**ISO/IEC JTC 1/SC 29/AG 03 N**Graphical user interface, text

Description automatically generated**0111**

**ISO/IEC JTC 1/SC 29/AG 03  
MPEG Liaison and Communication   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** White paper on G-PCC

**Status:** Approved

**Date of document:** 2023-04-30

**Source:** ISO/IEC JTC 1/SC 29/AG 03

**No. of pages:** 7 (with cover page)

**Email of Convenor:** kyuheonkim@khu.ac.kr

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29ag3>

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/AG 03 MPEG Liaison & Communication**

**ISO/IEC JTC 1/SC 29/AG 03 N0111**

**April 2023, Antalya**

|  |  |
| --- | --- |
| **Title** | **White paper on G-PCC** |
| **Authors** | **Ohji Nakagami, Sebastien Lasserre, Sugio Toshiyasu, Marius Preda** |
| **Source** | **AG 03, MPEG Liaison and Communication** |
| **Status** | **Approved** |
| **Serial Number** | **22804** |

**Abstract**

The Geometry-based Point Cloud Compression (G-PCC) standard, standardized in ISO/IEC as International Standard 23090-9, is the most recent international point cloud coding standard.

This white paper provides an overview of the ISO/IEC 23090-9 Information technology — Coded representation of immersive media — Part 9: Geometry-based Point Cloud Compression (G-PCC) standard. The G-PCC standard provides a general framework for compressing different types of point clouds and can be handled by G-PCC profiles.

**Introduction to the G-PCC standard**

The MPEG committee (ISO/IEC JTC 1/SC 29/WG 7) is developing the MPEG-I standard for encoding, encapsulation, and delivery of immersive media. Part 9 of this standard, Geometry-based Point Cloud Compression (G-PCC), provides a standard for coded representation of the point cloud media.

Point cloud may be created in various manners. Recently, 3D sensors such as Light Detection And Ranging (LiDAR) or Time of Flight (ToF) devices are widely used to scan dynamic 3D scenes.

To precisely describe 3D objects or real-world scenes, point clouds come with a large set of points in 3D space with geometry information and attribute information. The geometry information represents the 3D coordinates of each point in the point cloud; the attribute information describes the characteristics (e.g., color and reflectance) of each point. Point cloud requires a large amount of data, bringing huge challenges to data storage and transmission.

MPEG 3D Graphics Coding group (3DG) announced the Call for Proposal on point cloud compression in January 2017[1]. This project targets delivering efficient compression technologies for different types of point clouds that are acquired with different methods.

According to the contributions from industry and academia, the standardization activities progressed to two distinct technical solutions: V-PCC (ISO/IEC 23090 Part 5) which is suitable for coding dense content; and G-PCC (ISO/IEC 23090 Part 9) which is efficient for relatively sparse data. A detailed history of the early standard development process can be found in [2].

**G-PCC CODING DESIGN**

G-PCC standard provides a general compression framework for different kinds of point cloud as the examples shown in Figure 1.

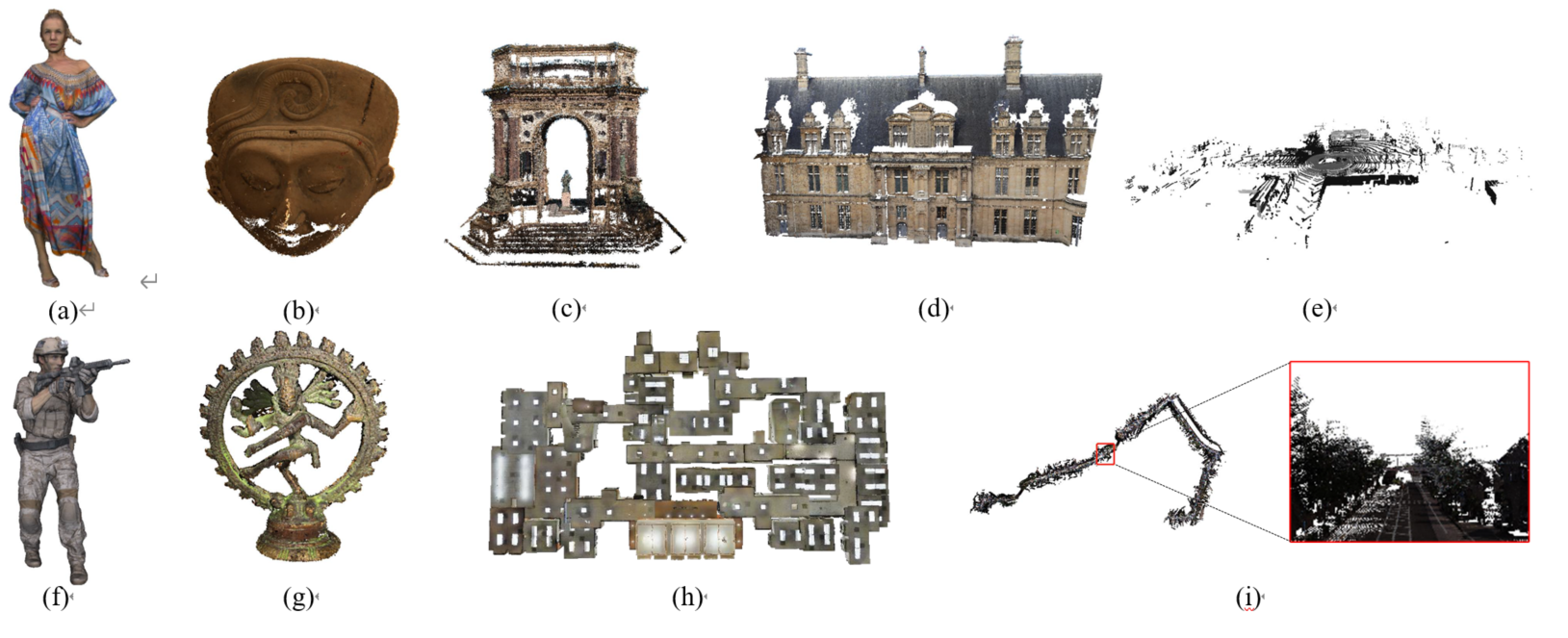


Figure Samples from the MPEG PCC dataset. (a) Longdress, (b) Egyptian mask, (c) Arco Valentino Dense, (d) Facade, (e) Ford, (f) Soldier, (g) Shiva, (h) Standford and (i) Citytunnel.

G-PCC encodes the geometry and attribute information of a point cloud separately. The geometry information is coded first while the attribute compression depends on the reconstructed geometry. The codec architectures of the encoder and decoder are illustrated in Figure 2 covers the major components in the standard.

Regarding the geometry, two coding tools are provided, Octree and Predtree (Predictive tree). The Octree coding provides general compression method while the Predtree coding intends to provide low-delay applications.

There are 3 attribute coding methods in G-PCC: Region Adaptive Hierarchical Transform (RAHT) coding, interpolation-based hierarchical nearest-neighbour prediction (Predicting Transform), and interpolation-based hierarchical nearest-neighbour prediction with an update/lifting step (Lifting Transform).

The detail information of the coding tools can be found in [2].

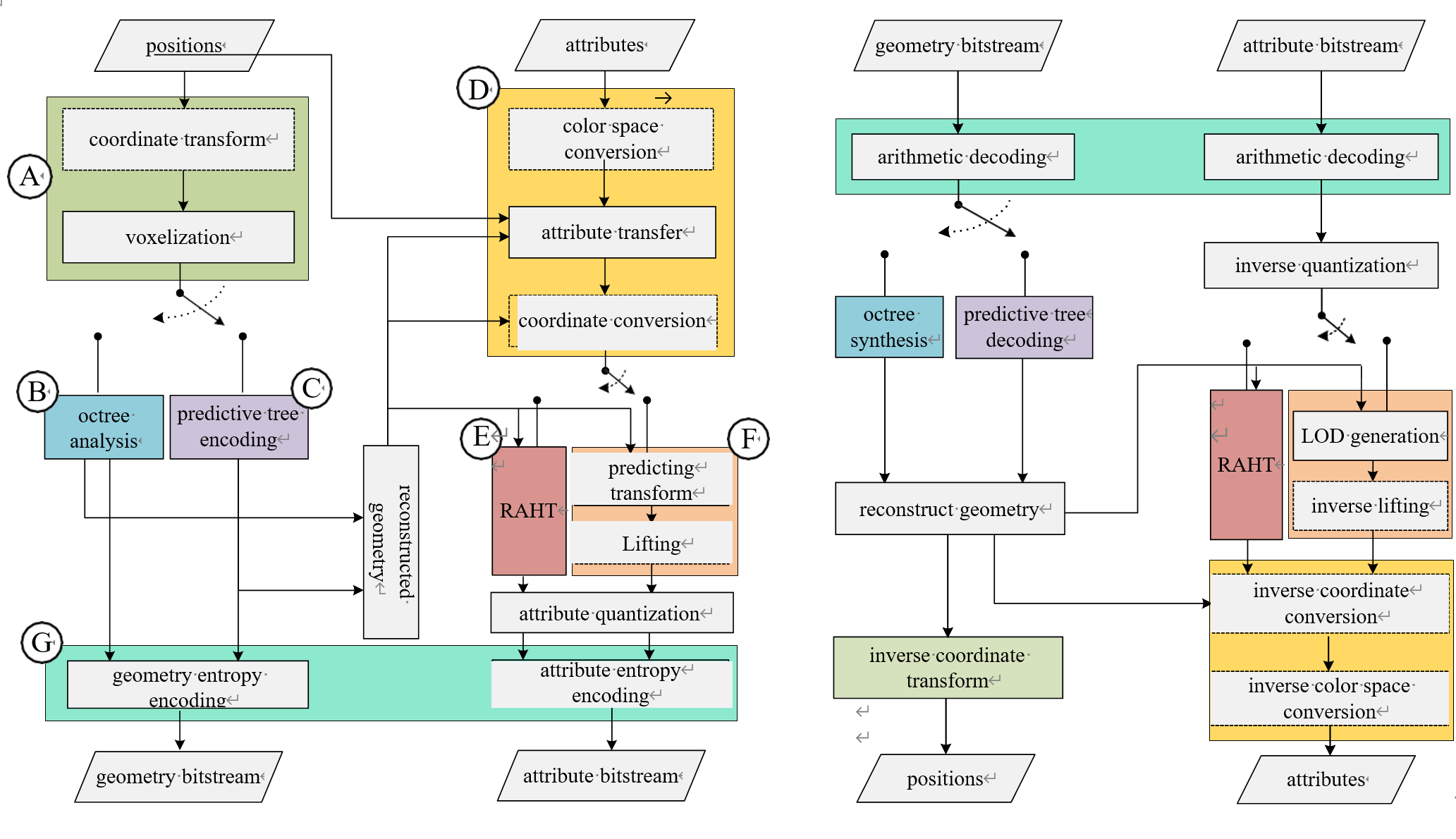


Figure G-PCC codec architecture: (a) encoder and (b) decoder. Key components indexed from A to G are geometry pre-processing, octree coding, predictive coding, attribute pre-processing, RAHT, predicting/lifting and arithmetic coding, respectively. Modules in dotted boxes are optionally performed.

**Functionalities**

* Parallel processing capability

As a recent coding standard, G-PCC supports parallel processing both in encoder and decoder with a slice coding. A single point cloud frame can be divided into multiple data unit as a slice to allow parallel processing within a single frame. A slice is a set of points comprising one geometry data unit and zero or more attribute data units. A group of slices may be identified by a common tile identifier. A tile inventory contains metadata that defines zero or more spatial regions. Each spatial region is identified by either an implicit or explicit tile ID. Each slice that belongs to the same tile may be allowed to have dependencies while the slices with different indices of the tile should be independent.

From one slice to another, there should be a process to re-initialize the state of the entropy coding engine to the default value at the end of each slice for parallelism. The initialization of the entropy context table will cause a decrease in encoding performance. However, if slice is implemented for data segmentation purposes in low latency application and not for parallelism, an entropy context continuation flag is introduced to save the context probability table at the end of slice and set it back at the beginning of the next slice for the sustainable use of the same context probability. This method will be able to improve the coding performance for slices.

* Quality control

In G-PCC, the geometry quality and attribute quality can be adjusted independently.

For geometry, the quality adjustment can be achieved by the non-normative voxelization in the pre-processing stage. The voxelized input can be further quantized with normative qual- ity control tools. For octree case, the in-tree quantization is used to adjust the voxel resolution for each spatial region. It cuts leaf nodes from specific depth with controlled granularity. For predicting geometry case, the geometry quality is adjusted by controlling the accuracy of the residual. These adjustments can be made independently for each data unit.

As for attribute, except for the slice-based quality control enabled by the slice and tile functionality, there are two alternatives to control the quality of the reconstructed signals: the layer-based quantization control and the region-wise quantization control. The layer-based quality control enables the quality adjustment for each refinement layer in the LoD scheme and the octree layer in the RAHT scheme. More precisely, a layer QP offset parameter is signalled in each attribute slice header. The effective QP value for each layer in a particular slice is obtained by adding the layer QP offset of that layer with the QP of that slice. Additionally, as the visual importance may vary across the 3D space, the quantization parameter can be further adjusted in a region-wise manner according to the point location in slice. For points within the specified box region where QP value is desired to be changed, a region QP offset is indicated and is added on top of the the slice QP. In conclusion, the region-based adjustment provides the quality control in spatial domain while the layer-based adjustment provides quality control in frequency domain.

* Scalability

The G-PCC supports regional scalability and spatial scalability. Regional scalability is the partial access to a region. This can be done with the slice and tile features. Spatial scalability is the access to data with small spatial resolutions. If octree coding is applied to geometry, the point cloud can be decoded with partial resolution as a thumbnail with less complexity and bandwidth. The attribute coding also provides an option to decode the partial resolution harmonized with the geometry. The harmonization allows immediate decoding of the smaller resolution data than the original with both geometry and attribute information present.

* Combined frame coding

In frame-based point cloud contents, each frame may be relatively smaller in file size which is less efficient for the I/O interface. The overhead of initializing decoder becomes more significant in the edge device. In this regard, combined frame coding technique is proposed to address both issues by introducing the encoding of frame index in the combined Group of Point cloud (GOP). As consecutive point cloud frames are highly correlated, this technique also improves the coding efficiency largely so that it could be also beneficial for storage usage of frame-based point cloud content. The frame indices, which are used in the decoder to reconstruct the input frames, are encoded in the bitstream.

**Profiles and levels of the standard**

G-PCC provides four profiles, Simple, Predictive, Dense and Main.

Simple profile defines the minimum set of the G-PCC coding tools targeting on low-complexity scenarios. Main profile targets on the general use case.

Predictive profile is specifically designed for LiDAR applications aiming low-delay scenarios.

Dense profile provides better coding performance on dense/solid point cloud content.

Main profile defines general tool sets that allows to customize the coding tool combination for own use cases.

For any profile defined above, a level generally corresponds to a particular decoder processing load and memory capability.

**G-PCC Performance**

The performance of G-PCC is obtained by experiments that are conducted following the G-PCC CTC [5]. Four test conditions are defined, including C1 as lossless geometry lossy attributes coding, C2 as near-lossless geometry lossy attributes coding, CW as lossless geometry lossless attributes coding, and CY as lossless geometry near-lossless attributes coding. For clarity, near-lossless implies a bounded error rather than the magnitude of the error.

As for the test dataset, variety of the contents are evaluated including, human models, buildings, objects, landscapes and LiDAR acquired data. The dataset is categories into six types, Solid, Dense, Sparse, Scant, Automotive-frame, and Automotive-fused respectively depending on the content density or capturing method.

The evaluation was made based on the rate-distortion (RD) performance. Bitrates are reported as bits per input point (bpp). For lossy compression, the geometry distortion is measured by the point-to-point (D1) and point-to-plane (D2) metrics. The color distortion is measured by PSNR for each channel:Y (Luma), Cb, and Cr (Chroma). The reflectance distortion is also computed by PSNR but only for a single channel.

To reflect the progress of G-PCC standardization activities, the current test model (i.e., TMC13v12) is compared to its early version (i.e., TMC13v3 released in 2018) and midterm version (i.e., TMC13v6 released in 2019). More specifically, TMC13v6 and TMC13v12 are benchmarked taking TMC13v3 as the anchor. It is noted that TMC13v1 is not chosen as the baseline since at that time a lot of tools in the model were immature to make a formative comparison.

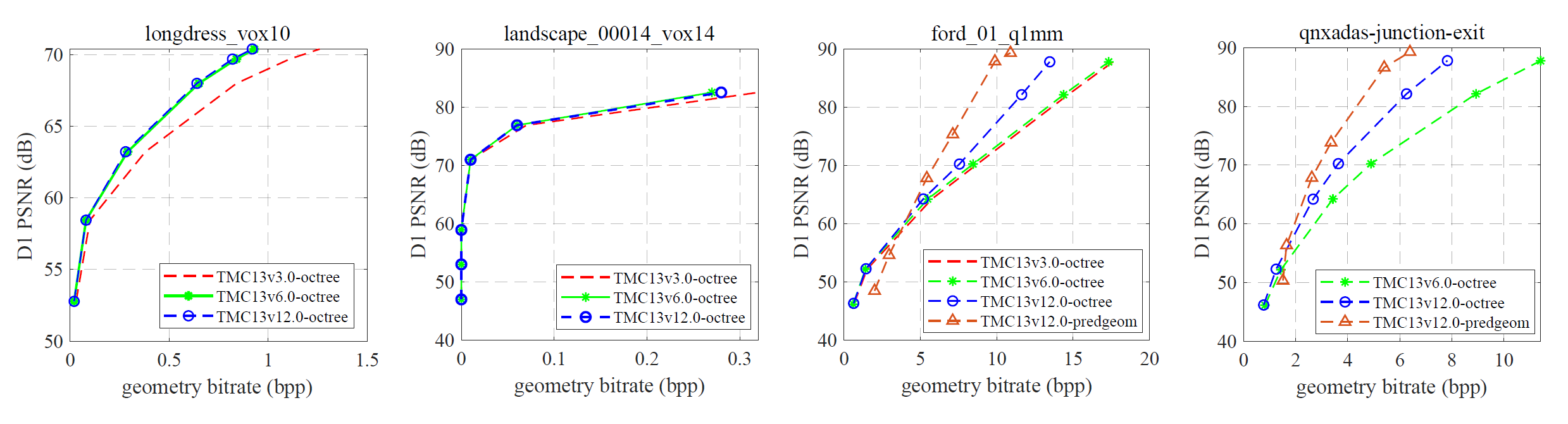


Figure G-PCC geometry coding performance examples

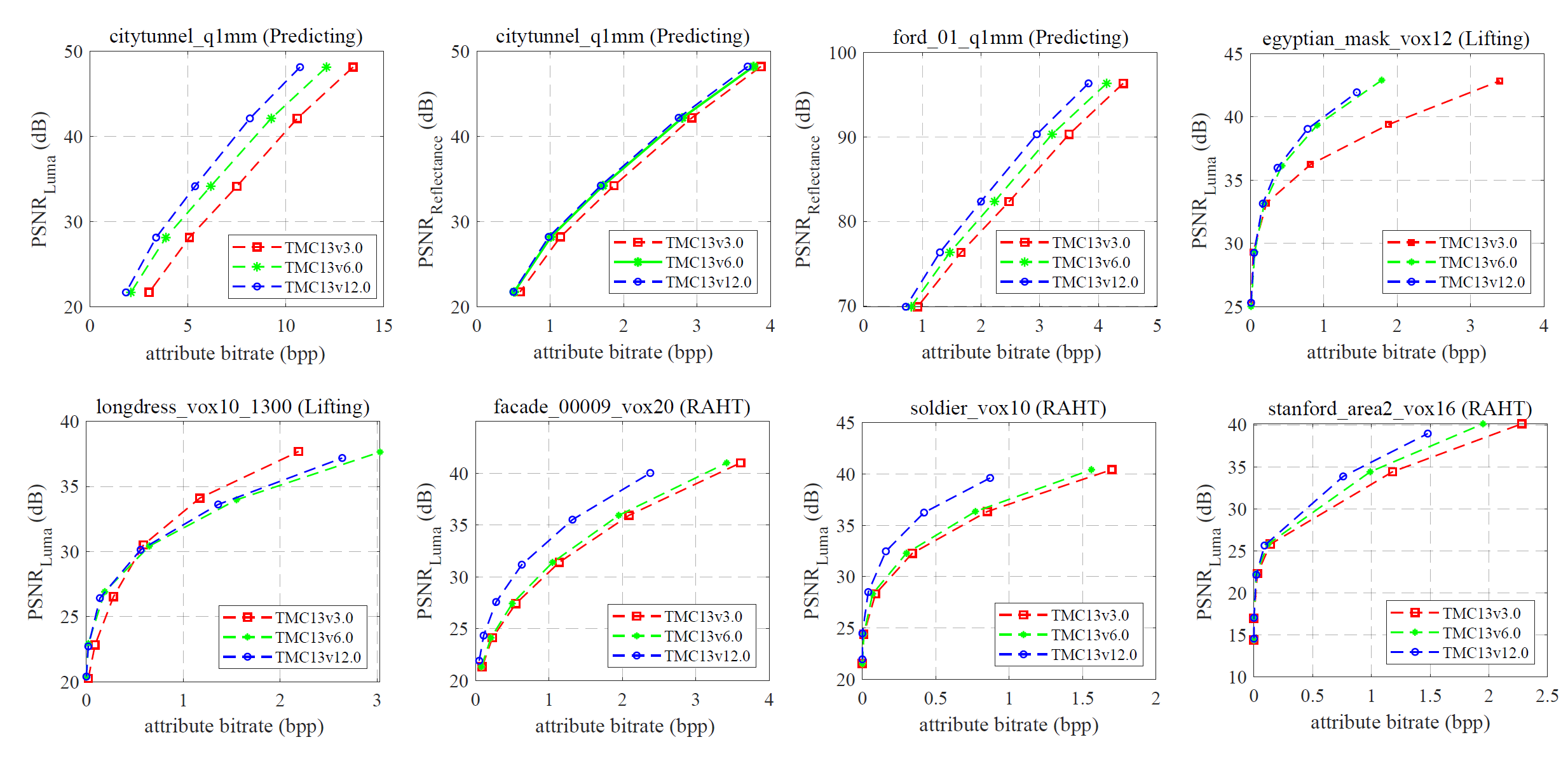


Figure G-PCC attribute coding performance examples

To benchmark G-PCC with existing point cloud compression schemes, Draco [8], an open-source library for com- pressing 3D meshes and point cloud is used for comparison. To make a fair comparison, special cares are required in correctly configuring the Draco codec. Specifically, in order to properly analyse the geometry coding performance, the Draco software is modified to strip colour attributes from input points. Two encode/decode passes are used for non-position attributes: the first codes geometry only, the second codes geometry plus the desired attribute. The difference in compressed sizes is used as an estimate of the bits required to code the given attribute. In order to evaluate the performance of the codec at different scales as per the G-PCC CTC, external pre- and post-processing stages are used to quantize the point cloud, permitting a direct comparison with G-PCC.

Table 1 and Table 2 compares Draco and G-PCC in lossy and lossless conditions respectively. As for G-PCC, TMC13v12 is configured as per the G-PCC CTC used to generate the anchor results. Note that predictive geometry coding is performed for Cat3-frame data while octree geometry coding is performed for the rest categories.

|  |  |  |
| --- | --- | --- |
|  | Draco 1.3.5-7 vs.TMC13v12 | |
|  | Geometry (D1) | Geometry (D2) |
| Cat1-A | -63.8% | -64.0% |
| Cat1-B | -30.1% | -30.3% |
| Cat3-fused | -29.6% | -29.5% |
| Cat3-frame | -38.3% | -52.9% |

Table PERFORMANCE COMPARISON BETWEEN DRACO AND G-PCC UNDER LOSSY CONDITION IN TERMS OF BD-RATE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Draco 1.3.5-7 vs.TMC13v12 | | | |
|  | Geometry | Colour | Reflectance | Total |
| Cat1-A | 63.0% | 66.9% | N/A | 65.6% |
| Cat1-B | 87.3% | 71.7% | N/A | 81.2% |
| Cat3-fused | 92.7% | 81.7% | 83.2% | 87.2% |
| Cat3-frame | 52.5% | N/A | 71.8% | 56.2% |

Table PERFORMANCE COMPARISON BETWEEN DRACO AND G-PCC UNDER LOSSLESS CONDITION IN TERMS OF BIT PER INPUT POINT (BPIP) RATIO.

**Publicly available information and resources**

MPEG has developed the G-PCC Test Model as its reference software encoder and decoder codebase [5]. It is intended to demonstrate coding efficiency capability and proper interpretation of the syntax and decoding process specified in the standard (but not as a speed-optimized implementation), and is intended to be usable as a starting basis for product implementations. The software is available under a BSD copyright license. G-PCC results are reported in various academic publications, few of them are listed in the Reference section.

**G-PCC vs V-PCC**

In parallel with the development of G-PCC, the MPEG consortium also published the V-PCC standard for encoding point clouds.

V-PCC follows a projection-based coding principle. The point coordinate is encoded as the distance with respect to a particular plane. The attribute associated to a 3D point is projected to a 2D map. 2D video codecs can be used to code both the geometry and attribute information. Different from V-PCC, G-PCC is a 3D oriented compression scheme in which the point cloud in 3D space is directly encoded. Data structures like octree and kd-tree are used to represent the point cloud. G-PCC encodes the geometry information and attribute information of point clouds separately. Geometry is coded first since the attribute coding depends on the reconstructed geometry.

Furthermore, G-PCC makes no assumption about the input point cloud coordinate representation. The points have an internal integer-based value, converted from a floating point value representation. This conversion is conceptually similar to voxelization of the input point cloud, and can be achieved by scaling, translation, and rounding. Another key concept for G-PCC is the definition of tiles and slices to allow parallel coding functionality. In G-PCC, a slice is defined as a set of points (geometry and attributes) that can be independently encoded and decoded. A tile is a group of slices with bounding box information. A tile may overlap with another tile and the decoder can decode a partial area of the point cloud by accessing specific slices.

In terms of compression performances, G-PCC provides state-of-the-art results for a majority of point cloud types. In contrast, V-PCC outperforms G-PCC for the particular case of dense points clouds with a low level of noise and motion.

**Future work in G-PCC**

One limitation of the current G-PCC standard is that it is only defined for intra prediction, that is, it does not currently use any temporal prediction tool. Nevertheless, techniques based on point cloud motion estimation and inter prediction are being considered for the next version of the standard. In addition, MPEG is also investigating the extension of G-PCC for dense point clouds.

# References

1. Call for proposals for point cloud coding v2, ISO/IEC JTC1/SC29/WG11, N16763, 2017.
2. S. Schwarz et al., “Emerging MPEG Standards for Point Cloud Compression,” in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133-148, March 2019.
3. G-PCC codec description, ISO/IEC JTC 1/SC 29/WG 7 N0057, January 2021
4. G-PCC performance evaluation and anchor results, ISO/IEC JTC 1/SC 29/WG 7 N0054, January 2021
5. Common Test Conditions for G-PCC, ISO/IEC JTC 1/SC 29/WG 7 N0368, October 2022
6. Guidelines to use G-PCC for achieving best compression performances, ISO/IEC JTC 1/SC 29/WG 7 N0460, October 2021
7. https://github.com/MPEGGroup/mpeg-pcc-tmc13
8. Google, "Draco: 3D Data Compression (2018)," https://github.com/google/draco, 2018.
9. Graziosi, D., Nakagami, O., Kuma, S., Zaghetto, A., Suzuki, T., & Tabatabai, A. (2020). An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing*, *9*, e13.
10. Yu, Y., Zhang, W., & Yang, F. (2022, December). Azimuth Adjustment Considering LiDAR Calibration for the Predictive Geometry Compression in G-PCC. In *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)* (pp. 1-5). IEEE.
11. Liu, H., Yuan, H., Liu, Q., Hou, J., & Liu, J. (2019). A comprehensive study and comparison of core technologies for MPEG 3-D point cloud compression. *IEEE Transactions on Broadcasting*, *66*(3), 701-717.
12. Zhang, X., Gao, W., & Liu, S. (2020, March). Implicit geometry partition for point cloud compression. In *2020 Data Compression Conference (DCC)* (pp. 73-82). IEEE.
13. Cao, C., Preda, M., Zakharchenko, V., Jang, E. S., & Zaharia, T. (2021). Compression of sparse and dense dynamic point clouds—methods and standards. *Proceedings of the IEEE*, *109*(9), 1537-1558.
14. S. Schwarz *et al*., "Emerging MPEG Standards for Point Cloud Compression," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133-148, March 2019, doi: 10.1109/JETCAS.2018.2885981.
15. Giuseppe Valenzise, Maurice Quach, Dong Tian, Jiahao Pang, Frédéric Dufaux, Chapter 13 - Point cloud compression, Editor(s): Giuseppe Valenzise, Martin Alain, Emin Zerman, Cagri Ozcinar, Immersive Video Technologies, Academic Press, 2023, Pages 357-385, ISBN 9780323917551,