INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 04 MPEG VIDEO CODING

ISO/IEC JTC 1/SC 29/WG 04 m64165

July 2023, Geneva, Switzerland

Title **[MIV] Patch margin signaling for low-bitrate rendering improvement**

Source **PUT, ETRI**
Authors **Adrian Dziembowski, Dawid Mieloch, Gwangsoon Lee, Jun Young Jeong**

Abstract

The document presents a proposal of centering the position of a cluster within a patch and adapting the rendering-side patch margin to the cluster size in order to decrease an impact of coding artifacts. The recommendation is to adopt the proposal.

Ver 2 (with tracked changes):

- added section 4, which answers to the question: "Why do you need new syntax instead of just signaling a smaller patch?"
- added braces in syntax structure.

1 Proposal

We propose to modify a TMIV encoder by putting the cluster in the center of a patch (Fig. 1A) instead of its top-left corner (Fig. 1B). Using this solution, valid cluster pixels are not removed during the rendering (because of the patch margin skipping algorithm).

Moreover, centralized position allows to increase the patch margin for some patches, decreasing an influence of coding artifacts on viewport quality. The size of the patch margin should be sent to the decoder (within patch data unit).

Potentially, the outer part of a patch may be modified in order to help a video encoder (not a part of this contribution).

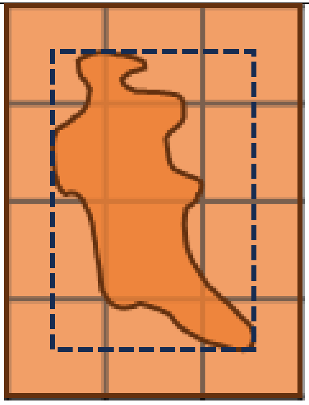
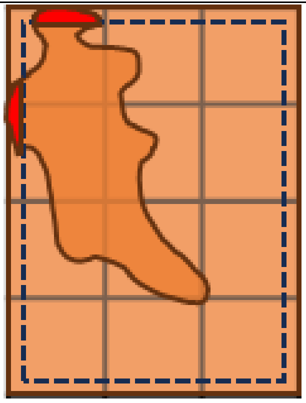
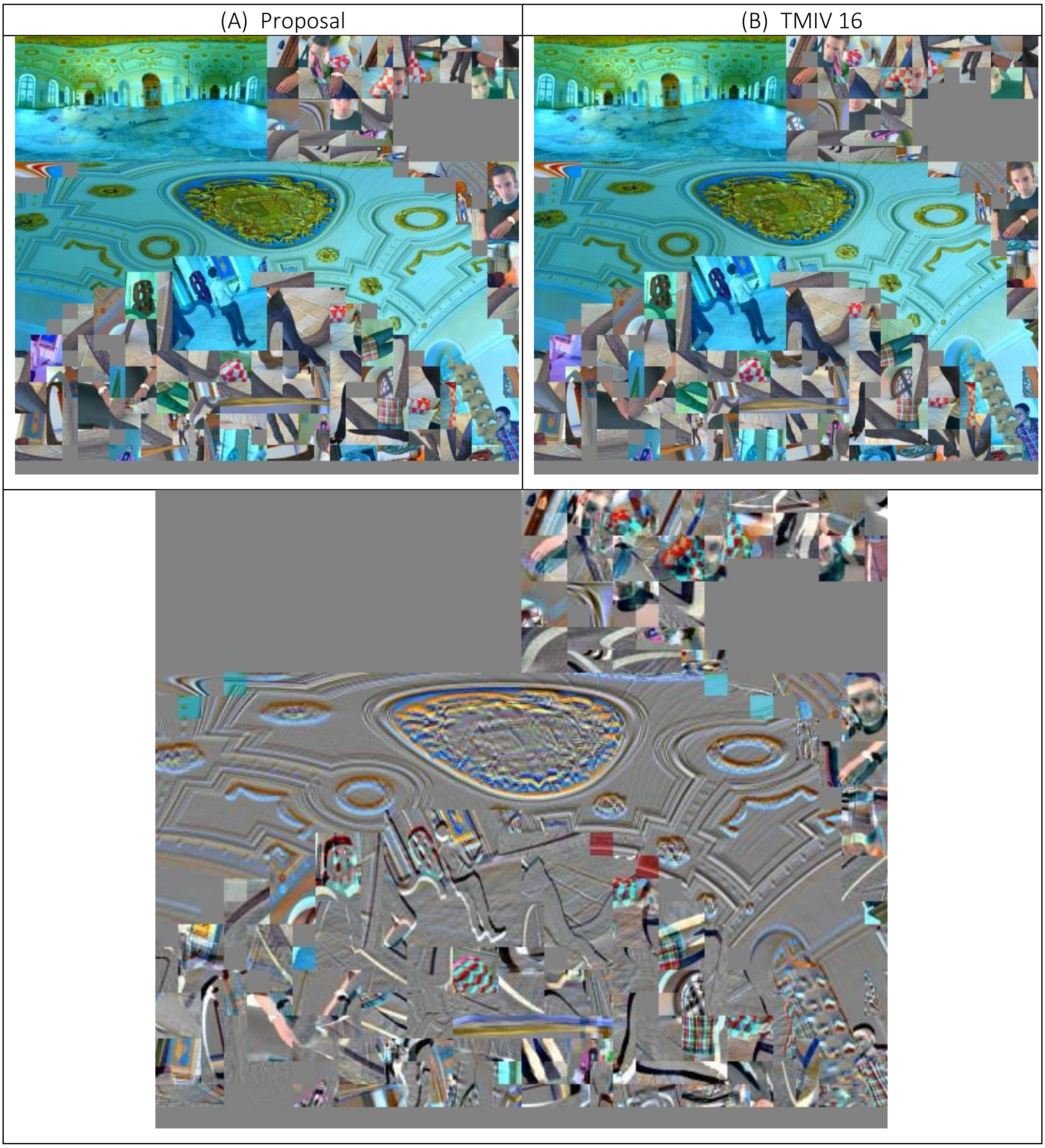
(A) Proposal	(B) TMIV 16
	

Fig. 1. Cluster within a patch in both approaches; dashed line: patch margin, areas not used in rendering; red area: valid information not used in rendering.



(A) Proposal

(B) TMIV 16

Fig. 2. Atlases in the proposed approach and TMIV 16.

As presented in Fig. 2, patches are packed in the same position as in TMIV 16, but the information within them is slightly shifted towards the bottom-right corner.

Some unoccupied 64x64 blocks became occupied (and vice versa) because of the shift of the cluster.

2 Results (A65)

Mandatory content - Proposal vs. Low/High-bitrate Anchors						Max delta Y-PSNR [dB]			Max delta IV-PSNR [dB]		
Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR	MIV Main	m64165	Difference [%]	MIV Main	m64165	Difference [%]
Chess	B02	127.5%	159.8%	-3.7%	-4.1%	8.91	10.62	19.3%	12.83	15.14	18.1%
Guitarist	B03	176.0%	99.5%	-0.7%	-0.7%	22.66	22.66	0.0%	21.03	20.99	-0.2%
Cadillac	J02	13.1%	14.9%	-0.8%	-0.8%	4.36	4.36	0.0%	5.16	5.01	-2.9%
Fan	J04	5.9%	6.9%	-0.6%	-0.7%	2.47	2.78	12.8%	2.74	3.47	27.0%
Group	W01	15.8%	9.0%	-0.7%	-0.6%	13.05	13.34	2.2%	13.42	13.43	0.1%
Painter	D01	4.7%	3.4%	-0.5%	-0.3%	5.46	5.92	8.4%	4.71	5.27	12.0%
Frog	E01	11.4%	9.5%	-1.0%	-0.9%	7.77	7.88	1.4%	4.83	4.88	0.8%
CBABasketball	L02	75.6%	65.0%	-1.8%	-1.8%	16.30	16.49	1.2%	14.10	14.36	1.9%
Average		**53.8%**	**46.0%**	**-1.2%**	**-1.2%**	**10.12**	**10.51**	**5.6%**	**9.85**	**10.32**	**7.1%**

Objective results for all content:

Class A

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
ClassroomVideo	A01	5.6%	3.7%	-0.1%	-0.1%
Average		**5.6%**	**3.7%**	**-0.1%**	**-0.1%**

Class B

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
Museum	B01	6.8%	7.0%	-0.5%	-0.6%
Chess	B02	127.5%	159.8%	-3.7%	-4.1%
Guitarist	B03	176.0%	99.5%	-0.7%	-0.7%
Average		**103.4%**	**88.8%**	**-1.6%**	**-1.8%**

Class C

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
Hijack	C01	27.5%	35.5%	-2.4%	-2.6%
Cyberpunk	C02	59.7%	19.3%	-0.8%	-0.5%
Average		**43.6%**	**27.4%**	**-1.6%**	**-1.5%**

Class J

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
Kitchen	J01	30.5%	31.5%	-1.2%	-1.2%
Cadillac	J02	13.1%	14.9%	-0.8%	-0.8%
Mirror	J03	23.5%	28.1%	-2.5%	-2.6%
Fan	J04	5.9%	6.9%	-0.6%	-0.7%
Average		**18.3%**	**20.4%**	**-1.3%**	**-1.3%**

Class W

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
Group	W01	15.8%	9.0%	-0.7%	-0.6%
Dancing	W02	73.9%	101.2%	-2.0%	-2.6%
Average		**44.9%**	**55.1%**	**-1.4%**	**-1.6%**

Class D

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
Painter	D01	4.7%	3.4%	-0.5%	-0.3%
Breakfast	D02	52.4%	32.3%	-2.2%	-1.3%
Barn	D03	50.0%	56.8%	-1.6%	-1.4%
Average		**35.7%**	**30.8%**	**-1.4%**	**-1.0%**

Class E

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
Frog	E01	11.4%	9.5%	-1.0%	-0.9%
Carpark	E02	4.3%	2.8%	-0.3%	-0.1%
Street	E03	32.0%	28.0%	-0.5%	-0.3%
Average		**15.9%**	**13.4%**	**-0.6%**	**-0.4%**

Class L

Sequence		BD-rate Y-PSNR	BD-rate IV-PSNR	BD-PSNR Y-PSNR	BD-PSNR IV-PSNR
Fencing	L01	18.9%	15.6%	-1.5%	-0.9%
CBABasketball	L02	75.6%	65.0%	-1.8%	-1.8%
MartialArts	L03	24.7%	30.4%	-0.5%	-0.3%
Average		**39.8%**	**37.0%**	**-1.3%**	**-1.0%**

Bitrates are very similar to the anchor, the BD-rate and BD-PSNR decreases are caused by lower objective quality:

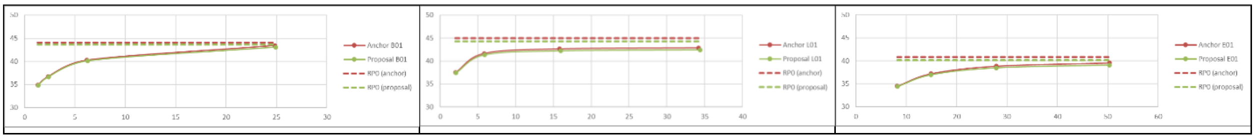


Fig. 3. RD-curves comparison for B01, L01, and E01.

Subjective comparison:

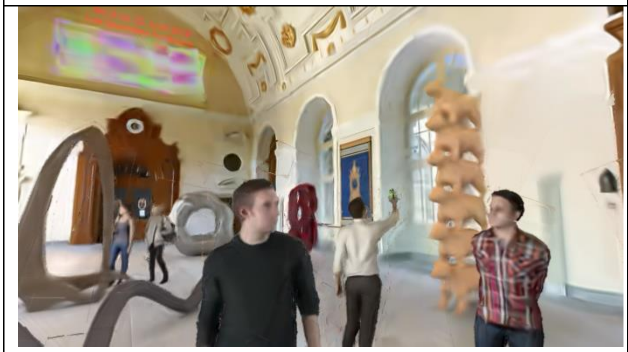
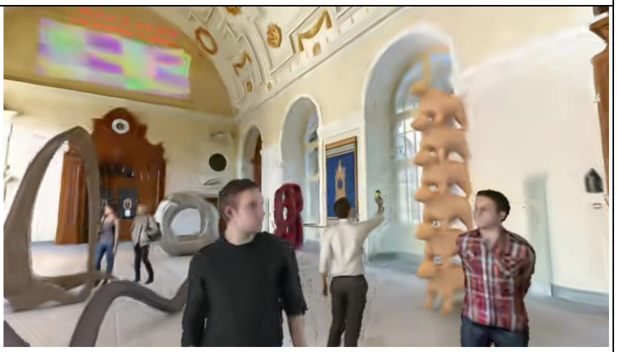
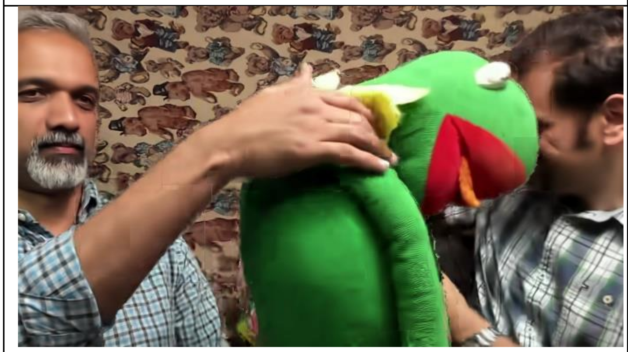
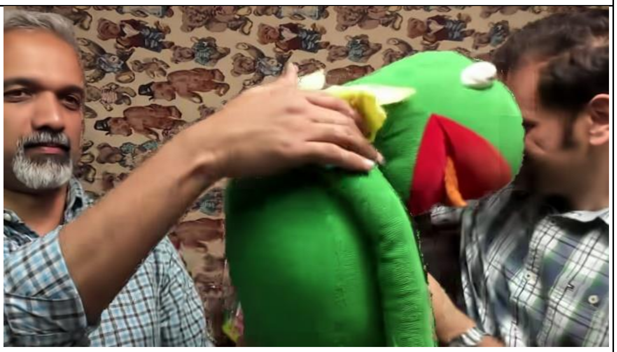
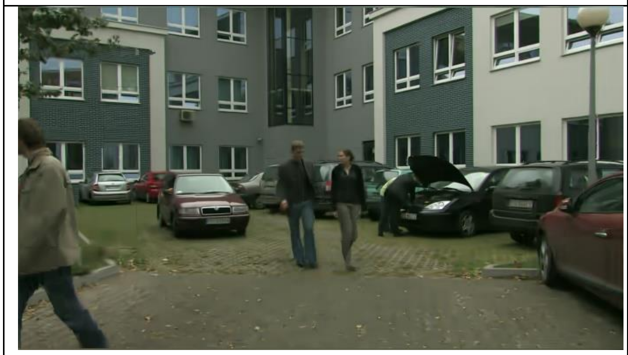
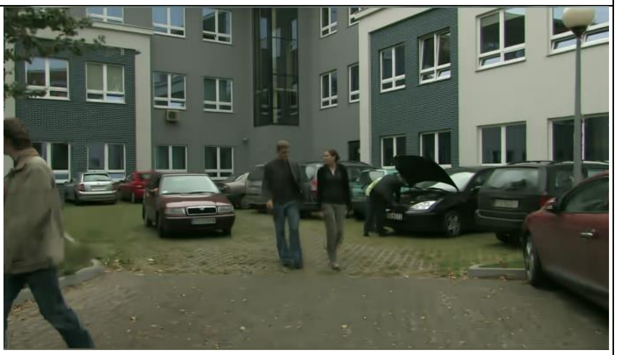
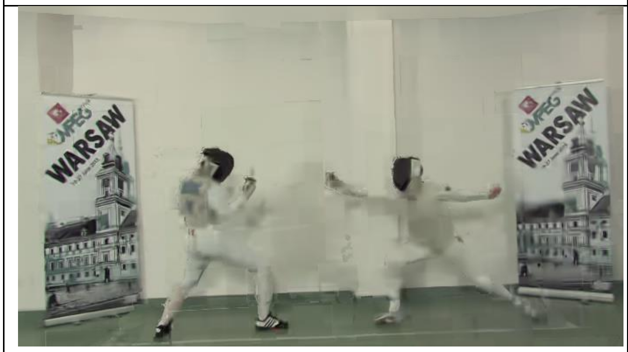
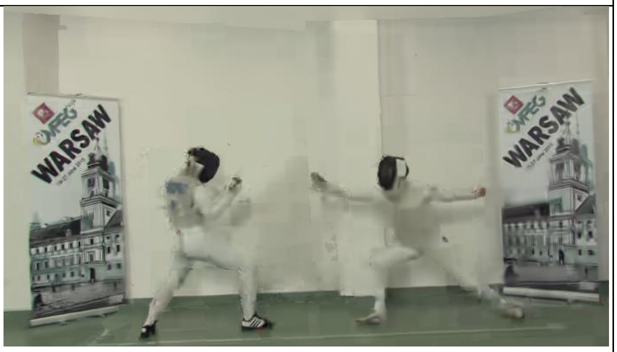
Anchor	Proposed
	
	
	
	

Fig. 4. Rendered viewports, RP4.

The proposed approach significantly reduces artifacts caused by strong encoding of patch boundaries. The differences are visible mostly for lowest bitrates.

In the proposed approach, more pixels are skipped during rendering. It reduces visible edges and decreases blurring.

3 Syntax & semantics

8.3.2.8 V3C parameter set MIV edition 2 extension syntax

vps_miv_2_extension( ) {	**Descriptor**
vps_miv_extension( )	
vme_reserved_zero_8bits	u(8)
vme_decoder_side_depth_estimation_flag	u(1)
vme_patch_margin_enabled_flag	u(1)
}	

vme_patch_margin_enabled_flag equal to 1 indicates that the patch margin parameters are present in the syntax structure. vme_patch_margin_enabled_flag equal to 0 indicates that the patch margin parameters are not present in the syntax structure. When not present, the value of vme_patch_margin_enabled_flag is inferred to be equal to 0.

8.3.2.7 Patch data unit MIV extension syntax

pdu_miv_extension(tileID, p) {	**Descriptor**
if(asme_max_entity_id > 0)	
pdu_entity_id[tileID][p]	u(v)
if(asme_depth_occ_threshold_flag)	
pdu_depth_occ_threshold[tileID][p]	u(v)
if(asme_patch_texture_offset_enabled_flag)	
for(c = 0; c < 3; c++)	
pdu_texture_offset[tileID][p][c]	u(v)
if(asme_inpaint_enabled_flag)	
pdu_inpaint_flag[tileID][p]	u(1)
if(vme_patch_margin_enabled_flag) {	
pdu_2d_margin_u[tileID][p]	u(v)
pdu_2d_margin_v[tileID][p]	u(v)
}	
}	

pdu_2d_margin_u[tileID][p] specifies the number of left-most and right-most columns in patch with index p of the current atlas tile, with tile ID equal to tileID, which contain only pruned pixels, which do not need to be decoded and used for rendering. The number of bits used to represent pdu_2d_margin_u[tileID][p] is asps_log2_patch_packing_block_size − 1.

pdu_2d_margin_v[tileID][p] specifies the number of top-most and bottom-most rows in patch with index p of the current atlas tile, with tile ID equal to tileID, which contain only pruned pixels, which do not need to be decoded and used for rendering. The number of bits used to represent pdu_2d_margin_v[tileID][p] is asps_log2_patch_packing_block_size − 1.

4 Why to signal patch margin instead of sending smaller patches?

Patch data unit in MIV 1 contains:

pdu_2d_pos_x	ue(v), / patchPackingBlockSize	<- 2^asps_log2_patch_packing_block_size
pdu_2d_pos_y	ue(v), / patchPackingBlockSize	<- 2^asps_log2_patch_packing_block_size
pdu_2d_size_x_minus1	ue(v), / patchSizeXQuantizer	<- 2^ath_patch_size_x_info_quantizer
pdu_2d_size_y_minus1	ue(v), / patchSizeYQuantizer	<- 2^ath_patch_size_y_info_quantizer

We propose to add two elements:

pdu_2d_margin_u	u(asps_log2_patch_packing_block_size - 1)
pdu_2d_margin_v	u(asps_log2_patch_packing_block_size - 1)

Below, we present bit savings caused by quantization / grid alignment for three approaches.
In the example, we assumed 64x64 grid and a cluster of size 7x7.

Case 1: MIV Main anchor (7x7 cluster have effective patch size 64x64):

pdu_2d_pos_x	/ 64	savings : 6 bits
pdu_2d_pos_y	/ 64	savings : 6 bits
pdu_2d_size_x_minus1	/ 64	savings : 6 bits
pdu_2d_size_y_minus1	/ 64	savings : 6 bits
~~pdu_2d_margin_u~~		nothing added
~~pdu_2d_margin_v~~		nothing added
TOTAL SAVINGS :		24 bits / patch

Case 2: Proposed (effective cluster grid size: 1x1):

pdu_2d_pos_x	/ 64	savings : 6 bits
pdu_2d_pos_y	/ 64	savings : 6 bits
pdu_2d_size_x_minus1	/ 64	savings : 6 bits
pdu_2d_size_y_minus1	/ 64	savings : 6 bits
pdu_2d_margin_u		added: 5 bits
pdu_2d_margin_v		added: 5 bits
TOTAL SAVINGS :		14 bits / patch

Case 3: MIV with effective patch grid size 1x1 (same rendering performance as in case 2):

pdu_2d_pos_x	/ 1	savings : 0 bits
pdu_2d_pos_y	/ 1	savings : 0 bits
pdu_2d_size_x_minus1	/ 1	savings : 0 bits
pdu_2d_size_y_minus1	/ 1	savings : 0 bits
~~pdu_2d_margin_u~~		nothing added
~~pdu_2d_margin_v~~		nothing added
TOTAL SAVINGS :		0 bits / patch

Proposed approach with patch margin signaling requires adding of 10 bits per patch, when compared to the MIV Main anchor. However, in order to obtain same rendering performance without adding proposed syntax (so using 1x1 grid), 24 additional bits per patch would be required.

Therefore, the proposed approach allows for saving 14 bits per patch for 64x64 grid. If the grid is set to 128x128, this number reaches 18 bits per patch.

5 Recommendation

We recommend watching provided posetraces and adopting the proposal.

6 Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory).