

V3C: V-PCC Encoder improvement for empty partition

Tianyu Dong Qiong Jia Euee S. Jang

Hanyang University

Abstract

In this paper, we proposed a method for Video-based point cloud compression reference software TMC2 encoder with an option for empty partitions in point cloud encoding. This encoder option allows tile initialization and process for an empty partition. The proposed method provides the TMC2 encoder the robustness to process dynamic point clouds.

1. Introduction

V-PCC is a Visual Volumetric Video-based Coding (V3C) technology of MPEG-Immersive standard. As more interest is seen in V-PCC, a variety of point cloud data are being tested in TMC2. Some of the point data are not preprocessed, which may be out of the range of the support of TMC2. One of the encoding methods, encoding of range of interest (ROI, also called as partitioning), that is a supported functionality by the 3DGH working group. However, due to the limitation of the current implementation, when a ROI has no point cloud on a certain frame the current TMC2 software cannot conduct the ROI encoding process.

For example, Figure 1. shows a capture of class-C data dancer. These dynamic point clouds have a dynamic value on the y-axis. When this data is evenly partitioned into eight parts on the y-axis, as shown in Table I. The last partition No. 7, where $y \in [1,792, 2,047]$, may have no points on the 25-th and 26-th frames. As shown in the figure, the number of vertices of partition 7 in frame 22 is 25,250, and that of partition 7 in frame 26 is none. Under the partitioning mode (i.e., *enablePointCloudPartitioning* = 1 *tileSegmentationType*

= 1), the corresponding tile cannot be generated for this partition. In the current TMC2 encoder implementation, it was forced to exit when the tile number did not match the partition number.

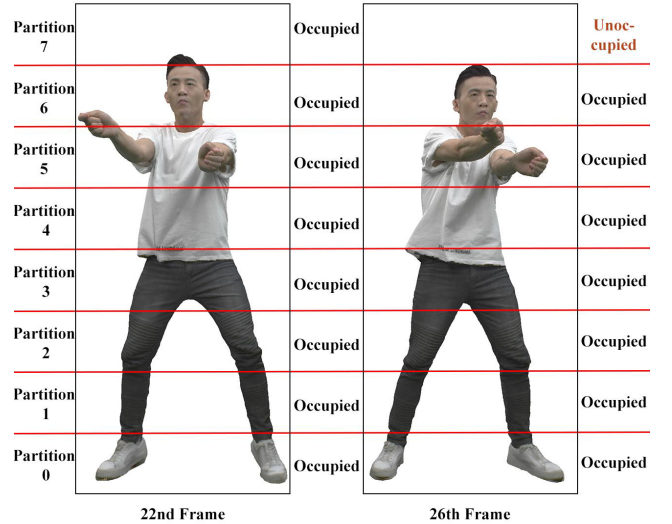


Figure 1. A captured image of frames 22 and 26 with their partition occupancy information.

Table I. ROI boundary box

	ROI #0	ROI #1	ROI #2	ROI #3	ROI #4	ROI #5	ROI #6	ROI #7
$x \in$	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047

y∈	0 ~ 255	256 ~ 511	512 ~ 767	768 ~ 1,023	1,024 ~ 1,279	1,280 ~ 1,535	1,536 ~ 1,791	1,792 ~ 2,047
z∈	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047	0 ~ 2,047

In this contribution, we proposed a TMC2 encoder option for allowing empty partitions in point cloud encoding. In our proposed method, the encoder can generate a tile as the previous frame or let the encoder generate the smallest allowed tile on the atlas. The result shows that this method provides an option for robust encoding support from the encoder and does not affect current common test condition (CTC) anchor results.

2. Proposed method

The TMC2 version 18.0 encoder has the packing process shown in Figure 2. When the *packingStrategy* of 1 is used, each partition will have its tile mapped inside *PackFlexibleMultipleTile()* and *spatialConsistencyPackFlexibleMultipleTile()* functions. After the tile-to-partition mapping process, the tile height and width are given where the tile position on the canvas has also been initialized. However, during the packing process in the current implementation, the empty partition (i.e., no points inside the partition) is not allocated to a correspondent tile. Furthermore, the tile size and position also need to be initialized.

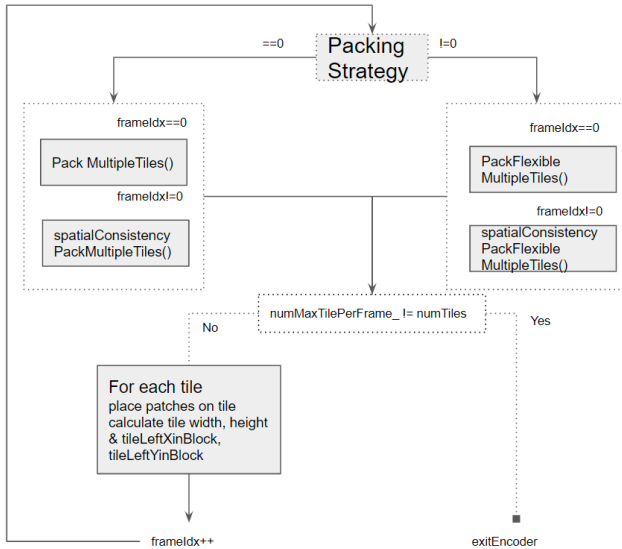


Figure 2. Anchor process diagram

Our proposal to the issue to provide partition and tile mapping and tile initialization in the empty partition as shown in Figure 3. These modifications of improved empty tile processing in packing can be categorized into three parts.

- A. *PackFlexibleMultipleTiles()* & *spatialConsistencyPackFlexible()* multiple tiles packing
 - a) Mapping the empty tile to the empty partition
 - b) Use the non-empty tile-partition mapping information from the last frame to initialize the mapping information
 - c) Use “last processed tile-partition mapping information and plus 1” to initialize the empty tile in 1st frame
- B. *generateTilesFromImage()* empty tile initialization
 - a) Use the last frame’s tile width/height for empty tile
 - b) Set the empty tile in 1st frame with the minimum possible tile size (64 as tile height and frameWidth/*numTilesHor* as tile width) and set leftTOP x and y of the tile according to the last tile
- C. Inside of *performDataAdaptiveGPAMethod* set *badPatchCount=true* when processing an empty partition.

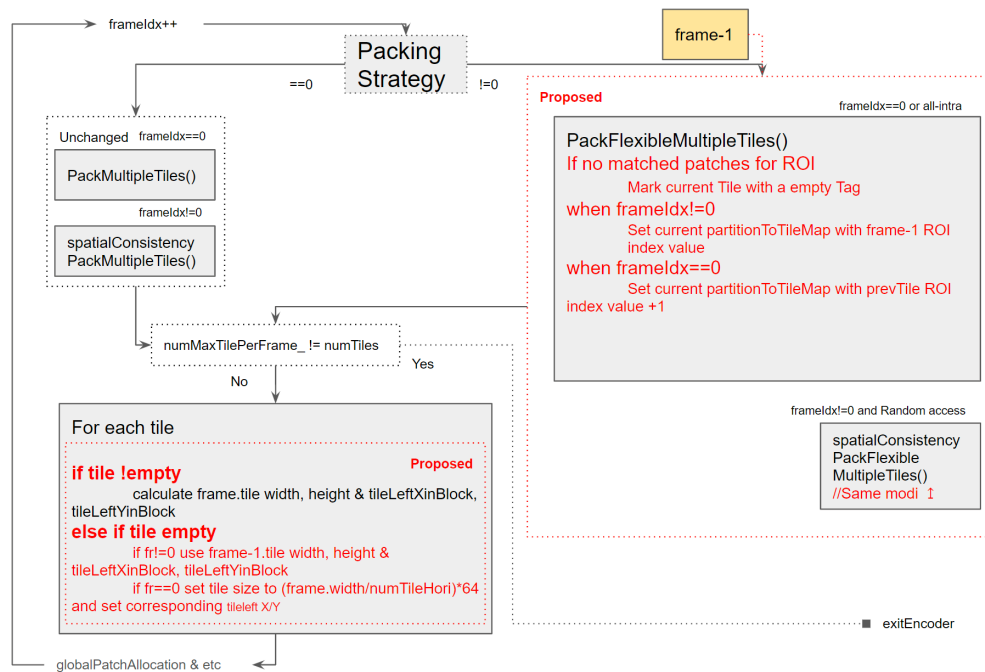


Figure 3. Diagram of proposed method.

Currently, when the *packingStrategy* of 0 is used, in function *generateTilesFromImage()*, the current design of *packMultipleTiles()* and *spatialConsistencyPackMultipleTiles()* does not map partitions to tiles. This will cause the encoder to exit always [1]. Furthermore, when the *globalPatchAllocation* method was set to two, the *doGlobalTetrisPacking()* could not normally work and it also caused a memory leak. Due to these two reasons, we have our method tested under *packingStrategy* = 1 and *globalPatchAllocation* = 1. The software code level difference could be found in the attached archive file of this contribution.

3. Result

This proposed method is implemented on TMC2 version 18.0 [2] with the anchor common test condition and default ROI condition [3]. We checked the results with empty ROI condition and CTC (non-ROI, ROI) condition.

a. Empty ROI result

The following capture images Figure 4 and Figure 5 shows the texture atlas of frame 22 and 26. On the right bottom of the atlas frame No. 42 shows the hair patches of ‘dancer’ which belong to the partition No. 7. On atlas frame

No. 50, the tile which mapped to partition No.7, is also generated.

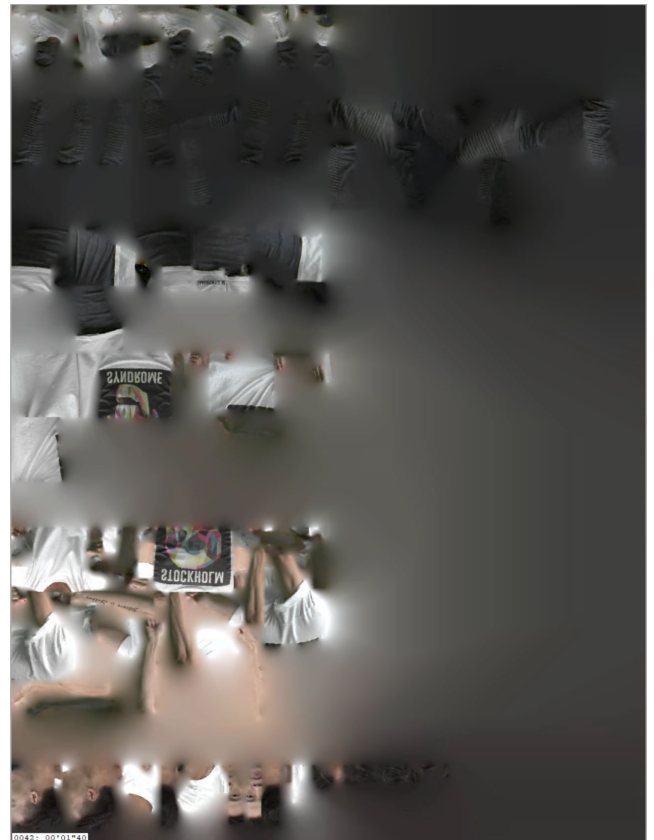


Figure 4. Texture atlas of frame No.22



Figure 5. Texture atlas of frame No.26

b. Conformance Results

Table II. shows the conformance test result of the proposed method versus anchor. The result shows that both CTC non-ROI result and 4 ROI default settings are matched. We also measured the encoding time differences between anchor and proposed method. Result shows in Table III are within 0.3%, which does not show significant differences.

Table II. Conformance test

Class	Sequence	With ROI		Non ROI	
		Random-Access	All-intra	Random-Access	All-intra
cat2-A	8ivfbv2_loot_vox10	checked	checked	checked	checked
	8ivfbv2_redandblack_vox10	checked	checked	checked	checked
	8ivfbv2_soldier_vox10	checked	checked	checked	checked
	queen	checked	checked	checked	checked
cat2-B	8ivfbv2_longdress_vox10	checked	checked	checked	checked
cat2-C	basketball_player_vox11	checked	checked	checked	checked
	dancer_player_vox11	checked	checked	checked	checked

Table III. Encoding time difference

Class	Sequence	With ROI		Non ROI	
		Random-Access	All-intra	Random-Access	All-intra
cat2-A	8ivfbv2_loot_vox10	checked	checked	checked	checked
	8ivfbv2_redandblack_vox10	checked	checked	checked	checked
	8ivfbv2_soldier_vox10	checked	checked	checked	checked
	queen	checked	checked	checked	checked
cat2-B	8ivfbv2_longdress_vox10	checked	checked	checked	checked
cat2-C	basketball_player_vox11	checked	checked	checked	checked
	dancer_player_vox11	checked	checked	checked	checked

4. Conclusions

Based on the result, the proposed method provides the TMC2 encoder the robustness to process point clouds and it does not change the behavior of the CTC result. Currently, this proposed method is under consideration as an encoder option to be includes into reference software. Future works such as robust ability of the encoder on processing non-CTC sequence could be studied.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00452, Development of Adaptive Viewer-centric Point cloud AR/VR(AVPA) Streaming Platform).

Reference

- [1]. "ROI(PointCloudPartitioning) packing issue (packingStrategy==0)", ISO/IEC JTC1/SC29/WG7, <http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2/-/issues/189>, Sep 19, 2022.
- [2]. "V-PCC Test Model v18", ISO/IEC JTC1/SC29/WG7, <http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2>, May 30, 2022.
- [3]. "Common Test Conditions for V-PCC", ISO/IEC JTC1/SC29/WG7 MDS19613_WG07_N00038, OnLine, October 2020.