ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.30, No-1-2, 2022

Spatiotemporal redundancy removal in immersive video coding

Adrian Dziembowski [1]
adrian.dziembowski@put.poznan.pl
Dawid Mieloch [1]
Marek Domański [1]
Gwangsoon Lee [2]
Jun Young Jeong [2]

[1] Institute of Multimedia Telecommunications, Poznań University of Technology
Polanka 3, 61-131 Poznań, Poland

[2] Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

ABSTRACT

In this paper, the authors describe two methods designed for reducing the spatiotemporal redundancy of the video within the MPEG Immersive video (MIV) encoder: patch occupation modification and cluster splitting. These methods allow optimizing two important parameters of the immersive video: bitrate and pixelrate. The patch occupation modification method significantly decreases the number of active pixels within texture and depth video produced by the MIV encoder. Cluster splitting decreases the total area needed for storing the texture and depth information from multiple input views, decreasing the pixelrate. Both methods proposed by the authors of this paper were appreciated by the experts of the ISO/IEC JTC1/SC29/WG11 MPEG and are included in the Test Model for MPEG Immersive video (TMIV), which is the reference software implementation of the MIV standard.

Keywords

Immersive video coding, multiview compression, virtual reality.

1. INTRODUCTION

Recently, there is a common interest in immersive video [Isg14] and virtual reality systems, where a user virtually immerses into the scene. Such systems are an evolution of previous free-viewpoint television and free navigation systems [Tan12], [Sta18], where a user may virtually navigate around the scene.

In the immersive video system, a scene is acquired by a set of multiple precisely calibrated [Tao21] cameras. The number of cameras may vary, depending on the system, from less than ten [Mie20b] to even hundreds of cameras [Fuj06].

However, even in the most expensive systems equipped with dozens of cameras, the user should not be limited to watch the videos explicitly captured by the cameras. In order to provide smooth virtual navigation, the user should be able to choose his or her own viewport, which has to be rendered [Ceu18], [Fac18], [Zhu19] using data from input cameras.

Such a rendering requires the creation of the 3D model of the scene, i.e., calculation of the exact position of each captured object. The 3D scene model can be stored using various representations, e.g., meshes, voxels, or point clouds [Cui19], [Zha20], but the most commonly used representation is the MVD (multiview video plus depth) [Mül11]. In the MVD representation each input view is complemented by the corresponding depth map (either captured by time-of-flight cameras or estimated based on input views [Mie20a]).

Obviously, the user of the immersive video system has to receive the multiview content, i.e., multiple views, corresponding depth maps, and exact camera parameters. Without efficient compression, such content would require hundreds of megabits per second of the video, making the system highly impractical. The most straightforward method of the compression of the multiview content is performing the simulcast encoding, i.e., using separate instances of the video encoder (e.g., the newest VVC [Bro21]) for each input view and depth map. However, such an approach does not reduce the inter-view redundancy of input videos, thus wastes bits for coding of unnecessary information.

A better solution is to use dedicated video encoders, which utilize the similarity between several input views, e.g., MVC [Nem10], MV-HEVC, or 3D-HEVC [Tec15], which are the multiview extensions of the AVC [Sul05] and HEVC [Sul12] encoders. However, these techniques either restrict the camera arrangement (3D-HEVC, which allows compression of multiview video captured by linear camera systems) or do not efficiently use the information about the 3D

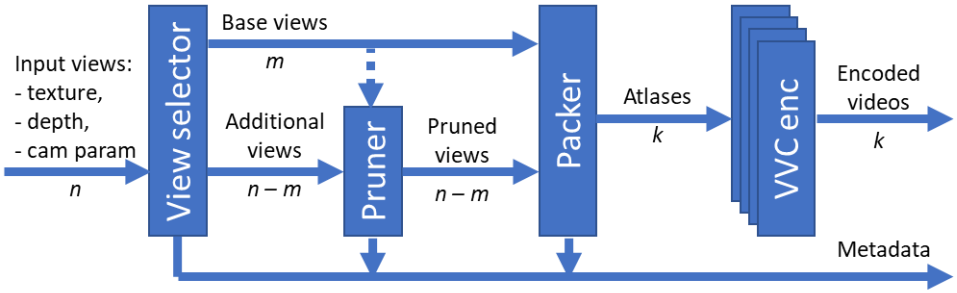

Figure 1. Simplified scheme of the MIV encoder.

scene model (MVC and MV-HEVC, which do not use depth maps for inter-view redundancy removal).

Mentioned flaws of existing multiview coding techniques motivated the development of the new technique – MPEG Immersive video (MIV) [Boy21], dedicated for any type of immersive video, including simple free navigation, free-viewpoint television, and virtual reality systems, where a user immerses into the 3D scene using the head-mounted device (HMD). The MPEG Immersive video is being developed by the ISO/IEC JTC1/SC29 WG04 MPEG VC group since 2019 and became a standard this year [ISO22].

2. MPEG IMMERSIVE VIDEO

The purpose of the MPEG Immersive video standard is to remove the inter-view consistency of the multiview video. As presented in Fig. 1, MIV is designed to be a preprocessing step before the actual video compression, which is performed using a typical video encoding algorithm, e.g., VVC. However, MIV is codec agnostic, so another video encoders (HEVC, AVC, or even M-JPEG) may be used as well.

The input data for the MIV encoder are n input views (including texture, depth map, and camera parameters for each input view).

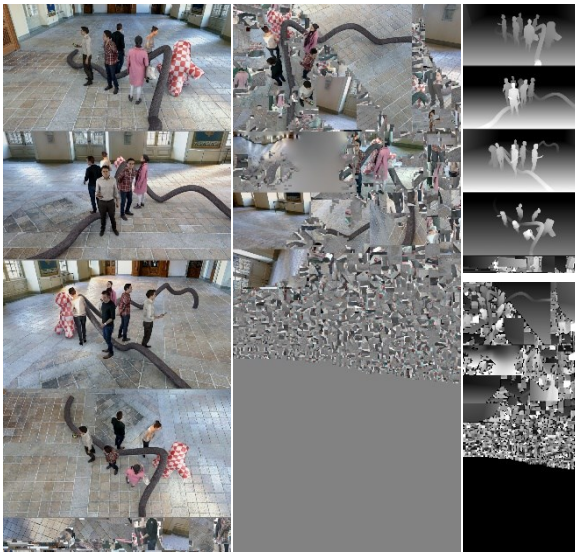

Figure 2. Atlases for sequence Group: two texture atlases and two depth atlases (reduced resolution).

Based on these data, MIV creates k atlases – videos containing information from n input views. An example of atlases is presented in Fig. 2.

At the first step, the MIV encoder chooses views, which will be sent without inter-view redundancy removal. These views are called "base views" and are pasted into the first atlas as full views. The base views are selected automatically based on camera parameters to cover the possibly largest part of the scene. Any other view ("additional view") is pruned in order to reduce the inter-view redundancy.

The pruning operation is performed by reprojecting pixels between views. Any pixel of an additional view is removed (pruned) if its depth and color are similar to the depth and color of the pixel reprojected to its position from base views and other (already pruned) additional views.

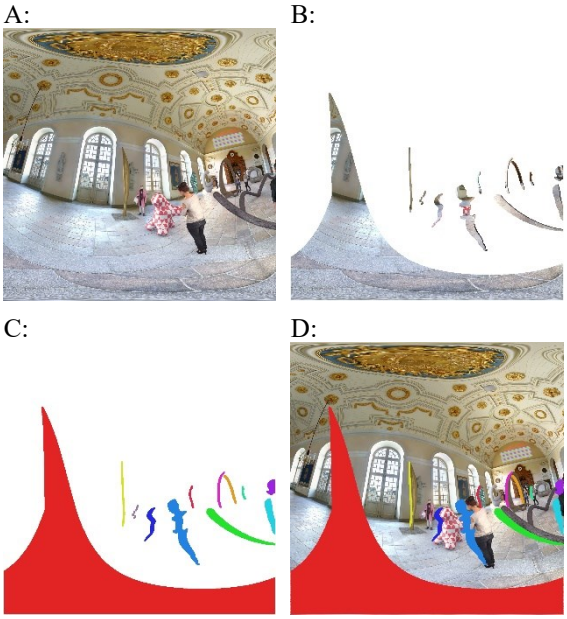

Figure 3. Pruning and clustering; A: input view (sequence Museum), B: view after pruning, C: preserved pixels after clustering, D: preview of clusters within input view.

All the preserved (non-pruned) pixels of each view are then merged into consistent clusters containing mutually connected pixels. An example of pruning

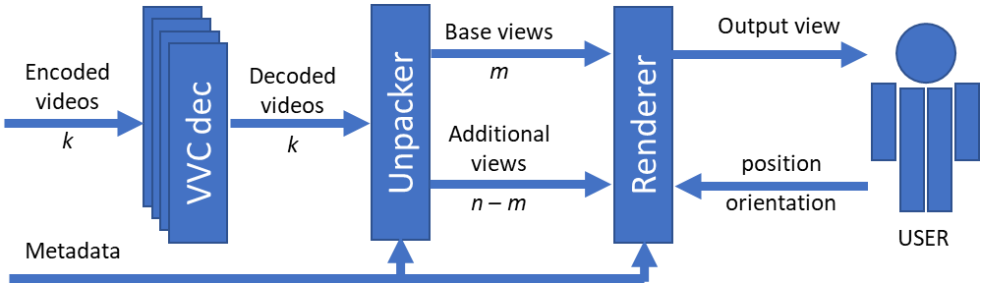

Figure 4. Simplified scheme of the MIV decoder.

and clustering is presented in Fig. 3. In Fig. 3A, an additional view is shown. All inter-view redundant pixels were pruned and only areas non-visible in other views were preserved (Fig. 3B). Fig. 3C presents the effect of pixel clustering, where each cluster was colored differently. In Fig. 3D, colored clusters were pasted into the input view to highlight, which areas of the view were preserved (disocclusions behind foreground objects and the bottom part of a view, which was out of field of view in other cameras).

In the next step, all the clusters are packed into atlases as "patches", containing a cluster together with its bounding box.

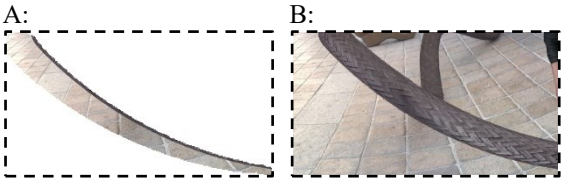

Figure 5. Cluster vs. patch; A: green cluster from Fig. 3, B: patch containing texture information for the cluster and its entire bounding box.

The packing process tries to efficiently fit non-pruned information from all input views in atlases, significantly reducing the total pixelrate (total number of pixels that have to be processed by the decoder) of the video (compared to the pixelrate of input video).

Of course, the packing operation has to be reversible in order to allow the unpacking of the atlas at the decoder side (Fig. 4). Reversibility of the packing process is provided by sending additional metadata for each patch, including its size, position within the input view, position within the atlas, and input view number.

In the last step, each atlas is separately encoded by the typical video encoder, e.g., VVC. The MIV standard [ISO22] describes also the process of multiplexing video bitstreams with metadata, as well as many other minor video processing techniques providing more efficient encoding of immersive video. However, this paper does not focus on them. The detailed description of the MIV encoder can be found in [Boy21] or [MPEG21].

On the decoder side, *k* video bitstreams are decoded using a typical video encoder (e.g., VVC). After video decoding, the input views are restored by unpacking patches from the atlases. Base views are restored completely. Restored additional views have many unoccupied areas, which were pruned in the encoder.

The last step of the decoding is the rendering of the view being watched by the user of the immersive video system. The user provides his or her position and orientation, and the renderer creates demanded virtual view.

3. PATCH OCCUPATION MODIFICATION

As presented in Fig. 5, a patch is a rectangular fragment of the input view, containing a cluster of non-pruned pixels together with its entire bounding box (Fig. 6A).

Such an approach has a major flaw: when the video encoder processes an entire patch, it wastes many bits for encoding useless texture information (pixels qualified as inter-view redundant by the pruner).

On the other hand, patches could contain only the non-redundant pixels (Fig. 6B), i.e., any pixel outside of the cluster could be greyed out and signaled as unoccupied by setting its depth value to a restricted level [MPEG21].

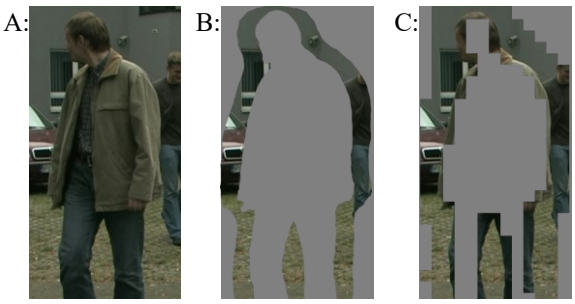

Figure 6. Various approaches to patch occupation; A: fully occupied patch, B: patch containing only non-pruned pixels, C: patch with modified occupation; sequence Carpark.

However, clusters have irregular shapes and thus are more difficult to efficiently encode by the video encoder, which has to handle many irregular edges between preserved and pruned (greyed out) areas. Moreover, the shape of a cluster changes in time because of the movement of objects in the scene, additionally reducing the efficiency of the inter-frame prediction.

We proposed an encoder-oriented solution, which adapts to the grid of the coding tree in the video encoder. In the proposed approach, the cluster is divided into blocks (e.g., blocks of size 16×16 pixels), and the entire block is greyed out if it does not contain any preserved pixels. Otherwise, all pixels within the block have a texture (Fig. 6C).

Figs. 7 and 8 compare both texture atlases created using two approaches: default with fully occupied patches (Fig. 7) and the proposed one (Fig. 8).

Figure 7. Texture atlases without patch occupation modification (sequence Frog).

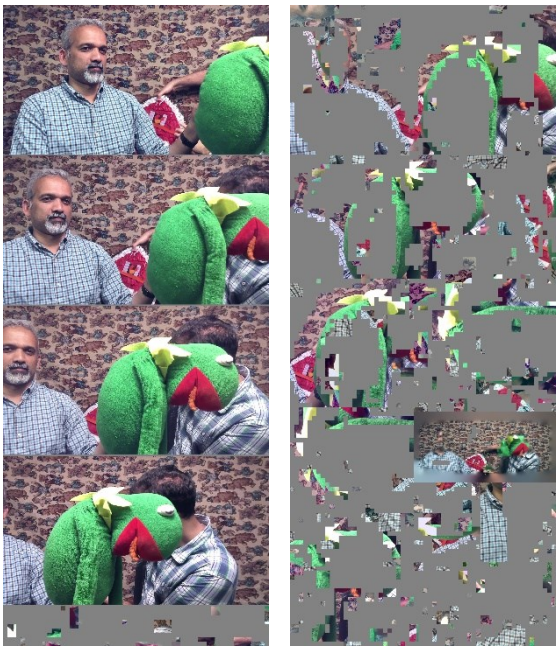

Figure 8. Texture atlases with patch occupation modification (sequence Frog).

As presented, the proposed modification significantly reduces the number of non-grey pixels within the second atlas. It does not change base views in the first atlas, as they are not pruned (cf. Fig. 1).

Regarding the temporal domain, the active blocks within atlases change over time, slightly decreasing the inter-frame prediction efficiency, but due to the fact, that the patch position does not change in consecutive frames, active blocks still have a similar texture and the decrease is very slight (Fig. 9).

Figure 9. Fragment of the second atlas from Figs. 7 and 8, frames 0 and 10; top: with fully occupied patches, bottom: with proposed patch occupation modification.

The method proposed by the authors of this paper was appreciated by the ISO/IEC MPEG VC experts [Dzi20b] and is included in the Test Model for MPEG Immersive video (TMIV) [MPEG21], which is the reference software implementation of MIV.

4. CLUSTER SPLITTING

The method of changing patch occupancy can decrease the bitrate needed for encoding of the atlases (especially the second one, as it does not contain base views), but it does not change the second crucial parameter of the practical immersive video system – the pixelrate, which defines the total number of pixels that have to be decoded. Therefore, we proposed a second technique, which allows reducing this parameter by allowing the splitting of large irregular clusters, e.g., the big red cluster presented in Fig. 3.

If a cluster is L-shaped (Fig. 10A), the patch containing this cluster has many unoccupied pixels (red area in Fig. 10B). If such a cluster will be split into two smaller clusters, the total area of two patches may be significantly smaller (red and blue areas in Fig. 10D).

The split line is parallel to the shorter side of the patch (Fig. 10C) and is placed in a position, which minimizes the total area of patches after the split. If the total area of patches after the split is similar to the area before splitting, the cluster is not split.

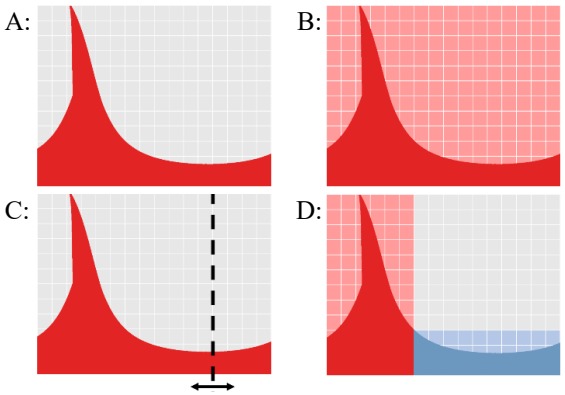

Figure 10. Splitting of the L-shaped cluster; A: initial cluster, B: patch for cluster A, C: the splitting of cluster A, D: cluster A split into two smaller clusters.

If the cluster has irregular contour but is not L-shaped (Fig. 11A), a different splitting algorithm is performed. For such a patch, occupied and non-occupied areas are being compared. If most pixels within the patch are non-occupied, the cluster is split into two halves, along the line parallel to the shorter side of the patch (Fig. 11B), resulting in two smaller clusters (Fig. 11C).

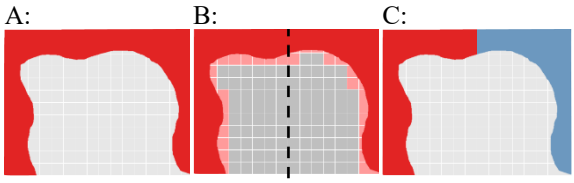

Figure 11. Splitting of the C-shaped cluster; A: initial cluster, B: the splitting of cluster A, C: cluster A split into two smaller clusters.

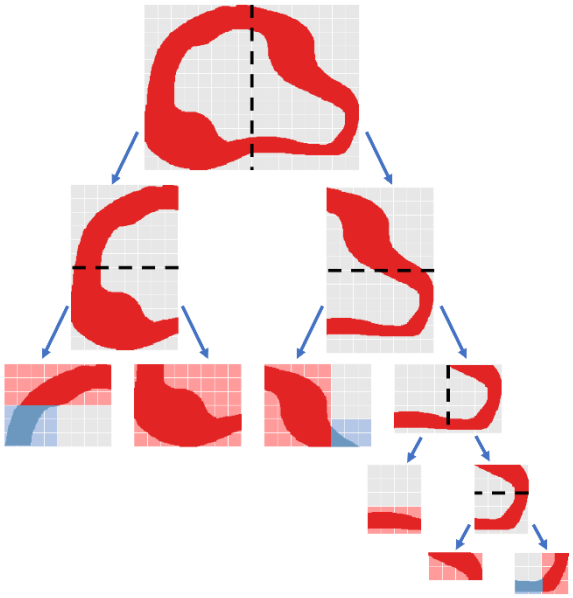

Figure 12. Recursive splitting of irregular cluster.

As presented in Fig. 11, the result of the splitting of the C-shaped cluster is two L-shaped clusters. To provide a reduction of the total area of patches, such clusters have to be split again, as shown in Fig. 10. Multiple splitting of a cluster is possible due to the recursiveness of the proposed method – each cluster is split until the splitting significantly minimizes the total area of the patches (Fig. 12).

As presented in Figs. 13 and 14, the proposed method of cluster splitting allows to significantly decrease the total occupied area of the atlas (the non-occupied, grey area in the second atlas in Fig. 14 is much bigger than the non-occupied area in Fig. 13).

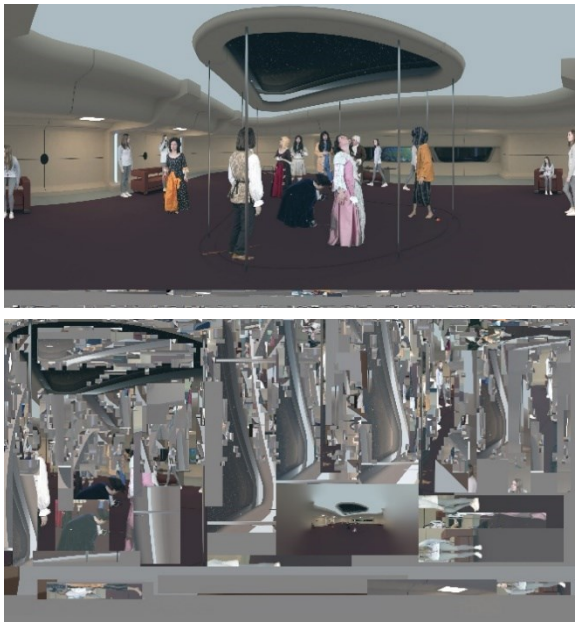

Figure 13. Two atlases without cluster splitting (sequence Hijack).

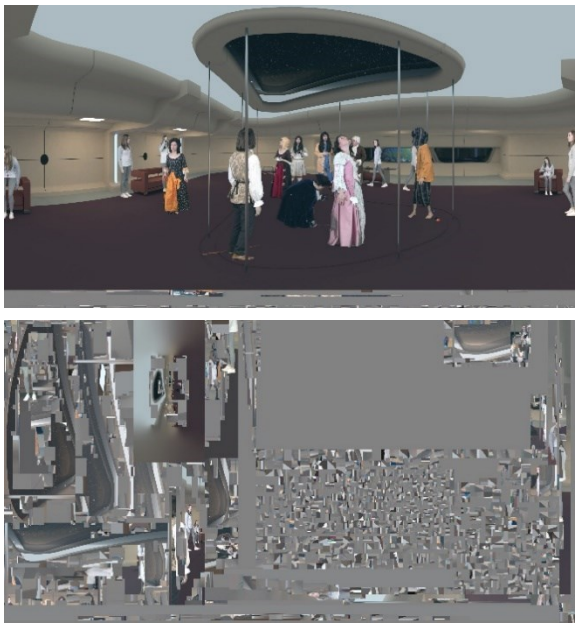

Figure 14. Two atlases with cluster splitting (sequence Hijack).

Similarly to the method presented in the previous section, also the cluster splitting method proposed by the authors of this paper was appreciated by the ISO/IEC MPEG VC experts [Dzi20a] and is included in the Test Model for MPEG Immersive video (TMIV) [MPEG21].

5. EXPERIMENTAL RESULTS

Both techniques presented in this paper were tested under the common test conditions for MPEG Immersive video (MIV CTC) [MPEG22], on 16 miscellaneous test sequences, including both natural content (NC) and computer-generated (CG) sequences of different resolutions, the number of cameras, and camera types (perspective and omnidirectional, represented in equirectangular projection – ERP). Key parameters of the test set are presented in Table 1.

Sequence name	Views	Type	Resolution	Source
Cadillac	15	NC/Persp.	1920×1080	[Dor21a]
Carpark	9	NC/Persp.	1920×1088	[Mie20b]
Chess	10	CG/Omni	2048×2048	[Ilo19]
ChessPieces	10	CG/Omni	2048×2048	[Ilo20]
ClassroomVideo	16	CG/Omni	1920×1080	[Kro18]
Fan	15	NC/Persp.	1920×1080	[Dor20a]
Fencing	10	NC/Persp.	1920×1080	[Dom16]
Frog	13	NC/Persp.	1920×1080	[Sal18]
Group	21	NC/Persp.	1920×1080	[Dor20b]
Hall	9	NC/Persp	1920×1088	[Mie20b]
Hijack	10	CG/Omni	4096×2048	[Dor18]
Kitchen	25	CG/Persp.	1920×1080	[Boi18]
Mirror	15	CG/Persp.	1920×1080	[Dor21b]
Museum	24	CG/Omni	2048×2048	[Dor18]
Painter	16	NC/Persp	2048×1088	[Doy18]
Street	9	NC/Persp	1920×1088	[Mie20b]

Table 1. Test sequences used in experiments.

To present a variety of content within the test set, Figs. 15 and 16 contain a single frame from each sequence.

Figure 15. Natural sequences. Left column: Carpark, Street, Frog; right column: Hall, Fencing, Painter.

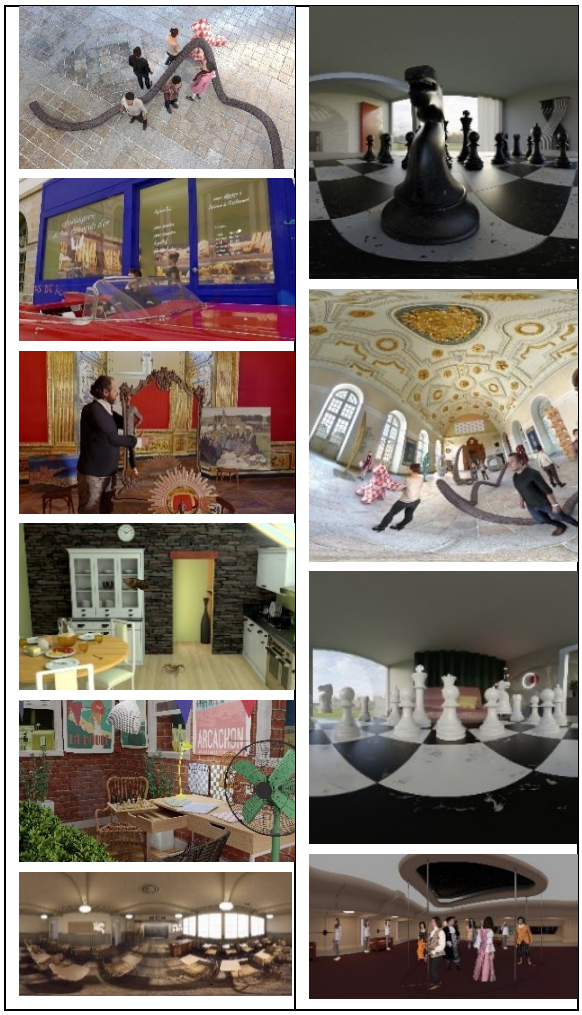

Figure 16. Computer-generated sequences. Left column: Group, Cadillac, Mirror, Kitchen, Fan, and ClassroomVideo; right column: Chess, Museum, ChessPieces, and Hijack.

Table 2 shows the gain of the proposed patch occupation modification method presented as bitrate reduction (compared to an atlas with fully occupied patches), separately for texture atlases, depth atlases, and total reduction for all video bitstreams.

As presented, the proposed method allows to significantly decrease the total bitrate needed for the representation of encoded video bitstreams, especially for higher bitrates (25 Mbps, on average). For low bitrates, the reduction is lower but noticeable.

Different efficiency between low and high bitrate was expected, as for higher bitrates, the encoder tries to encode all the high-frequency details of the video (and fully occupied patches have much more details than plain grey area), while for lower bitrates the details are destroyed (so the number of bits needed for encoding of grey area and highly compressed texture and depth is more similar).

Differences between various sequences, which can be spotted for low bitrates are caused by the sequence

characteristics. For example, texture of the Fan sequence is very detailed and has many areas with high frequencies which are very hard to be encoded at low bitrates. Therefore, when these areas are greyed out, the encoding is more efficient. On the other hand, for less-detailed sequences (e.g., ChessPieces) proposed technique increases a number of high frequencies (by adding edges between occupied and non-occupied regions), decreasing the efficiency of the VVC at higher QPs.

Test sequence	Bitrate reduction (patch occupation modification vs. fully occupied patches)					
	High bitrate (~25 Mbps)			Low bitrate (~7 Mbps)		
	Texture	Depth	All	Texture	Depth	All
Carpark	5.0%	5.0%	5.1%	0.9%	3.8%	3.0%
Fencing	4.6%	3.5%	4.3%	-6.1%	-1.3%	-2.6%
Frog	26.4%	14.1%	22.2%	24.2%	9.5%	17.4%
Hall	4.7%	9.5%	8.5%	-8.2%	7.1%	5.1%
Painter	10.5%	9.6%	10.1%	5.6%	5.6%	5.6%
Street	7.0%	8.4%	7.4%	6.0%	8.1%	7.0%
NC: Average	**9.7%**	**8.3%**	**9.6%**	**3.7%**	**5.5%**	**5.9%**
Cadillac	6.5%	4.8%	6.3%	-7.0%	-15.6%	-13.1%
Fan	25.0%	35.6%	32.5%	17.5%	34.2%	32.1%
Group	7.6%	3.3%	6.8%	-4.0%	-1.9%	-2.4%
Kitchen	14.6%	12.8%	14.3%	8.4%	9.0%	8.7%
Mirror	-2.1%	1.8%	-0.6%	-6.2%	0.1%	-1.7%
CG-P: Average	**10.3%**	**11.7%**	**11.9%**	**1.7%**	**5.2%**	**4.7%**
Chess	14.8%	4.3%	12.7%	2.8%	-6.5%	-1.0%
ChessPieces	9.6%	-2.5%	6.4%	-6.8%	-19.3%	-13.3%
ClassroomVideo	15.3%	10.7%	14.4%	1.4%	3.4%	2.6%
Hijack	32.3%	9.9%	27.6%	19.8%	6.8%	12.9%
Museum	20.3%	7.5%	18.0%	10.5%	-0.3%	5.3%
CG-O: Average	**18.5%**	**6.0%**	**15.8%**	**5.5%**	**-3.2%**	**1.3%**
All: Average	**12.6%**	**8.6%**	**12.3%**	**3.7%**	**2.7%**	**4.1%**

Table 2. Bitrate reduction caused by the proposed patch occupation modification method; NC: natural content, CG-O: computer-generated omnidirectional video, CG-P: computer-generated perspective video.

In Table 3, the influence of the second proposed method – cluster splitting – is presented. The cluster splitting purpose is to decrease the pixelrate of the immersive video. However, in the MIV CTC [MPEG22], the pixelrates are explicitly set to the limit defined for HEVC Level 5.2: 1,069,547,520 luma samples.

Therefore, to be compliant with the MIV CTC, we did not change the atlas size (thus pixelrate), but we have calculated the total area occupied by patches. This approach allows to estimate the possible pixelrate reduction without modifying the CTC.

As presented in Table 3, proposed cluster splitting allows to significantly reduce the total occupied area of the second atlas. The first atlas is practically unchanged, as it contains mostly the base views.

Test sequence	Occupied area in second atlas		
	No cluster splitting	Cluster splitting	Difference
Carpark	25.41%	25.62%	0.21%
Fencing	47.54%	42.61%	– 4.94%
Frog	15.63%	11.98%	– 3.65%
Hall	43.62%	41.66%	– 1.97%
Painter	20.98%	21.23%	0.25%
Street	8.99%	5.89%	– 3.10%
NC: Average	**27.03%**	**24.83%**	**– 2.20%**
Cadillac	96.47%	94.75%	– 1.72%
Fan	43.13%	38.65%	– 4.49%
Group	89.38%	79.44%	– 9.95%
Kitchen	39.87%	40.32%	0.45%
Mirror	92.60%	79.97%	– 12.63%
CG-P: Average	**72.29%**	**66.63%**	**– 5.67%**
Chess	97.70%	85.10%	– 12.60%
ChessPieces	98.66%	84.81%	– 13.85%
ClassroomVideo	21.62%	21.82%	0.20%
Hijack	93.27%	79.43%	– 13.84%
Museum	69.12%	40.26%	– 28.86%
CG-O: Average	**76.07%**	**62.28%**	– 13.79%
All: Average	**56.50%**	**49.60%**	**– 6.91%**

Table 3. Area occupied by patches in the second atlas with and without the proposed cluster splitting method.

The possible pixelrate for both approaches can be calculated as follows:

$$P\left[\frac{pix}{s}\right] = (1 + O) \cdot W_A \cdot H_A \cdot FPS \cdot 1.25$$

where: O is the occupied area percentage presented in Table 3, W_A and H_A are atlas width and height (defined in the MIV CTC [MPEG22]), FPS is the frame rate of the sequence (25 for Carpark, Fencing, Hall, and Street; 30 for other sequences). Multiplier 1.25 allows to include both texture and geometry atlas (1 for texture with full resolution and 0.25 for depth atlas, decimated by 2 in both directions [MPEG21]).

Figs. 17 and 18 present the influence of both proposed methods, in terms of both bitrate and pixelrate.

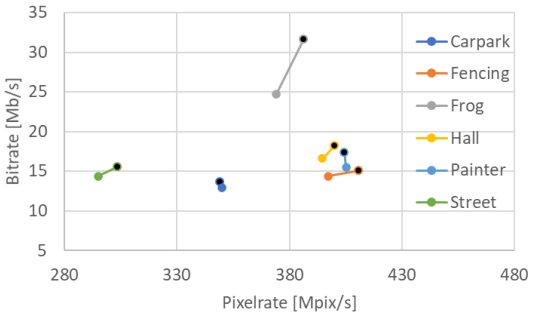

Figure 17. Bitrate vs. pixelrate for natural content; black dot: without proposed methods, color dot: with proposed modifications.

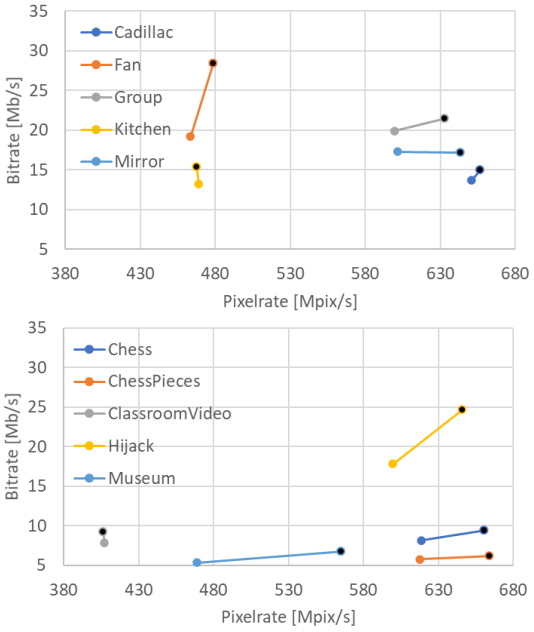

Figure 18. Bitrate vs. pixelrate for computer-generated sequences; black dot: without proposed methods, color dot: with proposed modifications.

As presented, the combination of patch occupation modification and cluster splitting allows to significantly decrease bitrate and pixelrate, irrespectively to the type of content.

Moreover, both proposed methods do not affect the rendering quality, as they do not modify the non-pruned pixels, which are used for rendering the virtual view watched by user of the immersive video system.

Regarding the computational time, video encoding (i.e., VVC) is much faster than without proposed techniques, while time needed for MIV encoding (i.e., atlas creation) is not influenced (Table 4).

Test sequence	MIV encoding time change			VVC encoding time change		
	A	B	C	A	B	C
Carpark	101.63%	113.66%	118.18%	93.16%	96.77%	84.80%
Fencing	90.76%	131.41%	99.88%	96.77%	95.49%	101.99%
Frog	111.34%	98.40%	116.30%	77.28%	106.93%	74.89%
Hall	86.38%	86.51%	88.32%	87.95%	98.06%	83.99%
Painter	116.85%	101.57%	107.15%	97.84%	91.31%	83.03%
Street	149.52%	101.37%	101.24%	92.34%	96.48%	84.43%
NC: Average	**109.41%**	**105.49%**	**105.18%**	**90.89%**	**97.51%**	**85.52%**
Cadillac	75.70%	75.40%	67.69%	84.10%	93.91%	85.65%
Fan	93.26%	139.61%	104.55%	74.90%	92.97%	76.62%
Group	98.46%	100.46%	105.20%	93.31%	90.58%	90.49%
Kitchen	99.52%	97.69%	117.75%	83.96%	97.52%	92.46%
Mirror	112.80%	113.05%	114.48%	111.43%	99.21%	97.06%
CG-P: Average	**95.95%**	**105.24%**	**101.93%**	**89.54%**	**94.84%**	**88.46%**
Chess	90.07%	86.19%	91.43%	96.11%	89.98%	90.54%
ChessPieces	97.00%	115.62%	102.29%	97.91%	92.34%	93.50%
ClassroomVideo	95.72%	104.38%	89.26%	80.23%	110.26%	89.30%
Hijack	85.47%	97.65%	94.85%	79.31%	90.07%	62.65%
Museum	100.37%	84.47%	101.29%	82.41%	89.66%	72.81%
CG-O: Average	**93.73%**	**97.66%**	**95.82%**	**87.19%**	**94.46%**	**81.76%**
All: Average	**100.30%**	**102.96%**	**101.24%**	**89.31%**	**95.72%**	**85.26%**

Table 4. Encoding time change (compared to the approach without proposed techniques); A: patch occupation modification, B: cluster splitting, C: both proposed techniques enabled.

6. CONCLUSIONS

The paper presents two techniques which allow to reduce the spatiotemporal redundancy of video within the MPEG Immersive video (MIV) encoder.

The first method is the patch occupation modification, which decreases the total bitrate of the immersive video encoded by MIV by decreasing the number of occupied pixels within texture and depth atlases.

The second method – cluster splitting allows to split large irregular clusters in order to decrease the total area of patches thus the pixelrate of the video.

Both ideas proposed by the authors of this paper were evaluated by experts of the ISO/IEC JTC1/SC29/WG 11 MPEG and are included in the Test Model for MPEG Immersive video (TMIV), which is the reference software implementation of the MIV.

7. ACKNOWLEDGMENTS

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).

8. REFERENCES

[Bro21] B. Bross et al. Overview of the Versatle Video Coding (VVC) standard and its applications. IEEE Tr. on Circ. and Syst. for Vid. Tech., 2021.

[Boi18] Boissonade P., and Jung J. Proposition of new sequences for Windowed-6DoF experiments on compression, synthesis, and depth estimation. Document ISO/IEC JTC1/SC29/WG11 MPEG/M43318, Ljubljana, Slovenia, Jul. 2018.

[Boy21] Boyce J., Doré R., Dziembowski A., Fleureau J., Jung J., Kroon B., Salahieh B., Vadakital V.K.M., and Yu L. MPEG Immersive Video Coding Standard. Proceedings of the IEEE, vol. 109, no. 9, pp. 1521-1536, Sep. 2021.

[Ceu18] B. Ceulemans et al. Robust Multiview Synthesis for Wide-Baseline Camera Arrays. IEEE Tr. on Multimedia, 2018.

[Cui19] L. Cui et al. Point-Cloud Compression: Moving Picture Experts Group's New Standard in 2020. IEEE Consumer Electronics Mag., 2019.

[Dor18] Doré R. Technicolor 3DoF+ Test Materials. Doc. ISO/IEC JTC1/SC29/WG11 MPEG/ M42349, San Diego, CA, USA, Apr. 2018.

[Dor20a] Doré R. et al. InterdigitalFan0 content proposal for MIV. Doc. ISO/IEC JTC1/SC29/ WG04 MPEG VC/ M54732, Online, Jul. 2020.

[Dor20b] Doré R. et al. InterdigitalGroup content proposal for MIV. Doc. ISO/IEC JTC1/SC29/ WG04 MPEG VC/ M54731, Online, Jul. 2020.

[Dor21a] Doré R. et al. Interdigital Mirror Content Proposal for advanced MIV investigations on reflection. Doc. ISO/IEC JTC1/SC29/WG04 MPEG VC/ M55710, Online, Jul. 2021.

[Dor21b] Doré R. et al. New Cadillac content proposal for advanced MIV v2 investigations. Doc. ISO/IEC JTC1/SC29/WG04 MPEG VC/ M57186, Online, Jan. 2021.

[Dom16] Domański M. et al. Multiview test video sequences for free navigation exploration obtained using pairs of cameras. Doc. ISO/IEC JTC1/SC29/WG11, MPEG M38247, 2016.

[Doy18] Doyen D. et al. [MPEG-I Visual] New Version of the Pseudo-Rectified Technicolor painter Content. Doc. ISO/IEC JTC1/SC29/ WG11 MPEG/M43366, Ljublana, 2018.

[Dzi20a] Dziembowski A. et al. Immersive Video CE3.1: Patch splitting. Doc. ISO/IEC JTC1/SC29/ WG11 MPEG/M51602, Brussels, Jan. 2020.

[Dzi20b] Dziembowski A. et al. Immersive Video CE3.2: Temporal patch redundancy removal. Doc. ISO/IEC JTC1/SC29/WG11 MPEG/M51603, Brussels, Belgium, Jan. 2020.

[Fac18] Fachada S. et al. Depth image based view synthesis with multiple reference views for virtual reality. 3DTV-Conf, Helsinki, Finland, Jun. 2018.

[Fuj06] Fujii T. et al. Multipoint measuring system for video and sound – 100-camera and microphone system, IEEE Int. Conf. on Mult. and Expo, 2006.

[Ilo19] Ilola L. et al. New test content for Immersive Video – Nokia Chess. Doc. ISO/IEC JTC1/SC29/ WG11 MPEG/M50787, Geneva, Oct. 2019.

[Ilo20] Ilola L. et al. Improved NokiaChess sequence. ISO/IEC JTC1/SC29/WG04 MPEG VC/ M54382, Online, Jul. 2020.

[Isg14] F. Isgro et al. Three-dimensional image processing in the future of immersive media. IEEE Tr. on Circuits and Systems for Video Tech., 2014.

[ISO22] Standard ISO/IEC FDIS 23090-12. Information technology – Coded representation of immersive media – Part 12: MPEG Immersive video. 2022.

[Kro18] Kroon B. Test sequence ClassroomVideo. Document ISO/IEC JTC1/SC29/WG11 MPEG/M42415, San Diego, CA, USA, Apr. 2018.

[Mie20a] Mieloch D., Stankiewicz O., and Domański M. Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation. 2020 IEEE Access, vol. 8, pp. 5760-5776, 2020.

[Mie20b] Mieloch D. et al. [MPEG-I Visual] Natural Outdoor Test Sequences. Doc. ISO/IEC JTC1/ SC29/WG11 MPEG/M51598, Brussels, Jan. 2020.

[MPEG21] Test Model 11 for MPEG Immersive video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0142, Online, Oct. 2021.

[MPEG22] Common Test Conditions for MPEG Immersive video. Document ISO/IEC JTC1/SC29/ WG04 MPEG VC, N0169, Online, Jan. 2022.

[Mül11] Müller K. et al. 3-D Video Representation Using Depth Maps. 2011 Proceedings of the IEEE, vol. 99, no. 4, pp. 643-656, 2011.

[Nem10] O. Nemcic et al. Multiview Video Coding extension of the H.264/AVC standard. ELMAR, Zadar, Croatia, Sep. 2010.

[Sal18] Salahieh B. et al. Kermit test sequence for Windowed 6DoF Activities. Doc. ISO/IEC JTC1/ SC29/WG11 MPEG/M43748, Ljublana, Jul. 2018.

[Sta18] O. Stankiewicz et al. A free-viewpoint television system for horizontal virtual navigation. IEEE Tr. on Multimedia, 2018.

[Sul05] G. Sullivan and T. Wiegand. Video compression – from concepts to the H.264/AVC standard. Proceedings of the IEEE, vol. 93, 2005.

[Sul12] G. Sullivan et al. Overview of the High Efficiency Video Coding (HEVC) standard. IEEE Tr. on Circ. and Syst. for Vid. Tech., vol. 22, 2012.

[Tan12] Tanimoto M. et al. FTV for 3-D Spatial Communication. 2012 Proceedings of the IEEE, vol. 100, no. 4, pp. 905-917, 2012.

[Tao21] L. Tao et al. A Convenient and High-Accuracy Multicamera Calibration Method Based on Imperfect Spherical Objects. IEEE Tr. on Instrumentation and Measurement, vol. 70, 2021.

[Tec15] G. Tech et al. Overview of the Multiview and 3D Extensions of High Efficiency Video Coding. IEEE Tr. on Circ. and Syst. for Vid. Tech. 2016.

[Zha20] J. Zhang et al. Point Cloud Normal Estimation by Fast Guided Least Squares Representation. IEEE Access, 2020.

[Zhu19] S. Zhu et al. An improved depth image based virtual view synthesis method for interactive 3D video. IEEE Access, 2019.