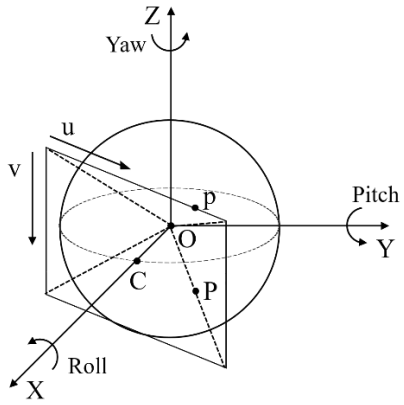

Fig. 2. Example of a viewport on a sphere

When 360 videos and the following camera parameters are given, TMIV renderer generates a viewport referencing the user's head movement, as shown in Figure 2. This section introduces a method of the viewport point detection on single 360 equirectangular projection (ERP) video based on Yu et al.[3]. In omnidirectional media format (OMAF)[4] and TMIV, the user's head rotation is represented by Euler angle. The Euler angle consists of yaw, pitch, roll, and they represents the rotation of the Z, Y, and X axes, respectively. Each angle is represented using degrees, which are converted into radians to compute the viewport area, as shown in Equation 1.

$$\begin{aligned} \alpha &= \alpha_d \ * \ \pi/180 \\ \beta &= \beta_d \ * \ \pi/180 \\ \gamma &= \gamma_d \ * \pi/180 \end{aligned} \tag{1}$$

Matrix R, as denoted in Equation 2, represents the rotation matrix that reflects the user's head movement. R consists of the product of the rotation matrices around α, β, and γ.

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Matrix K contains information about a viewport camera intrinsic matrix, as shown in Equation 3. Here, f_x and f_y are the focal lengths of the camera. Let fov_x be the horizontal FoV represented using radian, and w_v be the width of the viewport. f_x can be computed as $f_x = (w_v / 2) \cdot (1 / \tan(fov_x / 2))$. c_x and c_y are the principal point C in the viewport.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$p = [u, v, 1]^T$ is the homogeneous 2D coordinates of the viewport. P gives the viewport points on the 360 video, represented by Cartesian coordinates. This can be computed using Equation 4. In the equation, product of R, inverse matrix of K, and point p is divided by L2 norm of $K^{-1} \cdot \ p$.

$$P = R \cdot \frac{K^{-1}p}{||K^{-1}p||_2} \tag{4}$$

To obtain the coordinates of the 2D 360 ERP video, we need a point in the spherical coordinates. A point, P, in Cartesian coordinates can be converted to a point in spherical coordinates using Equation 5.

$$\begin{aligned} \phi &= atan2(P_y, P_x) \cdot 180/\pi \\ \Theta &= asin(P_z) \cdot 180/\pi \end{aligned} \tag{5}$$

The computed spherical coordinates can be converted to a point of the 2D 360 ERP video using Equation 6.

$$\begin{aligned} x &= \text{width} \cdot (0.5 + \phi/360) \\ y &= \text{height} \cdot (0.5 + \Theta/180) \end{aligned} \tag{6}$$

If the point which is computed by Equation 6, a tile which contains the point can be conducted by Equation 7, where t_i, p_w, p_h, t_w, and t_h represents tile index, picture width, picture height, tile width, and tile height, respectively.

$$t_i = (y/t_h) \cdot (p_w/t_w) + x/t_w \tag{7}$$

2.2 Warping based 3DoF+/6DoF 360 video viewport tile selection

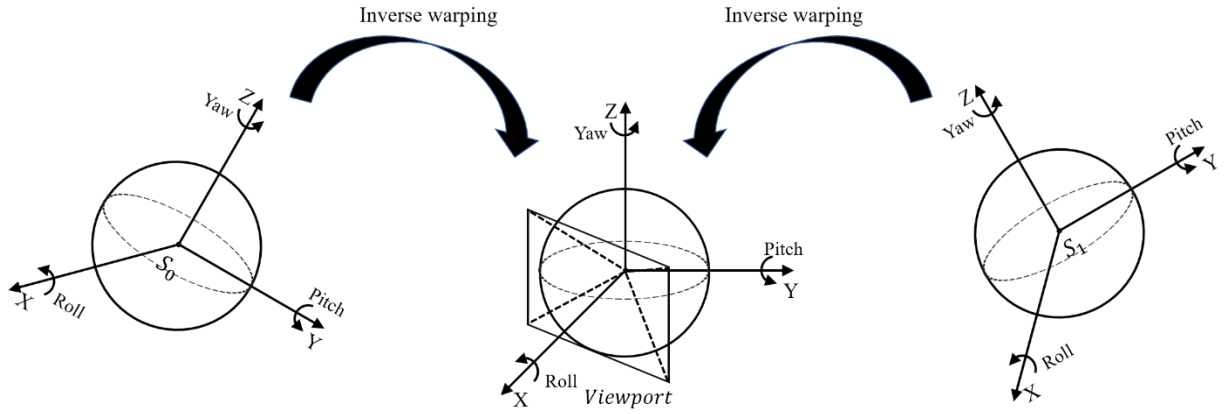

Fig. 3. Example of an inverse warping from multiview to a viewport

PW: Picture width
PH: Picture height
TW: Tile width
TH: Tile height
S: Source views
V: Viewport
p_s*: Points of a source view*
p_v*: Points of a viewport*
t_p*: Tile index of a point p*
L_t*: Lists of viewport tiles from a source view*
For each $S_i \in S$
  $R \leftarrow$ *Rotation matrix from* S_i *to V*
  $T \leftarrow$ *Translation matrix from* S_i *to V*
  For each $p_s \in S_i$
    $p'_s = R \cdot p_s + T$
    $t_{p_s} = (y_{p_s} / TH) \cdot (PW/TW) + x_{p_s} / TW$
    If t_{p_s} *is not in* L_t
      For each p_v
        If $p'_s = p_v$
          Add t_{p_s} *to* L_t

Alg. 1. Multiview 360 video viewport tile decision

In Section 2.1, viewport decision of a single 360 video method was introduced. This section describes a warping based 3DoF+/6DoF 360 video viewport tile decision method. Once a viewport at a user's position is decided, we can compute the points corresponding to the viewport of the source views with the use of warping.

Warping consists of three steps. First, the input image coordinates are unprojected into the world coordinates using the camera parameter and depth map. The depth map gives the distance between a camera and the object displayed in the image. During unprojection, the input image is projected onto the world coordinates. Secondly, an affine transformation is applied to each point in the world coordinates. As shown in Equation 8, the input image coordinates are rotated using rotation matrix R and moved by translation matrix t. In this step, the input image coordinates are transformed into the target image coordinates. Finally, the transformed world coordinates are projected onto the target image. During projection, the warped 2D input image at the target image position is generated.

$$x' = Rx + t \tag{8}$$

However, for some instances, warping from a viewport to each source view is not possible. As warping converts a 2D image into 3D coordinates, it requires the input image depth map. If the viewport is at the same position as one of the source views, the depth map can be obtained. However, in 3DoF+ and 6DoF, user can move in any directions, resulting the position of the viewport varying with respect to that of the source views, leading to non-existence of the texture and depth map for the viewport.
Alternatively, inverse warping from each source view to the viewport appears to be a solution for multiview viewport tile selection. Figure 3 shows the inverse warping method. Algorithm 1 describes the multiview viewport tile decision based on TMIV in detail. For the input, points p_v of a viewport V on 360 video, the image and tile information, and source view points p_s containing the depth map are given. p_v is computed prior to using Algorithm 1 with the help of the equations mentioned in Section 3.1. Rotation matrix R and translation matrix T from i th source view S_i to V is computed. For each point ps of S_i , a single affine transformation using the matrix R and T is conducted. The warped point p'_s is the transformed point from S_i to V. When point p'_s matches one of the points p_v , a tile of p'_s in S_i will be transmitted. To reduce the complexity, tile index t_{p_s} of point p'_s is evaluated, and if the tile index t_{p_s} is already included in the tile list L_t, the comparison of p'_s with every p_v is not conducted. However, if the tile index t_{p_s} is not in tile list L_t, point p'_s is compared with every pv . If both the points are at the same position, it indicates that point p_s belongs to the viewport area, which should be included. As a result, tile index t_{p_s} is included in L_t.

3 Experimental results

This section introduces the experimental results of the proposed 3DoF+/6DoF VTS on TMIV. The conditions of the experiment follows the common test conditions for immersive video [5]. As the proposed VTS is designed for 360 video, we selected *ClassroomVideo*, *TechnicolorMuseum*, *TechnicolorHijack*, and *NokiaChess* as test sequences. However, for the encoding / decoding, we have used HM 16.20 because of MCTS and tile extractor.
For the anchor, TMIV 3.0 was used and the atlases were encoded without MCTS. For the proposed method, basic views were encoded with MCTS, and the tile rows and columns are shown in Table 1. Further, Table 1 contains the average number of viewport tiles for each pose trace computed by the proposed VTS. CG1-A has 360° × 180° view FoV, and it was divided into 4 × 8 tiles. The other sequences, CG1-B, CG1-C, and CG1-N are divided into 4 × 4 tiles. For CG1-A, CG1-B, and CG1-N, the viewport tiles occupy 28.8%, 32.40%, and 32.28% of the full ERP video, respectively. However, CG1-C viewport tiles take 54.44% of the full ERP video, which increases the bitrate. The aforementioned problem is caused by Global FoV and the size of the tile. Global FoV of CG1-C is 180° × 180°, while the other sequences have 360°×180°. It means the cameras of CG1-C captured the videos in one directions, and the many tiles which have overlapping information are included in CG1-C. Further, we have observed that the viewport of CG1-C included the tiles at the edge of the ERP video. Even though the viewport area covers only a small area on a tile, the tile is included. If the tile size is small, the number of tiles which have unnecessary area will be decreased.

Table 1: List of used tools

Class	Tiling	Sequence X pose trace							
		MIV mode				MIV view mode			
		Xp01	Xp02	Xp03	Average	Xp01	Xp02	Xp03	Average
CG1-A	4×8	8.00	9.20	10.45	9.22	8.33	8.77	10.45	9.18
CG1-B	4×4	5.00	4.50	5.26	4.92	5.49	5.13	4.78	5.13
CG1-C	4×4	8.00	6.80	9.20	8.00	9.00	8.99	10.26	9.42
CG1-N	4×4	3.53	6.57	4.92	5.01	4.38	6.26	5.33	5.32

Considering the streaming scenario, the videos are divided into 33 frames chunks because 33 frames are the minimum frame length to reconstruct the bitstreams in random access structure, which is the encoding structure defined by CTC. For example, 97 frames of videos were divided into 0-32, 32-64, and 64-96 chunks and encoded for the streaming scenerio. The aforementioned chunk size definition may cause redundant I-frame transmission while the anchor does not. Nevertheless, the proposed VTS showed better results on BD-rate compared to the anchor.
Tiles corresponds to the user's viewport change frame by frame. Because of the chunk size, the extracted and transmitted tiles are decided by an union of the tiles in 33 frames. The experiment is conducted parts as follows. First, the test sequences are pre-processed by TMIV encoder. Second, the outputs of TMIV encoder are divided into 33 frames chunks to encode. Third, the chunks are encoded with HEVC encoder, while additional views are encoded without MCTS and basic views are encoded with MCTS. Fourth, basic views viewport tiles decision for each test sequence and pose trace is conducted using the proposed VTS. The viewport information is transmitted from the client HMD. Based on the tile lists of the VTS, viewport tiles are extracted from the bitstreams and delivered to the client. The client then decodes the tile chunks with HEVC decoder, and merges them. Finally, TMIV decoder generates the viewport using the merged videos and objective quality evaluation using peak-signal-to-noise ratio (PSNR) and immersive video PSNR (IV-PSNR) is conducted.

3.1 Objective evaluation on MIV mode and MIV view mode

Table 2: BD-rate savings of the proposed method in MIV mode

Test class	Anchor (ff)	High-BR BD rate Y-PSNR	Low-BR BD rate Y-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Pixel rate ratio
CG1	AA97 (MIV)	-23.5%	-21.3%	-31.1%	-24.8%	0.00%
	BA97 (MIV)	-24.9%	-24.1%	-24.1%	-23.5%	0.00%
	CA97 (MIV)	2.8%	3.5%	2.8%	2.7%	0.00%
	NA97 (MIV)	-7.3%	-4.4%	-8.1%	-5.0%	0.00%
	Average	-13.2%	-11.6%	-15.1%	-12.7%	0.00%

Table 2 shows the BD-rate savings of the proposed VTS compared to the anchor in MIV mode. As shown in the table, VTS is more efficient in high-quality video streaming. For Y-PSNR, *TechnicolorMuseum* showed 24.9% of high BD-rate saving. For IV-PSNR, *ClassroomVideo* showed 31.1% of high BD-rate saving. In average, VTS shows 13.2% of Y-PSNR high BD-rate saving in MIV mode.

Table 3: BD-rate savings of the proposed method in MIV view mode

Test class	Anchor (ff)	High-BR BD rate Y-PSNR	Low-BR BD rate Y-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Pixel rate ratio
CG1	AA97 (MIV)	-57.3%	-51.6%	-58.4%	-53.5%	0.00%
	BA97 (MIV)	-46.9%	-45.8%	-46.8%	-45.5%	0.00%
	CA97 (MIV)	19.3%	21.8%	17.4%	22.1%	0.00%
	NA97 (MIV)	-24.1%	-21.0%	-21.8%	-19.9%	0.00%
	Average	-27.2%	-24.2%	-27.4%	-24.2%	0.00%

As MIV view mode has more basic views than MIV mode, VTS shows more efficient results on MIV view mode, which is represented by Table 3. For Y-PSNR, *ClassroomVideo* showed 57.3% of high BD-rate saving. For IV-PSNR, *ClassroomVideo* showed 58.4% of high BD-rate saving. In average, VTS shows 27.2% of Y-PSNR high BD-rate saving in MIV view mode.

The rate-distortion curves (i.e. Y-PSNR [dB] vs. Bitrate [Mbps]) for ERP test sequences are shown in Figure 4. Note that the curves are represented with logarithmic horizontal axis.

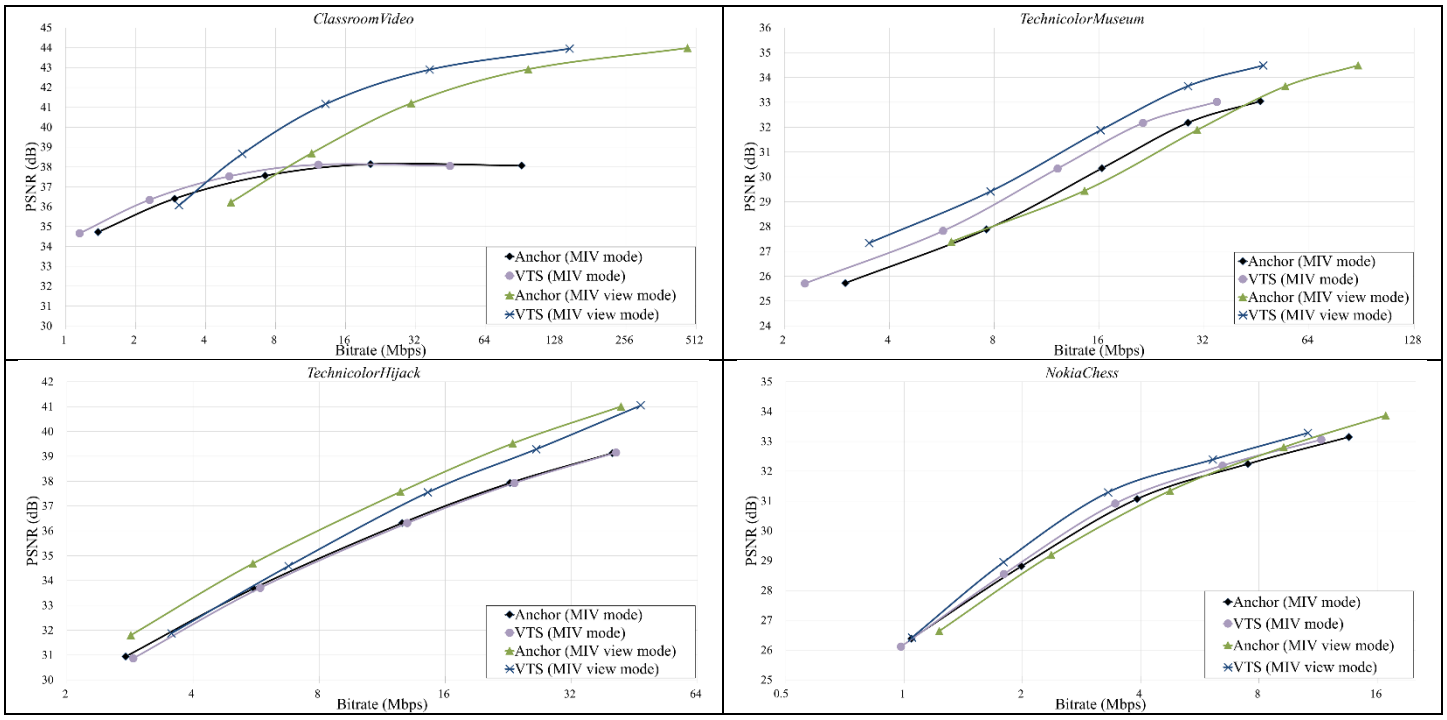

Fig. 4. Rate-distortion curves for VTS and anchor in MIV mode and MIV view mode

3.2 Subjective quality comparison with enlarged noticeable sections

Figure 5 shows the rendered viewports with enlarged noticeable sections on MIV view mode in CG1-A and CG1-B. As shown in the figure, there are several noticeable visual differences; the writings on the blackboard, arm, shoulder, and hair.

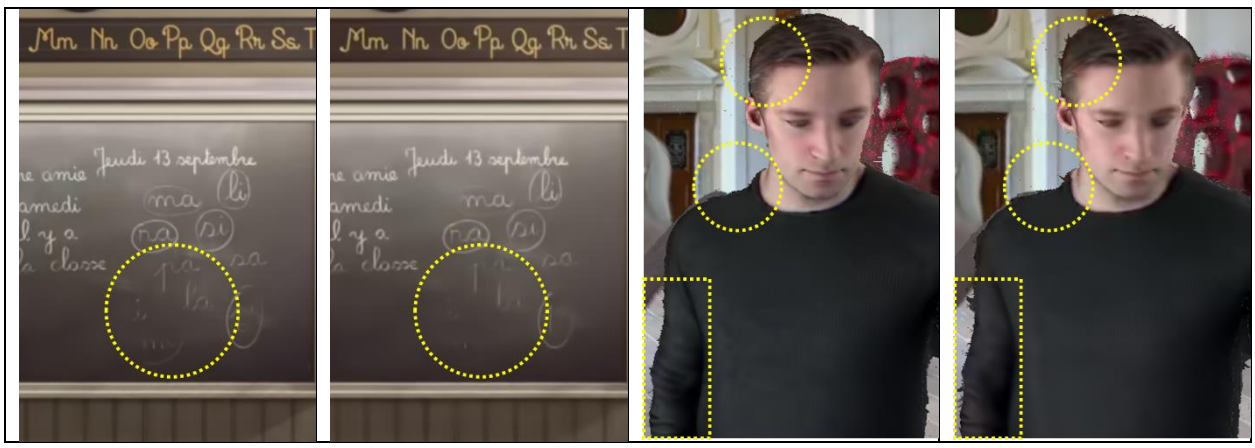

Fig. 5. Rendered viewport comparison with enlarged noticeable sections on MIV view mode in ClassroomVideo pose Ap02 anchor, QP4@11.43 Mbps, ClassroomVideo pose Ap02 VTS, QP3@13.12 Mbps, TechnicolorMuseum pose Bp02 anchor, QP4@14.53 Mbps, TechnicolorMuseum pose Bp02 VTS, QP3@16.14 Mbps

4 Conclusions and recommendations

This proposal introduces the viewport tile selector (VTS) method for 3DoF+/6DoF 360 video viewport-dependent rendering. When the camera parameter and depth map for each source view is given, VTS prints the viewport area tiles for each source view. The proposed VTS has advantages as follows:

- Compatible with existing MCTS-encoder and TMIV. Released HM 16.20 and TMIV 3.0 were used for this contribution.
- Significantly improved the objective and subjective quality. For CG1-B, there was a 24.9% BD-rate saving in MIV mode. For CG1-A, we were able to reduce the BD-rate by 57.3% in MIV view mode.
- Friendly with the personalized 3DoF+/6DoF streaming and rendering services.

We have implemented the proposed VTS under TMIV version 1.0, and the VTS implementation on the latest TMIV will be released later. We recommend to:

- Adopt this contribution into the latest TMIV.
- Keep CE-1 open for the next contributions.

References

[1] B. Salahieh, B. Kroon, J. Jung, M. Domański, “Test Model 3 for Immersive Video”, ISO/IEC JTC1/SC29/WG11 MPEG/n18795, October 2019, Geneva, Switzerland.

[2] R. Skupin, Y. Sanchez, K. Suehring, T. Schierl (HHI), E. Ryu, S. Jangwoo (Gachon University), “Temporal MCTS Coding Constraints Implementation”, ISO/IEC JTC1/SC29/WG11 MPEG/m41626, October 2017, Macau, China.

[3] M. Yu, H. Lakshman, B. Girod, “A framework to evaluateomnidirectional video coding schemes”, In 2015 IEEE International Symposium on Mixed and Augmented Reality. IEEE, 31–36.

[4] S. Deshpande, Y.K. Wang, M. M. Hannuksela, S. Deshpande, “Preliminary WD of ISO/IEC 23090-2 2nd edition OMAF”, ISO/IEC JTC1/SC29/WG11 MPEG/n18587, July 2019, Gothenburg, Sweden.

[5] J. Jung, B. Kroon, J. Boyce, “Common Test Conditions for Immersive Video”, ISO/IEC JTC1/SC29/WG11 MPEG/ n18789, October 2019, Geneva, Switzerland.