International Standard

ISO/IEC 23090-18

First edition
2024-01

Information technology — Coded representation of immersive media —

Part 18: Carriage of geometry-based point cloud compression data

Technologies de l'information — Représentation codée de média immersifs —

Partie 18: Transport des données de compression des nuages de points basée sur la géométrie

Reference number
ISO/IEC 23090-18:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23090 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Advances in 3D capturing and rendering technologies have unleashed a new wave of innovation in Virtual/Augmented/Mixed reality (VR/AR/MR) content creation and communication. Point clouds have arisen as one of the main representations for such applications. Geometry-based point cloud compression data is used for representing sparse dynamically varying point clouds such as those used in vehicular LiDAR or 3D mapping, as well as dense static point clouds used in cultural heritage, and industrial applications.

This document addresses technologies defining the carriage of geometry-based point cloud compression data for storage and delivery purposes. This document includes(but is not limited to):

— Storage of geometry-based point cloud compression data and the associated metadata using the ISO Base Media File Format (ISOBMFF) as specified in ISO/IEC 14496-12;

— Storage of non-timed geometry-based point cloud compression data and the associated metadata using HEVC Image File Format (HEIF) as specified in ISO/IEC 23008-12;

— Encapsulation, signalling, and streaming of geometry-based compression data in a media streaming system, for example, dynamic adaptive streaming over HTTP (DASH) as specified in ISO/IEC 23009-1 or MPEG media transport (MMT) as specified in ISO/IEC 23008-1.

Information technology — Coded representation of immersive media —

Part 18:
Carriage of geometry-based point cloud compression data

1 Scope

This document specifies a media format that enables the storage and delivery of geometry-based point cloud compression data. The geometry-based point cloud compression data can be timed or non-timed. It supports flexible extraction of geometry-based point cloud compression data at delivery or decoding time.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9834-1, *(Rec. ITU-T X.660), Information technology — Procedures for the operation of object identifier registration authorities — Part 1: General procedures and top arcs of the international object identifier tree*

ISO/IEC 9834-8, *(Rec. ITU-T X.667), Information technology — Procedures for the operation of object identifier registration authorities — Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers*

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 23008-1:2023, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 1: MPEG media transport (MMT)*

ISO/IEC 23008-12:2022, *Information technology — MPEG systems technologies — Part 12: Image File Format*

ISO/IEC 23009-1:2022, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*

ISO/IEC 23090-9:2023, *Information technology — Coded representation of immersive media (MPEG-I) — Part 9: Geometry-based point cloud compression*

IEEE 754-2019, *IEEE Standard for Floating-Point Arithmetic.*

W3C Recommendation, *XML schema part 1: Structures*

W3C Recommendation, *XML schema part 2: Datatypes*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23090-9 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
attribute track
volumetric visual track which carries ADUs of one instance of a particular attribute component of coded point cloud frames

3.2
attribute tile track
G-PCC tile track (3.10) which carries ADUs of one instance of a particular attribute component corresponding to one or more G-PCC tiles

3.3
G-PCC bitstream
sequence of bits of the coded point cloud sequence

Note 1 to entry: G-PCC bitstream is specified in ISO/IEC 23090-9.

3.4
G-PCC bitstream track
volumetric visual track which carries the entire coded point cloud sequence

3.5
G-PCC content
volumetric visual media that that consists of one or more point cloud frames

Note 1 to entry: Each point cloud frame includes a number of points, identified by their positions in 3D space, and the associated attributes (e.g. colour) at a particular time instance.

3.6
G-PCC component track
volumetric visual track which carries DUs of one instance of a particular G-PCC component

Note 1 to entry: There are two types of G-PCC component tracks: one is *geometry track* (3.7) and the other is *attribute track* (3.1).

3.7
geometry track
volumetric visual track which carries GDUs of the coded point cloud frames

3.8
geometry tile track
G-PCC tile track (3.10) which carries GDUs of one or more G-PCC tiles

3.9
G-PCC player
application responsible for receiving files/segments or accessing files locally, decapsulating files/segments, decoding the G-PCC bitstream, reconstructing point cloud frames from the decoded G-PCC bitstream, and rendering the point cloud frames

3.10
G-PCC tile track
volumetric visual track which carries either any of DUs corresponding to one or more G-PCC tiles

3.11
G-PCC tile base track
volumetric visual track which carries DUs which can be applied across the associated *G-PCC tile tracks* (3.10)

4 Abbreviated terms

ADU	attribute data unit (specified in ISO/IEC 23090-9)
APS	attribute parameter set (specified in ISO/IEC 23090-9)
DASH	dynamic adaptive streaming over HTTP (specified in ISO/IEC 23009-1)
DU	data unit (specified in ISO/IEC 23090-9)
FBDU	frame boundary marker data unit (specified in ISO/IEC 23090-9)
FSAP	frame-specific attribute properties (specified in ISO/IEC 23090-9)
GDU	geometry data unit (specified in ISO/IEC 23090-9)
GPS	geometry parameter set (specified in ISO/IEC 23090-9)
HTTP	Hyper-text transfer protocol
HEIF	HEVC image format (specified in ISO/IEC 23008-12)
ISOBMFF	ISO base media file format (specified in ISO/IEC 14496-12)
MMT	MPEG media transport (specified in ISO/IEC 23008-1)
SPS	sequence parameter set (specified in ISO/IEC 23090-9)

5 Overview

5.1 Overall architecture for carriage of geometry-based point cloud compression data

Geometry-based point cloud compression (G-PCC) provides the method for efficiently compressing the point cloud sequence which consists of one or more point cloud frames. Each point cloud frame consists of a number of points and each point is a tuple of a three-dimensional position and attribute values for every attribute present in the point cloud.

The coded point cloud sequence forms a G-PCC bitstream comprising of data represents a volumetric encoding of point clouds consisting of a sequence of point cloud frames. Each point cloud frame includes a number of points, identified by their positions in 3D space, and their associated attributes at a particular time instance. The number of points can vary from one frame to another.

parameter sets and slices of coded point cloud frames. Every slice includes a GDU which codes the slice geometry and ADUs or defaulted attribute DUs which code the slice attributes. The group of slices may be associated with spatial regions in a point cloud to aid spatial access.

Figure 1 shows the overall architecture for a typical content flow process for carriage of G-PCC data and it is applicable to both live and on-demand use cases.

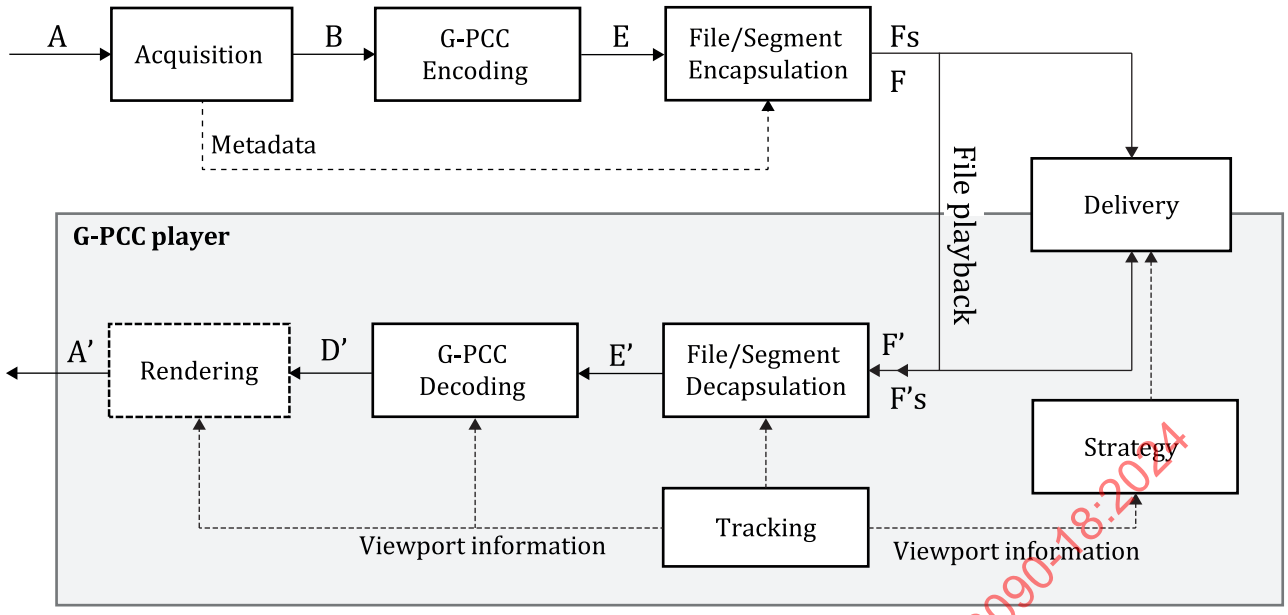

Figure 1 — Overall architecture flow process of for carriage of G-PCC data

A real-world visual scene (A) is captured by a set of cameras or a camera device with multiple lenses and sensors. A virtual visual scene (A) is also captured by virtual camera. The acquisition results in a point cloud sequence comprising of one or more point cloud frames (B). The point cloud sequence can be timed or non-timed. Each point cloud frame includes a set of points, identified by their positions in a three-dimensional Cartesian coordinate system, and associated attributes at a particular time instance. All points in the same point cloud frame have the same number of attributes. The number of points may vary from one frame to another. Each point cloud frame is coded as a sequence of slices and each slice comprises a sequence of encapsulated DUs. One or multiple point cloud frames are multiplexed into a G-PCC bitstream(E). The G-PCC bitstream are then encapsulated into a media file for file playback (F) or a sequence of an initialization segment and media segments for streaming (F_s), according to a particular media container file format. The metadata which can contribute to interpret and to consume the point cloud frames is encapsulated into the file or the segments. The point cloud metadata can describe, for example, the mapping between points to spatial regions within a point cloud. The segments F_s are delivered using a delivery mechanism to a G-PCC player.

The file that the file encapsulator outputs (F) is identical to the file that the file decapsulator inputs (F').

The G-PCC player processes the file (F') or the received segments (F'_s) and extracts the G-PCC bitstream (E') and parses the metadata. The G-PCC bitstream is then decoded into one or multiple point cloud frames (D') and the point clouds are reconstructed from the decoded point cloud frames (D'). The reconstructed point clouds are rendered and displayed onto the screen of a head-mounted display or any other display device based on the viewport information, such as the current viewing position, viewing orientation, or the field of view information, which is determined by various types of sensors. Besides being used by the G-PCC player to access the appropriate part of the point clouds, the viewport information can also be used for determining which tracks are extracted from the file. In viewport-dependent delivery, the viewport information is also passed to the strategy module, which determines the segments to be received based on the current viewport.

This process is applicable to both live and on-demand use cases.

The following interfaces are specified in this document:

— F/F': media file including the specification of the track formats in Clause 7 for timed G-PCC data, in Clause 8 for non-timed G-PCC data, and in Clause 9 for metadata in ISOBMFF.

— Clause 10 specifies the delivery related interfaces for DASH delivery.

— Clause 11 specifies the delivery related interfaces for MMT delivery.

The other interfaces in Figure 1 are not specified in this document.

5.2 Referenceable code points

5.2.1 Brands

The brands are used in this document to indicate conformance points to an encapsulation mode and a specific set of tools that are defined in this document. It may be indicated in the `FileTypeBox`.

The brands specified in this document are listed in Table 1 and defined in Annex A.

Table 1 — Brands specified in this document

Brand identifier	Subclause in this document	Description
gpst	A.2.1	Single track encapsulation
gpmt	A.2.2	Multiple track encapsulation
gppa	A.2.3	Encapsulation with partial access support
gpci	A.3	Non-timed G-PCC encapsulation

5.2.2 Sample entry type

The sample entry type specified in this document are listed in Table 2.

Table 2 — Sample entry types specified in this document

Sample entry type	Subclause in this document	Description
gpe1	7.3.2	For use with the single track encapsulation with all parameter set data units carrying SPS, GPS, and APS carried in decoder configuration record
gpeg	7.3.2	For use with the single track encapsulation with all parameter set data units carrying SPS, GPS, and APS carried in decoder configuration record and in track samples
gpc1	7.4.2	For use with the multiple track encapsulation with all parameter set data units carrying SPS, GPS, and APS carried in decoder configuration record
gpcg	7.4.2	For use with the multiple track encapsulation with all parameter set data units carrying SPS, GPS, and APS carried in decoder configuration record and in track samples
gpeb	7.5.2.1	For use with a tile base track with G-PCC tile track(s) containing DUs of all components
gpcb	7.5.2.1	For use with a tile base track with G-PCC tile track(s) containing DUs of one instance of a particular component
gpt1	7.5.3.1	For use with a G-PCC tile track
gpdr	9.1.3.2	For use with a timed metadata track indicating the dynamic spatial regions that are dynamically changing over time
gpdv	9.2.4.2	For use with a timed metadata track indicating viewport information that are dynamically changing over time

5.2.3 Box types

The box types specified in this document are listed in bold in Table 3. Mandatory boxes are marked with an asterisk. Box types without a four-character code are marked with '-' in the structure.

Table 3 — Box types specified in this document and their relation to boxes not specified in this document

Box types, structure, and cross-reference												
moov										*	ISOBMFF	*container for all the metadata*
	trak									*	ISOBMFF	*container for an individual track or stream*
		mdia								*	ISOBMFF	*container for the media information in a track*
			minf							*	ISOBMFF	*media information container*
				stbl						*	ISOBMFF	*sample table box, container for the time/space map*
					stsd					*	ISOBMFF	*sample descriptions (codec types, initialization etc.)*
						-					ISOBMFF	*visual sample entry*
						-					6.1.3	*volumetric visual sample entry*
							gpcC				7.2.2	*G-PCC decoder configuration box*
							ginf				7.2.3	*G-PCC component information box*
							gptC				7.5.3.1	*G-PCC tile configuration box*
							gpsr				8.3.3	*Static spatial region information box*
							gvpC				9.2.3	*Viewport information Configuration Box*
						gpdr					9.1.3.2	*Dynamic spatial region timed metadata sample entry*
							gpsr				8.3.3	*Static spatial region information box*
						gpdv					9.2.4.2	*Viewport information timed metadata sample entry*
							gvpC				9.2.3	*Viewport information Configuration Box*
meta											ISOBMFF	*Metadata*
	grpl										ISOBMFF	*group list box*
		vpta									8.4.1	*Viewport association box*
	iprp										ISOBMFF	*item properties box*
		ipco									ISOBMFF	*item property container box*
			gpcC								8.3.1	*G-PCC configuration item property*
			ginf								8.3.2	*G-PCC component information item property*
			gpsr								8.3.3	*G-PCC spatial region item property*
			gpti								8.3.4	*G-PCC tile information item property*

5.2.4 Track reference types

The track reference types specified in this document are listed in Table 4.

Table 4 — Track reference types specified in this document

Track reference type	Subclause in this document	Description
gpca	7.4.4	Referenced track is an attribute track
gpbt	7.5.5	Referenced track is a G-PCC tile track

5.2.5 Entity grouping types

The entity grouping types specified in this document are listed in Table 5.

Table 5 — Entity grouping types specified in this document

Entity groping type	Subclause in this document	Description
vpta	8.4.1	Viewport association between G-PCC items and the viewport information timed metadata track

5.2.6 Sample grouping types

The sample grouping types specified in this document are listed in Table 6.

Table 6 — sample grouping types specified in this document

Sample grouping type	Subclause in this document	Description
gtii	7.2.4	Tile inventory sample group

5.2.7 Uniform resource names

The URNs specified in this document are listed in Table 7.

Table 7 — URNs specified in this document

URN	Subclause in this document	Description
urn:mpeg:mpegI:gpcc:2023	10.2.2.1	Namespace for the XML elements and attributes specified in this document
urn:mpeg:mpegI:gpcc:2023:component	10.2.2.2	Scheme identifier for the G-PCC component DASH MPD descriptor
urn:mpeg:mpegI:gpcc:2023:gpc	10.2.2.3	Scheme identifier for the G-PCC content DASH MPD descriptor
urn:mpeg:mpegI:gpcc:2023:gpsr	10.3.1	Scheme identifier for the G-PCC static spatial region DASH MPD descriptor
urn:mpeg:mpegI:gpcc:2023:tileID	10.3.3.1	Scheme identifier for the G-PCC tile id DASH MPD descriptor
urn:mpeg:mpegI:gpcc:2023:rv	10.4.1	Scheme identifier for the static recommended viewports DASH MPD descriptor
urn:mpeg:mmt:app:gpcc:2023	11.3.1	Scheme identifier for the MMT G-PCC specific signalling messages

6 Volumetric media

6.1 Volumetric visual media

6.1.1 General

A volumetric visual track shall be identified by the volumetric visual media handler type `'volv'` in the `HandlerBox` of the `MediaBox`, as defined in ISO/IEC 14496-12, and by a volumetric visual media header as defined in subclause 6.1.2.

Multiple volumetric visual tracks can be present in the file.

6.1.2 Volumetric visual media header

6.1.2.1 Definition

Box Type: `'vvhd'`
Container: `MediaInformationBox`
Mandatory: Yes
Quantity: Exactly one

Volumetric visual tracks shall use a `VolumetricVisualMediaHeaderBox` in the `MediaInformationBox` as defined in ISO/IEC 14496-12.

6.1.2.2 Syntax

```
aligned(8) class VolumetricVisualMediaHeaderBox
   extends FullBox('vvhd', version = 0, 1) {
}
```

6.1.2.3 Semantics

`version` is an integer that specifies the version of this box.

6.1.3 Volumetric visual sample entry

6.1.3.1 Definition

Volumetric visual tracks shall use a `VolumetricVisualSampleEntry`.

6.1.3.2 Syntax

```
class VolumetricVisualSampleEntry(codingname)
   extends SampleEntry (codingname){
   unsigned int(8)[32] compressorname;
   // other boxes from derived specifications
}
```

6.1.3.3 Semantics

`compressorname` is a name, for informative purposes. It is formatted in a fixed 32-byte field, with the first byte set to the number of bytes to be displayed, followed by that number of bytes of displayable data encoded using UTF-8, and then padding to complete 32 bytes total (including the size byte). The field may be set to 0.

6.1.4 Volumetric visual sample group entry

```
abstract class VolumetricVisualSampleGroupEntry (unsigned int(32) grouping_type)
extends SampleGroupDescriptionEntry (grouping_type)
{
}
```

6.1.5 Volumetric visual samples

The format of a volumetric visual sample is defined by the coding system.

7 Timed G-PCC data storage in ISOBMFF

7.1 General

This clause specifies the below encapsulation of G-PCC bitstream in tracks within a file, and only one of the below encapsulations shall be used at the same time.

— Single track encapsulation of G-PCC bitstream, where one track carries the entire coded point cloud sequence, as specified in subclause 7.3.

— Multiple track encapsulation, where the coded bitstream of single G-PCC component of the point cloud sequence is encapsulated into a separate track, as specified in subclause 7.4.

— Encapsulation with G-PCC tile tracks, where DUs of one or more G-PCC tiles are encapsulated into a separate track, as specified in subclause 7.5.

The order of G-PCC slices and DUs of the output G-PCC bitstream after decapsulation should be maintained as the order of G-PCC slices and DUs of the original G-PCC bitstream.

Annex D provides a summary of the sample entry types and sample formats for tracks defined in this document.

7.2 Common boxes and data structures

7.2.1 G-PCC decoder configuration record

7.2.1.1 Definition

This subclause specifies the G-PCC decoder configuration information for ISO/IEC 23090-9 content.

This G-PCC decoder configuration record contains the information for configuration and initialization of G-PCC decoder. It contains a `configurationVersion` field. This edition of this document defines version 1 of this record. Incompatible changes to the record will be indicated by a change of configuration version number. Compatible extensions to this record will extend it and will not change the configuration version number. The G-PCC player shall not attempt to decode this record or the G-PCC bitstreams to which it applies if the value of `configurationVersion` is not supported or recognized.

The values for simple_profile_compliant, `dense_profile_compliant`, `predictive_profile_compliant`, and `main_profile_compliant` shall indicate a profile to which the G-PCC bitstream associated with this configuration record conforms. The value for `level_idc` shall indicate a level of capability equal to or greater than the highest level indicated for the highest tier in all the parameter sets. These values to indicate the conformance to a certain profile and `level_idc` shall be valid for all parameter sets that are activated when the G-PCC bitstream described by this record is decoded (referred to as "all parameter sets" below). The G-PCC player ignores the G-PCC bitstream if the indicated profiles are not supported or recognized.

The `setupUnit` array shall include data units that are constant for the bitstream referred to by the sample entry in which the decoder configuration record is present. The type of data units is restricted to indicate SPS, GPS, and APS.

7.2.1.2 Syntax

```
aligned(8) class GPCCDecoderConfigurationRecord {
   unsigned int(8)      configurationVersion = 1;
   unsigned int(2)      reseverd = 1;
   unsigned int(1)      simple_profile_compliant;
   unsigned int(1)      dense_profile_compliant;
   unsigned int(1)      predictive_profile_compliant;
   unsigned int(1)      main_profile_compliant;
   unsigned int(18)     reserved_profile_18bits
   unsigned int(8)      level_idc;
   unsigned int(8)      numOfSetupUnits;
   for (i=0; i<numOfSetupUnits; i++) {
      tlv_encapsulation setupUnit;    //as defined in ISO/IEC 23090-9, Annex B
   }
   // additional fields
}
```

7.2.1.3 Semantics

`configurationVersion` is a version field. Incompatible changes to the record are indicated by a change of the value of this field. In this edition of the document, it shall be equal to 1.

`simple_profile_compliant` equal to 1 specifies that the G-PCC bitstream conforms to the Simple profile defined in ISO/IEC 23090-9:2023, Annex A. `simple_profile_compliant` equal to 0 specifies that the G-PCC bitstream does not conform to the Simple profile.

`dense_profile_compliant` equal to 1 specifies that the G-PCC bitstream conforms to the Dense profile defined in ISO/IEC 23090-9:2023, Annex A. `dense_profile_compliant` equal to 0 specifies that the G-PCC bitstream does not conform to the Dense profile.

`predictive_profile_compliant` equal to 1 specifies that the G-PCC bitstream conforms to the Predictive profile defined in ISO/IEC 23090-9:2023, Annex A. `predictive_profile_compliant` equal to 0 specifies that the G-PCC bitstream does not conform to the Predictive profile.

`main_profile_compliant` equal to 1 specifies that the G-PCC bitstream conforms to the Main profile defined in ISO/IEC 23090-9:2023, Annex A. `main_profile_compliant` equal to 0 specifies that the G-PCC bitstream does not conform to the Main profile.

NOTE When none of `simple_profile_compliant`, `dense_profile_compliant`, `predictive_profile_compliant`, and `main_profile_compliant` set to 1, it means the G-PCC bitstream does not conform to profiles defined in ISO/IEC 23090-9:2023, Annex A.

`reserved_profile_18bits` shall be equal to 0 in this configuration version 1 of this record. Other values for `reserved_profile_18bits` are reserved for future use by ISO/IEC. The G-PCC player conforming to this edition of the document shall ignore the value of `reserved_profile_18bits`.

`level_idc` indicates a level to which the G-PCC bitstream conforms as specified in ISO/IEC 23090-9:2023, Annex A. The bitstream shall not contain values of `level_idc` other than those specified in ISO/IEC 23090-9:2023, Annex A. Other values of `level_idc` are reserved for future use by ISO/IEC.

`numOfSetupUnits` specifies the number of following `setupUnit` present in the decoder configuration record.

`setupUnit` contains one data unit carrying one of SPS, GPS, and APS as defined in ISO/IEC 23090-9.

7.2.2 G-PCC decoder configuration box

7.2.2.1 General

A G-PCC decoder configuration box includes the `GPCCDecoderConfigurationRecord` which contains the information for configuration and initialization of G-PCC decoder as defined in subclause 7.2.1.

This edition of this document, the `version` shall be equal to 0.

7.2.2.2 Syntax

```
aligned(8) class GPCCConfigurationBox extends FullBox('gpcC', 0, 0) {
   GPCCDecoderConfigurationRecord();
}
```

7.2.2.3 Semantics

`GPCCDecoderConfigurationRecord` is defined in subclause 7.2.1.

7.2.3 G-PCC component information box

7.2.3.1 Definition

This box specifies G-PCC component information including the type of G-PCC components (e.g. geometry, attribute). When this box is present in a sample entry of a track, it indicates the type of G-PCC components carried by the respective track. When this box is present in a sample entry of the G-PCC attribute track, it also provides the attribute name, index and an optional attribute type information by the attribute label or the international object identifier of G-PCC attribute component carried by the respective G-PCC attribute track.

The value of `flags` in this box indicates the presence of attribute type information and how the attribute type is indicated if the attribute type information presents as follows:

flags value	Description
0x000000	No attribute type information presents
0x000001	The attribute type information presents, and the attribute type is indicated by the value of attr_label.
0x000002	Reserved
0x000003	The attribute type information presents, and the attribute type is indicated by the value of attr_label_oid.
0x00004.. 0xFFFFFF	Reserved.

7.2.3.2 Syntax

```
aligned(8) class GPCCComponentInfoBox extends FullBox('ginf', 0, flags){
   GPCCComponentInfoStruct(flags);
}

aligned(8) class GPCCComponentInfoStruct(bit(24)   flags){
   unsigned int(8)   comp_type;
   if(comp_type == 4){
      unsigned int(8)   attr_index;
      if(flags == 0x000001){
         unsigned int(3)   attr_label;
      }elseif(flags == 0x000003){
         oid    attr_label_oid;
      }
      utf8string   attr_name;
   }
}
```

7.2.3.3 Semantics

comp_type identifies the type of G-PCC component as specified in Table 8. In this edition of the document, the value of this field shall be either 2 or 4.

Table 8 — G-PCC Component Type

comp_type value	Description
1	Reserved
2	Geometry component
3	Reserved
4	Attribute component
5..31	Reserved.

attr_index identifies an attribute component by indicating the order of the attribute signalled in the SPS. The first four bits of this value shall be equal to sps_seq_parameter_set_id of the SPS in G-PCC bitstream and the last four bits of this value shall be equal to attrIdx signalled in the SPS indicated by the sps_seq_parameter_set_id.

attr_label identifies the type of attribute component as specified in ISO/IEC 23090-9:2023, Table 9.

attr_label_oid specifies an ASN.1 object identifier value in the international object identifier tree. The international object identifier shall either be assigned by a registration authority in accordance with Rec. ITU-T X.660 | ISO/IEC 9834-1 or generated without registration using a universally unique identifier (UUID) as specified by Rec. ITU-T X.667 | ISO/IEC 9834-8. The syntax of object identifier is described in ISO/IEC 23090-9:2023, subclause 11.4.7.1.

attr_name specifies a human-readable name for the type of G-PCC attribute components.

7.2.4 Tile inventory information sample group

7.2.4.1 Definition

Group Types: 'gtii'
Container: Sample Group Description Box ('sgpd')
Mandatory: No
Quantity: Zero or more

The use of 'gtii' for the grouping_type in sample grouping represents the assignment of samples in G-PCC track to the tile inventory information carried in this sample group. When a SampleToGroupBox with grouping_type equal to 'vaps' is present, an accompanying SampleGroupDescriptionBox with the same grouping type shall be present and contains the ID of the group that the samples belong to.

When the tile inventory data units are present in the G-PCC bitstream, it is carried as the below.

— When the G-PCC bitstream is carried by using G-PCC tracks with track type 'gpc1', the tile inventory information sample group with grouping type 'gtii' shall be present in the G-PCC geometry track. The G-PCC attribute tracks shall not contain the sample group with grouping type 'gtii'.

— Under the 'gpcg' sample entry, the tile inventory information may be present in a tile inventory information sample group description entry with grouping type 'gtii' or in the samples of the G-PCC geometry track.

— Under the 'gpe1' sample entry, the tile inventory information sample group with grouping type 'gtii' shall be present.

— Under the 'gpeg' sample entry, the tile inventory information may be present in a tile inventory information sample group description entry with grouping type 'gtii' or in the samples of the G-PCC bitstream track.

— When the G-PCC bitstream is carried by using G-PCC tile tracks, sample grouping type 'gtii' shall not be present with a track with a sample entry 'gpcb', 'gpeb', and 'gpt1'. The tile inventory information shall be available in G-PCC tile base track samples.

7.2.4.2 Syntax

```
aligned(8) class TileInventoryInfoEntry()
   extends VolumetricVisualSampleGroupEntry ('gtii'){
   //tlv_encapsulation as defined in ISO/IEC 23090-9
   tlv_encapsulation    tile_inventory_DU;
}
```

7.2.4.3 Semantics

tile_inventory_DU contains a tile inventory data unit, as instance of TLV encapsulation structure of tlv_type equal to 5 as defined in ISO/IEC 23090-9, which applies to associated samples in the track.

7.3 Single track encapsulation

7.3.1 General

A single track encapsulation of G-PCC bitstream requires that the entire G-PCC bitstream is carried by a G-PCC bitstream track. Each sample in the track corresponds to a single point cloud frame and contains at least one GDUs, zero or more ADUs, and zero or more parameter set data units.

More than one track for same G-PCC bitstream shall not be present. Any G-PCC component tracks shall not be present.

7.3.2 Sample entry

7.3.2.1 Definition

Sample Entry Type: `'gpe1'` or `'gpeg'`
Container: `SampleDescriptionBox`
Mandatory: A `'gpe1'` or `'gpeg'` sample entry is mandatory.
Quantity: One or more sample entries may be present.

A G-PCC bitstream track shall use the `GPCCBistreamSampleEntry` with a sample entry type of `'gpe1'` or `'gpeg'`. A G-PCC bitstream track sample entry shall contain a `GPCCConfigurationBox` as defined in subclause 7.2.2.

Under the `'gpe1'` sample entry, all parameter set data units carrying SPS, GPS, and APS shall be in the `setupUnit` array. Under the `'gpeg'` sample entry, the parameter set data units can be present in the `setupUnit` array, or in the stream.

A G-PCC bitstream track sample entry shall not contain a `GPCCComponentInfoBox` as defined in subclause 7.2.3.

7.3.2.2 Syntax

```
aligned(8) class GPCCBitstreamSampleEntry()
   extends VolumetricVisualSampleEntry (unsigned int type) {
   // type is 'gpe1' or 'gpeg'
   GPCCConfigurationBox      config;   //mandatory
}
```

7.3.2.3 Semantics

`compressorname` in the base class `VolumetricVisualSampleEntry` indicates the name of the compressor used with the value `"\013GPCC Coding"` being recommended; the first byte is a count of the remaining bytes, here represented by `\013`, which (being octal 13) is 11 (decimal), the number of bytes in the rest of the string.

`config` includes G-PCC decoder configuration record information, as defined in subclause 7.2.2.

7.3.3 Sample format

7.3.3.1 Definition

Each G-PCC sample corresponds to a single point cloud frame and shall contain one or more data units which belong to the same presentation time. A sample may be self-contained (e.g. a sync sample).

Figure 2 depicts an example of the sample structure of a G-PCC bitstream track when the original G-PCC bitstream contains geometry component and one attribute component. In this case, each sample contains at least one GDU, one ADU, or zero or more parameter set data units.

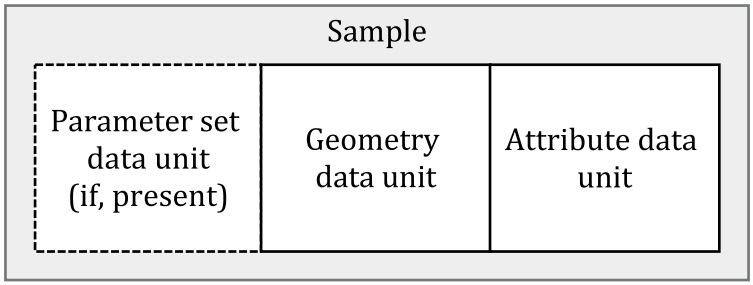

Figure 2 — Example of sample structure of a G-PCC bitstream track

When FSAP DUs are available in G-PCC bitstream, FSAP DUs shall be carried in the sample containing ADUs of the corresponding point cloud frame. If it is present, it shall precede any ADUs in the sample.

7.3.3.2 Syntax

```
aligned(8) class GPCCSample
{
   unsigned int GPCCLength = sample_size; //Size of Sample
   for (i=0; i< GPCCLength; )       // to end of the sample
   {
      unsigned int(8)    tlv_type;
      unsigned int(32)   tlv_num_payload_bytes;
      bit(tlv_num_payload_bytes*8)   data_unit;
      i += (1+4)+tlv_num_payload_bytes;
   }
}
```

7.3.3.3 Semantics

`tlv_type` identifies the syntax structure represented by the following `data_unit` as defined in ISO/IEC 23090-9:2023, Annex B.

`tlv_num_payload_bytes` specifies the length in bytes of the syntax element of the following `data_unit`.

`data_unit` contains a single DU which corresponds to the same point cloud frame.

7.3.3.4 Sub-sample

For the use of `SubSampleInformationBox`, a sub-sample is defined on the basis of the value of the `flags` field of the `SubSampleInformationBox`. The `flags` field specifies the type of sub-sample information given in this box as follows:

— 0: G-PCC data unit based sub-samples. A sub-sample contains only one DU.

— 1: Tile-based sub-samples. A sub-sample either contains one or more contiguous DUs corresponding to one G-PCC tile or contains one or more contiguous DUs which contain either each parameter set, tile inventory or frame boundary marker.

— Other values of `flags` are reserved.

The `subsample_priority` field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12.

The `discardable` field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded (e.g. the sub-sample consists of only tile inventory, frame boundary marker, and user data unit).

The `codec_specific_parameters` field of the `SubsampleInformationBox` is defined as follows:

```
if (flags == 0) {
   unsigned int(8)    payloadType;
   if (payloadType == 4) {
         unsigned int(6)     attrIdx;
         bit(18)        reserved = 0;
   }
   else
         bit(24)        reserved = 0;
} else if (flags == 1) {
   unsigned int(1)     tile_data_present;
   bit(7)             reserved = 0;
   if (tile_data_present)
         unsigned int(24)    tile_id;
   else
         bit(24)            reserved = 0;
}
```

`payloadType` provides the value of the `tlv_type` of the DU present in the sub-sample.

`attrIdx` indicates the `ash_attr_sps_attr_idx` of the `ADU` present in the sub-sample.

`tile_data_present` equal to 1 indicates that the sub-sample contains GDUs, defaulted ADUs, or ADUs corresponding to one G-PCC tile. `tile_data_present` equal to 0 indicates that the sub-sample contains `DUs` which contain either each parameter set, tile inventory, frame-specific attribute properties, or frame boundary marker.

`tile_id` indicates the index of the G-PCC tile, which the sub-sample is associated with, in the tile inventory.

For single track encapsulation, when sub-samples are present, the `SubSampleInformationBox` with `flags` equal to 0 in `SampleTableBox`, or in the `TrackFragmentBox` of each of its `MovieFragmentBoxes` shall be present.

7.4 Multiple track encapsulation

7.4.1 General

When the G-PCC bitstream is carried in multiple tracks, the coded bitstream of each G-PCC component is mapped to an individual track. There are two types of G-PCC component tracks: G-PCC geometry track and G-PCC attribute track. Each sample in each component track contains DUs of an instance of a single component, not both of GDUs and ADUs. When multiple attributes are present in the G-PCC bitstream, any attribute track shall not multiplex DUs of different attributes.

For multiple track encapsulation, the below conditions shall be satisfied:

— One geometry track shall present and shall be an entry point.

— One or more G-PCC attribute tracks can be present. The `track_in_movie` flags of each attribute track shall be set to 0.

— In each component track sample entry, one `GPCCComponentInfoBox` shall be present to indicate the role of the bitstream contained in the track.

— All component tracks shall have same values of `simple_profile_compliant`, `dense_profile_compliant`, `predictive_profile_compliant`, `main_profile_compliant`, and `level_idc` in a track sample entry.

— A `'gpca'` track reference from a geometry track to attribute tracks shall be present.

— All component tracks shall have the same implied or explicit edit lists.

— Any of G-PCC tile base track and tile tracks shall not be present.

Component tracks belonging to the same G-PCC bitstream are time-aligned. Samples that contribute to the same point cloud frame across different component tracks shall have the same presentation time. When any parameter sets are present in samples, the decoding time of parameter sets used for such samples shall be equal or prior to the decoding time of samples of corresponding DUs. When all parameter sets are present in samples across multiple tracks, the decoding time of samples containing the SPS shall be equal or prior to the decoding time of samples containing the GPS, or APS.

NOTE Synchronization between the elementary streams in the tracks are handled by the ISOBMFF track timing structures (`stts`, `ctts`, and `cslg`), or equivalent mechanisms in movie fragments.

When the G-PCC bitstream only contains the G-PCC geometry component and no G-PCC attribute component, this multiple track encapsulation shall not be applied.

7.4.2 Sample entry

7.4.2.1 Definition

Sample Entry Type: `'gpc1'` or `'gpcg'`
Container: `SampleDescriptionBox`
Mandatory: A `'gpc1'` or `'gpcg'` sample entry is mandatory
Quantity: One or more sample entries may be present

Component track shall use `GPCCSampleEntry` which extends `VolumetricVisualSampleEntry` with a sample entry type of `'gpc1'` or `'gpcg'`.

A component track sample entry shall contain `GPCCConfigurationBox` as defined in subclause 7.2.2 and `GPCCComponentInfoBox`, as defined in 7.2.3, respectively.

Under the `'gpc1'` sample entry, the `setupUnit` array of geometry track shall contain all DUs carrying SPS and GPS. The `setupUnit` array of attribute track shall contain all DUs carrying APS associated with samples in the track and shall not contain DUs carrying any of SPS and GPS. When multiple G-PCC attribute tracks present, the `setupUnit` array of G-PCC attribute tracks shall contain the same information.

Under the `'gpcg'` sample entry, DUs carrying SPS or GPS can be present in the `setupUnit` array of the sample entry or in the stream of geometry track. The DUs carrying APS can be present in the `setupUnit` array of the sample entry or in the stream of attribute track. Any DUs carrying SPS and GPS shall not be present in G-PCC attribute tracks.

7.4.2.2 Syntax

```
aligned(8) class GPCCSampleEntry()
      extends VolumetricVisualSampleEntry (unsigned int type) {
   //type is 'gpc1' or 'gpcg'
   GPCCConfigurationBox      config;
   GPCCComponentInfoBox      info;
   // optional boxes
}
```

7.4.2.3 Semantics

`compressorname` in the base class `VolumetricVisualSampleEntry` indicates the name of the compressor used with the value `"\013GPCC Coding"` being recommended, the first byte is a count of the remaining bytes, here represented by `\013`, which (being octal 13) is 11 (decimal), the number of bytes in the rest of the string.

`config` contains the information for configuration and initialization of G-PCC decoder as defined in subclause 7.2.2.

`info` indicates G-PCC component information carried in the respective track, as defined in subclause 7.2.3. It indicates the component type carried in the track. It also provides the attribute name, index, and optionally attribute type carried in the G-PCC attribute track.

7.4.3 Sample format

7.4.3.1 General

Each sample corresponds to a single point cloud frame and samples that contribute to the same point cloud frame across different component tracks shall have the same presentation time.

Each sample shall be comprised of one or more DUs of G-PCC component indicated in the `GPCCComponentInfoBox` in the sample entry and zero or more DUs carrying one of associated parameter sets. The DUs of G-PCC component which is not indicated in the sample entry shall not be present in samples. When DUs carrying any parameter sets are present in the sample, it shall be appeared before DUs of G-PCC component. When any FASP data units are available in the G-PCC bitstream, these FSAP data units shall be carried in the sample containing ADUs of the corresponding point cloud frame. If presents, it shall precede any ADUs in the sample.

Figure 3 describes an example of the sample structure of the geometry track. In this case, each sample contains at least one GDU, zero or more DUs carrying SPS or GPS. When DUs carrying SPS or GPS are present, it shall precede GDUs in the sample.

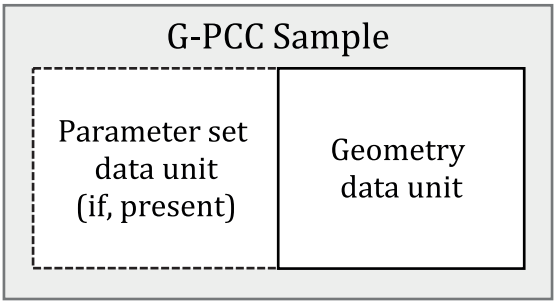

Figure 3 — Example of sample structure of geometry track

The syntax and semantics of the sample in component track as defined in subclause 7.3.3.2 and 7.3.3.3, respectively, are applied.

7.4.3.2 Sub-sample

For multiple track encapsulation without G-PCC tile tracks, when sub-samples are present, only `SubSampleInformationBox` with `flags` equal to 1 in `SampleTableBox`, or in the `TrackFragmentBox` of each of its `MovieFragmentBoxes` shall be present. The syntax and semantics of the sub-sample with `flags` equal to 1 as defined in subclause 7.3.3.4, shall be applied.

7.4.4 Track references

To link between different types of component tracks, the track reference tool of ISO/IEC 14496-12 shall be used.

One `TrackReferenceTypeBox` shall be added to a `TrackReferenceBox` within the `TrackBox` of the geometry track. The `TrackReferenceTypeBox` shall contain an array of `track_IDs` designating the tracks which the G-PCC track references. To link the geometry track to the attribute track, `reference_type` of a `TrackReferenceTypeBox` in the geometry track identifies the associated attribute tracks. The `4CCs` of these track reference types shall be:

— `'gpca'`: the referenced attribute track(s).

7.5 Encapsulation of tiled G-PCC bitstream

7.5.1 General

ISO/IEC 23090-9 provides the support for spatial random access with G-PCC tiles. The G-PCC tiles do not have coding dependencies with other G-PCC tiles in the same coded point cloud frame. Each G-PCC tile is described through a single 3D bounding box and an identifier, and it is associated with the group of slices.

There are cases where storing data units of G-PCC tiles in different tracks are useful for easy access to one or a few particular tile regions. For such cases, one or more G-PCC tiles are carried in separate tracks, the followings shall be satisfied:

— One G-PCC tile base track carrying all of parameter sets and tile inventory shall be present and shall be an entry point.

— One or more G-PCC tile tracks carrying data units of G-PCC tiles shall be present.

— A `'gpbt'` track reference from the G-PCC tile base track to G-PCC tile tracks shall be present.

— When DUs of each component corresponding to G-PCC tiles are mapped to separate tracks, one or more geometry tile tracks shall be present, and the same number of attribute tile tracks shall be present.

— When attribute tile tracks are present, the `track_in_movie` flag of each attribute tile track shall be set to 0.

— When attribute tile tracks are present, a `'gpca'` track reference shall be present in associated geometry tile tracks.

7.5.2 G-PCC tile base track

7.5.2.1 Sample entry

7.5.2.1.1 Definition

Sample Entry Type: `'gpeb'` or `'gpcb'`
Container: `SampleDescriptionBox`
Mandatory: No
Quantity: Zero or more sample entries may be present

The G-PCC tile base track shall use a `GPCCBitstreamSampleEntry`, which is defined in subclause 7.3.2, with sample entry type of `'gpeb'` or `'gpcb'`. Parameter sets, which apply to decode all samples of tile tracks, can be present in the `setupUnit` array of the sample entry or in the stream.

NOTE When all parameter sets are present in the sample entry of G-PCC tile base track and do not present in any of samples of G-PCC tile base track, these parameter sets in the sample entry of G-PCC tile base track are applied for decoding all samples of tile tracks.

When each G-PCC tile track carries DUs of all components present in the G-PCC bitstream, a G-PCC tile base track shall use a `GPCCBitstreamSampleEntry` with a sample entry type of `'gpeb'`.

When each G-PCC tile track carries DUs of one instance of a particular component corresponding to G-PCC tiles, a G-PCC tile base track shall use a `GPCCBitstreamSampleEntry` with a sample entry type of `'gpcb'`.

The sample entry names of a G-PCC tile base track and the corresponding G-PCC tile tracks shall be one of the rows in Table 9. Any types of combination of G-PCC tile base track and G-PCC tile tracks other than one defined in Table 9 are not considered in this edition of the document.

Table 9 — Sample entry type for a G-PCC tile base track according to associated G-PCC tile tracks

G-PCC tile base track sample entry type	Samples of associated G-PCC tile tracks
`gpeb`	Contain DUs of all components present in the G-PCC bitstream
`gpcb`	Contain DUs of one instance of a particular component indicated in the sample entry

The G-PCC tile base track sample entry can contain a `GPCCSpatialRegionInfoBox`, as defined in subclause 9.1.2. When the `GPCCSpatialRegionInfoBox` is not present in the sample entry, this G-PCC tile base track shall be associated with the dynamic spatial region timed metadata track.

7.5.2.2 Sample format

Each sample in G-PCC tile base track shall be comprised of DUs containing one of SPS, GPS, APS, and tile inventory. Any GDUs, defaulted ADUs or ADUs shall not be present in samples.

FSAP DUs shall be present in samples when the FSAP DUs are present in the G-PCC bitstream. Frame boundary marker DUs shall be present in samples when the frame boundary marker DUs are present in the G-PCC bitstream.

The syntax and semantics of the sample in G-PCC tile base track as defined in subclause 7.3.3.2 and 7.3.3.3, respectively, are applied.

7.5.2.2.1 Sub-sample

G-PCC tile base track shall not contain any sub-samples. Sub-samples shall not be present in G-PCC tile base track.

7.5.3 G-PCC tile tracks

7.5.3.1 Sample entry

7.5.3.1.1 Definition

Sample Entry Type: `'gpt1'`
Container: Sample Description Box
Mandatory: No
Quantity: Zero or more sample entries may be present

This sample entry describes samples of a G-PCC tile track. It shall use `GPCCTileSampleEntry` which extends `VolumetricVisualSampleEntry` with a sample entry type of `'gpt1'`.

The G-PCC tile identifiers present in samples of the respective G-PCC tile track are signalled in `GPCCTileSampleEntry`. The tile identifiers present in G-PCC tile track sample entry shall have the below constraints.

— When each G-PCC tile track carries DUs of all components present in the G-PCC bitstream, the G-PCC tile identifiers present in G-PCC tile track sample entry shall not overlap with the G-PCC tile identifiers present in other G-PCC tile track sample entry.

— When each G-PCC tile track carries DUs of one instance of a particular component, the G-PCC tile identifiers present in the geometry tile track sample entry shall not overlap with the G-PCC tile identifiers present in other geometry tile track sample entry. The G-PCC tile identifiers present in the attribute tile track sample entry shall be equals to the G-PCC tile identifiers in the associated geometry tile track sample entry.

When a G-PCC tile track carries DUs of all components, the `GPCCComponentInfoBox` shall not be present.

When a G-PCC tile track carries DUs of one instance of a particular component, the `GPCCComponentInfoBox` shall be present in the sample entry to indicate the type of the component carried in the track.

7.5.3.1.2 Syntax

```
aligned(8) class GPCCTileConfigurationBox extends Box('gptC',0,0) {
   unsigned int(1)    dynamic_num_tiles_flag;
   bit(7)             reserved = 0;
   unsigned int(16)   max_num_tile_ids_in_track;
   for(i=0; i < max_num_tile_ids_in_track; i++){
        unsigned int(16)   tile_id;
   }
}

aligned(8) class GPCCTileSampleEntry extends VolumetricVisualSampleEntry ('gpt1') {
   GPCCTileConfigurationBox   config;
   GPCCComponentInfoBox       info;      // optional
}
```

7.5.3.1.3 Semantic

`dynamic_num_tile_flag` indicates whether the number of G-PCC tiles or G-PCC tile identifiers present in samples of the respective G-PCC tile track are changed or not in the stream. Value 0 indicates the number of G-PCC tiles and G-PCC tile identifiers keep constant across all samples in the respective track and the number of G-PCC tiles and G-PCC tile identifiers corresponding to all samples are signalled in the sample entry. Value 1 indicates the number of G-PCC tiles or G-PCC tile identifiers presents in samples are changed in the stream.

`max_num_tiles_ids_in_track` indicates the maximum number of unique G-PCC tile identifiers present in the samples of the respective track. When `dynamic_num_tiles_flag` is 0, each sample in the respective G-PCC tile track shall contain `max_num_tile_ids_in_track` number of G-PCC tiles and the identifiers of those G-PCC tiles are not changing in the stream. When `dynamic_num_tiles_flag` is 1, each sample in the

G-PCC tile track contains at most `max_num_tile_ids_in_track` number of G-PCC tiles and the identifiers of those G-PCC tiles may be changed between samples.

`tile_id` indicates an identifier for a particular G-PCC tile corresponding to samples in the respective track. When `dynamic_tile_flag` is set to value 0, `tile_id` represents one of G-PCC tile identifier present in the tile inventory.

`info` is an instance of the `GPCCComponentInfoBox`, which indicates G-PCC component information carried in the respective track. When it is present, it indicates G-PCC component type carried in the track, and it also provides the attribute name, index, and optionally attribute type of G-PCC component carried in the attribute tile track.

7.5.3.2 Sample format

7.5.3.2.1 General

Each sample in G-PCC tile track is either

— One empty sample with the `sample_size=0` (representing a period of non-zero duration in which there is no data unit); or

— One G-PCC sample that the syntax and semantics as defined in subclause 7.3.3.2 and 7.3.3.3, respectively, are applied. Samples in G-PCC tile track shall not include DUs containing any parameter sets, tile inventory, and frame-specific attribute properties.

Samples corresponding to the same point cloud frame across G-PCC tile tracks shall have the same presentation time.

7.5.3.2.2 Sub-sample

When sub-samples are present, the `SubSampleInformationBox` with `flags` equal to 1 in `SampleTableBox`, or in the `TrackFragmentBox` of each of its `MovieFragmentBoxes` shall be present.

The syntax and semantics of the sub-sample with `flags` equal to 1 as defined in subclause 7.3.3.4, shall be applied.

When a G-PCC tile track carries DUs of all components and sub-samples are present, each sub-sample shall contain all DUs corresponding to each G-PCC tile. When a G-PCC tile track carries DUs of carries one instance of a particular component and sub-samples are present, each sub-sample shall contain DUs of component indicated in the `GPCCComponentInfoBox` in the sample entry.

7.5.4 Relationship between samples in G-PCC tile base track and tile track

The G-PCC tile base track sample required to decode the G-PCC tile track sample is identified using the presentation time of the sample. The presentation time of the corresponding G-PCC tile base track sample shall be either equal to or less than the presentation time of the G-PCC tile track samples. When the presentation time of samples of the G-PCC tile base track and G-PCC tile tracks are not equal, the G-PCC tile base track sample with a preceding presentation time closer to the presentation time of the G-PCC tile track sample is used for decoding the sample of G-PCC tile track or identifying the tile inventory information.

NOTE The presentation time of the sample in G-PCC tile base track or G-PCC tile track, can be derived by parsing the `'ctts'` table in the `CompositionOffsetBox` and `'stts'` table in the `TimeToSampleBox`, defined in ISO/IEC 14496-12, in the respective tracks.

The G-PCC tile base track sample carrying FSAP DUs required to decode samples in attribute tile tracks is identified using the presentation time of the sample. The presentation time of the corresponding G-PCC tile base track sample containing FSAP DUs shall be same as the presentation time of the attribute tile track samples carrying the corresponded ADUs.

7.5.5 Track references

The G-PCC tile base track is linked to the G-PCC tile tracks using track references tool of ISO/IEC 14496-12. One `TrackReferenceTypeBoxes` shall be added to a `TrackReferenceBox` within the `TrackBox` of the G-PCC tile base track. A new track reference type using the 4CC `'gpbt'` shall be used to link the G-PCC tile base track to the G-PCC tile tracks.

Under the G-PCC tile base track with a sample entry type of `'gpeb'`, the `'gpbt'` track reference shall be used to link the G-PCC tile base track to all G-PCC tile tracks.

Under the G-PCC tile base track with a sample entry type of `'gpcb'`, the `'gpbt'` track reference shall be used to link from the G-PCC tile base track to geometry tile tracks. Each geometry tile track is linked to corresponding attribute tile tracks using the `'gpca'` track reference type as defined in subclause 7.4.4.

7.6 Indication of alternatives

A point cloud sequence can be coded in alternatives. In such a case, alternatives of same point cloud sequence shall be indicated by the alternate track mechanism defined in ISO/IEC 14496-12 (i.e., `alternate_group` field of the `TrackHeaderBox`).

When each alternative coded bitstream of the same point cloud sequence is encapsulated in a single track, the G-PCC bitstream track carrying the G-PCC bitstream, which are alternatives of each other, shall have the same `alternate_group` value in their `TrackHeaderBox`.

When each alternative coded bitstream of a same point cloud sequence is encapsulated into multiple tracks without G-PCC tile tracks, geometry tracks of the alternatives shall have the same `alternate_group` value in their `TrackHeaderBox`.

When different encoded versions of the same attribute component are available, each version of the attribute component is carried in one single track. All G-PCC attribute tracks which are alternatives of each other shall have the same `alternate_group` value in their `TrackHeaderBox`.

When different encoded versions of the same point cloud sequence are available and each version is encapsulated in one tile base track and multiple G-PCC tile tracks, the G-PCC tile base track shall have the same `alternate_group` value in their `TrackHeaderBox`.

When different encoded versions of the same attribute component are available, each version of the attribute component is carried in or one or more attribute tile tracks. The G-PCC attribute tile tracks which are alternatives of each other shall have the same `alternate_group` value in their `TrackHeaderBox`.

Annex E contains examples of how alternatives are indicated according to different track encapsulations.

8 Non-timed G-PCC data storage in ISOBMFF

8.1 General

This clause specifies a format to encapsulate a non-timed G-PCC data, which is the G-PCC bitstream of a single point cloud frame, in a file.

The G-PCC bitstream of a single point cloud frame is encapsulated into one or more G-PCC items specified in subclause 8.2.1. When the G-PCC bitstream contains G-PCC tiles of a single point cloud frame, it can be encapsulated into G-PCC tile items.

Each G-PCC item shall be associated with the `GPCCConfigurationProperty`. When G-PCC tile items are present, each G-PCC tile item shall be associated with a `GPCCTileInfoProperty`.

The handler type for the `MetaBox` shall be `'volv'`.

8.2 Image item

8.2.1 G-PCC item

8.2.1.1 General

When the G-PCC bitstream of a single point cloud frame is stored in a single item, an item of type 'gpe1' shall be used. A 'gpe1' item contains a series of DUs of a single point cloud frame.

When the G-PCC bitstream of a single point cloud frame is stored in multiple items, DUs of one instance of a particular component is stored in a separate item with an item of type 'gpc1'. When it is encapsulated into multiple 'gpc1' items, following restrictions apply:

— One 'gpc1' item containing GDUs shall be present and shall be an entry point.

— One or more 'gpc1' items containing ADUs may be present. Each 'gpc1' items containing ADUs shall be marked as hidden items.

— A new item reference type with a 4CC code 'gpca' is used from the 'gpc1' item carrying GDUs to the 'gpc1' items carrying ADUs.

— Each 'gpc1' item shall be associated with GPCCConfigurationProperty and GPCCComponentInfoProperty.

— The associated GPCConfigurationProperty to each 'gpc1' item shall have same values of simple_profile_compliant, dense_profile_compliant, predictive_profile_compliant, main_profile_compliant, and level_idc.

— Any of 'gpe1' item, 'gpeb' item, and G-PCC tile items shall not be present.

A G-PCC item of type 'gpe1' or 'gpc1' can be associated with one image property of type 'subs'.

When the G-PCC bitstream contains G-PCC tiles of a single point cloud frame and it is encapsulated into G-PCC tile items, an item of type 'gpeb' shall be used with following restrictions:

— The 'gpeb' item shall include DUs containing tile inventory and shall not include any GDUs, defaulted ADUs, or ADUs.

— The 'gpeb' item shall contain FSAP DUs when the FSAP DUs are present in the G-PCC bitstream.

— When the 'gpeb' item is present, one or more G-PCC tile items shall be present.

— In order to indicate the relationship between a 'gpeb' item to G-PCC tile items, a new item reference type with 4CC codes 'gpbt' is used from the 'gpeb' item to the G-PCC tile items.

— The 'gpeb' item shall not be associated with one image property of type 'subs'.

Each G-PCC item shall be associated with a GPCConfigurationProperty containing the G-PCC decoder configuration information.

If PrimaryItemBox exists, item_ID in this box shall be set to indicate a 'gpe1' item, a 'gpc1' item storing GDUs, or a 'gpeb' item.

8.2.1.2 Syntax

The GPCCItemData is structurally identical to syntax for sample format as defined in subclause 7.3.3.2.

```
aligned(8) class GPCCItemData{
   unsigned int GPCCLength = item_size;     // Size of item
   for (i=0; i< GPCCLength; )               // to end of the item
   {
      unsigned int(8)        tlv_type;
      unsigned int(32)    tlv_num_payload_bytes;
      bit(tlv_num_payload_bytes*8)    data_unit;
      i += (1+4)+ tlv_num_payload_bytes;
```

```
    }
}
```

8.2.1.3 Semantics

The value of `item_size` is equal to the sum of the `extent_length` values of each extent of the item, as specified in the `ItemLocationBox`.

`tlv_type` identifies the syntax structure represented by the following `data_unit` as defined in ISO/IEC 23090-9:2023, Annex B.

`tlv_num_payload_bytes` specify the length in bytes of the syntax element of the following `data_unit`.

`data_unit` contains a single DU which corresponds to the same point cloud frame.

8.2.2 G-PCC tile item

To encapsulate G-PCC tiles of a single point cloud frame in a separate image item, an item of type `'gpt1'` shall be used. The G-PCC tile item is formatted as a series of DUs, as defined in subclause 8.2.1.2 and each DU corresponds to one G-PCC tile which represents a rectangular cuboid inside the bounding box of G-PCC data. This item shall not include DUs containing any parameter sets, tile inventory, and FSAP.

Each item of type `'gpt1'` shall be associated with a `GPCCTileInfoProperty` indicating the number of G-PCC tiles, identifiers of the G-PCC tiles present in the associated G-PCC tile item. A G-PCC tile item shall not be associated with a `GPCCConfigurationProperty`.

A G-PCC tile item can link to one image property of type `'subs'`. When `'subs'` item property is associated, G-PCC tile identifiers in `'subs'` item property shall be equal to G-PCC tile identifiers in the `GPCCTileInfoProperty` associated with the same G-PCC tile item.

NOTE In order to allow fast data fetching without analysing DU layout of G-PCC tile items, `'subs'` item property specified in ISOBMFF can be used.

8.3 Image properties

8.3.1 G-PCC configuration item property

8.3.1.1 Definition

Box type	`'gpcC'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per item):	Yes, for an image item of type `'gpe1'`,`'gpeb'`, or `'gpc1'`
Quantity (per item):	One for an image item of type `'gpe1'`,`'gpeb'`, or `'gpc1'`

`GPCCConfigurationProperty` indicates the information for G-PCC decoder configuration and initialization.

When `GPCCConfigurationProperty` is associated with a `'gpe1'` or a `'gpeb'` item, the `setupUnit` array shall include DUs containing SPS, GPS, and APS.

When `GPCCConfigurationProperty` is associated with a `'gpc1'` item carrying GDUs, the `setupUnit` array shall include DUs containing SPS and GPS. When it is associated with a `'gpc1'` item carrying ADUs, the `setupUnit` array shall include DUs containing APS.

`essential` shall be equal to 1 for a `'gpcC'` item property.

8.3.1.2 Syntax

```
aligned(8) class GPCCConfigurationProperty extends ItemFullProperty('gpcC', 0, 0)
{
   GPCCDecoderConfigurationRecord();
}
```

8.3.1.3 Semantics

The semantics of the syntax elements within the `GPCCConfigurationProperty` are the same as those specified for the syntax elements of `GPCCCConfigurationBox` as defined in subclause 7.2.2.3.

8.3.2 G-PCC component information item property

8.3.2.1 Definition

Box type:	`'ginf'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per item):	Yes, for an image item of type `'gpc1'`
Quantity (per item):	One for an image item of type `'gpc1'`

`GPCCComponentInfoProperty` indicates G-PCC component specific information including the type of G-PCC components stored in the respective G-PCC item. This also provides the attribute name, index and an optional attribute type of G-PCC attribute component stored in the respective G-PCC item. This item property is exactly identical to the `GPCCComponentTypeBox` as defined in subclause 7.2.3.

`GPCCComponentInfoProperty` shall not be associated with an image item of type `'gpe1'` or `'gpeb'`.

`essential` shall be equal to 1 for `'ginf'` item property associated with a G-PCC item of type `'gpc1'`.

8.3.2.2 Syntax

```
aligned(8) class GPCCComponentInfoProperty extends ItemFullProperty('gpcC', 0, flags)
{
   GPCCComponentInfoStruct(flags);
}
```

8.3.2.3 Semantics

The semantics of the syntax elements within the `GPCCComponentInfoProperty` are the same as those specified for the syntax elements of `GPCCComponentInfoBox` as defined in subclause 7.2.3.3.

8.3.3 G-PCC spatial region item property

8.3.3.1 Definition

Box type:	`'gpsr'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per item):	Yes, for an image item of type `'gpeb'` or `'gpe1'`
Quantity (per item):	Exactly One for an image item of type `'gpeb'` or `'gpe1'`

`GPCCSpatialRegionInfoProperty` item property is used to describe spatial region information of DUs presents in the respective G-PCC item. This item property includes the total number of 3D spatial regions present in the respective G-PCC item along with a region identifier, the anchor point, and the dimension of the 3D spatial region in the cartesian coordinate along to x, y, and z axes relative to the anchor point for each spatial region.

The GPCCSpatialRegionInfoProperty item property optionally include the G-PCC tile identifier(s) associated with each 3D spatial region. Examples of partial access support with GPCCSpatialRegionInfoProperty are explained in Annex G.

8.3.3.2 Syntax

```
aligned(8) class GPCCSpatialRegionInfoProperty extends ItemFullProperty('gpsr', 0, 0){
   unsigned int(16)    num_regions;
   for(int i=0; i< num_regions; i++){
       GPCCSpatialRegionStruct();
   }
}
```

8.3.3.3 Semantics

num_regions indicate the number of following 3D spatial regions.

GPCCSpatialRegionStruct() provide the 3D spatial region information by including the 3D spatial region identifier, the 3D bounding box information of the identified spatial region, or G-PCC tile information associated with the region. When the 3D bounding box information is not present, this data structure provides the identifier of a particular 3D spatial region and the G-PCC tile information associated with the identified region.

8.3.4 sub-sample item property

8.3.4.1 General

ISO/IEC 23008-12:2022, Annex B.2.3.2 shall be applied with the following constraints.

— This item property shall indicate an associated property that is exactly identical to SubSampleInformationBox with flags equal to 1 as defined in subclause 7.3.3.4.

The sub-sample item property shall not be associated with an image item of type 'gpeb'.

8.3.5 G-PCC tile information item property

8.3.5.1 Definition

Box type:	'gpti'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for an image item of type 'gpt1'
Quantity (per item):	Exactly one for an image item of type 'gpt1'

GPCCTileInfoProperty descriptive item property indicates the total number of tiles and tile identifiers of those tiles present in the associated G-PCC tile item.

When GPCCTileInfoProperty is present, it shall be associated with a G-PCC tile item.

8.3.5.2 Syntax

```
aligned(8) class GPCCTileInfoProperty extends ItemFullProperty('gpti', 0, 0){
   TileInfoStruct();
}
```

8.3.5.3 Semantics

TileInfoStruct() provides the G-PCC tile information associated with the G-PCC tile item by including the number of G-PCC tiles and those identifiers.

8.4 Entity grouping

8.4.1 Viewport association

8.4.1.1 Definition

Box Types:	'vpta'
Container:	GroupsListBox ('grpl')
Mandatory:	No
Quantity:	Zero or one

An EntityToGroupBox with grouping_type equal to 'vpta' specifies the association between the G-PCC item and the viewport information timed metadata track specified in subclause 9.2.4. The viewport information timed metadata track describes the dynamically changed viewport information for the associated G-PCC item. When it is associated with the G-PCC item, viewport_type indicated in the viewport information timed metadata track shall be 0, 1 or 2.

This entity group shall contain exactly one entity_id value that point to the G-PCC item of type of 'gpe1', 'gpeb' or 'gpc1' carrying GDUs and shall contain one or more entity_id value that point to the viewport information timed metadata track.

When a G-PCC item is associated with the 'gpdv' timed metadata track, the recommended viewport of the associated G-PCC item is determined by using signalled viewport information of the 'gpdv' timed metadata track. That is, non-timed G-PCC data recommends to be rendered using the signalled viewport position, rotation as well as viewport camera parameters according to the presentation timeline of this timed metadata track. As the signalled information of the 'gpdv' timed metadata track is changed, the recommended viewport of the non-timed G-PCC data is changed accordingly at the presentation time of the associated 'gpdv' timed metadata track sample.

8.4.1.2 Syntax

```
aligned(8) class ViewportAssociationBox extends EntityToGroupBox('vpta')
{
}
```

9 Signalling of metadata in ISOBMFF

9.1 G-PCC Spatial region information

9.1.1 Information structure

9.1.1.1 3D vector structure

9.1.1.1.1 Definition

The fields in this structure provides the x, y, and z coordinate values, respectively of a 3D point in the Cartesian coordinate systems.

9.1.1.1.2 Syntax

```
aligned(8) class Vector3(precision = 32){
   unsigned int(precision) x;
   unsigned int(precision) y;
   unsigned int(precision) z;
   int reserved_bits = 8 - (precision*3) % 8;
   if(reserved_bits != 8)
   {
```

```
        const bit(reserved_bits) reserved;
    }
}
```

9.1.1.1.3 Semantics

`x`, `y`, and `z` specify the x, y, and z coordinate values, respectively, of a 3D point in the Cartesian coordinate system. The `x`, `y`, and `z` values shall be in the range of 0 to $2^{precision}-1$.

9.1.1.2 3D bounding box information structure

9.1.1.2.1 Definition

The fields in this structure provide the 3D bounding box information in Cartesian space by using an anchor 3D point and the dimension along the x, y, and z axes. When the dimension information is not present, this structure provides the x, y, and z coordinate values of a particular 3D point.

9.1.1.2.2 Syntax

```
aligned(8) class GPCCBoundingBoxStruct(bit dimensions_included_flag){
   unsigned int(8)    bb_pos_precision;
   Vector3            bb_position(bb_pos_precision);
   if(dimensions_included_flag) {
      unsigned int(8)    bb_scale_precision;
      Vector3            bb_scale(bb_scale_precision);
   }
}
```

9.1.1.2.3 Semantics

`bb_pos_precision` indicates the precision of the following `bb_position` in a number of bits.

`bb_position.x`, `bb_position.y`, and `bb_position.z` indicate the x, y, and z coordinate values, respectively, of the anchor point of the 3D bounding box in the Cartesian coordinate system.

`dimensions_included_flag` equal to 1 indicates the dimension of the 3D bounding box along the x, y, and z axes relative to the anchor point is signalled in the structure. `dimensions_included_flag` equal 0 indicates the dimension of the 3D bounding box is not signalled in the structure.

`bb_scale_precision` indicates the precision of the following `bb_scale` in bytes.

`bb_scale.x`, `bb_scale.y`, and `bb_scale.z` indicate the dimension of the 3D bounding box in the Cartesian coordinates along the x, y, and z axes, respectively, relative to the anchor point. It indicates the width, height, and depth of the bounding box in the Cartesian coordinate system.

9.1.1.3 Tile information structure

9.1.1.3.1 Definition

The fields in this information structure indicates the tile information, which is expressed by the number of G-PCC tiles and identifiers of those G-PCC tiles.

9.1.1.3.2 Syntax

```
aligned(8) class TileInfoStruct(){
   unsigned int(16)       num_tiles;
   for  (j=0; j < num_tiles; j++) {
       unsigned int(16)   tile_id;
   }
}
```

9.1.1.3.3 Semantics

`num_tiles` indicate the number of G-PCC tiles signalled in this information structure.

`tile_id` is the identifier of the signalled G-PCC tile.

9.1.1.4 G-PCC spatial region information structure

9.1.1.4.1 Definition

The fields in this information structure provide the 3D spatial region information by including the 3D spatial region identifier, the 3D bounding box information of the identified spatial region, or G-PCC tile information associated with the region. When the 3D bounding box information is not present, this data structure provides the identifier of a particular 3D spatial region and the G-PCC tile information associated with the identified region.

9.1.1.4.2 Syntax

```
aligned(8) class GPCCSpatialRegionStruct() {
   unsigned int(32)    size;
   unsigned int(16)    region_id;
   unsigned int(1)     bounding_box_present_flag;
   unsigned int(1)     dimensions_included_flag;
   unsigned int(1)     tm_present_flag;
   unsigned int(5)     reserved;
   if(bounding_box_present_flag){
      GPCCBoundingBox(dimensions_included_flag);
   }
   if(tm_present_flag){
      TileInfoStruct();
   }
}
```

9.1.1.4.3 Semantics

`size` is an integer that specifies the number of bytes in this element, including all its fields and contained elements.

`region_id` is an identifier for the 3D spatial region.

`bounding_box_present_flag` indicates the presence of the 3D bounding box information of the signalled region.

`dimensions_included_flag` indicates the presence of the dimension information of the signalled spatial region. When `bounding_box_present_flag` is equal to 0, the `dimensions_included_flag` shall be set as 0 and when `bounding_box_present_flag` is equal to 1, the `dimensions_included_flag` shall be set as 1.

`tm_present_flag` indicates the presence of G-PCC tile information associated with the signalled spatial region. when `tm_present_flag` is equal to 1, `TileInfoStruct()` shall be present.

`GPCCBoundingBox(dimensions_included_flag)` specifies the 3D bounding box information of the signalled spatial region in Cartesian space by using an anchor 3D point and the dimension along the x, y, and z axes.

`TileInfoStruct()` provides the G-PCC tile information associated with the signalled region by including the number of the associated G-PCC tiles and identifiers of those G-PCC tiles.

9.1.2 Signalling of static spatial region information

9.1.2.1 Definition

Box Types: `'gpsr'`
Container: `GPCCSampleEntry` (`'gpe1'`, `'gpeg'`, `'gpc1'`, `'gpcg'`, `'gpeb'`, `'gpcb'`)
or `DynamicGPCCSpatialRegionSampleEntry`
Mandatory: No
Quantity: Zero or one

`GPCCSpatialRegionInfoBox` provides the information of one or more 3D spatial regions of G-PCC bitstream carried by the respective track.

When `GPCCSpatialRegionInfoBox` is present in the sample entry of the G-PCC bitstream track, it provides the static 3D spatial region information of G-PCC bitstream carried in the track.

When `GPCCSpatialRegionInfoBox` is present in the sample entry of the G-PCC geometry track, it indicates the static 3D spatial region information of G-PCC bitstream carried in the G-PCC geometry track and associated G-PCC attribute tracks. It shall not be present in the sample entry of any G-PCC attribute tracks.

When `GPCCSpatialRegionInfoBox` is present in the sample entry of the G-PCC tile base track, it provides the static 3D spatial region information of G-PCC bitstream carried in all G-PCC tile tracks. It shall not be present in the sample entry of any G-PCC tile tracks. Example of partial access support with `GPCCSpatialRegionInfoBox` in G-PCC tile tracks is described in Annex F.

9.1.2.2 Syntax

```
aligned(8) class GPCCSpatialRegionInfoBox extends FullBox('gpsr',0,0){
   unsigned int(16)     num_regions;
   for (int i=0; i < num_regions; i++) {
      GPCCSpatialRegionStruct();
   }
}
```

9.1.2.3 Semantics

`num_regions` indicate the number of the signalled 3D spatial regions.

`GPCCSpatialRegionStruct()` provides the 3D spatial region information by including the 3D spatial region identifier, the 3D bounding box information of the identified 3D spatial region, or G-PCC tile information associated with the 3D spatial region.

9.1.3 Signalling of dynamic spatial region information

9.1.3.1 General

This metadata track with a sample entry type `'gpdr'` indicates the dynamically changed 3D spatial region information corresponding to the part or all of G-PCC bitstream or the association between the 3D spatial regions and G-PCC tiles over time. If a G-PCC track is associated with the dynamic spatial region timed metadata track, the 3D spatial region information of the G-PCC bitstream carried by the track or the association between 3D spatial regions and G-PCC tiles is considered dynamic.

When this timed metadata track is present, it shall contain a `'cdsc'` track reference to one of G-PCC bitstream track, G-PCC geometry track, or G-PCC tile base track. When this timed metadata track is present, `GPCCSpatialRegionInfoBox` shall not be present in the sample entry of any associated G-PCC tracks.

When a tile base track and tile tracks are present and `GPCCSpatialRegionInfoBox` is not present in the sample entry of a G-PCC tile base track, the dynamic spatial region timed metadata track shall be present. It also shall contain a `'cdsc'` track reference to the G-PCC tile base track. Examples of partial access support with this metadata track are described in Annex F.

9.1.3.2 Sample entry

9.1.3.2.1 Syntax

```
aligned(8) class DynamicSpatialRegionMappingInfoBox
   extends FullBox('gpdm',0,0){
   bit(6)              reserved=0;
   unsigned int(1)     dynamic_dimension_flag;
   unsigned int(1)     dynamic_tile_mapping_flag;
}

aligned(8) class DynamicGPCCSpatialRegionSampleEntry
   extends MetaDataSampleEntry('gpdr')
{
   GPCCSpatialRegionInfoBox              region_info;
   DynamicSpatialRegionMappingInfoBox    mapping_info;
}
```

9.1.3.2.2 Semantics

`GPCCSpatialRegionInfoBox` specifies initial 3D spatial region information by including the 3D spatial region identifier, the 3D bounding box information of the 3D spatial region, or G-PCC tile information associated with the 3D spatial region.

`DynamicSpatialRegionMappingInfoBox` indicates that the dimension information or the associated G-PCC tile information of 3D spatial region information remains unchanged or is changed in samples referring to this sample entry.

`dynamic_dimension_flag` indicates that the dimension information of the signalled 3D spatial region remains unchanged or is changed in samples referring to this sample entry. When `dynamic_dimension_flag` is equal to 0, it indicates that the dimension of signalled 3D spatial region remains unchanged in all samples referring to this sample entry. When `dynamic_dimension_flag` equal to 1, it indicates that the dimension information of the signalled 3D spatial region is present in samples.

`dynamic_tile_mapping_flag` indicates that the associated G-PCC tile information of the signalled 3D spatial region remains unchanged or is changed in samples referring to this sample entry. When `dynamic_tile_mapping_flag` is equal to 0, it indicates that identifiers of G-PCC tiles associated with the 3D spatial region remains unchanged in all samples referring to this sample entry. When `dynamic_tile_mapping_flag` equal to 1, it indicates that identifiers of G-PCC tiles associated with the 3D spatial region are present in samples.

9.1.3.3 Sample format

9.1.3.3.1 General

A sample in the 3D spatial region information timed-metadata track shall be set as a sync sample or non-sync sample. At least one sync sample shall be present in the dynamic spatial region timed metadata track.

Sync samples in dynamic spatial region timed metadata track shall carry the dimension and the associated tile mapping information for all G-PCC 3D spatial regions. In sync samples for all 3D spatial regions, the value of `dimensions_included_flag` and `bounding_box_present_flag` shall be set to 1. The `tm_present_flag` flag shall be set to 1 when the tile inventory information is available in the G-PCC bitstream.

Non-sync samples in dynamic spatial region timed metadata track shall signal only the updated 3D spatial regions information with reference to the 3D spatial regions information available in the nearest preceding sync sample. Non-sync samples shall signal only the 3D spatial region information whose position, dimension or the identifiers of associated G-PCC tiles are updated, the added and cancelled 3D spatial regions with reference to the nearest sync sample.

For Non-sync samples in this timed-metadata track, the value of `canceled_region_flag` shall be set to 1 when a 3D spatial region is cancelled with reference to the preceding sync sample. The value of `dimensions_included_flag` shall only be set to 1, when the dimensions of a 3D spatial region in the current sample is

updated with reference to the preceding sync sample. The value of dimensions_included_flag shall be set to 0, when dynamic_dimension_flag in the referred sample entry equals to 0. The value of bounding_box_present_flag shall only be set to 1, when the position and/or dimensions of a 3D spatial region in the current sample is updated with reference to the preceding sync sample. The value of tm_present_flag shall only be set to 1 when the associated 3D tiles of a 3D spatial region in the current sample is updated with reference to the preceding sync sample. The value of tm_present_flag shall be set to 0, when dynamic_tile_mapping_flag in the referred sample entry equals to 0.

9.1.3.3.2 Sync samples

9.1.3.3.2.1 Syntax

```
aligned(8) DynamicGPCCSpatialRegionSample() {
   unsigned int(16)     num_regions;
   for (int i=0; i < num_regions; i++) {
      GPCCSpatialRegionStruct();
   }
}
```

9.1.3.3.2.2 Semantics

num_regions indicates the number of 3D spatial regions signalled in the sync sample.

GPCCSpatialRegionStruct() specifies the 3D spatial region information of G-PCC data when this sample is applied. In this information structure, the value of dimensions_included_flag and bounding_box_present_flag shall be set to 1. tm_present_flag shall be set to 1 when the tile inventory information is available. Otherwise, the tm_present_flag shall be set to 0.

Let the target be the point cloud data associated with samples in the referenced tracks with composition times greater than or equal to the composition time of this sample and less than the composition time of the next sample.

9.1.3.3.3 Non-sync samples

9.1.3.3.3.1 Syntax

```
aligned(8) DynamicGPCCSpatialRegionSample() {
   unsigned int(16)    num_regions;
   for (int i=0; i < num_regions; i++) {
      unsigned int(1)     canceled_region_flag;
      unsigned int(7)     reserved;
      if(canceled_region_flag == 0)
            GPCCSpatialRegionStruct();
      else
            unsigned int(16) region_id;
   }
}
```

9.1.3.3.3.2 Semantics

num_regions indicate the number of updated 3D spatial regions signalled in the sample. The 3D spatial regions whose dimensions and/or the associated 3D tiles are updated with reference to the previous sync sample are considered as updated regions. The 3D spatial regions that are cancelled in this sample with reference to the previous sync sample are also considered as updated regions.

canceled_region_flag indicates whether a 3D region is cancelled or updated in the current sample with reference to the previous sync sample. Value 1 indicates 3D regions is cancelled in this sample with reference to the previous sync sample. Value 0 indicates the 3D region dimensions and/or the associated 3D tiles are updated with reference to the previous sync sample.

GPCCSpatialRegionStruct() provides the 3D spatial region information of G-PCC data when this sample is applied. The value of dimensions_included_flag shall only be set to 1 when the dimensions of this

3D region are updated with reference to the previous sync sample. The value of `dimensions_included_flag` shall be set to 0 when `dynamic_dimension_flag` in the referred sample entry equals to 0. The value of `bounding_box_present_flag` shall only be set to 1 when the position and/or dimensions of this 3D region are updated with reference to the previous sync sample. The value of `tm_present_flag` shall only be set to 1 when the associated G-PCC tiles of this 3D region are updated with reference to the previous sync sample. The value of `tm_present_flag` shall be set to 0 when `dynamic_tile_mapping_flag` in the referred sample entry equals to 0.

`region_id` identifies a cancelled 3D spatial region with reference to the previous sync sample.

9.2 G-PCC viewport information

9.2.1 General

This subclause specifies signalling of viewport information for G-PCC data in ISOBMFF. The viewport information includes extrinsic camera information structure containing the position and rotation of the virtual camera. It also includes intrinsic camera information structure. The G-PCC player can render the G-PCC data based on the signalled viewport information.

9.2.2 Information structure

9.2.2.1 Extrinsic camera information structure

9.2.2.1.1 Definition

The fields in this structure provides the x, y, and z coordinate values, respectively of a virtual camera in the global reference coordinate system, and the x, y, and z components, respectively, of the rotation of the virtual camera.

9.2.2.1.2 Syntax

```
aligned(8) class ExtCameraInfoStruct() {
   unsigned int(32)      cam_pos_x;
   unsigned int(32)      cam_pos_y;
   unsigned int(32)      cam_pos_z;
   signed int(32)        cam_quat_x;
   signed int(32)        cam_quat_y;
   signed int(32)        cam_quat_z;
}
```

9.2.2.1.3 Semantics

`cam_pos_x`, `cam_pos_y`, and `cam_pos_z`, respectively, indicate the x, y, and z coordinates of the position of the camera in meters in the global reference coordinate system. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

`cam_quat_x`, `cam_quat_y`, and `cam_quat_z` indicate the x, y, and z components, respectively, of the rotation of the camera using the quaternion representation. The values shall be in the range of -2^{30} to 2^{30}, inclusive. When the component of rotation is not present, its value shall be inferred to be equal to 0. The value of rotation components may be calculated as follows:

`qX = cam_quat_x ÷ 230, qY = cam_quat_y ÷ 230, qZ = cam_quat_z ÷ 230`

The fourth component, `qW`, for the rotation of the current camera model using the quaternion representation is calculated as follows:

`qW = Sqrt( 1 − ( qX2 + qY2 + qZ2 ) )`

The point `(w, x, y, z)` represents a rotation around the axis directed by the vector `(x, y, z)` by an angle `2*cos ^{-1}(w)=2*sin ^{-1}(sqrt(x^{2}+y^{2}+z^{2}))`.

NOTE In the context of this document, `qW` is always positive. If a negative `qW` is desired, one can signal all three syntax elements, `cam_quat_x`, `cam_quat_y`, and `cam_quat_z` with an opposite sign, which is equivalent.

9.2.2.2 Intrinsic camera information structure

9.2.2.2.1 Definition

The fields in this information structure provides viewport camera parameters by including the projection information of the viewport camera and the near and far clipping plane information of the viewport.

9.2.2.2.2 Syntax

```
aligned(8) class IntCameraInfoStruct() {
   unsigned int(10)    camera_id;
   bit(3)              reserved = 0;
   unsigned int(3)     camera_type;

   if (camera_type == 0) {
     signed int(32)    erp_horizontal_fov;
     signed int(32)    erp_vertical_fov;
   }
   else if (camera_type == 1) {
      signed int(32)       perspective_horizontal_fov;
      unsigned int(32)     perspective_aspect_ratio;
   }
   else if (camera_type == 2) {
      unsigned int(32)     ortho_aspect_ratio;
      unsigned int(32)     ortho_horizontal_size;
   }

   unsigned int(32) clipping_near_plane;
   unsigned int(32) clipping_far_plane;
}
```

9.2.2.2.3 Semantics

`camera_id` is an identifier of a given viewport camera parameters.

`camera_type` indicates the projection method of the viewport camera. The value 0 specifies ERP projection. The value 1 specifies a perspective projection. The value 2 specifies an orthographic projection. Values in the range 3 to 255 are reserved for future use by ISO/IEC.

`erp_horizontal_fov` specifies the longitude range for an ERP projection corresponding to the horizontal size of the viewport region, in units of radians. The value shall be in the range 0 to 2π.

`erp_vertical_fov` specifies the latitude range for an ERP projection corresponding to the vertical size of the viewport region, in units of radians. The value shall be in the range 0 to π.

`perspective_horizontal_fov` specifies the horizontal field of view for perspective projection in radians. The value of shall be in the range of 0 and π.

`perspective_aspect_ratio` specifies the relative aspect ratio of viewport for perspective projection (horizontal/vertical). The value shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

`ortho_aspect_ratio` specifies the relative aspect ratio of viewport for orthogonal projection (horizontal/vertical). The value shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

`ortho_horizontal_size` specifies the horizontal size of the orthogonal in meters. The value shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

`clipping_near_plane` and `clipping_far_plane` indicate the near and far depths (or distances) based on the near and far clipping planes of the viewport in meters. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

9.2.2.3 Viewport information structure

9.2.2.3.1 Definition

The fields in this information structure provide either extrinsic camera parameters or intrinsic camera parameters of the viewport. When extrinsic camera parameters are present, it specifies whether the centre of the viewport or one of two stereo position of viewport is indicated and provides the position and the rotation of the viewport accordingly. When intrinsic camera parameters are present, it provides the projection information of the viewport camera and the near and far clipping plane information of the viewport.

9.2.2.3.2 Syntax

```
aligned(8) class ViewportInfoStruct(bit ext_camera_flag, bit int_camera_flag) {
   if (ext_camera_flag == 1) {
      unsigned int(1)       center_view_flag;
      bit(6) reserved = 0;
      if (center_view_flag == 0) {
         unsigned int(1)    left_view_flag;
      }else{
         bit(1) reserved = 0;
      }
      ExtCameraInfoStruct();
   }
   if (int_camera_flag == 1) {
      IntCameraInfoStruct();
   }
}
```

9.2.2.3.3 Semantics

`center_view_flag` is a flag indicating whether the signalled viewport position corresponds to the centre of the viewport or to one of two stereo positions of the viewport. When `center_view_flag` is equal to 1, it indicates that the signalled viewport position corresponds to the centre of the viewport. When `center_view_flag` is equal to 0, it indicates that the signalled viewport position corresponds to one of two stereo positions of the viewport.

`left_view_flag` is a flag indicating whether the signalled viewport information correspond to the left stereo position or the right stereo position of the viewport. When `left_view_flag` is equal to 1, it indicates that the signalled viewport information corresponds to the left stereo position of the viewport. When `left_view_flag` is equal to 0, it indicates that the viewport information signalled correspond to the right stereo positions of the viewport.

`ExtCameraInfoStruct()` specifies the position and rotation of the viewport.

`IntCameraInfoStruct()` specifies the projection information and clipping plane information for the viewport.

9.2.3 Signalling of static viewport information

9.2.3.1 Definition

Box Types: 'gvpC'
Container: GPCCSampleEntry ('gpe1', 'gpeg', 'gpc1', 'gpcg', 'gpeb', 'gpcb')
or DynamicGPCCViewportSampleEntry
Mandatory: No
Quantity: Zero or one

Static viewport information is signalled in GPCCViewportInfoConfigutationBox present in the sample entry of one of G-PCC bitstream track, G-PCC geometry track, or G-PCC tile base track.

9.2.3.2 Syntax

```
aligned(8) class GPCCViewportInfoConfigurationBox extends FullBox('gvpC', 0, 0) {
   unsigned int(7)   viewport_type;
   bit(1)            reserved = 0;
   unsigned int(1) dynamic_int_camera_flag;
   unsigned int(1) dynamic_ext_camera_flag;
   bit(6) reserved = 0;
   if (dynamic_int_camera_flag == 0) {
     IntCameraInfoStruct();
   }
   if (dynamic_ext_camera_flag == 0) {
     ExtCameraInfoStruct();
   }
   string             viewport_description;
}
```

9.2.3.3 Semantics

viewport_type specifies the type of the viewport as listed in Table 10 for viewport parameter set for all samples referring to this sample entry.

For viewport_type equal to 3, it indicates the recommended initial viewport information, composed of the initial viewport position and rotation, when playing the associated G-PCC media tracks. When the playback of a media track is intended to be started using another viewport than that indicated by initial viewport position (cam_pos_x, cam_pos_y, cam_pos_z) equal to (0, 0, 0) relative to the global coordinate axes and initial viewing rotation (cam_quat_x, cam_quat_y, cam_quat_z) equal to (0,0,0) relative to the global coordinate axes, it shall be present. In the absence of this type of viewport information, cam_pos_x, cam_pos_y, cam_pos_z, cam_quat_x, cam_quat_y, and cam_quat_z should all be inferred to be equal to 0 for the initial viewport.

Table 10 — Viewport Types

Value	Description
0	A recommended viewport per the director's cut, i.e., a viewport suggested according to the creative intent of the content author or content provider
1	A recommended viewport selected based on measurements of viewing statistics
2	A recommended viewport based on the selected viewport of another user
3	An initial viewport suggested to use when starting to play associated immersive media
4..239	Reserved
240..255	Unspecified (for use by applications or external specifications)

When dynamic_int_camera_flag is equal to 0, it indicates that intrinsic camera parameters are fixed for all samples referring to this sample entry. When dynamic_ext_camera_flag is equal to 0, dynamic_int_camera_flag shall also be equal to 0.

When dynamic_ext_camera_flag is equal to 0, it indicates that extrinsic camera parameters are fixed for all samples referring to this sample entry.

viewport_description is null-terminated UTF-8 string that provides a textual description of the recommended viewport for the viewport parameter set for all samples referring to this sample entry.

9.2.4 Signalling of dynamic viewport information

9.2.4.1 General

This subclause specifies the use of the timed metadata track to signal dynamically changed viewport information, composed of intrinsic and extrinsic camera parameters, including viewport position and rotation information as well as viewport camera parameters.

A viewport information timed metadata track containing a 'cdtg' track reference describes the referenced tracks and track groups collectively. When the timed metadata track is linked to one or more G-PCC tracks with a 'cdsc' track reference, it describes each G-PCC track individually.

When multiple viewport sequences of the same viewport type are present, it shall be stored in a separate metadata track.

9.2.4.2 Sample entry

```
class DynamicGPCCViewportSampleEntry extends MetaDataSampleEntry('gpdv') {
   GPCCViewportInfoConfigurationBox();
}
```

9.2.4.3 Sample format

9.2.4.3.1 General

Any sample in a viewport information timed metadata track is allowed to be marked as a sync sample. For a particular sample in the timed metadata track, if at least one of the media samples in the referenced G-PCC track having the same decoding time is a sync sample, the particular sample shall be marked as a sync sample, otherwise, that sample may or may not be marked as a sync sample.

In this edition of this document, num_viewports in sample shall be equal to 1.

9.2.4.3.2 Syntax

```
aligned(8) class DynamicViewportInfoSample() {
   unsigned int(8)       num_viewports;
   for (i=1; i <= num_viewports; i++){
     unsigned int(7)       viewport_id;
     unsigned int(1)       viewport_cancel_flag;
     if (viewport_cancel_flag == 0) {
       unsigned int(1)       camera_ext_flag;
       unsigned int(1)       camera_int_flag;
       bit(6) reserved = 0;
       ViewportInfoStruct(camera_ext_flag,camera_int_flag);
     }
   }
}
```

9.2.4.3.3 Semantics

num_viewports indicate the number of viewports signalled in the sample.

`viewport_id` is an identifier that is used to identify the viewport.

When `viewport_cancel_flag` is equals to 1, it indicates that the viewport with the `viewport_id` is cancelled. When `viewport_cancel_flag` is equals to 0, it indicates that viewport information is signalled.

When `camera_ext_flag` is equal to 1, it indicates that the extrinsic camera parameters of this viewport are present in the current sample. It shall be equal to 0 when `dynamic_ext_camera_flag` in the sample entry is equals to 0.

When `camera_int_flag` equal to 1 indicates that the intrinsic camera parameters are present in the viewport in the current sample. It shall be equal to 0 when `dynamic_int_camera_flag` in the sample entry is equals to 0. Moreover, it shall be set as 0 when `camera_extrinsic_flag` is equals to 0.

When the viewport information timed metadata track is present, extrinsic camera parameters expressed by `ExtCameraInfoStruct()` shall be present in either sample entry or samples. It is prohibited that both of the following concurrently happen; `dynamic_ext_camera_flag` equals to 0 and `camera_extrinsic_flag` equals to 0 for all samples.

10 Encapsulation and signalling in DASH

10.1 Single-track mode

10.1.1 General

The single-track mode in DASH enables streaming of G-PCC ISOBMFF files where the G-PCC bitstream is stored using single track encapsulation, The single-track mode in DASH should be represented as one Adaptation Set with one or more Representations. If a Representation consists of more than one Media Segment, an Initialization Media Segment shall be present. The Initialization Segment shall contain a `GPCCDecoderConfigurationRecord` with the G-PCC parameter sets (e.g. SPS, GPS, and APS) as defined in ISO/IEC 23090-9.

The restriction on some of the attributes shall be applied:

— The `@mimeType` parameter shall be `'application/mp4'`

— The `@codecs` shall be present on the adaptation set level and shall signal the maximum required capability to decode any Representation in the Adaptation Set. The `@codecs` parameter should be signalled on the representation level if different from the one on the adaptation set level.

— The `@codecs` parameter present on a representation level shall signal the required capability to decode any component in the Representation.

— When 'codecs' parameter of a MIME type is used, sub-parameters shall be used as defined in Annex C.

— The `@maxWidth` and `@maxHeight` parameters shall not be signalled for any Adaptation Set.

— The `@framerate` shall be signalled only in the Adaptation Set element, i.e., the value shall not be different for different representations in on Adaptation Set.

— The `@width` and `@height` shall not be signalled for any Representation.

10.2 Multi-track mode

10.2.1 General

Each G-PCC component shall be represented in the DASH manifest (MPD) file as a separate Adaptation Set, referred to as a Component Adaptation Set. The Adaptation Set containing the geometry information is also the Main GPCC Adaptation Set which serves as the access point for the G-PCC bitstream. The Main GPCC Adaptation Set shall contain either a single Initialization Segment at adaptation set level or multiple Initialization Segments at representation level (one for each Representation). Initialization Segments shall

contain the G-PCC parameter sets as defined in ISO/IEC 23090-9, which are needed to initialize the G-PCC decoder. Examples of DASH signalling are explained in Annex H.

10.2.2 DASH MPD descriptors

10.2.2.1 XML namespace and schema

A number of XML elements and attributes are defined in this document. These XML elements are defined in a separate namespace "`urn:mpeg:mpegI:gpcc:2023`". The namespace designator "`gpcc:`" is used to refer to this name space in this document. The namespace designator "`xs:`" shall correspond to namespace `https://www.w3.org/2001/XMLSchema` as defined in W3C Recommendation, XML schema part 1.

Some items in the "Data type" column of the tables in Clause 10 use datatypes and meaning as defined in W3C Recommendation, XML schema part 2 or in ISO/IEC 23009-1. Data types not defined in W3C Recommendation, XML schema part 2 or ISO/IEC 23009-1 shall be as defined in Annex B.

The GPCC DASH XML schema is provided in Annex B. When features specified in Clause 10 are present in a DASH MPD, Annex B shall be obeyed.

10.2.2.2 Signalling of G-PCC component information

An **`EssentialProperty`** element with the `@schemeIdUri` attribute equal to "`urn:mpeg:mpegI:gpcc:2023:component`" is referred to as a **`GPCCComponent`** descriptor.

At Adaptation Set level, one **`GPCCComponent`** descriptor shall be signalled for each component that is present in the Representations of the adaptation set. The **`GPCCComponent`** descriptor shall include elements and attributes defined in Table 11.

Table 11 — Elements and attributes for the `GPCCComponent` descriptor

Elements and Attributes for `GPCCComponent` descriptor	Use	Data type	Description
component	0..N	gpcc:gpccComponentType	An element that specify information for G-PCC component present in the representation(s) of the adaptation set.
component`@component_type`	M	xs:string	Indicates the type of the point cloud component. Value `'geom'` indicates a geometry component, and `'attr'` indicates an attribute component.
component`@att_type`	CM	xs:unsignedByte	Indicates the type of the attribute (refer to ISO/IEC 23090-9:2023, Table 9). Only values between 0 and 255, inclusive, are allowed. Shall be present only if the component is an attribute component (i.e., `@component_type` has the value `'attr'`) and `@attr_label_oid` is not present.
component`@attr_label_oid`	CM	xs:string	Indicates the international object identifier as specified in Recommendation ITU-T X.660 | ISO/IEC 9834-1. The syntax of object identifier is described in ISO/IEC 23090-9:2023, subclause 11.4.7.1. Shall be present only if the component is an attribute component(i.e., `@component_type` has the value `'attr'`) and `@attr_type` is not present.
component`@attr_index`	CM	xs:unsignedByte	Indicates the order of the attribute present in the SPS. The value of `@attr_index` shall be identical to the `ash_attr_sps_attr_idx` value of the TLV encapsulation structure carried by the Representations of the Adaptation Set. Shall be present only if the component is a point cloud attribute (i.e., `component_type` has the value `'attr'`).
component`@tile_ids`	CM	xs:UIntVectorType	A list of space-separated identifiers corresponding to the value of the `tile_id` field of each G-PCC tile present in the G-PCC tile track. Shall only be present if the Adaption Set is a Tile Component Adaptation Set. Shall only be present if the corresponding tile track carries a constant number of tiles and the tile identifiers do not change throughout the bitstream. i.e., `dynamic_num_tiles_flag` is set to 0 in the `GPCCTileSampleEntry` of the respective tile track.
Key: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs> (N=unbounded) Elements are `bold`; attributes are non-bold and preceded with an `@`.			

When more than one Representation is signalled in the Main GPCC Adaptation Set, the initialization segment of each Representation shall contain the G-PCC parameter sets for that Representation. The Representation(s) of a Component Adaptation Set shall list the corresponding Representation identifier from the Main GPCC Adaptation Set using the `@dependencyId` attribute defined in ISO/IEC 23009-1.

The data type for the attribute shall be as defined in the XML schema. An XML schema for G-PCC component information signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:gpcc:2023` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:gpcc:2023"
    xmlns:omaf="urn:mpeg:mpegI:gpcc:2023"
    elementFormDefault="qualified">
    <xs:element name=”component” type=”gpcc: gpccComponentType” />

    <xs:complexType name=”gpccComponentType”>
        <xs:attribute name=”type” type=”xs:string” use=”required” />
        <xs:attribute name=”attr_type” type=”xs:unsignedByte” use="optional" />
        <xs:attribute name=”attr_label_oid” type=”xs:string” use="optional" />
        <xs:attribute name=”attr_index” type=”xs:unsignedByte” use="optional" />
        <xs:attribute name=”tile_ids” type=”xs:UIntVectorType” use="optional" />
    </xs:complexType>
</xs:schema>
```

10.2.2.3 GPCC Descriptor

A G-PCC player can identify the type of point cloud component in an Adaptation Set or Representation by checking the **`GPCCComponent`** descriptor within the corresponding element. However, a G-PCC player also needs to distinguish between different geometry point cloud streams when multiple point cloud streams present in the MPD file.

A **`SupplementalProperty`** element with a `@schemeIdUri` attribute equal to `"urn:mpeg:mpegI:gpcc:2023:gpc"` is referred to as a GPCC descriptor. The attributes of **GPCC** descriptor are shown in Table 12.

At most one **`GPCC`** descriptor may be present at the Adaptation Set level for the Main GPCC Adaptation Set of the G-PCC media or at the Preselection level.

Table 12 — Attributes for the GPCC descriptor

Attributes for GPCC descriptor	Use	Data type	Description
`gpcc:@gpcId`	CM	xs:string	An id for the volumetric media. This attribute shall be present if multiple versions of the same volumetric media are signalled in separate AdapatationSets.
Key: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an `@`.			

The data type for the attribute shall be as defined in the XML schema. An XML schema for G-PCC component information signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:gpcc:2023` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:gpcc:2023"
    xmlns:omaf="urn:mpeg:mpegI:gpcc:2023"
    elementFormDefault="qualified">
```

```
    <xs:attribute name="gpcId" type="xs:string" use=”optional”/>
</xs:schema>
```

10.2.3 GPCC Preselection

A G-PCC Preselection is signalled in the MPD using a **Preselection** element (as defined in ISO/IEC 23009-1) with an id list for the `@preselectionComponents` attribute including the identifier of the Main GPCC Adaptation Set for the G-PCC media followed by the identifiers of other Component Adaptation Sets. The `@codecs` attribute for the Preselection shall be set to `'gpc1'`. The Preselection can either be signalled using a **Preselection** element within the **Period** element or a **Preselection** descriptor at the Adaptation Set level in the Main GPCC Adaptation Set.

10.2.4 Supporting multiple versions of GPCC data

Multiple versions of the same point cloud media shall be signalled using separate Preselections. Preselections that represent alternative versions of the same G-PCC content shall contain **a GPCC** descriptor with the same `@gpcId` value. At most one **GPCC** descriptor shall be present at the Preselection level. These Preselections are alternatives to each other. The identifier of the Main GPCC Adaptation Set is the first identifier in the list of Adaptation Set ids for the Preselection followed by the identifiers of Component Adaptation Sets.

10.3 Partial delivery and access

10.3.1 Signalling of static spatial regions

If the 3D spatial regions are static (i.e., the anchor point and dimensions of each spatial region do not change over the presentation time), the characteristics of the spatial regions and the mappings between those regions and G-PCC tiles shall be signalled using a **GPCCSpatialRegions** descriptor. A **SupplementalProperty** element with a `@schemeIdUri` attribute equal to `"urn:mpeg:mpegI:gpcc:2023:gpsr"` is referred as a **GPCCSpatialRegions** descriptor.

A single **GPCCSpatialRegions** descriptor shall be present at the Adaptation Set level or the Representation level in the Main GPCC Adaptation Set, or at the Preselection level for the G-PCC content. The mapping of the 3D region and the respective G-PCC tile component streams is signalled using the `@tileIds` attribute of the **GPCCSpatialRegions** descriptor. The `@value` attribute of the **GPCCSpatialRegions** descriptor shall not be present.

The **GPCCSpatialRegions** descriptor shall include elements and attributes as specified in Table 13.

Table 13 — Elements and attributes for the `GPCCSpatialRegions` descriptor

Elements and attributes for GPCCSpatialRegions descriptor	Use	Data type	Description
gpsr	0..1	gpcc:spatialRegionMapType	Container element whose attributes and elements specify a mapping between a 3D spatial region and G-PCC tiles.
gpsr.spatialRegion	1..N	gpcc:spatialRegionType	An element whose attributes define a 3D spatial region and provide a mapping between the defined region and a number of G-PCC tiles.
gpsr.spatialRegion@id	M	xs:unsignedShort	An identifier for the 3D spatial region. The value of this attribute shall match the value of the `3d_region_id` field signalled for the corresponding region in the ISOBMFF container.
gpsr.spatialRegion@x	OD	xs:int	The x-coordinate of the reference point for the bounding box defining the spatial region. If not present, the default value is 0.
gpsr.spatialRegion@y	OD	xs:int	The y-coordinate of the reference point for the bounding box defining the spatial region. If not present, the default value is 0.
gpsr.spatialRegion@z	OD	xs:int	The z-coordinate of the reference point for the bounding box defining the spatial region. If not present, the default value is 0.
gpsr.spatialRegion@dx	M	xs:int	The length of the bounding box along the x-axis (i.e., width). Negative values indicate a length that extends in the negative direction of the axis.
gpsr.spatialRegion@dy	M	xs:int	The length of the bounding box along the y-axis (i.e., height). Negative values indicate a length that extends in the negative direction of the axis.
gpsr.spatialRegion@dz	M	xs:int	The length of the bounding box along the z-axis (i.e., depth). Negative values indicate a length that extends in the negative direction of the axis.
gspr.spatialRegion@tileIds	O	xs:UIntVectorType	A list of space separated G-PCC tile identifiers. A G-PCC tile identifier corresponds to the `tile_id` field for a G-PCC tile in tile inventory and identifies the G-PCC tile associated with the 3D spatial region.
Key: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an `@`.			

The data type for the attribute shall be as defined in the XML schema. An XML schema for G-PCC component information signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:gpcc:2023` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:gpcc:2023"
    xmlns:omaf="urn:mpeg:mpegI:gpcc:2023"
    elementFormDefault="qualified">

    <xs:element name=”gpsr” type=”gpcc:spatialRegionMapType” />
```

```
    <xs:complexType name="spatialRegionMapType">
        <xs:element name="spatialRegion" type="gpcc:spatialRegionType" minOccurs="1" />
    </xs:complexType>

    <xs:complextType name="spatialRegionType">
        <xs:attribute name="id" type="xs:unsignedShort" use="required" />
        <xs:attribute name="x" type="xs:int" use="optional" default="0" />
        <xs:attribute name="y" type="xs:int" use="optional" default="0" />
        <xs:attribute name="z" type="xs:int" use="optional" default="0" />
        <xs:attribute name="dx" type="xs:int" use="required" />
        <xs:attribute name="dy" type="xs:int" use="required" />
        <xs:attribute name="dz" type="xs:int" use="required" />
        <xs:attribute name="tile_ids" type="xs:UIntVectorType" use="optional" />
    </xs:complexType>
</xs:schema>
```

10.3.2 Signalling of dynamic spatial regions

When the 3D partitions are dynamic, a timed-metadata track for signalling the anchor point and dimensions of each 3D region in the presentation timeline shall be carried in a separate Adaptation Set with a single Representation and associated (linked) with the Representations of the Main GPCC Adaptation Set using the `@associationId` attribute, defined in ISO/IEC 23009-1, with a `@associationType` value that includes the `4CC 'gpdr'` for the corresponding Adaptation Set or Representation.

10.3.3 Tiled G-PCC data encapsulation and signalling

10.3.3.1 Signalling of G-PCC tile base track

When multiple G-PCC tile tracks are present, the Main Tile Adaptation Set shall contain the parameter sets and the tile inventory from the G-PCC tile base track. G-PCC geometry and attribute data shall not be present in the Main Tile Adaptation Set and its Representations.

The `@codecs` attribute for the Main Tile Adaptation Set is set to `'gpcb'` or `'gpeb'`, indicating that the Adaptation Set contains the G-PCC tile base track data. When the Tile Component Adaptation Sets signal only a single G-PCC component, the `@codecs` attribute for the Main GPCC Adaptation Set is set to `'gpcb'`. When the Tile Component Adaptation Sets signal all the G-PCC components data, the `@codecs` attribute for the Main GPCC Adaptation Set is set to `'gpeb'`.

10.3.3.2 Signalling of G-PCC tile tracks

Each G-PCC tile track is signalled in the DASH MPD as a separate Adaptation Set. If the G-PCC tile track carries a single component, the corresponding Adaptation Set is referred to as a Tile Component Adaptation Set. When multiple versions of a component for the same tile or, set of tiles, are present and carried in separate tile tracks, each version shall be signalled in a Representation of the Tile Component Adaptation Set. The `@codecs` attribute for Tile Component Adaptation Sets representing tile tracks of G-PCC media content shall be set to `'gpt1'`.

When multiple tile tracks are present in the container, each Representation in the geometry Tile Component Adaptation Set refers to the corresponding Representation in the Main Tile Adaptation Set using the `@dependencyId` attribute defined in ISO/IEC 23009-1. Similarly, each Representation in an attribute Tile Component Adaptation Set refers to the corresponding Representation in the geometry Tile Component Adaptation Set using the `@dependencyId` attribute. Examples of DASH signalling are explained in Annex H.

10.3.3.3 DASH MPD Descriptors

10.3.3.3.1 GPCCTileID Descriptor

A **SupplementalProperty** element with a `@schemeIdUri` attribute equal to `"urn:mpeg:mpegI:gpcc:2023:tileID"` referred as **GPCCTileID** descriptor can be used to distinguish between different G-PCC tile tracks.

At most one **GPCCTileID** descriptor can be present at the Adaptation Set level for a G-PCC Tile Component Adaptation Set. The **GPCCTileID** descriptor shall be present at the Adaptation Set level if the Adaptation Set represents a G-PCC tile track carrying all the G-PCC components for the tiles in the track. One or more **GPCCTileID** descriptors can be present in a G-PCC Tile Preselection as defined in subclause 10.3.3.4. The **GPCCTileID** descriptor shall include the attributes defined in Table 14.

Table 14 — Attributes for the GPCCTileID descriptor

Attributes for GPCCTileID descriptor	Use	Data type	Description
`@tile_Ids`	M	xs:UIntVectorType	A list of space-separated identifiers corresponding to the value of the `tile_id` field of each G-PCC tile present in the G-PCC tile track. Shall only be present if the Adaption Set is a Tile Component Adaptation Set. Shall only be present if the corresponding tile track carries a constant number of tiles and the tile identifiers do not change throughout the bit-stream. i.e., `dynamic_num_tiles_flag` is set to 0 in the `GPCCTileSampleEntry` of the respective tile track.
Key: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an `@`.			

The data type for the attribute shall be as defined in the XML schema. An XML schema for G-PCC component information signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:gpcc:2023` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:gpcc:2023"
    xmlns:omaf="urn:mpeg:mpegI:gpcc:2023"
    elementFormDefault="qualified">

    <xs:attribute name="tile_Ids " type="xs:UIntVectorType" use=”required”/>

</xs:schema>
```

10.3.3.4 GPCC tile preselections

A G-PCC Tile Preselection is signalled in the MPD using a **PreSelection** element. The `@codecs` attribute for the Preselection shall be set to `'gpt1'`, indicating that the Preselection contains a set of G-PCC tiles. A G-PCC Tile Preselection can either be signalled using a **PreSelection** element within the **Period** element or a **Preselection** descriptor at the Tile Component Adaptation Set level.

Geometry Tile Component Adaptation Set and associated attribute Tile Component Adaptation Sets shall be part of a single Tile Preselection in the MPD with the geometry tile Component Adaptation Set being the main Adaptation Set for that Tile Preselection. The `@preselectionComponents` attribute for a G-PCC Tile Preselection shall contain the identifier of a geometry Tile Component Adaptation Set followed by the identifiers of corresponding attribute Tile Component Adaptation Set. The respective Representation of the Main Tile Adaptation Set corresponding to each Representation of the geometry Tile Component Adaptation Set is identified using `@dependencyId` attribute signalled in the Representation of the Tile Component Adaptation Set.

Each G-PCC Tile Preselection can contain one or more **`GPCCTileID`** descriptor(s). This allows the identification of the tiles referenced in each Preselection. When the **`GPCCTileID`** descriptor is not present, the tiles belonging to a G-PCC Tile Preselection shall be identified from the **`GPCCComponent`** descriptor signalled in the Tile Component Adaptation Sets.

10.4 Signalling recommended viewports

10.4.1 Signalling of static recommended viewports

A **`SupplementalProperty`** with a `@schemeIdUri` equal to "`urn:mpeg:mpegI:gpcc:2023:rv`" is defined for the **`Recommended Viewport(RV)`** descriptor in order to signal the recommended viewports of the G-PCC content. This descriptor may be used by content providers to signal a set of viewport position and rotation parameters recommended for rendering the G-PCC content. The **`RV`** descriptor indicates that each Representation in the Adaptation Set (for the multi-track case, this includes the Representations in the Main Adaptation Set and other related Adaptation Sets for the corresponding G-PCC components) is recommended to be rendered based on the provided set of viewport position (`@vp_pos`) and rotation (`@vp_quat`).

One or more **`RV`** descriptors may be present in each Adaptation Set for the single-track DASH mode. In the case of the multi-track DASH mode, one or more **`RV`** descriptors, if present, shall only be placed in the Main Adaptation Set. No other RV descriptor shall be present at the MPD representation level or any other level in both single-track and multi-track DASH modes.

The **`RV`** descriptor shall include elements and attributes as specified in Table 15.

Table 15 — Elements and attributes for the RV descriptor

Elements and attributes	Use	Data type	Description
`@viewport_id`	O	xs:integer	An identifier for the viewport.
ViewportInfo	1	gpcc:ViewportInfoType	Container element whose sub-elements and attributes provide information about the viewport.
ViewportInfo`@vp_pos`	M	gpcc:FloatVectorType	Indicates the x-, y- and z-coordinates of the position of the viewport in meters in the global reference coordinate system. The values in the array are in said order and the length of array is three. If position of the RV is dynamic, this attribute specifies the initial viewport position for this RV. Otherwise, this attribute specifies the static viewport position.
ViewportInfo`@vp_quat`	M	gpcc:IntVectorType	Indicates the x-, y- and z-components of the rotation of the viewport using the quaternion representation. The fourth component (w) may be calculated when other components are known. The integer values shall be mapped to range -1 and 1, inclusive. If the rotation of the RV is dynamic, this attribute specifies the initial viewport rotation for this RV. Otherwise, this attribute specifies the static viewport rotation.
ViewportInfo`@vp_center_view_flag`	O	xs:boolean	If equal to 1, this attribute indicates that the viewport position signalled corresponds to the centre of the viewport. If equal to 0, it indicates that the viewport position signalled corresponds to one of two stereo positions of the viewport.
ViewportInfo`@vp_left_view_flag`	O	xs:boolean	If equal to 1, this attribute indicates that the viewport information signalled correspond to the left stereo position of the viewport. If equal to 0, it indicates that the viewport information signalled correspond to the right stereo positions of the viewport.
ViewportInfo`@viewport_description`	O	xs:string	Null-terminated UTF-8 string describing the human-readable textual information associated with the viewport (e.g. “VIP Tribune View”, “Marathon Tribune View”, etc.)
ViewportInfo`@viewport_type`	O	xs:integer	Type of the recommended viewport as listed in Table 10
Key: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an `@`.			

The data type for the attribute shall be as defined in the XML schema. An XML schema for G-PCC component information signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:gpcc:2023` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:gpcc:2023"
    xmlns:omaf="urn:mpeg:mpegI:gpcc:2023"
    elementFormDefault="qualified">

    <xs:attribute name="viewport_id" type="xs:integer" use=”optional” />
    <xs:element name=”ViewportInfo” type=”gpcc:ViewportInfoType” />
```

```
    <xs:complexType name=”ViewportInfoType”>
      <xs:attribute name="vp_pos" type="gpcc:FloatVectorType" use="required" minLength="3"
maxLength="3" />
      <xs:attribute name="vp_quat" type="gpcc:IntVectorType" use="required" minLength="3"
maxLength="3"/>
      <xs:attribute name=”vp_center_view_flag” type="xs:boolean" use=”optional” />
      <xs:attribute name="vp_left_view_flag" type="xs:boolean" use="optional" />
      <xs:attribute name="viewport_description" type="xs:string" use="optional" />
      <xs:attribute name="viewport_type" type="xs:integer" use="optional" default="0" />
    </xs:complexType>
</xs:schema>
```

10.4.2 Signalling of dynamic recommended viewports

For dynamic viewports where the viewport position and rotation change over time, the RV descriptor shall be associated with a timed metadata Representation carrying a timed metadata track with sample entry type `'gpdv'`. In particular, the timed-metadata track for viewport signalling shall be carried in a separate Adaptation Set and associated with a Representation of the Main Adaptation Set using the `@associationId` attribute, defined in ISO/IEC 23009-1, with a `@associationType` value that includes the `4CC 'gpdv'` for the corresponding Adaptation Set or Representation.

11 Encapsulation and signalling in MMT

11.1 Encapsulation of G-PCC bitstream for MMT streaming

When the G-PCC bitstream is stored in the ISOBMFF file using multi-track encapsulation mode, each track is proposed to be encapsulated into a separate MMT asset, which is then packetized into MMTP packets as specified in ISO/IEC 23008-1. For the server and client to be able to identify a group of assets belonging to a certain G-PCC component, a G-PCC defined application message is proposed.

To support the delivery of G-PCC bitstream using MMT, each track inside the multi-track ISOBMFF G-PCC container will be encapsulated into a separate G-PCC Asset, where the number of assets equals the number of tracks inside the multitrack G-PCC ISOBMFF container. Multiple assets which correspond to a single G-PCC component are grouped and signalled as a G-PCC asset group in application specific message. Alternative component tracks can also be exposed using the G-PCC application-specific messages to enable efficient server and client selection decisions without the need for parsing the ISOBMFF file inside the MMTP packet.

11.2 MMT signalling descriptors

11.2.1 Asset reference descriptor

11.2.1.1 General

This descriptor provides the reference relationship information among the assets. This descriptor shall be added to the asset descriptor loop of the signalling messages or signalling tables (e.g. MP table) of an asset referencing other assets.

For a GPCC Content, asset reference descriptor is carried in the asset descriptor loop of signalling messages or signalling tables of an Asset for the G-PCC geometry track. The descriptor includes track references to the Assets carrying G-PCC attribute track with the track reference type for each GPCC attribute component. When G-PCC tile tracks are used, the asset descriptor loop of the Asset carrying G-PCC tile base track includes asset reference descriptor to the Assets carrying the G-PCC geometry tile track and the asset descriptor loop of the Assets carrying G-PCC geometry tile track carries the asset reference descriptor providing references to the Assets carrying G-PCC attribute tile tracks.

11.2.1.2 Syntax

Table 16 shows the syntax of the Asset Reference Descriptor.

Table 16 — Asset reference descriptor

Syntax	Values	No. of bits	Mnemonic
Asset_Reference_Descriptor (){			
descriptor_tag		16	uimsbf
descriptor_length		8	uimsbf
group_identification		8	uimsbf
number_of_reference	N	8	Uimsbf
for(i=0 ; N ; i++){			
reference_type		32	char
asset_id()			
}			
}			

11.2.1.3 Semantics

`descriptor_tag` indicates the type of a descriptor.

`descriptor_length` specifies the length in bytes counting from the next byte after this field to the last byte of the descriptor.

`group_identification` specifies the group identification which identifies a group of Assets in reference relationship.

`number_of_reference` specifies the number of referenced Assets by the Asset this descriptor is associated with.

`reference_type` specifies the type of reference. This is described in four-character code `("4CC")` type registered in ISO/IEC 14496-12 (see https://www.iso.org/mara).

`asset_id()` provides the identifier of the Asset referenced by the Asset this descriptor is associated with, as defined in ISO/IEC 23008-1:2023, subclause 10.6.2.

11.2.2 G-PCC Asset descriptor

11.2.2.1 General

This asset descriptor is used to inform the receiving entity and the consuming application about the content of an Asset that carries G-PCC bitstream. This descriptor shall be added to the asset descriptor loop of the signalling messages or signalling tables (e.g. MP table) of an asset comprising a G-PCC bitstream.

11.2.2.2 Syntax

Table 17 shows the syntax of the G-PCC Asset Descriptor.

Table 17 — G-PCC asset descriptor

Syntax	Values	No. of bits	Mnemonic
GPCC_descriptor () {			
descriptor_tag		16	uimsbf
descriptor_length		16	uimsbf
data_type		8	uimsbf
all_tiles_present_flag		1	bslbf
reserved	'1111111'	7	bslbf
if (!all_tiles_present_flag) {			
num_tiles	N1	16	uimsbf
for (i=0; i<N1; i++) {			
tile_id		16	uimsbf
}			
}			
}			

11.2.2.3 Semantics

`descriptor_tag` indicates the type of a descriptor.

`descriptor_length` specifies the length in bytes counting from the next byte after this field to the last byte of the descriptor.

`data_type` indicates the type of G-PCC data present in this Asset group. The values for this field are listed in Table 18.

Table 18 — Values of `data_type`

Value	Description
0x00	All G-PCC components (Geometry and Attributes) data
0x01	Geometry component
0x02	Attribute component
0x03	SPS, GPS, APS, FSAP and tile Inventory
0x04	3D Spatial region timed meta data information
0x05	Viewport timed metadata information
0x05-0xFF	Reserved

`all_tiles_present_flag` indicates weather all the tiles for the geometry component are part of an Asset or not. Value 1 indicates that data for all the G-PCC tiles is available in the Asset. Value 0 indicates that data for a sub-set of the G-PCC tiles is available in the Asset.

`num_tiles` indicates the number of tiles carried in this Asset.

`tile_id` indicates a unique identifier for a particular G-PCC tile.

11.3 MMT application-specific signalling messages

11.3.1 General

MMTP specification defines application-specific signalling messages which allow for the delivery of application-specific information. For streaming G-PCC bitstream using the MMT protocol, a set of G-PCC application-specific signalling messages are defined. These G-PCC-specific signalling messages shall have

an application identifier with a URN value of "urn:mpeg:mmt:app:gpcc:2023". The following application message types are defined.

— GPCCAssetGroupMetadataMessage

  — The sending entity sends this message to inform the client about the set of G-PCC assets available at the server, and to provide a list of those assets which are being streamed to the receiving entity.

— GPCCAssetSelectionMessage

  — The receiving client uses this message to request the set of G-PCC assets to be streamed by the sending entity.

— GPCCViewChangeFeedback

  — The receiving client uses this feedback message to send an indication of the user's current viewing space to the sending entity.

The GPCCAssetGroupMetadataMessage is mandatory when sending G-PCC bitstream via MMT. This message provides the receiving client with a list of G-PCC assets available at the server. It also informs the receiving client about which of these assets are currently being streamed. From this list, the client can request a unique subset of these G-PCC assets using the GPCCAssetSelectionMessage. The list of application message types is provided in Table 19.

Table 19 — G-PCC application message types

Application Message Type	Application Message Name
0x01	GPCCAssetGroupMetadataMessage
0x02	GPCCAssetSelectionMessage
0x03	GPCCViewChangeFeedback
0x04-0xFF	Reserved

11.3.2 GPCC Asset Group Metadata Message

11.3.2.1 Syntax

Table 20 shows the syntax of the GPCCAssetGroupMetadataMessage.

Table 20 — Syntax of GPCCAssetGroupMetadataMessage

Syntax	Values	No. of bits	Mnemonic
Application () {			
message_id		16	uimsbf
version		8	uimsbf
length		16	uimsbf
message_payload {			
application_identifier()			
if (application_identifier == "urn:mpeg:mmt:app:gpcc:2023") {			
app_message_type		8	uimsbf
if (app_message_type == 0x01) {			
num_gpcc_asset_groups	N1	8	uimsbf
start_time		32	uimsbf
for (i=0; i<N1; i++) {			
asset_group_id		16	uimsbf
num_assets	N2	16	uimsbf
3D_regions_info_flag	1	1	bslbf
reserved	"1111111"	7	bslbf
for (j=0; j<N2; j++)			
asset_id()		16	uimsbf
if (3D_regions_info_flag) {			
num_regions	N3	16	uimsbf
for (j=0; j<N3; j++)			
GPCCSpatialRegionStruct()			
}			
}			
}			
}			
}			

11.3.2.2 Semantics

`message_id` indicates the identifier of the G-PCC application message.

`version` indicates the version of the G-PCC application message.

`length` indicates the length of the G-PCC application message in bytes, counting from the beginning of the next field to the last byte of the message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as a urn that uniquely identifies the type of application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 19. The length of this field is 8 bits.

`num_gpcc_asset_groups` indicates the number of G-PCC asset groups, where each asset group contains the assets associated with a G-PCC component.

`asset_group_id` indicates the identifier of the asset group associated with a G-PCC component.

`num_assets` indicates the number of assets within this asset group which are associated with a G-PCC component.

`start_time` indicates the presentation time of the G-PCC component from which the state of the assets listed in this message are applicable.

`3d_regions_info_flag` indicates the 3D spatial regions are available in the G-PCC bitstream or not. This flag is set to 0 or 1 for only geometry asset group. This flag shall be set to 0 for all other asset groups.

`GPCCSpatialRegionStruct()` is an instance of GPCCSpatialRegionStruct and carries the information of the 3D spatial regions covered by the asset group.

`asset_id` provides the asset identifier of the asset as defined in ISO/IEC 23008-1:2023, subclause 10.6.2.

11.3.3 GPCC Asset Selection Message

11.3.3.1 Syntax

Table 21 shows the syntax of the GPCCAssetSelectionMessage.

Table 21 — Syntax of GPCCAssetSelectionMessage

Syntax	Values	No. of bits	Mnemonic
Application () {			
message_id		16	uimsbf
version		8	uimsbf
length		16	uimsbf
message_payload {			
application_identifier()			
if (application_identifier == "`urn:mpeg:mmt:app:gpcc:2023`") {			
app_message_type		8	Uimsbf
if (app_message_type == 0x02) {			
num_selected_asset_groups	N1	8	Uimsbf
for (i=0; i<N1; i++) {			
asset_group_id		16	uimsbf
reserved	"1111"	4	bslbf
switching_mode		4	bslbf
num_assets	N2	16	uimsbf
if (switching_mode == 0x01 || 0x02) {			
for (j=0; j<N2; j++) {			
asset_id()			
}			
}			
}			
}			
}			
}			
}			

11.3.3.2 Semantics

`message_id` indicates the identifier of the G-PCC application message.

`version` indicates the version of the G-PCC application message.

`length` indicates the length of the G-PCC application message in bytes, counting from the beginning of the next field to the last byte of the message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as a urn that uniquely identifies the type of application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 19. The length of this field is 8 bits.

`num_selected_asset_groups` indicates the number of asset groups for which there is an associated state change request by the receiving entity.

`asset_group_id` indicates the identifier of the asset group associated with a G-PCC content.

`switching_mode` indicates the switching mode used for the selection of assets as requested by the receiving entity (see Table 22).

Table 22 — Switching modes and definition of `switching_mode`

Value	Switching mode	Definition of switching mode
0x1	Refresh	For each asset listed as specified by its `asset_id`, its `state_flag` will be set to "1", and the `state_flag` for all other non-listed assets of the same `asset_group_id` will be set to "0". The states for assets of other non listed asset groups will remain unchanged.
0x2	Toggle	For each asset listed as specified by its `asset_id`, its `state_flag` will be changed (to "1", if originally "0", to "0" if originally "1"). The states for all non listed assets will remain unchanged
0x3	Send all	For the specified asset group, all associated assets within the group have their `state_flag` set to "1".
0x4~0xF	Reserved	Reserved

`num_assets` indicates the number of assets signalled for the state change according to the switching mode specified.

`asset_id` indicates the identifier for the asset as defined in ISO/IEC 23008-1:2023, subclause 10.6.2, for the state change according to the switching mode specified.

11.3.4 GPCC View Change Feedback Message

11.3.4.1 General

For view-dependent delivery of G-PCC bitstream through MMT, the G-PCC player uses the G-PCC view change feedback message to send its current viewport information to the server, after which the server can select and deliver the Assets corresponding to that viewport to the G-PCC player.

11.3.4.2 Syntax

Table 23 shows the syntax of the G-PCC View Change Feedback Message.

Table 23 — G-PCC View Change Feedback Message

Syntax	Values	No. of bits	Mnemonic
GPCC_view_change_feedback_message () {			
message_id		16	uimsbf
version		8	uimsbf
length		16	uimsbf
message_payload {			
application_identifier()			
app_message_type		8	uimsbf
vp_pos_x		32	uimsbf
vp_pos_y		32	uimsbf
vp_pos_z		32	uimsbf
vp_quat_x		32	uimsbf
vp_quat_y		32	uimsbf
vp_quat_z		32	uimsbf
clipping_near_plane		32	uimsbf
clipping_far_plane		32	uimsbf
horizontal_fov		32	uimsbf
vertical_fov		32	uimsbf
last_processed_media_timestamp		32	uimsbf
}			
}			

11.3.4.3 Semantics

`message_id` indicates the identifier of the G-PCC application message.

`version` indicates the version of the G-PCC application message.

`length` indicates the length of the G-PCC application message in bytes, counting from the beginning of the next field to the last byte of the message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as a URN that uniquely identifies the application to consume the contents of this message.

`app_message_type` indicates the type of the G-PCC application message.

`vp_pos_x`, `vp_pos_y`, `vp_pos_z` respectively indicates the x, y and z coordinates of the position of the viewport in meters in the global reference coordinate system. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

`vp_quat_x`, `vp_quat_y`, `vp_quat_z` indicates the x, y, and z components, respectively, of the rotation of the viewport region using the quaternion representation. The values shall be in the range of -2^{30} to 2^{30}, inclusive. When the component of rotation is not present, its value shall be inferred to be equal to 0. The value of rotation components may be calculated as follows:

$qX = cam\_quat\_x \div 2^{30}$, $qY = cam\_quat\_y \div 2^{30}$, $qZ = cam\_quat\_z \div 2^{30}$

The fourth component, `qW`, for the rotation of the current camera model using the quaternion representation is calculated as follows:

$qW = Sqrt(1 - (qX^2 + qY^2 + qZ^2))$

The point (w, x, y, z) represents a rotation around the axis directed by the vector (x, y, z) by an angle `2*cos^{-1}(w)=2*sin^{-1}(sqrt(x^{2}+y^{2}+z^{2}))`.

`clipping_near_plane, clipping_far_plane` indicates the near and far depths (or distances) based on the near and far clipping planes of the viewport in meters. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

`horizontal_fov` specifies the longitude range corresponding to the horizontal size of the viewport region, in units of radians. The value is in the range 0 to 2π.

`vertical_fov` specifies the latitude range corresponding to the vertical size of the viewport region, in units of radians. The value is in the range 0 to π.

`last_processed_media_timestamp` indicates the presentation timestamp of the last media unit that has been appended to the decoder buffer. This field is used by the MMT sending entity to determine the next media unit from the new asset that is sent to the player. The next media unit is the one with a timestamp or sequence number immediately following the indicated timestamp.

Annex A
(normative)

File format toolsets and brands

A.1 General

This annex defines what constitutes tools, for the purposes of branding files encapsulating the G-PCC bitstream. A specific brand may require some or all the tools indicated here. A brand should be chosen that indicates the full level of support required, including any requirements on other specifications (e.g. support for aspects of the ISO base media file format specification, ISO/IEC 14496-12).

At least one volumetric visual track with volumetric visual media handler type `'volv'` shall be present in the HandlerBox of the `MediaBox`, as defined in ISO/IEC 14496-12.

A.2 Timed G-PCC data storage in ISOBMFF

A.2.1 Single track encapsulation

The brand `'gpst'` may be present among the `compatible_brands` list of the `FileTypeBox`. File readers conforming to the `'gpst'` brand shall support single track encapsulation as specified in subclause 7.3.

A.2.2 Multiple track encapsulation

The brand `'gpmt'` may be present among the `compatible_brands` list of the `FileTypeBox`. File readers conforming to the `'gpmt'` brand shall support the track encapsulation in multiple tracks as specified in subclauses 7.4 and 7.5.

A.2.3 Encapsulation with partial access support

The brand `'gppa'` may be present among the `compatible_brands` of the `FileTypeBox`. File readers conforming to the `'gppa'` brand shall support the encapsulation with G-PCC tile tracks as specified in subclause 7.5, the tools for partial access of G-PCC data as specified in subclause 9.1.

A.3 Non-timed G-PCC data storage in ISOBMFF

The brand `'gpci'` may be present among the `compatible_brands` list of the `FileTypeBox`. File readers conforming to the `'gpci'` brand shall support the encapsulation with non-timed G-PCC data as specified in Clause 8. The brand `'gpci'` does not mandate a `MovieBox('moov')` in the file.

Annex B
(normative)

GPCC DASH Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="https://www.w3.org/2001/XMLSchema"
   targetNamespace="urn:mpeg:mpegI:gpcc:2023"
   xmlns:v3c="urn:mpeg:mpegI:gpcc:2023"
   elementFormDefault="qualified">

   <xs:attribute name="gpcId" type="xs:string" use="optional" />
   <xs:attribute name="tile_Ids" type="xs:UIntVectorType" use="required"/>

   <xs:element name="component" type="gpcc:gpccComponentType"/>

   <xs:complexType name="gpccComponentType">
   <xs:attribute name="type" type="xs:string" use="required" />
   <xs:attribute name="attr_type" type="xs:unsignedByte" use="optional" />
   <xs:attribute name="attr_label_oid" type="xs:string" use="optional" />
   <xs:attribute name="attr_index" type="xs:unsignedByte" use="optional" />
   <xs:attribute name="tile_ids" type="xs:UIntVectorType" use="optional" />
   </xs:complexType>

   <xs:element name="gpsr" type="gpcc:spatialRegionMapType" />

   <xs:complexType name="spatialRegionMapType">
   <xs:element name="spatialRegion" type="gpcc:spatialRegionType" minOccurs="1"/>
   </xs:complexType>

   <xs:complexType name="spatialRegionType">
   <xs:attribute name="id" type="xs:unsignedShort" use="required" />
   <xs:attribute name="x" type="xs:int" use="optional" default="0" />
   <xs:attribute name="y" type="xs:int" use="optional" default="0" />
   <xs:attribute name="z" type="xs:int" use="optional" default="0" />
   <xs:attribute name="dx" type="xs:int" use="required" />
   <xs:attribute name="dy" type="xs:int" use="required" />
   <xs:attribute name="dz" type="xs:int" use="required" />
   <xs:attribute name="tileIds" type="xs:UIntVectorType" use="optional" />
   </xs:complexType>

   <xs:attribute name="viewport_id" type="xs:integer" use="optional" />
   <xs:element name="ViewportInfo" type=" gpcc:ViewportInfoType"/>

    <!-- viewport -->
   <xs:complexType name="ViewportInfoType">
   <xs:attribute name="vp_pos" type="gpcc:FloatVectorType" use="required"
      minLength="3" maxLength="3"/>
```

```
   <xs:attribute name="vp_quat" type="gpcc:IntVectorType" use="required"
      minLength="3" maxLength="3"/>
   <xs:attribute name="vp_center_view_flag" type="xs:boolean" use="optional"/>
   <xs:attribute name="vp_left_view_flag" type="xs:boolean" use="optional"/>
   <xs:attribute name="viewport_description" type="xs:string" use="optional"/>
   <xs:attribute name="viewport_type" type="xs:integer" use="optional" default="0"/>
   <xs:anyAttribute processContents="skip"/>
   </xs:complexType>

   <!-- float and int vectors -->
   <xs:simpleType name="FloatVectorType">
   <xs:list itemType="xs:float"/>
   </xs:simpleType>
   <xs:simpleType name="IntVectorType">
   <xs:list itemType="xs:integer"/>
   </xs:simpleType>

</xs:schema>
```

Annex C
(normative)

MIME types and sub-parameters

C.1 MIME types and sub-types

When MIME type is associated with G-PCC data as described in this specification, MIME type of `'application'` shall be used along with the sub-type `'mp4'`.

Example

Content-Type: `application/mp4`;

C.2 Sub-parameters for 'codecs' parameter

C.2.1 General

When the `'codecs'` parameter of a MIME type is used, as defined in IETF RFC 6381, the sub-parameters in this annex apply when the MIME type identifies a file format of this family and the `'codecs'` parameter starts with a sample-entry code from this document.

C.2.2 G-PCC family

When the first element of a value is a code indicating a codec from ISO/IEC 23090-9, as documented in subclause 7.3 (`'gpe1'` or `'gpeg'`), subclause 7.4 (`'gpc1'` or `'gpcg'`) or subclause 7.5 (`'gpeb'`, `'gpcb'` or `'gpt1'`), the elements following are a series of values from `seq_parameter_set` syntax structure, as defined in ISO/IEC 23090-9, contained in SPS of the G-PCC decoder configuration record, separated by period characters (“.”). In all numeric encodings, leading zeroes may be omitted.

— the `simple_profile_compliant` encoded as a decimal number;

— the `dense_profile_compliant` encoded as a decimal number;

— the `predictive_profile_compliant` encoded as a decimal number;

— the `main_profile_compliant` encoded as a decimal number;

— the `level_idc` encoded as a decimal number;

Example

codecs=`gpe1.1.0.0.0.4` (Simple profile, Level 4.)

For `'gpe1'`, `'gpeg'`, `'gpeb'` or `'gpcb'`, the referenced G-PCC decoder configuration record is taken form the respective track. For `'gpc1'` or `'gpcg'`, the referenced G-PCC decoder configuration record is taken from the respective track when it is geometry track, otherwise, when it is attribute track, the referenced G-PCC decoder configuration record is taken from the geometry track referenced by the attribute track. For `'gpt1'`, the referenced G-PCC decoder configuration record is taken from the G-PCC tile base track referenced by the respective tracks.