A black and white logo

Description automatically generated **ISO/IEC JTC 1/SC 29/WG 2 N365**

**ISO/IEC JTC 1/SC 29/WG 2  
MPEG Technical Requirements   
Convenorship: SFS (Finland)**

**Document type:** Output Document

**Title:** Call for Proposals for AI-based Point Cloud Coding

**Status:** Approved

**Date of document:** 2024-04-26

**Source:** ISO/IEC JTC 1/SC 29/WG 2

**Expected action:** None

**Action due date:** None

**No. of pages:** 56 (with cover page) + 1 XLS + 1 requirement document + 1 use cases document

**Email of Convenor:** igor.curcio@nokia.com

**Committee URL:** <https://sd.iso.org/documents/ui/#!/browse/iso/iso-iec-jtc-1/iso-iec-jtc-1-sc-29/iso-iec-jtc-1-sc-29-wg-2>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 2**

**MPEG TECHNICAL REQUIREMENTS**

**ISO/IEC JTC 1/SC 29/WG 2 N365**

**Rennes, France, April 2024**

|  |  |
| --- | --- |
| **Title** | **Call for Proposals for AI-based Point Cloud Coding** |
| **Source** | **WG2, MPEG Technical Requirements** |
| **Status** | **Approved** |
| **Serial Number** | **23940** |

Table of Contents

[1 Introduction 4](#_Toc165019267)

[2 Scope 5](#_Toc165019268)

[3 Timeline 7](#_Toc165019269)

[3.1 Timeline of the calls, deadlines, and evaluation of the response 7](#_Toc165019270)

[3.2 Preliminary Development Plan 9](#_Toc165019271)

[4 Definitions 10](#_Toc165019272)

[5 Use Cases and Requirements 12](#_Toc165019273)

[5.1 Conventions 12](#_Toc165019274)

[5.2 3D Point Cloud Input Representation Requirements 12](#_Toc165019275)

[5.3 3D Point Cloud Specific Compression Requirements 13](#_Toc165019276)

[5.4 Hybrid-coding Requirements 13](#_Toc165019277)

[5.4.1 AI-based Geometry and non-AI-based Attribute coding 14](#_Toc165019278)

[5.4.2 Non-AI-based Geometry and AI-based Attribute coding 14](#_Toc165019279)

[6 Test Conditions 15](#_Toc165019280)

[6.1 Target Bit Rates 15](#_Toc165019281)

[6.2 Anchors 19](#_Toc165019282)

[7 Evaluation Procedure and Reporting 20](#_Toc165019283)

[7.1 Overview of Evaluations 20](#_Toc165019284)

[7.2 Objective Evaluation 21](#_Toc165019285)

[7.2.1 Bit rate reporting 21](#_Toc165019286)

[7.2.2 Distortion reporting 21](#_Toc165019287)

[7.2.3 Complexity reporting 22](#_Toc165019288)

[7.3 Subjective Evaluation 25](#_Toc165019289)

[8 Training and testing materials 27](#_Toc165019290)

[8.1 Dense Point Clouds for Immersive Applications 27](#_Toc165019291)

[8.1.1 Dynamic Objects 28](#_Toc165019292)

[8.1.2 Static Objects and Scenes 28](#_Toc165019293)

[8.2 Sparse Point Clouds from LiDAR 30](#_Toc165019294)

[9 Submission Logistics 32](#_Toc165019295)

[9.1 Documentation 32](#_Toc165019296)

[9.2 Source code 32](#_Toc165019297)

[9.3 Coded test material 34](#_Toc165019298)

[9.4 Submission package structure 34](#_Toc165019299)

[9.4.1 Folder app/ description 35](#_Toc165019300)

[9.4.2 Folder enc/ description 35](#_Toc165019301)

[9.4.3 Folder cfg/ description 35](#_Toc165019302)

[9.4.4 Folder src/ description 36](#_Toc165019303)

[9.4.5 Folder dec/ description 36](#_Toc165019304)

[9.4.6 Examples 36](#_Toc165019305)

[9.5 Test Coordination 37](#_Toc165019306)

[9.5.1 Test coordination on objective evaluation 37](#_Toc165019307)

[9.5.2 Test coordination on subjective evaluation 37](#_Toc165019308)

[9.5.3 Testing fee 38](#_Toc165019309)

[9.5.4 Data delivery method 38](#_Toc165019310)

[9.6 Non-MPEG Member Participation 38](#_Toc165019311)

[10 Source code and IPR 39](#_Toc165019312)

[11 Contacts 40](#_Toc165019313)

[References 41](#_Toc165019314)

[Annex A Hybrid Framework 42](#_Toc165019315)

[A.1 AI-based Geometry and Non-AI-based Attribute Encoder 42](#_Toc165019316)

[A.2 Non-AI-based Geometry and AI-based Attribute Encoder 42](#_Toc165019317)

[Annex B Anchor 43](#_Toc165019318)

[B.1 Anchor for C1 Dense Point Clouds for Immersive Applications 43](#_Toc165019319)

[B.2 Anchor for C2 Sparse Point Clouds for Autonomous Navigation and Robotics 44](#_Toc165019320)

[B.3 Anchor Bitstreams 44](#_Toc165019321)

[Annex C Objective Evaluation Metrics 45](#_Toc165019322)

[C.1 Geometric Distortions 45](#_Toc165019323)

[C.2 Attribute Distortions 46](#_Toc165019324)

[C.3 Metric Software Usage 46](#_Toc165019325)

[Annex D Docker and GPU Complexity Anchor 48](#_Toc165019326)

[D.1 Docker 48](#_Toc165019327)

[D.2 GPU Runtime Complexity Anchor 48](#_Toc165019328)

[Annex E Subjective Evaluation 50](#_Toc165019329)

[E.1 Test Design 50](#_Toc165019330)

[E.2 Renderer and generation of video sequences 50](#_Toc165019331)

[E.3 Delivery of submissions for subjective testing 51](#_Toc165019332)

[E.4 Laboratory setup 51](#_Toc165019333)

[E.5 Selection and training of test subjects 51](#_Toc165019334)

[Annex F Information Form 52](#_Toc165019335)

[F.1 Information Form 52](#_Toc165019336)

[Annex G Excel Reporting Template 54](#_Toc165019337)

[G.1 Introduction 54](#_Toc165019338)

[G.2 "Cover" sheet 54](#_Toc165019339)

[G.3 "Summary” sheet 54](#_Toc165019340)

[G.4 "Complexity” sheet 54](#_Toc165019341)

[G.5 “Graphs” sheet 54](#_Toc165019342)

[G.6 Raw data sheet 55](#_Toc165019343)

# Introduction

3D graphics contents are growing significantly with the rapid development of 3D sensors e.g., 3D scanners, LiDAR, and vision-based 3D reconstruction technologies. As an outcome of sensor-based data acquisition, point cloud – a raw 3D data format defined as a set of points, has become popular. Each point is depicted by its position using a 3D coordinate. As a set, the point cloud represents the geometry of an object or a scene. Optionally, each point may be associated with attributes that is application dependent. For example, RGB color for Immersive Applications (such as VR and AR), and reflectance from LiDAR for Autonomous Navigation and Robotics. The point cloud data allows further processing for more complete 3D modeling, e.g., mesh representation. They can be also consumed directly by machine-oriented downstream perception tasks, for example, detection, recognition, etc.

Often point clouds are composed of a sequence of frames representing a dynamic object or dynamic scene. This is especially the case for Immersive Applications (such as VR and AR), and LiDAR for Autonomous Navigation and Robotics.

AI-based eco-systems have been flourishing in a few application domains such as autonomous driving, robotics and metaverse in the past decade. AI techniques have also recently demonstrated state-of-the-art performance in point cloud compression. However, unlike e.g., processing, enhancement, rendering and perception tasks, there is no wide deployment of AI-based point cloud compression, and one reason is the lack of standards that prevents interoperability between different vendors.

The Call for Proposal solicits AI-based technologies to compress dynamic point clouds. There are two competition tracks: The 1st track focuses on lossy geometry-only compression, and the 2nd track focuses on lossy geometry+attribute compression.

Attachments of this CfP include,

* Use cases for AI-based point cloud coding [10]
* Requirements for AI-based point cloud coding [11] and
* Excel reporting template.

# Scope

The Call for Proposal (CfP) seeks an AI-based Point Cloud Compression (AI-PCC) framework. Specifically, deep learning-based technologies are to be solicited to compress point cloud data. A codec, with such deep learning-based point cloud compression technologies, typically allows for an end-to-end training or fine-tuning of different point cloud data types.

Please refer to Section 4 “Definitions” for the explanation of the terminologies used in this section.

The Call for Proposal (CfP) extends invitations for submissions and solicits technologies in Table 1.

Table 1. Technologies

|  |  |  |
| --- | --- | --- |
| **Track number** | **Technology** | **Description** |
| **Track 1** | **AI-based point cloud geometry compression** | Technology focused on AI-based compression techniques specifically designed for point cloud geometry. |
| **Track 2** | **AI-based point cloud attribute compression** | Technology focused on AI-based compression techniques specifically designed for point cloud attributes. |
| **AI-based point cloud geometry and attribute compression** | Technology focused on AI-based compression techniques specifically designed to encode both the point cloud geometry as well as attributes. |

The scope of the CfP is shown in Table 2. Proponents are encouraged to submit solutions that encompass lossy compression codecs:

* **Technology**: Proponents may submit geometry only compression solutions, attribute only compression solutions, or a compression solution that encodes both the geometry and attribute using AI-based technique. A geometry and attribute compression solution could be a joint geometry and attribute solution that uses a single (non-separable) AI-based technique to encode both the geometry and attribute, or it could have two different pipelines to encode geometry and attribute separately but employing AI-based techniques in at least one of the two pipelines.
* **Sparsity**: Proponents may submit category-specific codec or a unified codec (category is defined here with respect to point cloud sparsity). If the performance difference of the codecs is marginal, preference will be given to the unified codec.

Table 2. Call for Proposal scope categories.

|  |  |  |
| --- | --- | --- |
| 1. Technology |  | 1. Sparsity |
| Geometry only |  | Dense and Sparse (unified) |
| Geometry and Attribute | Dense only (category-specific) |
| - | Sparse only (category-specific) |

It is noted that a unified approach is recommended across differing sparsity levels among point cloud categories, e.g., dense point clouds for Immersive Applications and sparse point clouds from LiDAR sensors for Autonomous Navigations and Robotics. With a unified codec, the tool sets used to compress different point cloud data categories are assumed to have some level of overlaps or interactions. The tool sets need not however be totally identical: the framework could be configured specifically for different categories of point cloud data.

On other hand, category-specific codecs are also welcome for this CfP. That is, a codec can be tailored to compress dense point clouds only for Immersive Applications, or a codec can be tailored for sparse point clouds only as the ones obtained from LiDAR sensors for Autonomous Navigation and Robotics.

There are pros and cons for both the unified and the category-specific solutions. For example, in anticipation of the expensive cost in chip design and manufacturing, a unified codec solution has the potential to cut down the overall deployment cost by maximizing the possibility or the extent to re-use certain hardware assets. In contrast, because a category specific design is less restrictive, it could demonstrate a better coding performance /complexity tradeoff benefit.

In this CfP, higher preference will be given to a unified framework rather than a category-specific framework, provided that the performance/complexity benefits of category-specific solution become marginal. However, considering the possibility of a category-specific codec yielding substantial coding improvements over a unified approach, the CfP allows the submission of category-specific solutions.

Finally, AI-based techniques focused solely on geometry or attribute coding may utilize a hybrid-coding approach to achieve a comprehensive codec. The proposals of AI-based attribute coding only may use a hybrid-coding approach (and are therefore evaluated under Track 2).

# Timeline

This section consists of two subsections:

1. "Timeline of the calls, deadlines, and evaluation of the responses" outlines the chronological sequence of events related to the call for proposals, including the release of the call, submission deadlines, and the evaluation process. This subsection serves as a reference point for participants, ensuring clarity and transparency throughout the process.
2. "Preliminary Development Plan" outlines the initial steps and a broad roadmap for the development of the AI-based point cloud coding project. It provides a high-level overview of the key objectives, deliverables, and milestones anticipated during the project's implementation. This subsection aids in setting expectations and alignment in the understanding of the project's trajectory.

## Timeline of the calls, deadlines, and evaluation of the response

Table 3 outlines the milestones associated with the CfP process. It provides a comprehensive overview of the milestones, with corresponding dates, and noteworthy remarks involved in the CfP.

Table 3. Timeline of the calls, deadlines and evaluation of the response

|  |  |  |
| --- | --- | --- |
| Milestone | Date | Remarks |
| CfP releasing stage | | |
| Release of the CfP | 2024.04.26 | (Release the CfP when MPEG 146 is being closed. Friday of MPEG 146.) |
| Registration: Declaration of intention of answering to the CfP. | 2024.06.24 | (8 weeks after CfP release, and 3 weeks before MPEG 147)  An email should be sent to the Test Coordinators as listed in Subsection 9.5 to be registered. |
| Confirmation of registration | 2024.07.15 | (Monday of MPEG 147 meeting)  An individual uploading account and a proponent number for subjective testing will be provided to each proponent as confirmation. |
| Invoice | 2024.07.26 | (1 week after MPEG 147)  Test Coordinator issues an invoice to those who have declared the intention to answer the CfP. |
| Experimenting with the benchmarking machine | 2024.07.29  -  2024.09.06 | (6 weeks)  Account per proponent will be created on the benchmark machine by the Test Coordinator. Proponents can upload/decode example sequences. |
| CfP evaluation stage | | |
| Define camera path for subjective viewing | 2024.09.09 | (8 weeks before MPEG 148)  Define camera path used for subjective viewing by Test Coordinator. |
| Submission of testing material for subjective and objective evaluation, as well as the decoder | 2024.09.13 | (7 weeks before MPEG 148)  Testing material to be uploaded on the uploading site. Windows, MacOS or Linux decoder and the compressed bitstreams and decoded point clouds must be provided. The reconstructed point cloud sequences by the decoder should follow the CfP naming convention.  Alternatively, if the uploading site encounters problems proponents are encouraged to provide a private link for the Test Coordinator to download their data. |
| Submission of the completed objective evaluation spreadsheets | 2024.09.20 | (6 weeks before MPEG 148)  To be uploaded on the uploading site. Raw PSNR and rate are included in the spreadsheet for crosscheck. |
| Release of camera path for subjective viewing | 2024.09.20 | (6 weeks before MPEG 148)  Action performed by 3DGH Chair or Test Coordinator. |
| Feedback that PSNR check/rendering results for decoded bitstreams are matched | 2024.09.27 | (5 weeks before MPEG 148)  Action performed by 3DGH Chair or Test Coordinator.  PSNR difference threshold is set as 0.2%. The difference check is performed and verified per frame.  Rendering results by 3DGH Chair or Test Coordinator are shared with proponents on a 1:1 basis. |
| Compilation of submitted data in a unique spreadsheet and submission as an input contribution | 2024.10.18 | (2 week before MPEG 148)  Action performed by 3DGH Chair or Test Coordinator.  With decoding time and complexity against PCGCv2. Objective evaluations. |
| Subjective evaluation with naive viewers | 2024.09.27  -  2024.10.25 | (4 weeks for subjective evaluation. Start after test material uploading deadline and finish Friday before MPEG 148)  Test Coordinator: Mathias Wien  Viewing sequence generation. Subjective sessions.  Reports of subjective results. |
| Submission deadline for proposals documentation | 2024.10.28 | (1 week before MPEG 148)  To be uploaded as input to the MPEG document management system. |
| Evaluation of responses | 2024.11.04  -  2024.11.06 | (MPEG 148 week)  Action performed during the MPEG meeting week. Proponents are required to present their proposals in person. |
| Nominate the best proposal(s) | 2024.11.06 | (Wednesday of MPEG 148 week)  Preference is to select a single best proposal. If for any reason, there is no agreement on a single best proposal, the group will then nominate a reduced number of candidates (e.g. 2 or 3). Contributors would work to combine candidate solutions and present a combined proposal in MPEG 149. The combined proposal will be benchmarked against the initial best proposals in MPEG 148. |
| Submission deadline of source code for encoder/decoder of selected best proposal or the selected candidates | 2024.11.25 | (2 weeks after MPEG 148)  Source code shall allow to reproduce identical results to those submitted on the same platform that the technology was submitted. The source code shall match with the submitted documentation. In addition, the execution environment shall be provided. |
| Establishment of the first test model based on the best proposal(s) | 2025.01.22 | (Wednesday of MPEG 149)  Create the initial test model.  Identify potential technologies to be evaluated complementing the best proposal(s) |

## Preliminary Development Plan

Table 4 provides a detailed overview of the milestones and deliverables for the development of the AI-based graphics coding project.

Table 4: Preliminary development plan.

|  |  |  |  |
| --- | --- | --- | --- |
| Year | Month | MPEG meeting | Stage |
| 2025 | 01 | 149 | Establishment of the first test model based on the best proposal(s)  Identify potential technologies to be evaluated complementing the best proposal(s). |
|  | 04 | 150 | Issue WD 1.0. |
|  | 07 | 151 | Issue WD 2.0. |
|  | 10 | 152 | Issue WD 3.0. |
| 2026 | 01 | 153 | Issue CD. |
|  | 04 | 154 |  |
|  | 07 | 155 | Issue DIS (prefer stable text). |
|  | 10 | 156 |  |
| 2027 | 01 | 157 | Issue FDIS. |
|  | 04 | 158 |  |
|  | 07 | 159 | Approval of IS. |

# Definitions

This section provides a centralized reference for key terms and concepts used throughout the document, ensuring consistent interpretation, and facilitating effective communication.

**AR (Augmented Reality):** Technology that superimposes a computer-generated content on a user's view of the real world.

**Training/Validation/Test:** Splitting the dataset into a training, a validation and a test set for choosing the best network model.

**VR (Virtual Reality):** Computer-generated environment with scenes and objects that appear to be real, making the user feel they are immersed in their surroundings.

**LiDAR (Light Detection and Ranging):** Capturing method that uses light in the form of a pulsed laser to measure ranges/variable distances to the 3D sample positions.

**Point cloud frame:** A static point cloud representation at a given time instant or, for dynamically acquired point clouds, during a time range.

**Geometry:** The locations in 3D space of the points in the point cloud, i.e. the (x, y, z) coordinates of the points. Also referred to as 3D positions.

**Attribute**: A value or set of values, other than geometry, associated with a point, e.g., (R, G, B) color values for Immersive Applications point clouds, or r for reflectance from LiDAR point clouds for Autonomous Navigation and Robotics.

**Dense** **point cloud:** Refers to a point cloud dataset that contains a high density of points, where points are sampled at a relatively small spacing or resolution. In the context of this CfP, the term dense refers to the representation of an object that is captured dense enough to produce reasonable rendering for human viewing for Immersive Applications use case.

**Sparse** **point cloud**: Refers to a point cloud dataset that contains a relatively low density of points. In a sparse point cloud, points are sampled at a larger spacing or resolution, resulting in a sparser representation of the object or scene. In the specific context of this CfP, the term sparse refers to automotive frame-based spinning LiDAR acquired data for machine tasks.

**Lossy geometry compression**: The decoded compressed geometry is not necessarily numerically identical to the uncompressed geometry. The number of points in the output cloud can differ from the number of points in the input cloud.

**Lossless geometry compression:** The decoded compressed geometry is numerically identical to the uncompressed geometry, in terms of (x, y, z) values. The number of points in the output cloud is identical to the number of points in the input cloud.

**Lossy attribute compression:** The decoded compressed attribute values are not necessarily numerically identical to the uncompressed attribute values.

**Lossless attribute compression:** The decoded compressed attribute values are numerically identical to the uncompressed attribute values.

**Lossless point cloud compression**: A point cloud is losslessly encoded if and only if its geometry and attributes are compressed in a lossless manner.

**Lossy point cloud compression**: Lossy point clouds compression permits different tradeoffs between bitrate and quality. Such trade-offs could be achieved by encoding in lossy manner any components of the input point cloud (geometry and attributes).

**Mesh:** A fundamental data structure used to represent the surface of a three-dimensional objects and scenes with a collection of geometry, connectivity, and mapping information, as well as, vertex attributes, and attribute maps. In the current CfP, meshes are not used to evaluate inference performance but sampled point clouds from meshes can optionally be used during training.

**Near-lossless geometry compression**: The number of points after compression remains the same as the original, but the point locations after compression may not be mathematically identical, but the error between the original and compressed points is always less than the given error margin.

**Near-lossless attribute compression:** The number of points after compression remains the same as the original, but the point attributes after compression may not be mathematically identical, but the error between the original and compressed attributes is always less than the given error margin.

For clarity**, near-lossless** implies a bounded error rather than the magnitude of the error (i.e., near-lossless does not necessarily imply nearly lossless).

**AI-PCC**: Artificial intelligence based-point cloud compression (AI-PCC). A neural network-based point cloud compression scheme trained either in an end-to-end manner or has some portion of the solution employing neural network-based solution.

**Unified codec**: A codec capable of compressing both the dense and sparse point clouds.

**Category-specific codec**: A codec tailored specifically to compress one of the dataset categories: either the dense point clouds, or the sparse point clouds.

**Joint geometry and attribute codec**: Non-separable AI-based compression technique that jointly encodes point cloud geometry and attribute using a single module.

**Separate geometry and attribute coding**: An AI-based compression technique that independently encodes point cloud geometry and attributes using two distinct modules, employing AI-based compression in both the geometry and attribute segments of the codec.

**Geometry only coding**: An AI-based compression technique that compresses only the point cloud geometry using a neural network-based compression scheme. Proponents opting for a geometry-only technique should complete their codecs by employing a Hybrid approach.

**Attribute only coding**: An AI-based compression technique that compresses only the point cloud attributes using a neural network-based compression scheme. Proponents opting for an attribute-only technique should complete their codecs by employing a Hybrid approach that includes the compression of geometry component.

**Hybrid codec**: A compression methodology that combines both AI-PCC (Artificial Intelligence based-Point Cloud Compression) and non-AI-PCC techniques. This approach may involve employing an AI-based solution for geometry compression and a non-AI-based solution for attribute compression, or vice versa.

# Use Cases and Requirements

Two primary use cases have been identified: Immersive Technologies, characterized by dense point clouds capturing rich details, and Autonomous Navigation and Robotics, represented by sparse point clouds. These use cases are further elaborated in document [10] . Furthermore, while general requirements are outlined in [11] , additional specific requirements are detailed in the following subsections.

## Conventions

This subsection describes a set of conventions used throughout the document. Table 5 provides an explanation of the meaning associated with each convention.

Table 5: Conventions used in the document.

|  |  |
| --- | --- |
| Convention | Meaning |
| shall, shall support | Used for mandatory requirement |
| should, should support | Used for recommended requirement |
| may, may support, can | Used for optional requirement |
| shall enable | The functionality shall be specified but its support is optional. |
| should enable | The functionality is recommended to be specified and its support is optional. |
| may enable | The functionality may be specified and if it is, then its support is optional, and it shall not have any weight in the selection or exclusion of any solution. |
| shall not preclude | The functionality shall not be prevented. |
| should not preclude | It is recommended not to prevent the functionality. |

## 3D Point Cloud Input Representation Requirements

This subsection outlines the specific specifications regarding the representation of 3D point cloud data as input to the proposed system or solution. The 3D point cloud representation is listed in Table 6.

Table 6: Point cloud representation requirements.

|  |  |  |
| --- | --- | --- |
| Requirement | Use Case | Description |
| Point Cloud Geometry | Dense and Sparse | These are 3D positions represented as (x, y, z) coordinates of points with a specification of their precision and dynamic range. Fixed-point values shall be used. |
| Dense | 10 to 12 bits geometry bit-depth  Between 500,000 and 15,000,000 points |
| Sparse | 18 bits geometry bit-depth  Between 50,000 and 200,000 points per frame |
| Pre-defined Attributes | Dense and Sparse | Point attributes associated with each 3D position. Fixed-point values shall be used. |
| Dense | 8 bits color bit-depth  Normal to support the rendering using a shader |
| Sparse | 16 bits reflectance bit-depth |
| Static and Dynamic Point Clouds | Dense and Sparse | Support for both static and dynamic point clouds, accommodating scenarios where the point cloud data may remain unchanged or undergo temporal variations. |

Different types of point clouds may require specific considerations for attributes. For Dense point clouds used in Immersive applications, the primary attribute of interest is color information. For Sparse point clouds generated by LiDAR technology, the primary attribute of interest is reflectance. In this CfP color attribute in the RGB color space for Dense point clouds and reflectance for Sparse point clouds will be considered.

## 3D Point Cloud Specific Compression Requirements

This subsection defines the requirements pertaining to the AI-based compression techniques applied to 3D point cloud data. The basic intention of this CfP is to seek deep learning-based techniques to compress point clouds. Proponents have the discretion to determine the application of deep learning techniques in optimizing point cloud coding. This may range from an end-to-end deep learning-based codec to the utilization of deep learning for specific modules or functionalities within a point cloud codec. However, the proposals for the CfP shall incorporate deep learning-based modules that enhance point cloud compression.

MPEG AI-PCC shall support the requirements listed in Table 7.

Table 7: Compression requirements

|  |  |
| --- | --- |
| Requirement | Description |
| AI-based Compression | Proposed solutions shall be an AI-PCC solution. |
| Efficient Compression of Static and Dynamic 3D Point Clouds | The proposed solution shall support efficient compression techniques for both static and dynamic 3D point clouds. |
| Encoding of Point Clouds Geometry | The proposed solution shall support, at a minimum, the encoding of point clouds with 3D positions. Attribute-only solutions shall support employing a Hybrid solution to encode the geometry. |
| Encoding of Point Clouds Attributes | The proposed solution shall support, at a minimum, the encoding of point clouds attributes. Geometry-only solutions shall support employing a Hybrid solution to encode the attributes. |
| Fixed-Point Representation for Geometry and Attributes | The proposed solution shall support fixed-point input and output values for geometry and attributes. Floating point data shall be converted to a fixed-point representation according to input/output bit depth requirements. |
| Rate Points for Lossy Compression | The proposed lossy compression solution shall support four rate points for lossy compression as specified in Section 6. Each rate point shall be visually different when evaluating the subjective quality. Parameter control of the bitrate should be supported. |

## Hybrid-coding Requirements

As explained in Section 2, proponents can provide AI-based solutions that address either the geometry coding, or both geometry and attribute coding. The solutions that perform geometry or attribute coding may employ a hybrid-coding approach. It will enable participation to Track 2 of this CfP. This is shown in Table 8.

Table 8 Technologies allowed for solutions targeting geometry and attribute coding.

|  |  |
| --- | --- |
| Technologies | AI-involvement |
| Hybrid Approaches | AI-based Geometry and non-AI-based Attributes |
| Non-AI-based Geometry and AI-based Attributes |
| Non-Hybrid Approaches | AI-based Geometry and AI-based Attributes |

### AI-based Geometry and non-AI-based Attribute coding

If proponents choose to submit an AI-based geometry compression solution, either for static or dynamic point clouds, a non-AI-based attribute compression technique may be employed to participate in Track 2 of this CfP in addition to Track 1. In this case, the attribute coding may be supported using G-PCC’s intra-frame RAHT or V-PCC within a hybrid compression framework. See Annex A for more information. This requirement ensures that the attribute information can be incorporated into the overall compression process, allowing for a comprehensive representation of the point cloud data.

### Non-AI-based Geometry and AI-based Attribute coding

Proponents focusing exclusively on attribute compression, for either static or dynamic point clouds, shall be required to complete their codec by implementing a geometry compression solution. Such solutions may be based on G-PCC's Octree or V-PCC for geometry encoding within a hybrid compression framework.

**Note**. No hybrid framework implementation is provided in this document for either “AI-based geometry and non-AI-based attribute coding”, or “non-AI-based geometry and AI-based attribute coding”.

# Test Conditions

Though the CfP is seeking a wide range of technologies, the test condition is set with a focus on lossy dynamic point cloud compression. See Table 9.

Table 9 Dynamic AI-PCC test conditions for both dense and sparse point clouds

|  |  |
| --- | --- |
| Competition track | Test condition |
| Track 1 (T1) | Lossy Geometry-only |
| Track 2 (T2) | Lossy Geometry + Attribute |

With Track 1 (T1), lossy geometry-only compression is conducted. With Track 2 (T2), lossy geometry + attribute compression is performed. In both tracks, random access configuration is in use. The inclusion of T1 track is to satisfy geometry-only use cases, for example, BIM and navigation system. It also provides insights of the behavior of geometry codec that is necessary to avoid a poorly performing geometry coder that cannot be fully discovered by existing evaluation method.

For objects with single frame to be coded, no inter prediction will be applied.

For dynamic sequences with multiple frames to be coded, the Intra frame shall be inserted at least for every 32 frames.

## Target Bit Rates

For dense point clouds, the target bit rates for both T1 (geometry-only) and T2 (geometry + attribute) are set in Table 10 . For sparse point clouds, their target bit rates are set in Table 11.

The target rates are set per rate point, per sequence, per test track.

Note that the data submitted to test condition T1 (geometry-only) will be evaluated objectively as described in Subsection 7.2. Subjective evaluation will be additionally conducted for test condition T2 (geometry + attribute) for Dense point clouds for Immersive applications.

For the description of the sequences /objects, refer to Section 8.

Table 10 Target rates for dense point clouds

|  |  |  |  |
| --- | --- | --- | --- |
| **Sequence** | **Target Rates** | | |
| **Rate Points** | **T1** | **T2** |
| exercise\_vox10 | R1 (Mbps) | 0.72 | 0.83 |
| R2 (Mbps) | 0.85 | 1.04 |
| R3 (Mbps) | 1.07 | 1.40 |
| R4 (Mbps) | 1.41 | 2.06 |
| model\_vox10\* | R1 (Mbps) | 0.91 | 1.07 |
| R2 (Mbps) | 1.05 | 1.30 |
| R3 (Mbps) | 1.25 | 1.70 |
| R4 (Mbps) | 1.57 | 2.40 |
| basketball\_player\_vox11\* | R1 (Mbps) | 2.10 | 2.44 |
| R2 (Mbps) | 2.42 | 2.92 |
| R3 (Mbps) | 2.89 | 3.72 |
| R4 (Mbps) | 3.63 | 5.03 |
| dancer\_vox11 | R1 (Mbps) | 2.46 | 2.98 |
| R2 (Mbps) | 2.94 | 3.81 |
| R3 (Mbps) | 3.69 | 5.18 |
| R4 (Mbps) | 4.81 | 7.58 |
| thaidancer\_viewdep\_vox12\* | R1 (Mbps) | 2.48 | 2.83 |
| R2 (Mbps) | 2.74 | 3.09 |
| R3 (Mbps) | 3.15 | 3.78 |
| R4 (Mbps) | 3.77 | 5.03 |
| staue\_klimt\_vox12 | R1 (bpip) | 2.89 | 3.40 |
| R2 (bpip) | 3.90 | 4.89 |
| R3 (bpip) | 5.53 | 7.30 |
| R4 (bpip) | 8.11 | 11.45 |
| shiva\_00035\_vox12 | R1 (bpip) | 2.07 | 2.59 |
| R2 (bpip) | 2.77 | 3.80 |
| R3 (bpip) | 3.97 | 6.03 |
| R4 (bpip) | 5.94 | 10.17 |
| arco\_valentino\_dense\_vox12 | R1 (bpip) | 3.10 | 3.60 |
| R2 (bpip) | 4.13 | 5.19 |
| R3 (bpip) | 5.76 | 8.11 |
| R4 (bpip) | 8.23 | 13.26 |
| facade\_00009\_vox12 | R1 (bpip) | 0.57 | 0.80 |
| R2 (bpip) | 0.74 | 1.13 |
| R3 (bpip) | 1.04 | 1.68 |
| R4 (bpip) | 1.58 | 2.69 |
| house\_without\_roof\_00057\_vox12 | R1 (bpip) | 0.20 | 0.26 |
| R2 (bpip) | 0.26 | 0.40 |
| R3 (bpip) | 0.39 | 0.67 |
| R4 (bpip) | 0.62 | 1.22 |
| RWTT\_vishnu\_156\_vox10\* | R1 (bpip) | 0.07 | 0.10 |
| R2 (bpip) | 0.08 | 0.15 |
| R3 (bpip) | 0.09 | 0.23 |
| R4 (bpip) | 0.12 | 0.41 |
| RWTT\_foxstatue\_211\_vox10\* | R1 (bpip) | 0.06 | 0.08 |
| R2 (bpip) | 0.07 | 0.12 |
| R3 (bpip) | 0.08 | 0.22 |
| R4 (bpip) | 0.10 | 0.45 |
| RWTT\_tomb\_059\_vox10\* | R1 (bpip) | 0.10 | 0.13 |
| R2 (bpip) | 0.12 | 0.19 |
| R3 (bpip) | 0.14 | 0.35 |
| R4 (bpip) | 0.18 | 0.87 |

\*: The sequences/objects with a mark \* are selected to perform subjective tests.

Table 11 Target rates for sparse point clouds

|  |  |  |  |
| --- | --- | --- | --- |
| **Sequence** | **Target Rates** | | |
| **Rate Points** | **T1** | **T2** |
| KITTI\_11 | R1 (Mbps) | 0.46 | 0.47 |
| R2 (Mbps) | 2.97 | 3.00 |
| R3 (Mbps) | 5.79 | 6.34 |
| R4 (Mbps) | 12.93 | 14.85 |
| KITTI\_12 | R1 (Mbps) | 1.21 | 1.22 |
| R2 (Mbps) | 5.20 | 5.24 |
| R3 (Mbps) | 8.30 | 8.70 |
| R4 (Mbps) | 15.35 | 16.93 |
| KITTI\_13 | R1 (Mbps) | 0.48 | 0.49 |
| R2 (Mbps) | 2.57 | 2.61 |
| R3 (Mbps) | 4.98 | 5.58 |
| R4 (Mbps) | 11.24 | 13.14 |
| KITTI\_14 | R1 (Mbps) | 0.82 | 0.83 |
| R2 (Mbps) | 3.93 | 3.95 |
| R3 (Mbps) | 6.92 | 7.30 |
| R4 (Mbps) | 14.36 | 15.89 |
| KITTI\_15 | R1 (Mbps) | 0.56 | 0.57 |
| R2 (Mbps) | 3.26 | 3.30 |
| R3 (Mbps) | 6.05 | 6.64 |
| R4 (Mbps) | 12.98 | 14.90 |
| KITTI\_16 | R1 (Mbps) | 0.80 | 0.80 |
| R2 (Mbps) | 3.85 | 3.89 |
| R3 (Mbps) | 6.58 | 7.07 |
| R4 (Mbps) | 13.51 | 15.19 |
| KITTI\_17 | R1 (Mbps) | 1.32 | 1.33 |
| R2 (Mbps) | 5.26 | 5.31 |
| R3 (Mbps) | 8.21 | 8.70 |
| R4 (Mbps) | 15.04 | 16.71 |
| KITTI\_18 | R1 (Mbps) | 0.65 | 0.66 |
| R2 (Mbps) | 3.24 | 3.28 |
| R3 (Mbps) | 5.95 | 6.52 |
| R4 (Mbps) | 12.92 | 14.87 |
| KITTI\_19 | R1 (Mbps) | 0.73 | 0.74 |
| R2 (Mbps) | 3.30 | 3.34 |
| R3 (Mbps) | 5.86 | 6.41 |
| R4 (Mbps) | 12.50 | 14.27 |
| KITTI\_20 | R1 (Mbps) | 0.70 | 0.71 |
| R2 (Mbps) | 3.80 | 3.87 |
| R3 (Mbps) | 6.60 | 7.30 |
| R4 (Mbps) | 13.39 | 15.32 |
| KITTI\_21 | R1 (Mbps) | 0.82 | 0.83 |
| R2 (Mbps) | 4.01 | 4.06 |
| R3 (Mbps) | 6.98 | 7.44 |
| R4 (Mbps) | 14.10 | 15.74 |

## Anchors

A set of anchors is defined to establish reference points for evaluation.

For dense point clouds, *ISO/IEC 23090-5:2021 - Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)* is selected as the anchor codec. For sparse point clouds, G-PCC Octree + RAHT, a configuration in *ISO/IEC 23090-9:2023 - Geometry-based Point Cloud Compression*, is selected as the anchor codec.

To be specific, the V-PCC anchor codec is chosen as TMC2v22.0 that is configured as random access, i.e., “vvenc-slow-ra”. The G-PCC Octree + RAHT anchor codec is chosen as TMC13v12.8. Note that the T1 anchor bitstream is a subset of its corresponding T2 anchor bitstream. For details about the anchor generation, see Annex B.

Table 12 Anchor codec selection

|  |  |  |
| --- | --- | --- |
| **Track** | **C1: Dense** | **C2: Sparse** |
| **T1 (Geometry-only)** | V-PCCgeometry | G-PCC Octree |
| **T2 (Geometry + Attribute)** | V-PCC geometry + attribute | G-PCC Octree + RAHT |

# Evaluation Procedure and Reporting

A clear and well-defined evaluation procedure is of utmost importance in ensuring a fair and comprehensive assessment of the proposed solutions. It provides a structured procedure for evaluating the performance and effectiveness of different approaches, thus facilitating a meaningful comparisons and informed decision-making. To achieve this, the evaluation metrics and methodologies are to be described in detail. Specifically, the distortion evaluation, including geometry and attribute distortion, needs to be clearly outlined, specifying the algorithms, formulas, and methodologies employed. The document provides comprehensive descriptions and guidelines for the calculation of the specified distortions in Annex C. Software usage details to aid in replicability and transparency are additionally provided, allowing participants for an effective application of specified tools and methodologies.

To be a qualified proposal, target bit rates shall not be exceeded. Proposals will be evaluated based on bit rates, objective quality metrics, subjective evaluation (Track 2 for Immersive applications), and complexity. An Excel reporting template, described in Annex G, is provided to facilitate the result report.

## Overview of Evaluations

Since there are two competition tracks, namely, T1) geometry-only and T2) geometry + attribute; hence two evaluations are organized. T1 competition focuses on the geometry compression techniques only; while T2 competition focuses on a combined evaluation, that includes both geometry and attribute compression.

T1 shall apply if a proposal is based on a separable coding method between geometry and attribute. T1, however, shall not be applicable if a proposal is based on a non-separable geometry and attribute codec. Moreover, T1 is applied to both C1) dense point clouds for Immersive Applications and C2) sparse point clouds for Autonomous Navigation and Robotics. Objective evaluation applies to T1.

T2 shall apply to proposals that code both geometry and attribute. This track covers C1) dense point clouds for Immersive Applications with color, as well as C2) sparse point clouds for Autonomous Navigation and Robotic with reflectance. Subjective evaluation, in addition to objective evaluation, is conducted for Immersive Applications dense point clouds. For sparse point clouds for Autonomous Navigation and Robotics, only objective evaluation is applied. Below in Table 13, the evaluation method described above are summarized.

Table 13 Evaluation methods

|  |  |  |
| --- | --- | --- |
| - | **C1: Dense** | **C2: Sparse** |
| **T1: Geometry-only** | Objective evaluation applies | Objective evaluation applies |
| **T2: Geometry + Attribute** | Both objective and subjective evaluation apply | Objective evaluation applies |

The conditions of the two competition tracks are summarized as follows.

**Conditions of competition T1**

* Geometry shall be coded. No attribute is required to be coded.
* Four (4) geometry target bit rates per sequence are defined in Subsection 6.1.
* A set of geometry-only anchors is generated as specified in Subsection 6.2.
* Objective evaluation shall be done based on the geometry PSNRs against geometry bit rates.

**Conditions for competition T2**

* Both geometry and attribute shall be coded.
* Four (4) target bit rates per sequence are defined in Subsection 6.1.
  + All four rate points are used for objective evaluation for both C1 and C2.
  + For selected C1 sequences, the four (4) target rate points listed in Table 10 used for objective evaluation are also used for subjective evaluation.
* For a given target bit rate, how to allocate bit rates between geometry and attribute is not specified in this document and is up to the proponents.
* A set of geometry+attribute anchors is generated as specified in Subsection 6.2.

Since it is possible to have different top performance codec solutions for the two competition tracks, the group will examine the possibility of potential integration of the two solutions. This is not however mandatory and it may end up with two standard tracks.

## Objective Evaluation

This section presents the objective evaluation method to be used in this CfP. An Excel reporting template is provided with the Call for Proposal. See Annex G for details of the template.

### Bit rate reporting

Bitrates shall be reported overall, as well as separately for geometry and attributes. All the data and high-level syntax (HLS) required to decode geometry shall be counted towards the geometry bit rate. Similarly, all the data and HLS needed to decode attribute shall be counted for the attribute bit rate. Note this latter implies excluding the data and HLS that have already been counted for geometry.

Bit rates shall be reported as megabits per second (Mbps) for dynamic point clouds. Bits per input point (bpip) shall be reported for static objects/scenes. The evaluation will be made based on the rate-distortion (RD) performance, and RD curves shall be plotted using PSNR as the quality measure for all categories. Bjontegaard rates (BD-rates) against the anchors should be calculated using D1/D2 Y-Cb-Cr/reflectance PSNR vs Mbps/bpip RD-curves.

### Distortion reporting

The metrics used for evaluation is shown in Table 14. These metrics are essential for assessing the performance of the proposed method and will be given priority in the evaluation process. In the subsequent subsections, details for each metric are provided.

Table 14 Evaluation metrics

|  |  |
| --- | --- |
|  | Metrics |
| Geometry | D1 point-to-point distance in dB  D2 point-to-plane distance in dB |
| Attributes | Y-Cb-Cr in dB  Reflectance in dB |

#### Quality metrics

For geometric distortions D1 and D2, the PSNR calculated from the symmetric distortions shall be reported. The distortion metrics and calculation of PSNR is specified in Annex C.1 of this document.

Color distortion is measured in Y-Cb-Cr space with 3 separate MSE distortions, which are reported as PSNRs for each channel: Y (Luma), Cb and Cr (Chroma). The distortion metrics and calculation of PSNR is specified in Annex C.2 of this document.

The reflectance distortion is also computed using MSE, but only for a single component, and the corresponding PSNR values are reported.

The PSNRs for each component will be individually used to compare methods. All attribute PSNRs are also calculated from the symmetric distortions.

The metric software for calculating the mandatory objective distortion can be accessed from the MPEG GitLab:

https://git.mpeg.expert/MPEG/3dgh/v-pcc/software/mpeg-pcc-dmetric/

Note that there are other MPEG repositories of the software with specific purposes. Be sure to use the above V-PCC repository, and the software version used in this CfP shall be 0.14.1. Details for running the software are provided in Annex C.3 of this document.

Any possible duplicated points in the reconstructed point cloud shall be reported. For distortion metrics reporting, the average over duplicated points shall be imposed (setting dropdups=2 for the distortion metric software).

In the case of lossy dynamic content, the reported distortion measures, the number of output points and the number of duplicated points shall be averaged over all coded frames.

Proponents are advised that upon selection of their codec proposal, they may be required to provide additional metrics for further evaluation. These include distortion metrics averaged over all coded frames (i.e., averaged over Intra-coded frames, averaged over Inter-coded frames, averaged over all frames) and a report of the PSNR per frame.

### Complexity reporting

In addition to reporting the rate-distortion coding performance, proponent shall report the complexity parameters as listed in Table 15 for each codec model. This helps selecting a technology with a light-weight design while achieving competitive rate-distortion coding performance.

Table 15 Complexity assessments of a learning-based point cloud codec (also in Excel template)

|  |  |  |
| --- | --- | --- |
| **Context** | **Parameter** | **Description** |
| **Model** | Number of model parameters of encoder | Total number of learnable parameters in the neural network-based encoder model. |
| Number of model parameters of decoder | Total number of learnable parameters in the neural network-based decoder model. |
| Parameter precision | Level of precision or accuracy used to represent the values of the neural network model parameters (e.g., integer 8 bits, float 32 bits). |
| **Training** | Batch size | Number of training samples used in each iteration of the training process before the before the neural network model is updated. |
| Training time (days) | Total amount of time taken to train the neural network model on a specific dataset. |
| Learning rate | The learning rate used to govern the pace at which the neural network model parameters are updated. |
| Epoch size | Complete pass through the entire training dataset during the training process. |
| Peak training memory (MB) | Training memory consumption (GPU/TPU & CPU). |
| Hyperparameter selection | Other hyperparameters or configurations for training |
| **Inference** | Encoding time | Inference encoding runtime (GPU/TPU & CPU). |
| Decoding time | Inference decoding runtime (GPU/TPU & CPU). |
| Peak encoding memory (MB) | Inference encoding memory consumption (GPU/TPU & CPU). |
| Peak decoding memory (MB) | Inference decoding memory consumption (GPU/TPU & CPU). |

#### Neural network model

Since this Call for Proposal solicits AI technologies for point cloud compression, a prospective response shall involve at least one neural network model. The number of neural network parameters and the parameter precisions (e.g., 8-bit integer, 32-bit floating number) shall be disclosed.

#### Training complexity

Training complexity is another factor indicating the complexity level of a neural network model. The batch size, epoch size and learning rate shall be disclosed.

The time to train the neural network model from scratch shall be reported as training time. Absolute GPU runtime for training shall be reported.

Total training time shall be reported. If multiple stages of training are used, the training time of all training stages shall be summed up and reported as the total training time. For example, if a pre-training on a particular module is needed before training AI-based codec, the pre-training time shall be counted and included in the total training time.

#### Inferencing complexity (Coding complexity)

The inferencing complexity is also referenced as the coding complexity. The inferencing of the neural network-based codec is composed of two steps: encoding and decoding. Proponents shall report encoding and decoding complexity separately.

The GPU environment used for the inferencing shall be disclosed to assess the inferencing time.

**Memory**

Peak memory consumption shall be reported for the encoder and decoder, per rate point, and per sequence. This include both CPU memory and GPU memory.

Runtime per rate point and per sequence shall be reported for the encoder and decoder. To make it easier to compare runtime between proposals, relative encoding and decoding runtime shall be collected as described next.

**Runtime**

Basically, one needs to collect the runtime of anchor software first, and then compute the relative runtime based on the anchor software T\_proposal / T\_anchor. The same computing environment shall be in use for the anchor software and the proposed codec to make the relative runtime meaningful.

As V-PCC and G-PCC are non-AI codec, their total runtime is counted as CPU time T\_CPU\_VPCC and T\_CPU\_GPCC. Hereinafter, they are represented as T\_CPU\_anchor. They are collected per rate point and per sequence.

**GPU Runtime Anchor**

However, for AI-based technologies, especially for a hybrid framework with both AI-based functions and non-AI based functions, there is a need to analyze GPU runtime and CPU runtime. When computing a meaningful relative runtime, the GPU and CPU time shall be collected and computed separately, T\_proposal\_GPU / T\_anchor\_GPU and T\_proposal\_CPU / T\_anchor\_CPU.

For a prospective answer to this CfP, the following steps should be followed to collect GPU time and CPU time per rate point and per sequence:

T\_total: The total time used from the encoding/decoding

T\_GPU: The GPU runtime to run the GPU modules

T\_CPU = T\_total – T\_GPU: Estimated CPU runtime

PCGCv2, an AI-based point cloud codec, is used as a GPU complexity anchor software, and shall be used to collect the GPU anchor runtime. Note that the GPU anchor runtime T\_GPU\_anchor needs to be run on a single specified configuration rather than per rate point and per sequence configurations. A detail description on how to derive the GPU complexity of the PCGV2 anchor is provided in Annex D.

Finally the total relative runtime is computed as:

T\_total\_proposal = (T\_GPU\_proposal / T\_GPU\_anchor) + (T\_CPU\_proposal / T\_CPU\_anchor)

Additional notes on the runtime report:

* Between wall time and user time, proponent shall report user time.
* The timer shall be set by excluding the data loading/unloading from a hard drive. That is, the timer shall start immediately after the data is loaded and stop right before the data is to be written to hard drives.

#### Computing environment

The GPU environment used for the training and inferencing shall be disclosed to assess the training and inferencing complexity. See Table 16 for the details that should be reported.

Table 16 Computing environment for training and inference (also in Excel template)

|  |  |  |
| --- | --- | --- |
| **Context** | **Parameter** | **Description** |
| **Hardware** | GPU/TPU type | Specific type or model of the Graphics Processing Unit (GPU) or Tensor Processing Unit (TPU) being used. |
| CPU type | Specific type or model of the Central Processing Unit (CPU) being used. |
| Framework | Neural network development framework (PyTorch/TensorFlow). |
| Number of GPUs/TPUs | Number of GPUs/TPUs jointly used during training or inferencing. |
| CPU memory | The total physical CPU memory. |
| GPU/TPU memory | The total physical GPU/TPU memory. |
| **Software** | Learning infrastructure | Infrastructure used to facilitate the AI models, e.g., PyTorch, TensorFlow, etc., and their version |

#### Additional reporting

Additionally, one should report the following analysis results on a proposed AI technology.

* Theoretical study of the complexity on encoder and decoder, including number of operations, memory access patterns, floating point operations, etc.
* Suitability of the hardware implementation, i.e., avoid floating points, memory usage, potential for parallelization.
* Degree of capability for parallel processing.
  + 1. *Summary reporting*

The summary reporting shall include 1) RD plots and 2) BD differences for each sequence and an averaged result per category, dense static PCs for Immersive Applications, dense dynamic PCs for Immersive Applications and sparse dynamic PCs from LiDAR for Autonomous Navigation and Robotics:

* For both T1 and T2, report D1/D2 PSNR vs. geometry bitrate.
* For T2, additionally report Y-Cb-Cr/Reflectance PSNR vs. geometry+attribute total bitrate.
* For both T1 and T2, report relative encoding and decoding time.

If the proposed technology is separable between geometry and attribute, the geometry-only statistics shall be reported in the summary reporting sheet. Otherwise, if the proposed technology is non-separable between geometry and attribute coding, the geometry-only statistics shall be ignored.

## Subjective Evaluation

This section presents the subjective evaluation method to be used in this CfP.

Subjective evaluation will only be conducted on selected test material for T2 lossy geometry and attribute test conditions. The subset of test material as listed in Table 17 will be used for subjective evaluation: Model, Basketball\_player, Thaidancer\_viewdep,RWTT\_vishnu\_156, RWTT\_foxstatue\_211, and RWTT\_tomb\_059.

Table 17 Test Sequences for Subjective Evaluations

|  |  |  |
| --- | --- | --- |
| Class | Sequence identifier | Test material dataset filename |
| Dense | D02 | model\_vox10 |
| D04 | basketball\_player\_vox11 |
| D05 | thaidancer\_viewdep\_vox12 |
| Static | S06 | RWTT\_vishnu\_156\_vox10 |
| S07 | RWTT\_foxstatue\_211\_vox10 |
| S08 | RWTT\_tomb\_059\_vox10 |

The bitrates for the subjective evaluation are listed in Table 10. Note that the bitrates for the subjective evaluation of selected test sequences shall be the same with the ones used for the objective evaluations, that is, the same bitstreams shall be used for both objective and subjective evaluation tests. Additional details on the subjective evaluation can be found in Annex E.

# Training and testing materials

The training and testing materials are selected based on the guidelines provided in [3] . All test sequences are accessible under the following URLs:

https://content.mpeg.expert/data/CfP/AI-PCC/dataset/

As specified in Section 5 “Use Cases and Requirements" and Section 6 “Test Conditions”, two categories of point clouds are specified: *dense* point clouds for Immersive Applications and *sparse* point clouds for Autonomous Navigation and Robotics. Their respective definitions are given in Section 4 “Definitions”. Accordingly, the materials for training/validation and test will be described in the following two subsections.

Note that to enrich the training dataset options, additional mesh datasets are also included. See Table 18 for a summary of the different types of graphics content used in this CfP.

Table 18 Point Cloud Dataset Classification

|  |  |  |
| --- | --- | --- |
| Dataset Category | Content Type | Description |
| Dense PCs for Immersive Applications | Mesh | 3D models represented as vertices, edges and textures, used to generate mesh-derived sampled point clouds for training only. |
| Static Objects and Scenes | *Dense* voxelized point cloud frames representing surfaces. |
| Dynamic Objects | Sequence of multiple *dense* voxelized point cloud frames. |
| Sparse PCs for Autonomous Navigation and Robotics | Dynamic Acquisition | Sequence of multiple *sparse* voxelized point cloud frames in automotive LiDAR frame-based applications. |

To facilitate experiments on learning-based methods, two types of datasets are specified in Table 19: “TrainVal” dataset for training and validation and “Test” set for testing.

The data in TrainVal dataset are reserved exclusively for training and validation. Type of pre-processing strategy applied to point clouds or meshes is not made mandatory and may be freely specified by the proponents. Means to reproduce the training procedure for crosschecking purposes needs to be provided, however.

Table 19 Dataset experiment usages

|  |  |
| --- | --- |
| **Experiment Usage** | **Description** |
| TrainVal set | Used to train and validate the learning-based models |
| Test set | Exclusively reserved for benchmarking the performance of the learning-based point clouds compression model |

## Dense Point Clouds for Immersive Applications

The dynamic datasets utilized for dense point clouds category are primarily comprised of dynamic human bodies with moderate motions, and a synthetic dynamic dataset with computer generated objects. To enhance the variety of shapes in dense point clouds and to avoid potential bias toward human body applications, we have also incorporated static non-human datasets. This inclusion of diverse shapes aims to promote and ensure a more balanced technological development in response to the CfP.

### Dynamic Objects

The dynamic objects datasets are diversly composed of a few dynamic human bodies with moderate motions and a synthetic dynamic dataset with computer generated objects in simulated motions. With the exception of Queen (computer graphics content), other sequences listed under this category are scanned from real humans.

The *TrainVal set of dynamic objects* for Immersive Applications is specified in Table 20.

Table 20 TrainVal material - Dynamic Objects, Immersive Applications Dense Point Clouds (https://content.mpeg.expert/data/CfP/AI-PCC/dataset/training/dense\_dynamic)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class | TrainVal material identification | Frames | fps | No. Points /fr | Geometry Precision | Peak Value (p) | Attributes |
| Dense | Queen | 250 | 50 | ~1,000,000 | 10 bit | 1023 | R,G,B |
| 8i VFB – Loot | 300 | 30 | ~780,000 | 10 bit | 1023 | R,G,B |
| 8i VFB – Red\_and\_Black | 300 | 30 | ~700,000 | 10 bit | 1023 | R,G,B |
| 8i VFB – Soldier | 300 | 30 | ~1,500,000 | 10 bit | 1023 | R,G,B |
| 8i VFB – Long\_dress | 300 | 30 | ~800,000 | 10 bit | 1023 | R,G,B |

The *coding test set of dynamic objects* for Immersive Applications is specified in Table 21.

Table 21 Coding test material - Dynamic Objects, Immersive Applications Dense Point Clouds (https://content.mpeg.expert/data/CfP/AI-PCC/dataset/testing/dense\_dynamic)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class | Test material identification  (sequence ID: sequence name) | Frames | fps | No. Points / fr | Geometry Precision | Peak Value (p) | Attributes |
| Dense | D01: Exercise\_vox10 | 300 | 30 | ~608,000 | 10 bit | 1023 | R,G,B |
| D02: Model\_vox10 | 300 | 30 | ~810,000 | 10 bit | 1023 | R,G,B |
| D03: Dancer\_vox11 | 300 | 30 | ~2,600,000 | 11 bit | 2047 | R,G,B |
| D04: Basketball\_player\_vox11 | 300 | 30 | ~2,900,000 | 11 bit | 2047 | R,G,B |
| D05: Thaidancer\_viewdep\_vox12 | 300 | 30 | ~3,100,000 | 12 bit | 4095 | R,G,B |

### Static Objects and Scenes

The *TrainVal set of static objects and scenes* for Immersive Applications is specified in Table 22.

A few non-human objects/scenes, obtained from museum scans, are also considered. Compared to human bodies in the dynamic objects, they exhibit different shapes and have a higher bit depth than most dynamic object point clouds.

RWTT [4] data are provided as 3D meshes which can be sampled to obtain point clouds.

Real World Textured Things (RWTT) is a collection of publicly available textured 3D models, generated with modern photo-reconstruction tools. The dataset aims to provide a challenging benchmark for geometry processing algorithms targeting real-world.

RWTT originally consists of 568 textured models generated with various 3D reconstruction pipelines, with 109 models containing multiple texture features. Out of all the models, only the ones with permissive licenses are selected that are listed in Table 24 .

Moreover, out of all selected ones, three RWTT objects are reserved for testing as shown in Table 24 while the rest are used for training.

Table 22 TrainVal material - Static Objects and Scenes, Immersive Applications Dense Point Clouds (https://content.mpeg.expert/data/CfP/AI-PCC/dataset/training/dense\_static)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class | TrainVal material identification | Frames | fps | No. Points | Geometry Precision | Peak Value (p) | Attributes |
| Mesh | RWTT Train Set | 1 |  | To be sampled | | | R,G,B |
| Dense | Head\_00039\_vox12 | 1 |  | 13,903,516 | 12 bit | 4095 | R,G,B |
| Frog\_00067\_vox12 | 1 |  | 3,614,251 | 12 bit | 4095 | R,G,B |
| Egyptian\_mask\_vox12 | 1 |  | 272,684 | 12 bit | 4095 | R,G,B |
| ULB\_Unicorn\_vox13 | 1 |  | 1,995,189 | 13 bit | 8191 | R,G,B |

The *Coding test set of static objects and scenes* for Immersive Applications is specified in Table 23.

Table 23 Coding test material – Static objects and scenes, Immersive Applications, Dense Point Clouds (https://content.mpeg.expert/data/CfP/AI-PCC/dataset/testing/dense\_static)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | Test material identification  (sequence ID: sequence name) | No. Points | Geometry Precision | Peak Value (p) | Attributes |
| Dense | S01: Facade\_00009\_vox12 | 1,596,085 | 12 bit | 4095 | R,G,B |
| S02: House\_without\_roof\_00057\_vox12 | 4,848,745 | 12 bit | 4095 | R,G,B |
| S03: Arco\_Valentino\_Dense\_vox12 | 1,481,746 | 12 bit | 4095 | R,G,B |
| S04: Statue\_Klimt\_vox12 | 499,660 | 12 bit | 4095 | R,G,B |
| S05: Shiva\_00035\_vox12 | 1,009,132 | 12 bit | 4095 | R,G,B |
| S06: RWTT\_vishnu\_156\_vox10 | 1,155,954 | 10 bit | 1023 | R,G,B |
| S07: RWTT\_foxstatue\_211\_vox10 | 1,348,767 | 10 bit | 1023 | R,G,B |
| S08: RWTT\_tomb\_059\_vox10 | 1,671,340 | 10 bit | 1023 | R,G,B |

Table 24 RWTT dataset selection

|  |  |  |
| --- | --- | --- |
| Class | Material identification | Object Index |
|  | RWTT All  (Training Set) | 1-5, 7, 9-14, 16, 18-26, 30-32, 34-36, 38, 39, 42, 43, 45, 47, 48, 52-54, 56-58, 61-63, 65-67, 69-72, 76-79, 82-85, 88-91, 94-97, 99-111,113-118, 120-127, 129, 130, 132-136, 138, 140-142, 144-155, 157-160, 163-166, 168, 170, 172-174, 176-189, 191, 193, 194, 196, 198-203, 206-208, 210, 212-214, 217-220, 222-227, 229, 232-234, 236, 237, 241, 242, 244-247, 249, 252, 253, 255, 256, 258-262, 264, 265, 267-269, 272, 274, 275, 279-289, 291-294, 296-305, 307-309, 311, 312, 314, 316, 317, 319-321, 324-326, 329-331, 333-335, 337-342, 344-352, 354-358, 360-363, 365, 371-380, 382-389, 391-398, 400-404, 406-423, 425-433, 435, 436, 439-441, 443-453, 456-459, 461-463, 465-469, 471-478, 482, 483, 485-487, 491-495, 498-506, 508-514, 516, 519-523, 526, 528-533 |
| RWTT Test Set | 059 (Tomb), 156 (Vishnu), 211 (Fox Statue), |

## Sparse Point Clouds from LiDAR

KITTI [5] dataset is a large-scale dataset with LiDAR scans obtained in outdoor self-driving settings containing large amount of raw LiDAR data, together with other data modalities (e.g., RGB) and labels. The original geometry positions are represented in floating numbers. For geometry, quantization to 18 bit is done for compression purpose. The quantization is done by dividing the floating-point values by 0.001 followed by rounding to the nearest integer values. This makes the data lie in range [-131072, 131072], which is further shifted by adding 131072 to each data point making the final values to be in range (0, 218-1). All frames go through the same processing which makes the dataset amenable to dynamic compression pipelines.

The *TrainVal set* for LiDAR is specified in Table 25.

Table 25 TrainVal material – Sparse Point Clouds from LiDAR for Autonomous Navigation and Robotics (https://content.mpeg.expert/data/CfP/AI-PCC/dataset/training/sparse\_dyanamic/KITTI18)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class | TrainVal material identification | Frames | fps | No. Points / fr | Geometry Precision | Peak Value (p) | Attributes |
| Sparse | KITTI\_00 | 4541 | 10 | 121494 | 18 bit | 30000 | I |
| KITTI\_01 | 1101 | 10 | 105682 | 18 bit | 30000 | I |
| KITTI\_02 | 4661 | 10 | 125627 | 18 bit | 30000 | I |
| KITTI\_03 | 801 | 10 | 123971 | 18 bit | 30000 | I |
| KITTI\_04 | 271 | 10 | 125718 | 18 bit | 30000 | I |
| KITTI\_05 | 2761 | 10 | 125038 | 18 bit | 30000 | I |
| KITTI\_06 | 1101 | 10 | 122300 | 18 bit | 30000 | I |
| KITTI\_07 | 1101 | 10 | 121331 | 18 bit | 30000 | I |
| KITTI\_08 | 4071 | 10 | 122594 | 18 bit | 30000 | I |
| KITTI\_09 | 1591 | 10 | 124384 | 18 bit | 30000 | I |
| KITTI\_10 | 1201 | 10 | 125910 | 18 bit | 30000 | I |

The Coding t*est set* from LiDAR for Autonomous Navigation and Robotics is specified in Table 26.

Table 26 Coding test material - Sparse Point Clouds from LiDAR for Autonomous Navigation and Robotics (https://content.mpeg.expert/data/CfP/AI-PCC/dataset/testing/sparse\_dynamic/KITTI18)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class | Test material identification  (sequence ID: sequence name) | Frames | fps | Average No. Points / fr | Geometry Precision | Peak Value (p) | Attributes |
| Sparse | K11: KITTI\_11 | 300 | 10 | 124,022 | 18 bit | 30000 | I |
| K12: KITTI\_12 | 300 | 10 | 120,196 | 18 bit | 30000 | I |
| K13: KITTI\_13 | 300 | 10 | 112,360 | 18 bit | 30000 | I |
| K14: KITTI\_14 | 300 | 10 | 126,201 | 18 bit | 30000 | I |
| K15: KITTI\_15 | 300 | 10 | 121,138 | 18 bit | 30000 | I |
| K16: KITTI\_16 | 300 | 10 | 123,551 | 18 bit | 30000 | I |
| K17: KITTI\_17 | 300 | 10 | 116,455 | 18 bit | 30000 | I |
| K18: KITTI\_18 | 300 | 10 | 123,796 | 18 bit | 30000 | I |
| K19: KITTI\_19 | 300 | 10 | 119,697 | 18 bit | 30000 | I |
| K20: KITTI\_20 | 300 | 10 | 117,725 | 18 bit | 30000 | I |
| K21: KITTI\_21 | 300 | 10 | 123,897 | 18 bit | 30000 | I |

Note that all downloaded training, validation and test material datasets should be verified with MD5 checksums. Each zip file of a training, validation and test material dataset contains an MD5 file (with the corresponding md5 sums for each file in the archive). A README.md file is also available with further details of the files in the MPEG repository.

# Submission Logistics

To prepare a submission to the CfP, a prospective proponent is required to do the following:

* Encode the test datasets as specified in Section 8 “Training and testing materials” using their codec, at the target bitrates specified as mandatory in Subsection 6.1 “Target Bit Rates”.
* Provide their bitstreams, decoded point clouds, along with their decoder following the timeline as specified in Subsection 3.1.

The proponents shall provide sufficient material in a written document to validate the performance of their submission. A submission should address at least one of the competition tracks as defined in “Table 9 Dynamic AI-PCC test conditions for both dense and sparse point clouds”. Proponents should present their proposals in person at MPEG 148 meeting in Turkey.

In the following subsections, the procedure to prepare a submission is provided in detail.

## Documentation

Complete submissions shall include the following elements:

1. Information forms shall be submitted within each proposal. See the form as specified in Annex F of this document.
2. A technical description shall be provided as an input document to MPEG 148 meeting for full conceptual understanding of the encoding and decoding algorithms for the proposal to be considered. It should include all data processing components used to generate the bitstreams and reconstructed point clouds. The uploading deadline of the description is set by the usual MPEG document uploading deadline – Oct 30.
3. The technical description shall include results of the objective tests. The results shall be reported as specified in Subsection 7.2 and using the Excel spreadsheet as provided in Annex G. The uploading deadline of the spreadsheet containing the objective results to uploading site is set on Sep 27.
4. The technical description shall contain information suitable to assess the complexity of the implementation of the technology, see Subsection 7.2.3 for details.
5. The technical description shall contain a statement about the programming language in which the software is written, e.g., C/C++/Python, the deep learning library PyTorch/TensorFlow, and the platform(s) on which the binaries were compiled. Note that PyTorch is recommended to be used in the responses for easier software integration after CfP.
6. The technical description shall state how the proposed technology behaves in terms of random access to any frame within the sequence. For example, a description of the GOF (Group of Frames), such as structure and the maximum number of frames that are required to be decoded to access any frame, shall be given.
7. The technical description shall specify the expected encoding and decoding delay characteristics of the technology, including structural delay, e.g., due to the amount of frame reordering and buffering and the degree by which the delay can be minimized by parallel processing.

## Source code

Complete submissions shall additionally agree with the following elements related to crosschecks:

1. **Training crosscheck**:
   1. Training crosscheck is not required for Test Coordinator to run on all answers.
   2. However, proponents shall be ready to support running training crosscheck upon request, e.g., for the winner answer. Upon request, proponents shall immediately provide the source code pertaining to the training process employed for developing the proposal, including details on the training datasets, loss functions, hyper-parameter values, and any other pertinent information necessary for reproducing the proposal's results. It should be noted that only proposals trained exclusively with the datasets specified in “Section 8 Training and test material” will be accepted.
2. **Inference crosscheck**:
   1. Decoding crosscheck is required to be performed by the Test Coordinator. Upon submission, proponents shall provide the source code of their decoder along with hardware and software requirements.
   2. Encoding crosscheck is not required for the Test Coordinator to run on all answers. However, proponents shall provide the source code of their encoder along with hardware and software requirements to support running encoding crosscheck upon request, e.g., for the winner answer.
   3. The encoder and decoder implementation shall support standalone inference/testing on a standard computer (CPU and/or GPU) within a reasonable timeframe. The description of the standard computer is depicted in Subsection 9.5.
   4. For AI-based codec, the model parameters shall be provided together with the source code. It would allow the Test Coordinator to run inference test without conducting the training.
   5. The encoder shall produce a bitstream as its output, which will be processed by the decoder to reconstruct the point cloud. For either a sequence with multiple frames or a static object, one and only one bitstream file shall be supplied to the decoder.
   6. Docker image shall be provided to support running their encoder / decoder.

Proponents shall additionally agree:

1. **Open-source requirement**: Proponents are advised that, upon acceptance for further evaluation or development, certain parts of any proposed technology shall be made available in source code format to the WG 7 with sufficient rights granted to the participants. It would allow WG 7 to use their source code to build an initial test model as a starting point for a potential new standard. It would also allow WG 7 to plan either exploration experiments (EEs) or core experiments (CEs) to facilitate the standardization procedure. When a particular technology becomes a candidate for further evaluation, commitment to provide such software in source code format is a condition to proceed.
2. **Software version requirement**: The software shall produce identical results to those submitted to the test on the same platform that the technology was submitted. Additionally, for software of the winner answer, submission of improvements (bug fixes, etc.) once after being selected is strongly encouraged. The new software should be provided as a patch on top of their initially submitted software.

## Coded test material

Complete submissions shall additionally include the following elements:

1. Bitstreams for all datasets in target test scenarios, which follow the associated test conditions and satisfy the rate constraints specified in this document. Proponents shall use the following naming scheme:
   1. PnnTx[S/D/K]mmRy.bin. The name is composed of 4 substrings that are illustrated in Table 27. For example, the bitstream file of “Dancer” for rate point 2 of geometry-only track from proponent P03 shall be named as P03T1D03R2.bin.

Table 27 Naming scheme

|  |  |
| --- | --- |
| Variable | Description |
| Pnn | Proponent number; to the Anchor, typically assigned the code P00. |
| Tx | A code identifying the test track which is defined in Table 9. That is, T1 is for geometry-only coding track, and T2 is for geometry+attribute coding track. |
| [S/D/K]mm | Sequence number. It appears as “Dmm” in Table 21 for dynamic dense sequences, “Smm” in Table 23 for static dense objects, and “Kmm” in Table 26 for sparse sequences. |
| Ry | The rate number, range from 1 to 4 as specified in Table 10 and Table 11. |

1. Reconstructed point clouds for all datasets in target test scenarios.
   1. PnnTx[S/D/K]mmRy.zip. The ZIP file contains the PnnTx[S/D/K]mmRy(\_04d%) .ply that are the reconstructed point clouds. The .ply files shall be placed in the root folder of the zip package. That is, no subfolder structures shall be created inside the zip package. For example, the reconstructed point cloud of “Dancer” for rate point 2 of geometry-only track from proponent P03 shall be named as P03T1D03R2\_%04d.ply and directly packaged into P03T1D03R2.zip.
   2. The index number (%04d) in file name shall use the same numbers of corresponding frame in the input sequence. For example, Thaidancer sequence starts from 6487 to 6786, and the index number shall remain the same range from 6487 to 6786.

## Submission package structure

Proponents shall deliver their submission in the form described in this section. Note that submissions deviating from that format cannot be properly evaluated.

The directory shall contain seven (7) folders under the root of the uploading site as described in Subsection 9.5 of the document.

Table 28 Submission package structure

|  |  |
| --- | --- |
| Folder | Contents |
| app/ | A script named as “decode.sh” shall be provided to allow a single click to decode all the bitstreams by the Test Coordinator. No command line parameters are allowed to run the script “decode.sh”.  Any decoder executable or binary library (Windows 64 bits, MacOS, or Linux) – precompiled if applicable, supporting scripts (if required), and usage documentation for decoding the bitstreams. |
| enc/ | All encoded bitstreams and decoded point clouds. See Subsection 9.4.2 for details.  A text file md5check.txt containing the md5 number of all bitstreams and all decoded point cloud frames. |
| cfg/ | AI model parameters trained by the proponents.  Configuration files (if required). |
| src/ | Decoder source code shall be provided upon submission.  Encoder source code, as well as the source code for training, upon request after submission.  The source code shall be wrapped up in a docker image along with an environment. A copy of the docker image may be placed directly in this folder. Alternatively, if it is shared via a docker registry, the link shall be provided under this folder. |
| dec/ | Placeholder reserved for all decoded files by 3DG Chair / Test Coordinator. |

Note that the delivery of the material follows the timeline as described in Subsection 3.1 of this document.

### Folder app/ description

To facilitate the evaluation and the decoding processes, proponents shall provide a bash script decode.sh (without parameters) in the app/ directory to decompress the encoded bitstreams. The decoding script takes bitstreams from enc/, and store the decoded point clouds into the reserved folder dec/.

Upon request, for example for a winner answer, all encoding- and training- related executable, library, and supporting scripts shall be provided in this folder.

### Folder enc/ description

Folder enc/ hosts the encoded bitstreams and decoded point clouds from the proponents for all sequences and all rate points. A subfolder is created per rate point per sequence. The files shall follow the rules:

* enc/PnnTx[S/D/K]mmRy/PnnTx[S/D/K]mmRy.bin
* enc/PnnTx[S/D/K]mmRy/dec/PnnTx[S/D/K]mmRy.zip containing enc/PnnTx[S/D/K]mmRy/dec/PnnTx[S/D/K]mmRy(\_%04d).ply. The optional 4-digit field (\_04d%) is appended to indicate the frame number in the dynamic sequence.
* enc/PnnTx[S/D/K]mmRy/md5check.txt contains the md5 checksum of the bitstream and each decoded frame.

For an answer to the CfP Track 1 (or Track 2), there will be 24 (number of sequences) x 4 (number of rate points) = 96 subfolders under folder enc/.

### Folder cfg/ description

Folder cfg/ hosts the AI models that are required to be loaded before running encoding and decoding. Note that the AI models for decoding shall be made available upon submission. The AI models for encoding shall be made available upon request later.

The configuration files required to run the codec shall be stored under cfg/ folder. They shall not contain essential data that need be counted during rate reporting or complexity reporting, for example, look up tables.

### Folder src/ description

Folder src/ is used to host the source code in a docker image to set up computing environments. It can take a large space up to several GBs for the raw docker image. If docker registry can be utilized, a link to retrieve the docker image would be sufficient. See Annex D for details.

### Folder dec/ description

Folder dec/ is reserved to host the decoded point clouds by the Test Coordinator. By launching the decoding script app/decode.sh, the decoded point clouds shall be put under folder dec/. The file naming follows the same rule as the decoded point clouds under folder enc/, i.e., dec/PnnTx[S/D/K]mmRy/PnnTx[S/D/K]mmRy(\_%04d).ply.

### Examples

For example, for each proponent, the folder of each contribution would have the structure as shown in Table 29:

Table 29 Folder structure of a submission

|  |  |  |
| --- | --- | --- |
| P05/ | app/ | decode.sh |
| … |
| enc/ | P05T1S01R1/P05T1S01R1.bin |
| P05T1S01R1/dec/P05T1S01R1.zip // contains the decoded PLY |
| P05T1S01R1/md5check.txt |
| P05T1S01R2/P05T1S01R2.bin |
| P05T1S01R2/dec/P05T1S01R2.zip // contains the decoded PLY |
| P05T1S01R2/md5check.txt |
| … |
| cfg/ | DecoderModel1.pt |
| … |
| doc/ | … |
| src/ | … |
| dec/a | P05T1S01R1/P05T1S01R1(\_%04d).ply a |
| P05T1S01R2/P05T1S01R2(\_%04d).ply a |
| … |

a Note the dec/ folder is reserved for Test Coordinator to run the decoding and save the decoded point clouds. The proponents shall leave the dec/ folder empty.

## Test Coordination

The formal objective and subjective evaluations are coordinated by Test Coordinators. They are responsible for all required logistic, technical, and design activities in the context of the formal objective and subjective evaluations for the verification tests.

### Test coordination on objective evaluation

The Test Coordinator for the objective evaluation is:

Dr. Marius Preda

MPEG 3D Graphics Chair

Institut Mines-Télécom

37-39 rue Dareau 75014 Paris, France

Email: marius.preda@it-sudparis.eu

The verification test will be conducted by the ARTEMIS laboratory of Institut Mines Telecom. The provided bitstreams from the proponents will be decoded and assessed together, with a verification of the distortion measures as reported by the proponents. Should any discrepancies be observed, the respective proponents will be immediately notified, and a joint investigation will be initiated between the test lab and the proponent to determine the cause. Once all tests are confirmed, the test lab will prepare a comprehensive Excel file summarizing the performance of all submitted solutions. The document will be submitted as an input contribution to the MPEG 148 meeting in November 2024.

**CPU/GPU HW platform**. The run environment used by the Test Coordinator is specified as below.

* CPU: Intel (R) Core (TM) i9-10900K CPU @ 3.70GHz
* GPU: 1x NVIDIA GeForce RTX 4090, 24 GB
* OS: AlmaLinux 9.3
* Memory: 64 GB DDR4

For each CfP registered participants, the Test Coordinator will create an account on the above specified machine. The CfP participants will have an opportunity to perform experiments, e.g., to decode example bitstreams during a scheduled period as in Table 3.

### Test coordination on subjective evaluation

The Test Coordinator for the subjective evaluation is:

Mathias Wien

RWTH Aachen University

Lehrstuhl für Bildverarbeitung

52056 Aachen

Germany

Email: wien@lfb.rwth-aachen.de

Tel.: +49-241-80-27867

The Test Coordinator will select two or more ITU-R BT.500 compliant [1] evaluation laboratories, depending on the number of answers to be received. Subjective evaluation shall be in line with guidelines described in [2] .

### Testing fee

The testing fee when responding to the CfP is set to EUR 4500 per candidate codec in Track 2 for dense point clouds. The fee is used to compensate the efforts of the recruited test subjects. The fee in addition covers the cost from the dryrun performed during the preparation of the CfP.

No testing fee for responding to Track 1 will be charged.

### Data delivery method

The Test Coordinator will share instructions and credentials for data uploading to each proponent after they register for the CfP.

Alternatively, if problems are encountered during uploading, proponents are encouraged to provide a private link for the Test Coordinator to download their data before the deadline.

## Non-MPEG Member Participation

Current non-MPEG members are in addition encouraged to participate this Call for Proposal. During the Call for Proposal stage, they would be provided access to the MPEG GIT space, DMS system, and data repository. They should send their request to WG 7 convenor Marius Preda.

# Source code and IPR

By responding to a CfP, the proponent affirms that they are willing to make source code available for use as the starting point for collaborative standardization if their solution is selected.

It is the responsibility of the proponent to obtain any necessary internal approvals in a timely manner.

Furthermore, proponents are advised that this Call is being made subject to the common patent policy of ITU-T/ITU-R/ISO/IEC (refer to www.itu.int/ITU-T/dbase/patent/patent-policy.html or Appendix I of ISO/IEC Directives Part 1).

The copyright of the selected solution shall be transferred to ISO/IEC at the time of releasing the first version of test model.

# Contacts

If any questions the Call for Proposals, please reach out to the contact persons:

Dr. Marius Preda

MPEG 3D Graphics Chair

Institut Mines-Télécom

37-39 rue Dareau 75014 Paris, France

Email: marius.preda@it-sudparis.eu

Dr. Igor Curcio

MPEG Requirements Chair

Nokia Technologies

Hatanpäan Valtatie 30

33100 Tampere

Finland

Email: igor.curcio@nokia.com

# References

1. Recommendation ITU-R BT.500-14 (2019), Methodologies for the subjective assessment of the quality of television images
2. “Guidelines for Verification Testing of Visual Media Specifications,” ISO/IEC JTC 1/SC 29/AG 5 Doc. w20975/N0039, Online, October 2021
3. “Guidelines for test material for CfEs and CfPs,” JTC 1/SC 29 WG2 (MPEG Technical requirements) Document w20116 / N32, Online, January 2021
4. Real-world Textured Things, Retrieved from https://texturedmesh.isti.cnr.it/, January 2024
5. SemanticKITTI, Retrieved from http://www.semantic-kitti.org/, July 2023
6. “EE 5.5 on Hybrid framework for AI-PCC related coding experiments,” JTC 1/SC 29 WG 7 (MPEG 3D Graphics Coding) Document w22802, Antalya, April 2023
7. “Common Test Conditions for V-PCC,” JTC 1/SC 29 WG 7 (MPEG 3D Graphics Coding) Document w19613, Online, October 2020
8. “Common Test Conditions for G-PCC,” JTC 1/SC 29 WG 7 (MPEG 3D Graphics Coding) Document w19642, Online, October 2020
9. “3D graphics renderer for the CfP on dynamic mesh compression,” ISO/IEC JTC 1/SC 29 WG 7 Doc. N0298, Online, January 2022
10. “Use cases for AI-based point cloud coding,” JTC 1/SC 29 WG2 (MPEG Technical Requirements) Doc. N0364, Rennes, April 2024
11. “Requirements for AI-based point cloud coding,” JTC 1/SC 29 WG 2 (MPEG Technical Requirements) Doc. N0358, Rennes, April 2024
12. ITU-T Rec. P.910, “Subjective video quality assessment methods for multimedia applications,” Oct. 2023
13. Tian, Dong, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro. "Geometric distortion metrics for point cloud compression." In 2017 IEEE International Conference on Image Processing (ICIP), pp. 3460-3464. IEEE, 2017.
14. Hybrid Framework

This annex outlines the hybrid framework that was selectively studied in the MPEG WG 7 exploration on hybrid frameworks for AI-PCC [6] . The goal of the exploration was to investigate hybrid approaches that incorporate both AI-based and non-AI-based solutions and to assess the performance of such hybrid solutions.

A proponent may employ a hybrid framework with AI-based geometry + non-AI-based attribute codec or a non-AI-based geometry + AI-based attribute codec. See Section 5.4.

Both G-PCC and V-PCC are acceptable non-AI-based solution candidates.

* 1. AI-based Geometry and Non-AI-based Attribute Encoder

One example of AI-based geometry and non-AI-based attribute encoder is recoloring-based framework as illustrated in Fig. A.1.

First, the geometry provided to an “AI-based Geometry Encoder” is reconstructed.

A recoloring is performed such that the points in the reconstructed geometry are associated with attributes based on the attributes associated with the input geometry.

In the end, the recolored attributes are encoded using a non-AI-based attribute encoder.

A diagram of a algorithm

Description automatically generated

Figure A.1 – Hybrid framework with AI-based geometry and non-AI-based attribute encoder.

* 1. Non-AI-based Geometry and AI-based Attribute Encoder

One example of non-AI-based geometry and AI-based attribute encoder is illustrated in Fig. A.2. It is a similar recoloring-based framework as in Fig. A.1. In this case, the reconstructed geometry is obtained by a non-AI-based geometry encoder, and the recolored attributes are coded by an AI-based encoder.

A diagram of a algorithm

Description automatically generated

Figure A.2 – Hybrid framework with non-AI-based geometry and AI-based attribute encoder.

1. Anchor

A set of anchors is defined to establish reference points for evaluation.

For dense point clouds for Immersive applications, *ISO/IEC 23090-5:2021 - Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)* is selected as the anchor codec. For sparse point clouds for Autonomous Navigation and Robotic, G-PCC Octree + RAHT, a configuration in *ISO/IEC 23090-9:2023 - Geometry-based Point Cloud Compression*, is selected as the anchor codec. Both anchors for dense and sparse point clouds are chosen as the best performing anchor configuration operating under lossy conditions.

* 1. Anchor for C1 Dense Point Clouds for Immersive Applications

The V-PCC anchor codec is chosen as TMC2v22.0 configured as random access, i.e., “vvenc-slow-ra”. This anchor codec can be accessed from the MPEG GitLab:

https://git.mpeg.expert/MPEG/3dgh/v-pcc/software/mpeg-pcc-tmc2/-/tree/release-v22.0

The configuration of anchors for dense point clouds basically follows V-PCC's Common Test Conditions [7] with the following added encoder parameters (Extended-Rec2 configuration):

* profileCodecGroupIdc=3
* profileToolsetIdc=1
* profileReconstructionIdc=2
* mapCountMinus1=1
* pointLocalReconstruction=1 # for dense dynamic
* pointLocalReconstruction=0 # for dense static
* pbfEnableFlag=1
* useEightOrientations=1
* additionalProjectionPlaneMode=5

The enabled tools/profiles in these configurations are:

* Extended profile
* Reconstruction 2 profile
* VTM encoder
* 2 maps
* Point local reconstruction
* Patch border filtering
* Use eight orientations
* Additional projection plane

Additionally, parameters minimumImageWidth and minimumImageHeight require sequence (or object) specific values to successfully run the V-PCC encoding and decoding.

The configuration files for each dense point cloud and each test point are available at:

https://content.mpeg.expert/data/CfP/AI-PCC/anchor/cfg/v-pcc/

Lossy Geometry + Attributes anchor: The bitstream created following the above V-PCC configurations are referenced as the lossy geometry + attribute anchor for dense point clouds.

Lossy Geometry-only anchor: The sub-bitstream from the above anchor bitstream including all necessary data required to decode the geometry, e.g., high level syntax, occupancy map, etc., is referenced as the lossy geometry-only anchor.

* 1. Anchor for C2 Sparse Point Clouds for Autonomous Navigation and Robotics

The G-PCC Octree + RAHT anchor codec refers to TMC13v12.8. This anchor can be accessed from the MPEG GitLab:

https://git.mpeg.expert/MPEG/3dgh/g-pcc/software/tm/mpeg-pcc-tmc13/-/tree/release-v12.8

The configuration of anchors for sparse point clouds basically follows G-PCC's Common Test Conditions [8] . For geometry coding, the octree is used with angular mode disabled (angularEnabled=0). On attribute coding, the region-adaptive hierarchical transform is in use (transformType=0) with inter-level transform domain prediction enabled (rahtPredictionEnabled=1).

The configuration files for each sparse point cloud and each test point are available at:

https://content.mpeg.expert/data/CfP/AI-PCC/anchor/cfg/g-pcc/

Lossy Geometry + Attributes anchor: The bitstream created following the above G-PCC configurations are referenced as the lossy geometry + attribute anchor for sparse point clouds.

Lossy Geometry-only anchor: The sub-bitstream from the above anchor bitstream including all necessary data required to decode the geometry, e.g., high level syntax, is referenced as the lossy geometry-only anchor.

* 1. Anchor Bitstreams

The compressed anchor bitstreams for different settings are available at:

https://content.mpeg.expert/data/CfP/AI-PCC/anchor/bitstreams/

The MD5 checksum of each anchor bitstream and decoded point clouds are provided in addition for proponents to verify their anchor generation process.

Objective evaluation results of the anchor are available in the Excel template provided with this CfP. The distortion is calculated according to Annex C. Note that the anchor runtime in the Excel template shall be finished by the proponents, based on the same platform that run their proposed codec, to report the relative runtime of a proposed codec.

Note: For non-MPEG members, the software and the anchors will be available upon request to the contact persons. See Subsection 9.6 for details.

1. Objective Evaluation Metrics
   1. Geometric Distortions

Let and denote the original and the compressed point cloud, respectively. Consider evaluating the compression errors, denoted as in point cloud relative to reference point cloud. The steps to compute both point-to-point error (D1) and point-to-plane error (D2) for geometric errors are summarized in the following and illustrated in the below figure.

For each point in point cloud, *i.e.*, the black point in the figure, identify a corresponding point in point cloud, *i.e.*, the red point in the figure. Nearest neighbor is used to locate the corresponding point. In particular, a KD-tree search is used to perform the nearest neighbor search in order to reduce the computation complexity.

* + 1. Computing D1

Determine an error vector by connecting the identified point in reference point cloud to point in point cloud . The length of the error vector is the point-to-point error, *i.e.*,

|  |  |  |
| --- | --- | --- |
|  |  | (C-1) |

Based on the point-to-point distances for all points , the point-to-point error (D1) for the whole point cloud, withas the number of points in point cloud, is defined as:

|  |  |  |
| --- | --- | --- |
|  |  | (C-2) |

For near-lossless geometry encoding, the maximum point-to-point error, denoted , should be considered instead:

|  |  |  |
| --- | --- | --- |
|  |  | (C-3) |

* + 1. Computing D2

Project the error vector along the normal direction and get a new error vector. In this way, the point-to-plane error is computed as,

|  |  |  |
| --- | --- | --- |
|  | . | (C-4) |

The point-to-plane error (D2) for the whole point cloud is then defined as,

|  |  |  |
| --- | --- | --- |
|  |  | (C-5) |

For near-lossless geometry encoding, the maximum plane-to-plane error, denoted , should be considered instead:

|  |  |  |
| --- | --- | --- |
|  | . | (C-6) |

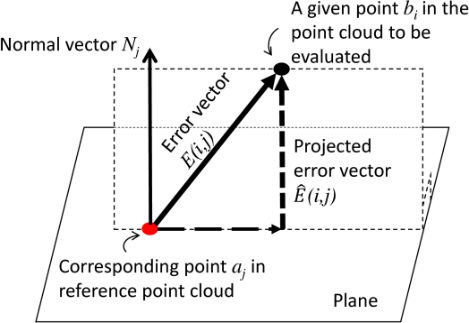


Figure C.1 – Illustration of point-to-point distance (D1) and point-to-plane distance (D2) [13]

* + 1. PSNR Calculation

For reporting distortion values, the PSNR is defined as the peak signal over the symmetric distortion, i.e. for geometry distortions D1 or D2 computed as:

|  |  |  |
| --- | --- | --- |
|  |  | (C-7) |

where is the peak constant value defined for each reference point cloud as specified in Section 8, and is the mean squared point-to-point (D1) or point-to-plane (D2) error. For dynamic content, the peak value is unchanged over the frames of a sequence.

Note that the metric software dynamically determines the intrinsic resolution and uses it as normalizer if it is not specified in the command line (option -r).

* 1. Attribute Distortions

For lossy attribute coding, the attribute PSNR value is computed as:

|  |  |  |
| --- | --- | --- |
|  |  | (C-8) |

For color attributes, the MSE for each of the three color components is calculated. A conversion from RGB space to ITU-R BT.709, since *Y′CbCr* spaces correlate better with human perception. A symmetric computation of the distortion is utilized, in the same way as is done for geometric distortions. The maximum distortion between the two passes is selected as the final distortion. Since the colour attributes for all test data have a bit depth of 8 bits per point, the peak value p for PSNR calculation is 255.

For reflectance attributes, the MSE for a single component is calculated. Since the reflectance attribute for all test data has a bit depth of 16 bits per point, the peak value p for PSNR calculation is 65535.

For near-lossless coding, the maximum error should be considered instead of MSE.

* 1. Metric Software Usage

The metric software dmetric that implements the presented method is available at:

https://git.mpeg.expert/MPEG/3dgh/v-pcc/software/mpeg-pcc-dmetric/

The metric software version 0.14.1 shall be used to report PSNR values.

The specified command line below shall be used to recreate the objective distortion metrics for T2 lossy geometry+attribute coding response:

./pc\_error --fileA=pointcloudOrg.ply --fileB=pointcloudDec.ply \

--inputNorm=normalOrg.ply --resolution=<peak value> \

--color=1 --neighborsProc=1 --dropdups=2

For T1 lossy geometry coding, the parameter --color=1 should be removed.

For near-lossless, the modified command line is:

./pc\_error --fileA=pointcloudOrg.ply --fileB=pointcloudDec.ply \

--inputNorm=normalOrg.ply --resolution=<peak value> \

--color=1 --neighborsProc=1 --dropdups=2 -d

These command lines will produce D1, D2 as well as color and reflectance distortion metrics, using an imported intrinsic resolution for PSNR computation rather than an internally determined intrinsic resolution. The option --dropdups=2 indicates that duplicate points are averaged and reports the number of points with the same coordinates. The (default) option --neighborsProc=1 indicates, for the color metrics, that neighbouring points at the same minimum distance are averaged using equal weights. The option -d indicates the use of the maximum error (Hausdorff distance).

Further example command lines to use the evaluation metric tool are provided below.

This will load the point cloud and report nearest neighbor distances – the intrinsic resolution to be used for PSNR computation:

./pc\_error --fileA=pointcloud1.ply

This will produce D1:

./pc\_error --fileA=pointcloudOrg.ply --fileB=pointcloudDec.ply

This will produce both D1 and D2. The normal is provided in normalOrg.ply for the original point cloud. It could be the same as pointcloudOrg.ply.

./pc\_error --fileA=pointcloudOrg.ply --fileB=pointcloudDec.ply \

--inputNorm=normalOrg.ply

This will produce D1, D2, as well as the color distortion metric.

./pc\_error --fileA=pointcloudOrg.ply --fileB=pointcloudDec.ply \

--inputNorm=normalOrg.ply --color=1

This will produce D1, D2 as well as color distortion metric, using an imported peak value for PSNR computation rather than an internally determined peak value.

./pc\_error --fileA=pointcloudOrg.ply --fileB=pointcloudDec.ply \

--inputNorm=normalOrg.ply --color=1 --resolution=<peak value>

1. Docker and GPU Complexity Anchor

This annex describes the GPU complexity anchor.

* 1. Docker

Docker containers are lightweight, portable, and self-sufficient units that encapsulate an application and its dependencies. They enable consistent and efficient deployment across different environments.

Docker container is recommended to be used to prepare the CfP answers. It helps leveraging the burden to create an exact deep learning environment and hence to improve the reproducibility. It can finally make crosscheck work easier. That is, the proponents shall encapsulate their environment as a docker image, and provide the docker image for crosscheckers.

It is recommended that the proponents to provide a Docker image file in their responses to the CfP. Having this eases the burden on the Test Coordinators and cross-checkers and facilitates the creation of the same running instances as to the one run by the proponents.

* + 1. Docker registry

Docker registry is helpful for access control and versioning. Proponents can use their choice of registry, for example, GitHub platform.

* 1. GPU Runtime Complexity Anchor

The runtime anchors presented in Subsection 7.2.3.3 attempt to provide benchmarks to evaluate the runtime. For algorithm modules implemented on CPU, the relative encoding/decoding time shall be reported against a corresponding anchor codec, V-PCC or G-PCC, per rate point per sequence/object. For algorithm modules implemented on GPU, the relative encoding/decoding time shall be reported against a GPU runtime anchor module that is specified next.

PCGCv2 is selected as the GPU runtime anchor model. The source code, trained network models and docker image for a required environment is available at the CfP dataset repository for downloading:

https://content.mpeg.expert/data/CfP/AI-PCC/anchor/pcgcv2/

PCGCv2 needs not be run all sequences/objects, it shall be run once on “House\_without\_roof\_00057\_vox12” only using a single configuration r3\_0.10bpp.pth, as listed in the packed PCGCv2 above. To get this anchor GPU runtime, the following steps are taken:

1. Download the packed PCGCv2.
2. Load docker image.

docker load -i /PATH/TO/IMAGE/pcgcv2\_docker.tar

docker run -it --gpus all --ipc=host --ulimit memlock=-1 --ulimit stack=67108864 -p 8888:8888 -v [HOST FOLDER]:[CONTAINER FOLDER] pcgcv2:init

Note that, [HOST FOLDER]:[CONTAINER FOLDER] needs to be replaced by the source (host) and the destination (container) folders with the source folder containing PCGCv2.

1. Run PCGCv2 for encoding and decoding, with the following command:

python coder.py –filedir=’[PATH to PC]/House\_without\_roof\_00057\_vox12.ply’ –ckptdir=’ckpts/r3\_0.10bpp.pth’ –scaling\_factor=0.375 –rho=1.0 –res=4096

1. Collect the printed “Enc Time” and “Dec Time” as shown in Figure D.1 below.

A screenshot of a computer

Description automatically generated

Figure D.1 Example snapshot of the output screen of the PCGCv2 runtime anchor

1. Compute “anchor GPU runtime” by adding the above “Enc Time” and “Dec Time”.
2. Repeat 10 times of the step 2 to 4.
3. Compute the average of the 10 “anchor GPU runtime”, that is to be used as the final “anchor GPU runtime”, referenced as T\_GPU\_anchor in Subection 7.2.3.

Note: The encoding and decoding time of a proposal shall be reported separately while the same anchor GPU runtime T\_GPU\_anchor is shared.

1. Subjective Evaluation

This annex describes the details on how subjective evaluation will be performed.

The Test Coordinator is responsible for all required logistic, technical, and design activities in the context of the formal subjective evaluation for the verification test. These include:

* The selection, the direction, the coordination and the instructions to the test laboratory(ies).
* The design, the supervision of the conduction of the formal subjective test.
* The collection and the statistical analysis of the data resulting from the test, to be submitted in a report to AG 5.
  1. Test Design

The test is conducted following the Degradation Category Rating (DCR) Rec. ITU-T P.910 [12] method using naive test subjects.

* 1. Renderer and generation of video sequences

The video sequences used in the test are generated using the MPEG PCC renderer software [9] . The renderer can be configured to output video sequences with the following video parameters:

* Video resolution: progressive uncompressed 1920x1080p Full HD format
* Frame rate: the frame rate is fixed to 30 frames per second
* Length: 10 seconds
* Color space: ITU-R BT.709
* Sub-sampling: 4:2:0 Y-Cb-Cr 8 bit

The background color shall be set to a medium grey (RGB 128/128/128) and the floor color shall be set to a slightly darker grey such as (RGB 95/95/95).

The MPEG PCC renderer software used for rendering purposes can be accessed via MPEG GIT with the following link:

https://git.mpeg.expert/MPEG/3dgh/v-pcc/software/mpeg-pcc-renderer

Use this commit version: e42aff092d7217de5cb6bb615b1c3b30ccf359d8.

The rendering viewpoint/camera path will follow a pre-defined path selected by the MPEG subjective Test Coordinator and the WG 7 chair for each test material. Selected camera paths are not known to the proponents before CfP answer submission deadline thus preventing any potential of optimizing a codec for these tests.

The generation of camera paths is described in subsection 2.3 of [9] with the resulting path stored in a camerapath.txt file.

The generation of the videos with default parameters as described above by using the example shell script below:

./mpeg-pcc-renderer/scripts/renderer.sh -i ./dec/PnnTx[S/D/K]mmRy/ \  
 --camera=camerapath\_xxxx.txt

* 1. Delivery of submissions for subjective testing

Proponents shall provide decoder executable (including everything that is needed for using the decoder, such as decoding scripts), compressed bitstreams, and corresponding MD5 checksums. Decoded point clouds need to be provided for crosscheck purposes. See the timeline planned in Table 3.

* 1. Laboratory setup

The Test Coordinator will select the test laboratories and make sure that these comply with ITU-R BT. 500 [1] and MPEG guidelines [2] . Test Coordinator will agree on handling of scoring tool (e.g. paper sheet or electronic). Test Coordinator will deliver a consolidated document with all test results to WG 7.

* 1. Selection and training of test subjects

Test subjects will be naive viewers and each subject will be carefully screened for visual acuity (Snellen Chart) and color blindness (Ishihara tables). The Test Coordinator will make sure that the test panel has a state-of-the-art number of test subjects.

Before participating in a test experiment, all subjects will participate in a training activity during which a detailed explanation of the test scope, the test method, the voting procedure, and the kind of assessment they are expected to do, will be provided.

The training activity will include a short demonstration session conducted after the training explanation, to let the subjects practice with the scoring procedure and to allow them to familiarize themselves with the video sequences; some explanations will be provided about the kind of impairments to look for in the videos that they will see.

1. Information Form

Prospective proponent shall finish the “Information Form” specified in the annex and submit it together with the documentation.

* 1. Information Form

1. Title of the proposal
2. Organization (name of proposing company, name and email of contact person)
3. Which test conditions are covered by your proposal?

Table 30 Check the appropriate box that applies to the submission.

|  |  |  |
| --- | --- | --- |
| Competition track | Use Case | |
| Dense | Sparse |
| Track 1: Geometry-only |  |  |
| Track 2: Geometry + Attribute |  |  |

1. Proponents shall provide information on how the requirements for AI-based graphics coding of dynamic point clouds in [11] are fulfilled, in the form of a table. This table should list the requirement, the fulfillment (yes/no). A short rationale why the requirement is fulfilled should be described.

Table 31 Fulfillment of requirements for AI-based graphics coding of dynamic point clouds

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Technology | Category 1  (Dense Static PCs for Immersive Applications) | | Category 2  (Dense Dynamic PCs for Immersive Applications) | | Category 3.A  (Sparse Dynamic PCs for LiDAR) | |
| **Req.** | **Fulfilled?** | **Req.** | **Fulfilled?** | **Req.** | **Fulfilled?** |
| a) Lossy compression | P |  | P |  | P |  |
| b) Lossless geometry compression | P |  | P |  | P |  |
| c) Lossless attribute compression | P |  | P |  | P |  |
| d) Near-lossless geometry compression | *o* |  | *o* |  | P |  |
| e) Near-lossless attribute compression | *o* |  | *o* |  | *o* |  |
| f) Temporal variations | - |  | P |  | P |  |
| g) Low latency | P |  | P |  | P |  |
| h) Low complexity | P |  | P |  | P |  |
| i) Temporal scalability | - |  | P |  | P |  |
| j) Spatial scalability | *o* |  | *o* |  | *o* |  |
| k) Region-based scalability | *o* |  | *o* |  | *o* |  |
| l) Quality scalability | *o* |  | *o* |  | *o* |  |
| m) Spatial random access | *o* |  | *o* |  | *o* |  |
| n) Temporal random access | - |  | P |  | P |  |
| o) Error resilience | *o* |  | P |  | *o* |  |
| p) Parallel encoding and decoding | *o* |  | *o* |  | *o* |  |
| q) Separable attribute and geometry coding | *o* |  | *o* |  | *o* |  |
| q-1) Geometry only coding | P |  | P |  | P |  |
| q-2) Multiple attribute coding | P |  | P |  | P |  |
| r) Geometry precision | At least Up to 20 |  | At least Up to 12 |  | At least Up to 18 |  |
| s) Model architecture | P |  | P |  | P |  |
| t-1) On the fly Model Update | P |  | P |  | P |  |
| t-2) On demand Model Update & download | P |  | P |  | P |  |
| u) Inference Reproducibility | P |  | P |  | P |  |

(‘P’ = Required ‘*o*’ = Optional ‘-’ = Not applicable)

Explanation of the requirement items above can be found in the requirement document [11] .

1. Excel Reporting Template
   1. Introduction

An Excel reporting template is provided together the Call for Proposal document. It is composed of six reporting sheets: Cover sheet, Summary sheet, Complexity sheet, Graphs sheet, T1 lossyG-only sheet and T2 lossyG-lossyA sheet.

* 1. "Cover" sheet

The cover sheet has the tables required to indicate basic information of the proposed technology. The tables in this sheet is described in Annex F.

* 1. "Summary” sheet

The summary sheet is composed of two tables. Each table corresponds to one competition track or test condition. The purpose of this sheet is to provide a global understanding of the coding performance in terms of bitrate savings relative to anchor codec. Algorithm complexity is roughly reflected by the relative runtime.

All the numbers in this sheet are automatically filled up. Proponents should not manually edit this sheet.

* 1. "Complexity” sheet

The first table in the complexity sheet is for complexity assessment. It has three parts: model, training and inference as specified in Table 15 of this document.

The other two tables are for computing environments as specified in Table 16. If inferencing environment is different from training, they can be separately reported.

Note that the number of columns to be finished in each table depends on the proponent’s technology. In the worksheet, four (4) models are listed as examples in case that the proposed codec has four (4) neural network models from four (4) different training / refining.

* 1. “Graphs” sheet

The graphs sheet is to provide RD plots. One can first choose the “competition track” in Cell B1, then select the sequence/object from a list in Cell B2. Six (6) RD plots can be shown at one time. For each RD plot, the axis X and Y can be customized.

Table G1. Main variables in the “graphs” sheet

|  |  |
| --- | --- |
| Variable | Description |
| eqn.bpp.geometry | Geometry bitrate |
| eqn.bpp | Geomery + attribute bitrate |
| dec.d1-psnr | D1 Geometry PSNR |
| dec.d2-psnr | D2 Geometry PSNR |
| dec.y-psnr | Luma End-to-end PSNR |
| dec.cb-psnr | Chroma Cb End-to-end PSNR |
| dec.cr-psnr | Chroma Cr End-to-end PSNR |
| dec.reflectance-psnr | Reflectance End-to-end PSNR |

* 1. Raw data sheet

The last two sheets are for importing raw data, and computing the reporting data. The first is for “T1 geometry-only coding”. And the second is for “T2 geometry and attribute coding”. We take T2 sheet as example to introduce the structure of the sheet.

Column A and B are for the list of sequences / objects. The first row-wise block is for dynamic dense point cloud sequences. The second row-wise block is for static dense point cloud objects. The third is for dynamic sparse point clouds.

Column C to G are basic information about each sequence. For each sequence, there are four rows, each for one rate point.

The Column I to AB is finished with the results of the anchor results. Proponents shall not edit the data in this section. The exception is that proponents shall run the anchors on the same platform where their proposal will run, and collect their anchor running time in column Y to AB. Note that the GPU complexity anchor time shall be typed into Cell Z118 based on the instruction in Annex D. In addition, data from column N to X in blue background are data from logged information of the anchor codecs. Data from column I to M in gray are automatically computed in this sheet.

Similarly, the data for a proposal is structured from column AD to AW.

The last section from column AY to BS is to prepare for reporting. The BD bitrate savings are computed for each sequence based on the 4 rate points. Relative runtime is also computed. The category specific reporting results are finally managed from row 113 and 116.