Efficient Patch Merging for Atlas Construction in 3DoF+ Video Coding

Hyun-Ho KIM†, Sung-Gyun LIM†, Gwangsoon LEE††, Jun Young JEONG††, Nonmembers, and Jae-Gon KIM††, Member

SUMMARY The emerging three degree of freedom plus (3DoF+) video provides more interactive and deep immersive visual experience. 3DoF+ video introduces motion parallax to 360 video providing omnidirectional view with limited changes of the view position. A large set of views are required to support such 3DoF+ visual experience, hence it is essential to compress a tremendous amount of 3DoF+ video. Recently, MPEG is developing a standard for efficient coding of 3DoF+ video that consists of multiple videos, and its test model named Test Model for Immersive Video (TMIV). In the TMIV, the redundancy between the input source views is removed as much as possible by selecting one or several basic views and predicting the remaining views from the basic views. Each unpredicted region is cropped to a bounding box called patch, and then a large number of patches are packed into atlases together with the selected basic views. As a result, multiple source views are converted into one or more atlas sequences to be compressed. In this letter, we present an improved clustering method using patch merging in the atlas construction in the TMIV. The proposed method achieves significant BD-rate reduction in terms of various end-to-end evaluation metrics in the experiment, and was adopted in TMIV6.0.

key words: MPEG-I, immersive video, 360 video, VR, 3DoF+, TMIV

1. Introduction

Recently, with the increase of commercial interest in deploying virtual reality (VR) applications, 360 video has become popular as a new media type that gives immersive experience. In order to support immersive experiences with up to six degree of freedom (6DoF), MPEG is actively working on standardization of “Coded Representation of Immersive Media,” called MPEG-I [1], [2]. MPEG-I is a project initiated at the 116th MPEG meeting in October 2016 and is a standard package including architecture for immersive media, transmission format, audio/video compression, metadata, etc.

360 video, also called omnidirectional video, provides three degree of freedom (3DoF) which allows viewers to look around in all directions at a fixed viewing position in an observed scene. Compared to 3DoF video, 3DoF+ video is an enhanced 3DoF video which introduces limited translational movements in addition to rotation and provides viewers with enhanced immersive visual experience with interactive parallax feature. In other words, 3DoF+ video allows viewers to look around in all directions with slight body and head movements in a sitting position.

To support interactive parallax feature, limited alterations of view position in virtual space of 3DoF+ requires rendering the virtual view at any point of view. Depth based rendering is used in 3DoF+ for this purpose. Therefore, a 3DoF+ video consists of dozens of videos plus depths taken at many points of views within a limited space at the same time. These multiple videos of 3DoF+ video can be natural sequences captured by a real camera array or synthetic sequences projected by virtual camera array using computer rendering, and they have very large volume to improve rendering quality of high resolutions such as 4K or 8K, etc. Therefore, it becomes more important to efficiently compress 3DoF+ video to provide immersive visual experience.

2. Test Model for Immersive Video (TMIV)

In the MPEG-I Visual workgroup, to efficiently compress and transmit large volume of 3DoF+ video, various compression methods are being studied. They recently released the committee draft (CD) [3] and the test model for immersive video (TMIV6.0) [4]. The main approach on the compression of 3DoF+ video in the TMIV is to reduce the total amount of pixel data to be compressed by eliminating the redundancy between all available source views which are highly correlated in advance of compression. In other words, one or more views are selected as basic views, and then only unpredictable regions of the remaining views are cropped to rectangular regions called patches. The selected basic views and the patches are packed into one or more frames called atlases, which are low pixel rate 3DoF+ videos to be compressed using HEVC [5].

Figure 1 shows an overall architecture of the TMIV encoder [4]. A set of source views with different view positions are grouped into one or two groups, and classified into basic view or additional view within a group in a view labeling module firstly. After that, in a pruner, an additional view is re-projected to the basic view’s view position, and then the pixels that can be predicted from basic views regarded as redundant are removed. After one additional view is pruned, another additional view can be pruned by basic view and already pruned additional views as well. Finally, all the additional views are pruned into multiple patches each of which contains unpredictable regions remained after removing the
redundant parts of the views.

As shown in Fig. 2, for each view, the pruning process is applied at the frame level, which results in multiple remaining regions. Then, the remaining regions are temporally aggregated over one intra period in an aggregator. Then, the aggregated regions are clustered into multiple separated regions each of which is bounded by a rectangular box called patch. Therefore, in each rectangular patch, ‘valid region’ which contains real residual pixels that should be transmitted are included as well as ‘invalid region’ which contains redundant pixels.

After that, the TMIV generates metadata for each patch which describes its position of the top left corner and its size with width and height values. The patches generated from all additional views are packed into single or multiple frames called atlases in a frame level. This packing process sequentially packs each patch into atlas in a raster scan order to occupy the packing area as small as possible while allowing rotation of a patch and overlapping each other unless valid regions are invaded. The placement of patch begins from the top left of the atlas in the unit of 8 pixels with search for the best packing position.

This process of atlas generation is done in depth video, and texture atlas is separately generated according to the packing results of the depth atlas. Each time a patch is packed, packing position and rotation information is added to each patch’s metadata. As a result, such metadata indicates where in which view each patch came from when rendering additional views in the TMIV decoder. Finally, as shown in Fig. 1, video sub bitstreams of depth and texture atlases are multiplexed with metadata into an output bitstream.

3. Proposed Patch Merging

In the TMIV, each clustered region is generated as a rectangular-shaped patch one-by-one. Because of this process, when a large clustered region is made of one patch and even if small clustered regions are included in that patch, the small regions are generated as patches separately. As a result, these generated patches are packed into atlas separately and take up more space, even if the small patches can be included in another patch. In addition, the amount of metadata describing each patch’s information will be increased. Figure 3 shows such example of patch generation in the pruned view.

Furthermore, in the TMIV, there is a process of splitting generated patch which is containing a lot of invalid regions inside to reduce the space occupied in the atlas[6]. However, if the patch is split and packed in a separate place in atlas, it may cause visual artifacts in the section where the patch is split when rendering the viewport. Figure 4 (Left) shows that the patches generated by TMIV are unnecessarily split into small pieces marked as yellow circle areas.

To address these problems, we prevent the generation of unnecessary patches and their associated metadata by
merging patches into one patch if there are small patches within in the generated patch’s internal area as the example shown in Fig. 5. By the way, as shown in Fig. 5 (cluster C), when one patch is not completely included in a parent patch that contains it, and spans multiple patches, merging is not applied. This is because when the patch is cut off at patch boundaries due to merging, visual artifacts may be occurred at that position.

The proposed method is implemented in the clustering step of the TMIV encoding process, and which consists of two parts of processes as shown in the flowchart of Fig. 6. The first process generates a separate patch for valid pixels placed on the boundary of the large patch. In detail, when one patch is generated, we search rectangular boundaries of that patch to find any valid pixels not yet clustered. If such a pixel is detected, the cluster containing that pixel is generated as a separate patch. The second process merges a patch which is completely contained within the large patch into that large patch. In other words, when valid pixels are searched within the patch area, if there exist valid and not clustered pixels, indicate the pixels as valid ones and merge with the patch which currently being checked instead of generating separate patch as in the existing method.

As a result, the proposed method can improve coding efficiency by merging regions in the same additional view which are spatially close and reduce the amount of metadata describing patch information. In addition, the proposed method suppresses the unnecessary patch splitting and consequently reduce tiny pieces of patch generated by splitting. Without the patch merging, it is likely that small patches spatially distant or come from different views are placed in the vacant area of a larger patch. This may not only decrease the coding efficiency of the atlas but also degrade the subjective quality of the final rendered view.

4. Experimental Results

To evaluate the performance of the proposed patch merging in terms of coding efficiency, the end-to-end performance which compares the source views and rendered views synthesized at the same view positions was checked. In the experiments, quality evaluations were carried out with various metrics such as weighted spherical (WS)-PSNR [7], video multi-method assessment function (VMAF) [8], and immersive video (IV)-PSNR [9] according to the common test conditions (CTCs) for immersive video [10].

Table 1 shows the end-to-end coding performance of 3DoF+ videos with the atlas constructed by the proposed method in comparison with TMIV as an anchor in terms of Bjontegaard-Delta rate (BD-rate). The proposed method was implemented on the TMIV5.0, and the HEVC test model HM 16.16 [11] was used to compress the constructed atlas according to the CTCs. It can be observed that there is a significant coding gain in the proposed method over the anchor of TMIV, on average 1.4% BD-rate savings on luma for high-bitrate.

Table 2 shows the amount of metadata generated by the proposed method compared with the anchor. It is observed that the amount of metadata is decreased 11.3% in the proposed method on average.

Table 3 shows the comparison of the computation time for atlas generation. On average, the computation time was increased about 7.5% in the proposed method. Especially
in natural sequences (SD, SE, SL), the computation time increase is quite high. This is because there are many small patches in natural sequences, which results in the increase of the occurrences of searching and merging processes than those of CG sequences.

Table 3

| Sequence | Anchor (sec) | Proposed (sec) | Ratio   |
|----------|--------------|----------------|---------|
| SA       | 10745.822    | 10721.296      | 99.8%   |
| SB       | 15454.73     | 15545.728      | 100.6%  |
| SC       | 6382.234     | 6360.58        | 99.7%   |
| SJ       | 4606.000     | 4594.977       | 99.8%   |
| SD       | 2329.31      | 2742.43        | 117.7%  |
| SE       | 1654.163     | 1950.056       | 117.9%  |
| SL       | 1164.116     | 1365.056       | 117.3%  |

Average 107.5%

Figure 7 and 8 shows the comparison of subjective quality between the anchor and the proposed method with viewports of “SD” and “SL” sequences. In Fig. 7, some artifacts which are shown in the anchor viewport are disappeared in the viewport that rendered with the proposed method. Also, in Fig. 8, we can find that artifacts around the fencer in the anchor viewport are significantly reduced in the proposed viewport. As shown in the example viewports, it is noted that the proposed method significantly improves visual quality as well as objective coding gain shown in Table 1.

5. Conclusions

In this letter, we proposed the application of patch merging in the atlas construction to improve coding efficiency of 3DoF+ video by reducing the number of patches to be packed into atlases which results in enhancing the correlation of atlases. The experimental results show that the proposed method gives significant coding gain in terms of BD-rate saving and considerably improves visual quality as well.

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).

References

[1] “MPEG-I use cases for omnidirectional 6DoF, windowed 6DoF, and 6DoF,” ISO/IEC JTC1/SC29/WG11, N16768, April 2017.
[2] M. Wien, J.M. Boyce, T. Stockhammer, and W.-H. Peng, “Standardization Status of Immersive Video Coding,” IEEE J. Emerg. Select. Topics Circuits Syst., vol.9, no.1, pp.5–17, March 2019.
[3] J. Boyce, R. Dore, and V. Vadakital, “Text of ISO/IEC CD 23090-12 MPEG immersive video,” ISO/IEC JTC1/SC29/WG11, N19482, Jan. 2020.
[4] B. Salahieh, B. Kroon, J. Jung, and M. Domänski (Eds.), “Test model 6 for immersive video,” ISO/IEC JTC1/SC29/WG11, N19483, July 2020.
[5] B. Wang, Y. Sun, and L. Yu, “Low Pixel Rate 3DoF+ Video Compression Via Unpredictable Region Cropping,” Proc. 2019 Picture Coding Symp., Nov. 2019.
[6] A. Dziembowski, D. Mieloch, A. Grzelka, J. Stankowski, M. Domainski, and G. Lee, “Immersive video CE3.1: Patch splitting,” ISO/IEC JTC1/SC29/WG11, m51602, Jan. 2020.
[7] “WS-PSNR software manual,” ISO/IEC JTC1/SC29/WG11, N18069, Oct. 2019.
[8] VMAF: Perceptual video quality assessment based on multi-method fusion, [Online] available at: https://github.com/Netflix/vmaf
[9] A. Dziembowski, “Software manual of IV-PSNR for immersive video,” ISO/IEC JTC1/SC29/WG11, N18709, July 2019.
[10] J. Jung, B. Kroon, and J. Boyce, “Common test conditions for immersive video,” ISO/IEC JTC1/SC29/WG11, N19214, April 2020.
[11] HM reference software, [Online] available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/