MPEG Point Cloud Compression First Standard for Immersive Media

Ohji Nakagami[1]

[1]Sony Corporation
Keywords: Point cloud, Compression, MPEG, Standard

ABSTRACT

This paper introduces recent MPEG activity on Point Cloud Compression (PCC) standard planned to be released in 2020 as a part of ISO/IEC 23090 series. The paper explains two complementary technologies, Video-based PCC and Geometry-based PCC. The coding algorithm, the compression performance, and the use-cases are discussed.

1 INTRODUCTION

Recent progress in visual capture technology enables the digitization of a real 3D world. Point Cloud (PC) is one of the major 3D visual data representation formats. In an AR/VR application, multi-camera array generates a dynamic PC as a 3D video [8]. For autonomous driving, a LiDAR sensor on a vehicle captures the surrounding environment as a time-varying PC [9]. For a cultural heritage digital archive, an object is scanned with a 3D sensor for a high-resolution static PC [10].

A PC is composed of a collection of points in a 3D space. Each point has the geometry position in the 3D space as well as the associated attribute information (e.g. color, reflectance). A point in a PC can be considered as a 3D extension of a pixel in the 2D video, and is usually called as a voxel. While the new format gives lots of flexibility for the content representation, the raw data size becomes huge for a realistic 3D quality.

MPEG has explored PCC technology since 2014 and started the standardization since 2017 [1][2][3]. After the CfP (Call for Proposal) was issued in January 2017, the first Test Models were developed in October 2017. Over the last two years, technical polishing work is ongoing. The PCC standard is planned to be finalized in 2020.

This paper introduces two compression technologies for the upcoming PCC standard; Video-based PCC and Geometry-based PCC. The two standards are planned to be ISO/IEC 23090-5 and -9, respectively.

The goal of this paper is to give an insight into the MPEG PCC. The paper discusses the current compression methods, their performances, and the use cases.

2 VIDEO-BASED PCC (V-PCC)

2.1 Overview

As the name indicates, the concept of V-PCC is to leverage the existing 2D video coding technology for a PCC.

Figure 1 shows a simplified V-PCC encoder architecture. In the encoder, the input PC frames are mapped into three 2D videos; geometry, texture, and occupancy videos. The videos are compressed by a conventional video codec (e.g. AVC, HEVC). The decoder basically conducts an inverse process of the encoder.

In the encoder, the 3D PC frames to 2D video mapping is conducted with a patch generation process. It is a 2D approximation of a local flat surface in the PC. In the V-PCC Test Model, such flat area is derived by merging the points having similar normal vectors.

To generate a patch, the encoder projects the points on a flat surface onto a face of the local bounding box. The bounding box face can be rotated in 45 degree to improve the projection quality.

The point distance from the box face is mapped as a geometry video. The corresponding point color is mapped as a texture video. The patch size is defined by the box face. The patch location is aligned between the geometry and the texture video to simplify the decoder implementation.

The patch position in the mapped 2D videos is signaled as an occupancy video. With the occupancy, the patch boundary in the texture video can be blurred to improve the coding performance. The occupancy video can be scaled down to reduce the occupancy bitstream size.

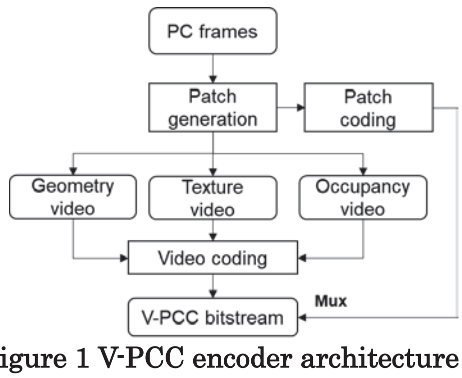

Figure 1 V-PCC encoder architecture

For the better 3D to 2D mapping quality, two or more layers can be used for the depth and the texture videos. The overlapping points from the bounding box face are assigned to different video frames as the layers so that the projection quality improves.

Lossless compression is also supported in V-PCC.

The points in a sparse area or in an occluded area are mapped to a special patch where the absolute 3D position and the attribute values are signaled.

Finally, the three video bitstreams are combined into a V-PCC bitstream with a header. The mapping information is also multiplexed into the bitstream as a patch data.

2.2 V-PCC coding performance

Figure 2 shows coding performance of the initial and the latest V-PCC Test Models (TM) [6] for a sequence "Longdress" in the PCC CTC (Common Test Condition) [7][11].

The sequence is a 30fps,10-seconds human point cloud video. Each PC frame has roughly 1 million points with RGB color. The raw data size is about 1.5 Gbps.

With the latest TM version 6, the 10-15 Mbps bitstream provides a good visual quality over the 70 dB geometry PSNR while the initial TM v1 needed 30-40 Mbps for the equivalent quality.

As V-PCC relies on the existing video codec for the compression performance, the improvement mostly came from the encoder optimization such as a patch generation, a patch allocation among frames in time consistency manner, or a motion estimation guidance for the video encoder from the patch generator.

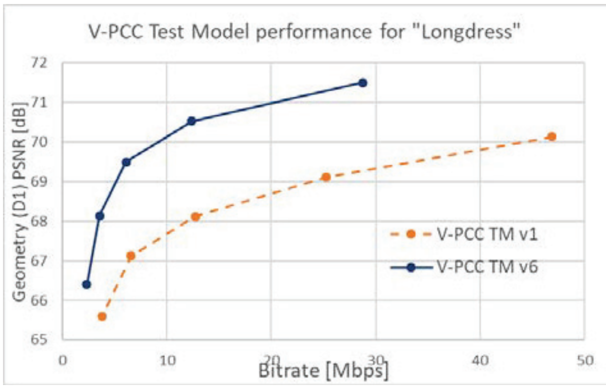

Figure 2 V-PCC coding performance

2.3 Use cases

AR/VR

A PC video provides a 6DoF (Degree of Freedom) user experience. One can see the content from any view point. The characteristic is familiar with AR/VR applications.

On a smartphone, a video decoder is available as well as an integrated GPU [15]. It is reported that the V-PCC decoder is implemented on the existing smartphone for a real time playback. Combined with an AR framework (e.g. ARCore, ARkit), the decoded PC video is overlayed on a real world [14].

Telecommunication

V-PCC enables the transmission of a PC video over the network with the high compression performance. Once a real-time encoder is implemented, an immersive communication is achieved [16]. With HMD (Head Mount Display), one can interact in the virtual world. This can be extended to tele-presence applications.

3 GEOMETRY-BASED PCC (G-PCC)

3.1 Overview

Figure 3 shows a simplified encoder diagram for G-PCC. In G-PCC, the geometry is encoded first, then the associated attribute is encoded second if needed. The order is same for the decoding process.

For the geometry encoding, the input PC is voxelized as a pre-process. The positions are quantized to a fixed-point representation. The quantization is conducted for a bounding box that covers the input PC with the given bitdepth. The input points may be merged in the quantization process.

The voxelized PC is converted to Octree in a lossless manner. An Octet, which represents an Octree node occupancy state in 1 byte, is compressed by the entropy coder considering the correlation with the neighbor Octet [19].

Trisoup is an optional coding tool. The points on the object surface is approximated by a series of mesh. It improves the subjective quality in lower bitrate [17].

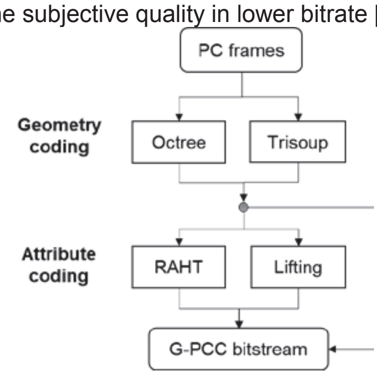

Figure 3 G-PCC encoder architecture

For the attribute coding, two coding options are provided; RAHT and Lifting.

RAHT is a Haar based hierarchical transform [17]. Instead of the fixed transform weight, RAHT changes the weight depending on the point distribution. Higher weight is applied to the dense area point transform. RAHT shows better coding efficiency than the non-adaptive transform.

Lifting is a LoD (Level of Detail) based prediction approach [18]. The input points are grouped into LoDs depending on the spatial distance between the points. The prediction is conducted based on the group. Higher weight is applied to the predictors from the near point. Controlling the prediction algorithm, the Lifting scheme supports lossless to lossy compression.

3.2 G-PCC coding performance

The latest G-PCC TM v6 [5] coding performance is compared to an open source codec Google Draco [4] in MPEG [12].

Figure 4 shows the geometry coding performance for a PC frame "Longdress" in CTC. For the dense PC, the

G-PCC bitstream size is 30% of the Draco in the equivalent quality. In the lossless compression, the G-PCC compresses in 1 bit per input point (bpp) while the Draco does in 4 bpp.

While the geometry coding method is similar (Octree for the G-PCC and KDtree for the Draco), the coding performance in G-PCC gains by the entropy coding algorithm utilizing the neighbor point distribution.

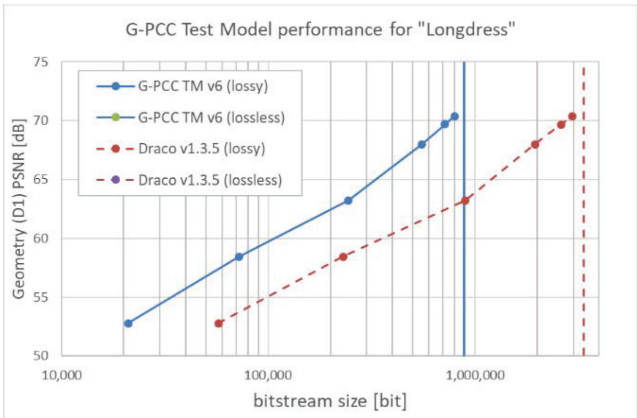

Figure 4 G-PCC coding performance

3.3 Use cases

Autonomous vehicle

In autonomous driving, a vehicle implements multiple visual sensors to recognize the surrounding environment. G-PCC is capable of compressing a sparse Lidar sequence. It helps to improve the dataflow bottleneck with a light and efficient compression algorithm.

World heritage

A historical architecture or object can be preserved once it is digitized into a virtual world. Lots of high-quality PCs are generated by academic/research projects. Laser range scanner or SfM (Structure from Motion) techniques are often employed in the content generation process.

G-PCC can store a high-resolution data with a lossless to a near-lossless quality. Based on the CTC result, such PC is compressed into 1/10 to 1/20 of the original data size.

4 CONCLUSION

Recent technology progress opens a door for a capture-based 3D visual world. It is the convergence of a high-quality 2D video and a high-performance 3D Computer Graphics technology.

In MPEG, experts from the two different compression background (i.e. Video and CG) jointly discuss toward the PCC standardization.

It is expected that the two PCC standards provide competitive solutions to a new market, satisfying various application requirements or use cases.

For the interested reader, the MPEG PCC website [13] provides more informative resources as well as the latest TM software.

REFERENCES

[1] MPEG 3DG, "Requirements for Point Cloud Compression," ISO/IEC JTC 1/SC 29/WG 11 N16330, 2016.

[2] MPEG 3DG, "Use Cases for Point Cloud Compression (PCC)," ISO/IEC JTC 1/SC 29/WG 11 N16331, 2016.

[3] MPEG 3DG, “Call for proposals for point cloud compression v2,” ISO/IEC JTC 1/SC 29/WG 11 N16763, 2017.

[4] Google, “Google Draco - 3D data compression,” https://github.com/google/draco.

[5] MPEG 3DG, “G-PCC Test Model v6,” ISO/IEC JTC 1/SC 29/WG 11 N18473, 2019.

[6] MPEG 3DG, “V-PCC Test Model v6,” ISO/IEC JTC 1/SC 29/WG 11 N18475, 2019.

[7] MPEG 3DG, “Common Test Conditions for PCC,” ISO/IEC JTC 1/SC 29/WG 11 N18474, 2019.

[8] P. Chou et al, “8i Voxelized Full Bodies – A Voxelized Point Cloud Dataset,” ISO/IEC JTC 1/SC 29/WG 11 m40059, 2017.

[9] D. Flynn and S. Lasserre, “PCC Cat3 test sequences from BlackBerry|QNX,” ISO/IEC JTC 1/SC 29/WG 11 m43647, 2018.

[10] C. Tulvan and M. Preda, “Point Cloud compression for cultural objects,” ISO/IEC JTC 1/SC 29/WG 11 m37240, 2015.

[11] S. Schwarz et al, “[V-PCC] On the status of V-PCC standardization,” ISO/IEC JTC 1/SC 29/WG 11 m48266, 2019

[12] D. Flynn and S. Lasserre, “G-PCC EE13.4 Draco performance comparison,” ISO/IEC JTC 1/SC 29/WG 11 m49383, 2019

[13] MPEG 3DG, “MPEG Point Cloud Compression,” http://www.mpeg-pcc.org/

[14] M. Pesonen and S. Schwarz, “On standard demo implementation,” ISO/IEC JTC 1/SC 29/WG 11 m46074, 2019

[15] J. Ricard et al, “Mobiles device decoder and considerations for profiles definition,” ISO/IEC JTC 1/SC 29/WG 11 m49238, 2019

[16] R. Mekuria et al., “Point Cloud Codec for Tele-immersive Video,” ISO/IEC JTC 1/SC 29/WG 11 m38136, 2016

[17] P. Chou et al, “Point Cloud Compression using a Blockable Geometry Representation and Region Adaptive Hierarchical Transform,” ISO/IEC JTC 1/SC 29/WG 11 m41646, 2017

[18] K. Mammou et al, “Video-based and Hierarchical Approaches Point Cloud Compression,” ISO/IEC JTC 1/SC 29/WG 11 m41649, 2017

[19] S. Lasserre and D. Flynn, “Neighbour-dependent entropy coding of occupancy patterns in TMC3,” ISO/IEC JTC 1/SC 29/WG 11 m42238, 2017