INTERNATIONAL STANDARD

ISO/IEC 23090-10

First edition
2022-05

Information technology — Coded representation of immersive media —

Part 10:
Carriage of visual volumetric video-based coding data

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

Partie 10: Transport de données de codage basé sur la vidéo volumétrique

Reference number
ISO/IEC 23090-10:2022(E)

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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

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

This document addresses the storage of visual volumetric video-based coding data in files based on ISO/IEC 14496-12, reusing existing tools for storage of video-coded components. Another important aspect considered by this document is supporting flexible extraction of component streams at delivery or decoding time, or both.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents or patents.iec.ch.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Coded representation of immersive media —

Part 10:
Carriage of visual volumetric video-based coding data

1 Scope

This document specifies carriage of coded media representations which comply with visual volumetric video-based coding and video-based point cloud compression (specified in ISO/IEC 23090-5).

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.

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

IETF RFC 6381, *The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types*

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

ISO/IEC 14496-15, *Information technology — Coding of audio-visual objects — Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO based media file format*

ISO/IEC 23008-1:2017, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 1: MPEG media transport (MMT)*

ISO/IEC 23009-1:2019, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*

ISO/IEC 23090-5:2021, *Information technology — Coded representation of immersive media — Part 5: Visual Volumetric Video-based Coding (V3C) and Video-based Point Cloud Compression (V-PCC)*

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 23090-5 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
atlas parameter sets
non-ACL NAL units that have nal_unit_type equal to NAL_ASPS, NAL_AAPS, or NAL_AFPS.

3.2
V3C content
volumetric media that is encoded

Note 1 to entry: For the purposes of this document, the media shall be encoded using ISO/IEC 23090-5.

3.3
volumetric visual track
track with a handler type reserved to describe volumetric visual track

3.4
V3C track
V3C bitstream track, V3C atlas track or V3C atlas tile track

3.5
V3C bitstream track
volumetric visual track containing V3C bitstream in case of single-track container

3.6
V3C atlas track
volumetric visual track containing V3C atlas bitstream in case of multi-track container

3.7
V3C atlas tile track
volumetric visual track containing portion of V3C atlas bitstream corresponding to one or more tiles in case of multi-track container

3.8
V3C video component track
video track which carries 2D video encoded data for any of the occupancy, geometry, or attribute component video bitstreams of the V3C bitstream

4 Abbreviated terms

2D	two-dimensional
3D	three-dimensional
CVS	coded V3C sequence
DASH	dynamic adaptive streaming over HTTP
HTTP	Hyper-text transfer protocol
IRAP	intra random access point
ISOBMFF	ISO base media file format
LoD	level of detail
PCC	point cloud compression
SEI	supplemental enhancement information
V3C	visual volumetric video-based coding
VPS	V3C parameter set
V-PCC	video-based Point Cloud Coding

5 Overview

5.1 General

Visual volumetric video-based coding (V3C) provides mechanism for coding visual volumetric frames. Visual volumetric frames are coded by converting the 3D volumetric information into a collection of 2D images and associated data. The converted 2D images are coded using widely available video and image coding specifications and the associated data, i.e., atlas data, is coded according to ISO/IEC 23090-5. The coded images and the coded atlas data are multiplexed and form a V3C bitstream.

A V3C bitstream consists of one or more CVSs. A CVS starts with a VPS, included in at least one V3C unit or provided through external means, and contains one or more V3C units carrying V3C sub-bitstreams, with each V3C sub-bitstream associated with a V3C component., e.g., atlas, occupancy, geometry, or attribute.

5.2 Overall architecture for carriage of V3C data

Figure 1 shows a typical content flow process for V3C media.

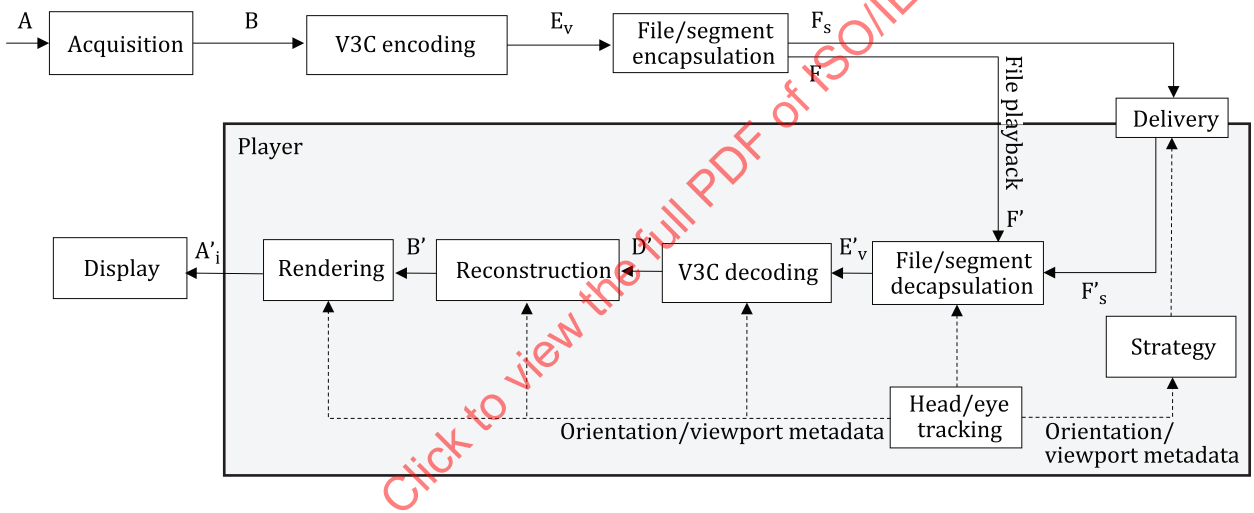

Figure 1 — Content flow process for V3C media

A real-world or synthetic visual scene (A) is captured by a set of cameras, a camera device with multiple lenses and sensors, or by virtual cameras. The acquisition results in source volumetric data (B). One or multiple volumetric frames are encoded as a coded V3C bitstream including an atlas bitstream, at most one occupancy bitstream, a geometry bitstream, and zero or more attribute bitstreams (E_v). One or more coded bitstreams are then packaged into a media file for local playback (F) or a sequence of an initialization segment and media segments for streaming (F_s), according to a particular media container file format. In this document, the media container file format is the ISO Base Media File Format specified in ISO/IEC 14496-12. The file encapsulator may also include metadata into the file or the segments. The segments F_s are delivered using a delivery mechanism to a player.

The file that the file encapsulator outputs (F) is identical to the file that the file decapsulator takes as input (F'). A file decapsulator processes the file (F') or the received segments (F'_s) and extracts the coded bitstreams (E'_v) and parses the metadata. The V3C bitstream is then decoded into a decoded signal (D'). The decoded volumetric data (D') are reconstructed, rendered, and displayed onto the screen of a head-mounted display or any other display device based on the current viewing orientation or viewport. The current viewing orientation is determined by the head tracking and possibly also eye tracking functionality. In viewport-dependent delivery, the current viewing orientation is also passed to the strategy module, which determines the tracks to be received based on the viewing orientation.

The process described above is applicable to both live and on-demand use cases.

The following interfaces are normatively specified in this document:

- F/F': media file including the specification of the track formats, which may contain constraints on the elementary streams contained within the samples of the tracks; see Clause 7 for timed V3C content and Clause 8 for non-timed V3C data.
- Clause 11 specifies the delivery related interfaces for DASH delivery.
- Clause 12 specifies the delivery related interfaces for MMT delivery.

5.3 Summary of referenceable code points

5.3.1 Brands

ISO/IEC 14496-12 defines the concept of brands, which may be indicated in the FileTypeBox. Brands are used in this document to indicate conformance to an encapsulation mode and a specific set of tools, as well as requirements on other specifications (e.g., ISO/IEC 14496-12).

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

Table 1 — Brands specified in this document

Brand	Clause	Informative description
v3st	A.2	Single track encapsulation mode
v3mt	A.3	Multi-track encapsulation mode
v3mp	A.3	Multi-track encapsulation mode with partial access support
v3nt	A.4	Non-timed V3C data

5.3.2 Uniform resource names

The URNs specified in this document are listed in Table 2.

Table 2 — URNs specified in this document

URN	Clause	Informative description
urn:mpeg:mpegI:v3c:2020	11.3.1	Namespace for the XML elements and attributes specified in this document
urn:mpeg:mpegI:v3c:2020:component	11.3.2	Scheme identifier for the V3C component DASH MPD descriptor
urn:mpeg:mpegI:v3c:2020:v3c	11.3.3	Scheme identifier for the V3C content DASH MPD descriptor
urn:mpeg:mpegI:v3c:2020:v3sr	11.6.1	Scheme identifier for the V3C static spatial region DASH MPD descriptor

5.3.3 Restricted scheme types

The restricted scheme types specified in this document are listed in Table 3.

Table 3 — Restricted scheme types specified in this document

Restricted scheme type	Clause	Informative description
vvvc	7.4.5.1	V3C component video

5.3.4 Sample entry types

The sample entry types specified in this document are listed in Table 4.

Table 4 — Sample entry types specified in this document

Sample entry type	Clause	Informative description
`v3e1`	7.3.2.2	For use with the single-track mode with all atlas parameter sets and SEI messages carried in decoder configuration record
`v3eg`	7.3.2.2	For use with the single-track mode with atlas parameter sets and SEI messages carried in decoder configuration record and in track samples
`v3c1`	7.4.2	For use with the multi-track mode with a single atlas and all atlas parameter sets and SEI messages carried in decoder configuration record
`v3cg`	7.4.2	For use with the multi-track mode with a single atlas and atlas parameter sets and SEI messages carried in decoder configuration record and in track samples
`v3cb`	7.4.2	For use with a base track in the multi-track mode with multiple atlases
`v3a1`	7.4.2	For use with an atlas track in the multi-track mode with multiple atlases and all atlas parameter sets and SEI messages carried in decoder configuration record
`v3ag`	7.4.2	For use with an atlas track in multi-track mode with multiple atlases and atlas parameter sets and SEI messages carried in decoder configuration record and in track samples
`v3t1`	7.4.3	For use with an atlas tile track in the multi-track mode
`dyvm`	9.7.1	For use with a timed metadata track indicating the dynamic spatial regions that are dynamically changing over time
`6vpt`	10.3.2	For use with a timed metadata track indicating viewport information that are dynamically changing over time

5.3.5 Box types

The box types specified in this document are listed in bold in Table 5. Mandatory boxes are marked with an asterisk. Box types without a four-character code are marked with '-' in the structure.

Table 5 — Box types specified in this document

Box types, structure, and cross-reference (Informative)												
moov										*	ISOBMFF	*container for all the metadata*
	trak									*	ISOBMFF	*container for an individual track or stream*
		trgr									ISOBMFF	*track grouping indication*
			potg								7.4.8.2	*playout track group box*
			vtcg								9.4	*atlas tile components track group box*
		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	*sample entry or restricted sample entry*
							rinf				ISOBMFF	*restricted scheme info box*
								frma			ISOBMFF	*original format box*
								schm			ISOBMFF	*scheme type box*
								schi			ISOBMFF	*scheme information box*
									vunt		7.2.3	*V3C unit header box*
									mmvi		7.4.5.2	*Multimap video box*
						dyvm					9.7.2	*dynamic volumetric metadata sample entry*
						6vpt					10.3.2	*viewport information sample entry*
							6vpC				10.3.2	*viewport information configuration box*
						-					ISOBMFF	*visual sample entry*

Table 5 *(continued)*

Box types, structure, and cross-reference (Informative)												
						-					6.4	*volumetric visual sample entry*
							v3cC				7.2.2	*V3C decoder configuration box*
							vunt				7.2.3	*V3C unit header box*
							v3tC				7.4.3	*V3C atlas tile configuration box*
							vpbb				9.5	
							v3sc				9.6	*Static spatial region collection box*
meta											ISOBMFF	*Metadata*
	grpl										ISOBMFF	*group list box*
		eply									8.5.5.2	*playout entity group box*
		swpc									7.2.5	*object switch alternatives box*
	iprp										ISOBMFF	*item properties box*
		ipco									ISOBMFF	*item property container box*
			v3cp								8.5.2	*V3C configuration item property*
			vutp								8.5.3	*V3C unit header item property*
			v3tp								8.5.4	*V3C atlas tile configuration item property*

5.3.6 Track reference types

The track reference types specified in this document are listed in Table 6.

Table 6 — Track reference types specified in this document

Track reference type	Clause	Informative description
`v3cs`	7.4.6.1	Referenced track is a V3C atlas track
`v3ct`	7.4.6.2	Referenced track is a V3C atlas tile track
`v3vo`	7.4.6.3	Referenced track is a V3C video component track carrying occupancy data
`v3vg`	7.4.6.3	Referenced track is a V3C video component track carrying geometry data
`v3va`	7.4.6.3	Referenced track is a V3C video component track carrying attribute data

5.3.7 Track grouping types

The track grouping types specified in this document are listed in Table 7.

Table 7 — Track grouping types specified in this document

Track grouping type	Clause	Informative description
`potg`	7.4.8.2	Playout track grouping
`vtcg`	9.4	V3C tile components track grouping

5.3.8 Entity grouping types

The entity grouping types specified in this document are listed in Table 8.

Table 8 — Entity grouping type specified in this document

Entity grouping type	Clause	Informative description
`eply`	8.5.5.2	Playout entity grouping
`swpc`	7.2.5	Object switch alternatives box

5.3.9 Sample grouping types

The sample grouping types specified in this document are listed in Table 9.

Table 9 — Sample grouping types specified in this document

Sample grouping type	Clause	Informative description
`vaps`	7.2.4	V3C atlas parameter set sample grouping

6 Volumetric media

6.1 General

This clause contains descriptions and definitions, which are intended to be defined in ISO/IEC 14496-12.

6.2 Volumetric visual media

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 this document.

Multiple volumetric visual tracks may be present in the file.

6.3 Volumetric visual media header

6.3.1 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.

6.3.2 Syntax

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

6.3.3 Semantics

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

6.4 Volumetric visual sample entry

6.4.1 Definition

Volumetric visual tracks shall use a `VolumetricVisualSampleEntry`

6.4.2 Syntax

```
class VolumetricVisualSampleEntry(unsigned int(32) codingname)
   extends SampleEntry (codingname) {
```

```
   unsigned int(8)[32] compressorname;
   // other boxes from derived specifications
}
```

6.4.3 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.

6.5 Volumetric visual sample group entry

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

6.6 Volumetric visual samples

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

7 Carriage of visual volumetric video-based coding data

7.1 General

This clause defines the storage for a V3C bitstream utilizing the existing capabilities of the ISO base media file format and defining extensions, when necessary.

Two methods of the storage for V3C data are specified; a single track encapsulation, where a V3C data is stored as one track, and multi-track encapsulation, where each V3C component is stored as a separate track.

7.2 Common boxes and data structures

7.2.1 V3C decoder configuration record

7.2.1.1 Definition

The V3C decoder configuration record provides V3C bitstream's decoding specific information (i.e. parameter sets and SEI messages) for further configuration and initialization of the V3C decoder. This document sets the following restrictions for V3C content encapsulation:

— `num_of_v3c_parameter_sets` in `V3CDecoderConfigurationRecord` shall be equal to 1. Under the '`v3a1`' and '`v3ag`' sample entry, `num_of_v3c_parameter_sets` shall be equal to 0.

— For a given atlas the tile IDs shall remain unique for the duration of the sequence. Tile IDs shall not be re-used between atlas frame parameter sets.

— V3C data is naturally represented as variable bit rate in the file format and should be filled for transmission if needed. Filler Data NAL units and Filler Data SEI messages shall not be present in the file format stored stream when the sample entry does not also permit in-stream parameter sets.

If the version number is not supported or recognized by the reader, then it shall not attempt to decode this configuration record or the bitstreams to which it applies.

7.2.1.2 Syntax

```
aligned(8) class V3CDecoderConfigurationRecord {
   // version 0
   unsigned int(3) unit_size_precision_bytes_minus1;
   unsigned int(5) num_of_v3c_parameter_sets;
   for (int i=0; i < num_of_v3c_parameter_sets; i++) {
      unsigned int(16) v3c_parameter_set_length;
      // v3c_unit() as defined in ISO/IEC 23090-5
      v3c_unit v3c_parameter_set(v3c_parameter_set_length);
   }
   unsigned int(8) num_of_setup_unit_arrays;
   for (int j=0; j < num_of_setup_unit_arrays; j++) {
      unsigned int(1) array_completeness;
      bit(1) reserved = 0;
      unsigned int(6) nal_unit_type;
      unsigned int(8) num_nal_units;
      for (int i=0; i < num_nal_units; i++) {
         unsigned int(16) setup_unit_length;
         // nal_unit(size) as defined in ISO/IEC 23090-5
         nal_unit setup_unit(setup_unit_length);
      }
   }
   // additional fields
}
```

7.2.1.3 Semantics

unit_size_precision_bytes_minus1 plus 1 specifies the precision, in bytes, of the sample stream NAL unit or sample stream V3C unit to which this configuration record applies. The value of this field shall be conditional on the 4CC-code of the sample entry. For V3C atlas tracks unit_size_precision_bytes_minus1 shall be equal to ssnh_unit_size_precision_bytes_minus1 in sample_stream_nal_header(). For V3C bitstream tracks unit_size_precision_bytes_minus1 shall be equal to ssvh_unit_size_precision_bytes_minus1 in sample_stream_v3c_header().

num_of_v3c_parameter_sets specifies the number of V3C parameter set units signalled in the decoder configuration record.

v3c_parameter_set_length indicates the size, in bytes, of the v3c_parameter_set field.

v3c_parameter_set is a V3C unit payload for V3C unit of type V3C_VPS, as defined in ISO/IEC 23090-5.

num_of_setup_unit_arrays indicates the number of arrays of atlas NAL units of the indicated type(s).

array_completeness when equal to 1 indicates that all atlas NAL units of the given type are in the following array and none are in the stream; when equal to 0 indicates that additional atlas NAL units of the indicated type may be in the stream; the default and permitted values are constrained by the sample entry name.

nal_unit_type indicates the type of the atlas NAL units in the following array (which shall be all of that type); it takes a value as defined in ISO/IEC 23090-5; it is restricted to take one of the values indicating a NAL_ASPS, NAL_AAPS, NAL_AFPS, NAL_PREFIX_ESEI, NAL_PREFIX_NSEI, NAL_SUFFIX_ESEI, or NAL_SUFFIX_NSEI atlas NAL unit.

num_nal_units indicates the number of atlas NAL units of type nal_unit_type included in the configuration record for the stream to which this configuration record applies.

setup_unit_length indicates the size, in bytes, of the setup_unit field. The length field includes the size of both the NAL unit header and the NAL unit payload but does not include the length field itself.

setup_unit contains a NAL unit according to related nal_unit_type. When present in setup_unit, NAL_PREFIX_ESEI, NAL_PREFIX_NSEI, NAL_SUFFIX_ESEI, or NAL_SUFFIX_NSEI contain SEI messages of a ‘declarative’ nature, that is, those that provide information about the stream as a whole. An example of such an SEI could be a user-data SEI.

The setup_unit arrays shall include atlas sub-bitstream parameter sets that are constant for the CVS referred to by the sample entry in which the decoder configuration record is present as well as atlas sub-bitstream SEI messages.

7.2.2 V3C decoder configuration box

7.2.2.1 Definition

A V3C decoder configuration box includes a `V3CDecoderConfigurationRecord` as defined in subclause 7.2.1.

In this this document `version` shall be equal to 0.

7.2.2.2 Syntax

```
class V3CConfigurationBox extends FullBox('v3cC', version = 0, 0) {
    V3CDecoderConfigurationRecord();
}
```

7.2.2.3 Semantics

`V3CDecoderConfigurationRecord` is defined in subclause 7.2.1

7.2.3 V3C unit header box

7.2.3.1 Definition

Box Types:	'vunt'
Container:	Sample Entry ('v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag') and SchemeInformationBox
Mandatory:	Yes
Quantity:	one

A single `V3CUnitHeaderBox` is present in the sample entry of a V3C atlas track and in `scheme_specific_data` Box array of the `SchemeInformationBox` of all V3C video component tracks. `V3CUnitHeaderBox` contains the V3C unit header describing the data carried by the respective track.

7.2.3.2 Syntax

```
aligned(8) class V3CUnitHeaderBox extends FullBox('vunt', version = 0, 0){
   v3c_unit_header header(); // 4-bytes as defined in ISO/IEC 23090-5
}
```

7.2.3.3 Semantics

`header` contains a single instance of the 32-bit V3C unit header as defined in ISO/IEC 23090-5.

7.2.4 V3C atlas parameter set sample group

7.2.4.1 Definition

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

The use of `'vaps'` for the `grouping_type` in sample grouping represents the assignment of samples in V3C atlas track or V3C bitstream track to the atlas parameter sets 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.

Sample grouping type 'vaps' shall not be used with a track with a sample entry 'v3c1', 'v3e1' or 'v3a1'.

NOTE V3C atlas parameter set sample group can be used to improve random access of atlas tracks, by removing the need to replicate parameter sets and SEI messages for sync samples.

7.2.4.2 Syntax

```
aligned(8) class V3CAtlasParamSampleGroupDescriptionEntry() extends VolumetricVisualSample
GroupEntry('vaps') {
   unsigned int(8) num_of_setup_units;
   for (int i=0; i < num_of_setup_units; i++) {
      unsigned int(16) setup_unit_length;
      // nal_unit(size) as defined in ISO/IEC 23090-5
      nal_unit setup_unit(setup_unit_length);
   }
}
```

7.2.4.3 Semantics

num_of_setup_units specifies the number of setup units signalled in the sample group description.

setup_unit_length indicates the size, in bytes, of the setup_unit field. The length field includes the size of both the NAL unit header and the NAL unit payload but does not include the length field itself.

setup_unit is a NAL unit of type NAL_ASPS, NAL_AAPS, NAL_AFPS, NAL_PREFIX_ESEI, NAL_PREFIX_NSEI, NAL_SUFFIX_ESEI, or NAL_SUFFIX_NSEI carrying data associated with this group of samples.

7.2.5 Object switch alternatives box

7.2.5.1 Definition

Box Types: 'swpc'

Container: Groups List Box ('grpl')

Mandatory: No

Quantity: Zero or more

EntityToGroupBox with grouping_type equal to 'swpc' specifies tracks and items that are associated with each other based on a logical context and are user switchable alternatives of each other. If ObjectSwitchAlternativesBox is absent, there is no information on which items or tracks should be played out together. When ObjectSwitchAlternativesBox contains alternatives, only items or tracks from one point-cloud object within an alternate group should be played or streamed at any one time.

7.2.5.2 Syntax

```
aligned(8) class ObjectSwitchAlternativesBox extends EntityToGroupBox(‘swpc') {
}
```

7.3 Single track encapsulation of V3C data

7.3.1 General

A single-track encapsulation of V3C data requires the V3C bitstream to be represented by a single-track declaration, referred to as V3C bitstream track.

Single-track encapsulation of V3C data is utilized in the case of direct ISOBMFF encapsulation of a V3C bitstream. V3C bitstream is directly stored as a single track without further processing. V3C unit header data structures are kept in the bitstream as-is. A single-track encapsulated V3C data could be provided to media workflows for further processing (e.g., multi-track file generation, transcoding, DASH segmentation, etc.).

7.3.2 V3C bitstream sample entry

7.3.2.1 Http

7.3.2.2 Definition

Sample Entry Type: 'v3e1', 'v3eg'

Container: SampleDescriptionBox

Mandatory: A 'v3e1' or 'v3eg' sample entry is mandatory

Quantity: One or more

V3C bitstream tracks shall use VolumetricVisualSampleEntry with a sample entry type of 'v3e1' or 'v3eg'.

A V3C bitstream track sample entry shall contain a V3CConfigurationBox, as defined in subclause 7.2.2.

Under the 'v3e1' sample entry, all atlas parameter sets and SEI messages, as defined in ISO/IEC 23090-5, shall be in the setup_unit array. Under the 'v3eg' sample entry, the atlas parameter sets and SEI messages may be present in the setup_unit array, or in the samples of the V3C bitstream track.

An optional BitRateBox as defined in ISO/IEC 14496-12 may be present in the V3C bitstream sample entry to signal the bit rate information of the V3C bitstream track.

7.3.2.3 Syntax

```
aligned(8) class V3CBitstreamSampleEntry() extends VolumetricVisualSampleEntry (type) {
   // type is 'v3e1' or 'v3eg'
   V3CConfigurationBox config;
}
```

7.3.2.4 Semantics

compressorname in the base class VolumetricVisualSampleEntry indicates the name of the compressor used with the value "\012V3C Coding" being recommended; the first byte is a count of the remaining bytes, here represented by \012, which (being octal 12) is 10 (decimal), the number of bytes in the rest of the string.

7.3.3 V3C bitstream track sample format

7.3.3.1 Definition

A V3C bitstream sample shall contain one or more V3C units which belong to the same presentation time, i.e., one V3C composition unit. A sample may be self-contained (e.g., a sync sample) or decoding-wise dependent on other samples of V3C bitstream track.

7.3.3.2 Syntax

```
aligned(8) class V3CBitstreamSample {
   // sample_size size of sample from SampleSizeBox
   for (int i=0; i < sample_size; ) {
      sample_stream_v3c_unit ss_v3c_unit;  // as defined in ISO/IEC 23090-5
```

```
        i += ss_v3c_unit.ssvu_v3c_unit_size +
                V3CDecoderConfigurationRecord.unit_size_precision_bytes_minus1 + 1;
    }
}
```

7.3.3.3 Semantics

ss_v3c_unit contains a single V3C unit in V3C unit sample stream format as defined in ISO/IEC 23090-5:2021, Annex C.

ssvu_v3c_unit_size specifies the size, in bytes, of the sample stream V3C unit. The number of bits used to represent ssvu_v3c_unit_size is equal to (V3CDecoderConfigurationRecord.unit_size_precision_bytes_minus1 + 1) * 8.

7.3.3.4 V3C bitstream track sync sample

A V3C bitstream sync sample shall satisfy all the following conditions:

- It shall be independently decodable.
- None of the samples that come after the sync sample (in decoding order) have any decoding dependency on any sample prior to the sync sample.
- All samples that come after the sync sample (in decoding order) are successfully decodable.

7.3.3.5 V3C bitstream track sub-sample

A V3C bitstream track sub-sample is a V3C unit which is contained in a V3C bitstream track sample.

A V3C bitstream track shall contain one SubSampleInformationBox in its SampleTableBox, or in the TrackFragmentBox of each of its MovieFragmentBoxes, which lists the V3C bitstream track sub-samples.

The 32-bit unit header of the V3C unit which represents the sub-sample shall be copied to the 32-bit codec_specific_parameters field of the sub-sample entry in the SubSampleInformationBox. The V3C unit type of each sub-sample shall be identified by parsing the codec_specific_parameters field of the sub-sample entry in the SubSampleInformationBox.

7.4 Multi-track encapsulation of V3C data

7.4.1 General

There may be three types of tracks in a multi-track encapsulated V3C data container: V3C atlas track, V3C atlas tile track, and V3C video component track. A multi-track encapsulated V3C data container shall include at least one V3C atlas track that references zero or more V3C atlas tile tracks or zero or more V3C video component tracks. A V3C atlas tile track, when present, references zero or more V3C video component tracks. The number of V3C video component tracks in a multi-track encapsulated V3C data container is dependent on the V3C toolset profile, defined in ISO/IEC 23090-5, that is used.

To indicate the association of V3C video component tracks to a V3C atlas track, or V3C atlas tile track, ISOBMFF track referencing is utilized, where the V3C atlas track, or V3C atlas tile track, contain track references to the V3C video component tracks.

Tracks belonging to the same CVS are time-aligned. Samples that contribute to the same volumetric frame across the different V3C video component tracks, V3C atlas track and V3C atlas tile tracks shall have the same composition time. Atlas parameter sets used for such samples shall have a decoding time equal or prior to the composition time of the volumetric frame. In addition, all tracks belonging to the same CVS shall have the same implied or explicit edit lists.

NOTE 1 Synchronization between the elementary streams in V3C atlas track, V3C atlas tile tracks and V3C video component tracks is handled by the ISOBMFF track timing structures (stts, ctts, and cslg), or equivalent mechanisms in movie fragments.

NOTE 2 The sync samples in the V3C atlas track, V3C atlas tile track and V3C video component tracks can be time-aligned. In the absence of time-alignment, random access can involve pre-rolling the various tracks from different sync start-times, to enable starting at the desired time. In the case of time-alignment (e.g., required by a V3C profile such as the V-PCC Basic toolset profile as defined in ISO/IEC 23090-5), the sync samples of the V3C atlas track can be considered as the random-access points for the V3C content, and random access can be done by only referencing the sync sample information of the V3C atlas track.

An example layout of a multi-track encapsulated V3C data container is shown in Figure 2. The boxes in the figure map to corresponding ISOBMFF boxes, as defined in ISO/IEC 14496-12. Payloads of V3C units of a V3C bitstream are mapped to individual tracks within the multi-track container file based on their types.

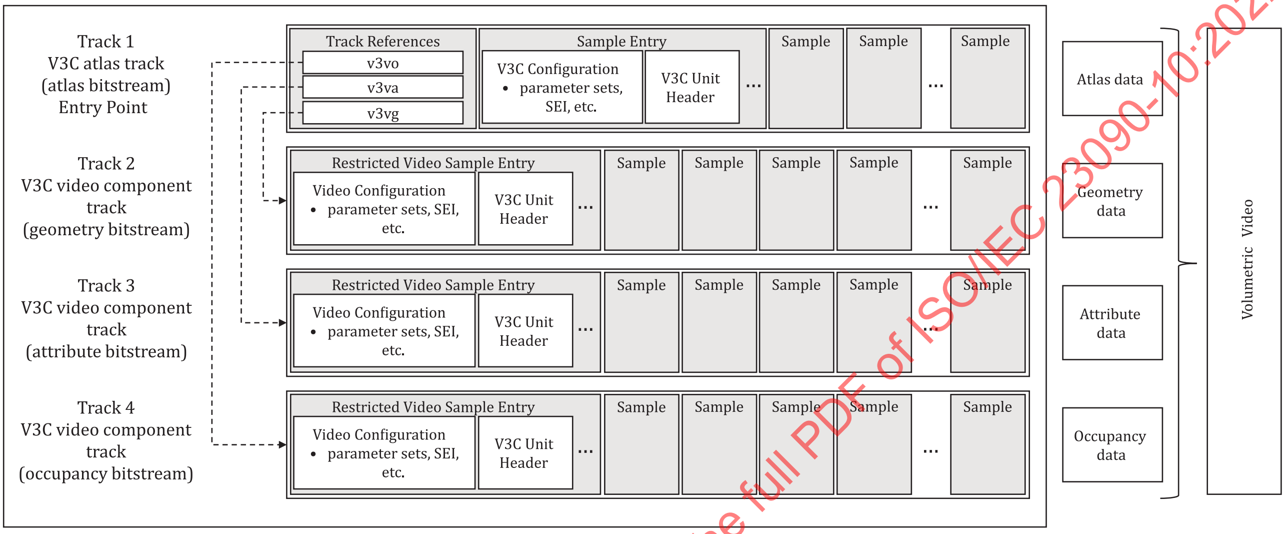

Figure 2 — Visualization of a multi-track encapsulation of V3C data container file

A multi-track encapsulated V3C data container shall include the following:

— A V3C atlas track which contains a V3C parameter set and may contain atlas parameter sets in the sample entry and atlas component bitstream NAL units in the samples. A V3C atlas track may also include track references to other tracks carrying the payloads of video compressed V3C units (i.e., V3C unit types equal to V3C_OVD, V3C_GVD, and V3C_AVD) or to V3C atlas tile tracks.

— Zero or more V3C video component tracks where the samples contain access units of a video-coded elementary stream for occupancy data (i.e., payloads of V3C units of type equal to V3C_OVD).

— Zero or more V3C video component tracks where the samples contain access units of video-coded elementary streams for geometry data (i.e., payloads of V3C units of type equal to V3C_GVD).

— Zero or more V3C video component tracks where the samples contain access units of video-coded elementary streams for attribute data (i.e., payloads of V3C units of type equal to V3C_AVD).

— Zero or more V3C atlas tile tracks where the samples contain only ACL NAL units for a sub-set of atlas tiles. V3C atlas tile track may also contain track references to other tracks carrying the payloads of video compressed V3C units (i.e., V3C unit types equal to V3C_OVD, V3C_GVD, and V3C_AVD) for the indicated sub-set of atlas tiles.

7.4.2 V3C atlas sample entry

7.4.2.1 Definition

Sample Entry Type: `'v3c1'`, `'v3cg'`, `'v3cb'`, `'v3a1'`, or `'v3ag'`

Container: SampleDescriptionBox

Mandatory: A 'v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag' sample entry is mandatory

Quantity: One or more

V3C atlas tracks use V3CAtlasSampleEntry which extends VolumetricVisualSampleEntry with a sample entry type of 'v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag'. Following restrictions are set for V3C atlas tracks:

- A V3C atlas track shall not carry ACL NAL units belonging to more than one atlas.
- In track header the track_in_movie flag is set to 1.
- A V3C atlas track sample entry contains a V3CConfigurationBox, as defined in subclause 7.2.2, and a V3CUnitHeaderBox, as defined in subclause 7.2.3.

  NOTE V3C atlas tracks with sample entry type 'v3cb', 'v3a1' and 'v3ag' are relevant for carrying multiple atlases.

Depending on V3C content or sample entry type of the atlas track, following further restrictions may be placed on atlas tracks

- When the V3C bitstream contains a single atlas, a V3C atlas track with sample entry 'v3c1' or 'v3cg' shall be used.
- When the V3C bitstream contains multiple atlases, each atlas bitstream shall be stored as a separate V3C atlas track. One of those tracks shall use sample entry type 'v3cb', while the other tracks shall use sample entry type 'v3a1', or 'v3ag'.
- A V3C atlas track with sample entry type 'v3cb' shall not include any ACL NAL units.
- Under the 'v3a1' and 'v3ag' sample entry, no V3C parameter set shall be stored in the v3c_parameter_set array.
- Under the 'v3c1' sample entry, all atlas parameter sets shall be stored in the setup_unit array. SEI messages that apply to the stream as a whole may be stored in the setup_unit array as well. Under the 'v3cg' and 'v3cb' sample entry, the parameter sets and SEI messages may be present in the setup_unit array, or in the samples of V3C atlas track.
- Under the 'v3a1' sample entry, all atlas parameter sets shall be stored in the setup_unit array. SEI messages that apply to the stream as a whole and are associated with the same vuh_atlas_id indicated in the V3C unit header box may be stored in the setup_unit array as well. Under the 'v3ag' sample entry, atlas parameter sets and SEI messages, associated with the same vuh_atlas_id indicated in the V3C unit header box, may be present in the setup_unit array, or in the samples of V3C atlas track.
- Under the 'v3c1' or 'v3a1', the default and mandatory value of array_completeness in V3CDecoderConfigurationRecord is 1 for arrays of all types of atlas parameter sets and for SEI messages that apply for the stream as a whole and are carried in the V3CDecoderConfigurationRecord. For all other arrays array_completeness is set to 0. When the sample entry 4CC is 'v3cg' or 'v3ag' or 'v3cb', the default value of array_completeness is 0 for all arrays.

An optional BitRateBox may be present in the V3C atlas sample entry to signal the bit rate information of the V3C atlas track.

7.4.2.2 Syntax

```
aligned(8) class V3CAtlasSampleEntry() extends VolumetricVisualSampleEntry (type) {
    // type is 'v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag'
    V3CConfigurationBox config;
    V3CUnitHeaderBox unit_header;
}
```

7.4.2.3 Semantics

`compressorname` in the base class `VolumetricVisualSampleEntry` indicates the name of the compressor used with the value `"\012V3C Coding"` being recommended; the first byte is a count of the remaining bytes, here represented by `\012`, which (being octal 12) is 10 (decimal), the number of bytes in the rest of the string.

7.4.3 V3C atlas tile sample entry

7.4.3.1 Definition

Sample Entry Type:	'v3t1'
Container:	SampleDescriptionBox
Mandatory:	Yes
Quantity:	One or more

V3C atlas tile tracks use `V3CAtlasTileSampleEntry` which extends `VolumetricVisualSampleEntry` with a sample entry type of `'v3t1'`.

A V3C atlas tile track samples shall contain only ACL NAL units, which belong to the same atlas. V3C atlas tile tracks shall contain ACL NAL units of at least one tile, indicated by `tile_id` in `V3CAtlasTileConfigurationBox`.

The `V3CAtlasTileSampleEntry` shall not contain `V3CConfigurationBox` or `V3CUnitHeaderBox`. Information provided by these boxes is found in the V3C atlas track sample entry that references the V3C atlas tile track. Other optional boxes may be included.

7.4.3.2 Syntax

```
class V3CAtlasTileConfigurationBox extends FullBox('v3tC', version = 0, 0) {
    unsigned int(3) unit_size_precision_bytes_minus1;
    unsigned int(1) spatial_scalability_enabled_flag;
    bit(4) reserved = 0;
    if (spatial_scalability_enabled_flag) {
            unsigned int(8) lod_index;
    }
    unsigned int(16) num_tiles;
    for(int i=0; i < num_tiles; i++){
            unsigned int(16) tile_id;
    }
}
aligned(8) class V3CAtlasTileSampleEntry() extends VolumetricVisualSampleEntry ('v3t1') {
    V3CAtlasTileConfigurationBox tile_info;
}
```

7.4.3.3 Semantics

`unit_size_precision_bytes_minus1` plus 1 specifies the precision, in bytes, of the sample stream NAL unit to which the sample entry containing this configuration box applies. The value of this field shall be equal to `ssnh_unit_size_precision_bytes_minus1` in `sample_stream_nal_header()` for the atlas component bitstream.

`spatial_scalability_enabled_flag` is a flag indicating whether the LoD-based scalability is supported by the carried V3C content.

`lod_index` indicates the LoD index value associated with the tiles carried by the atlas tile track. An atlas tile track with a certain LoD index (if present) should be selected with all atlas tile tracks with lower lod_index values carrying corresponding tiles. An LoD tile set associated with a lower lod_index value should be processed first.

`num_tiles` number of tiles contained in this track

`tile_id` specifies the tile ID of the tile present in the track. The value of `tile_id` is equal to value of `afti_tile_id` syntax element in atlas frame tile information, defined in ISO/IEC 23090-5.

`compressorname` in the base class `VolumetricVisualSampleEntry` indicates the name of the compressor used with the value "`\017V3C Atlas Tiles`" being recommended; the first byte is a count of the remaining bytes, here represented by `\017`, which (being octal 17) is 15 (decimal), the number of bytes in the rest of the string.

7.4.4 V3C atlas sample format

7.4.4.1 Definition

Each sample in a V3C atlas track or V3C atlas tile track corresponds to a single coded atlas access unit.

When `'v3cb'` sample entry is used, each sample in the V3C atlas track corresponds to one or more non-ACL NAL units.

When `'v3c1'`, `'v3cg'`, `'v3a1'` or `'v3ag'` sample entry is used and one or more V3C atlas tile tracks are present, each sample in the V3C atlas track corresponds to one or more non-ACL NAL units associated with same `vuh_atlas_id` indicated in V3C unit header box in sample entry.

NOTE When V3C atlas sample contains no reconstruction SEI message as defined in ISO/IEC 23090-5, it can be marked as non-output sample as defined in ISO/IEC 14496-12.

7.4.4.2 Syntax

```
aligned(8) class V3CAtlasSample {
    // sample_size value is the size of the sample from the SampleSizeBox
    for (int i=0; i < sample_size; ) {
        sample_stream_nal_unit ss_nal_unit;  // as defined in ISO/IEC 23090-5
        i += ss_nal_unit.ssnu_nal_unit_size +
                V3CDecoderConfigurationRecord.unit_size_precision_bytes_minus1 + 1;
    }
}
```

7.4.4.3 Semantics

`ss_nal_unit` contains a single NAL unit in NAL unit sample stream format as defined in ISO/IEC 23090-5:2021, Annex D.

`ssnu_nal_unit_size` specifies the size, in bytes, of the sample stream NAL unit. The number of bits used to represent `ssnu_nal_unit_size` is equal to (`V3CDecoderConfigurationRecord.unit_size_precision_bytes_minus1` + 1) * 8.

7.4.4.4 V3C atlas track and V3C atlas tile track sync sample

A sync sample in a V3C atlas track or V3C atlas tile track is a sample that contains an intra random access point (IRAP) coded atlas access unit as defined in ISO/IEC 23090-5.

NOTE Atlas parameter sets and SEI messages can be repeated, if needed, at a sync sample to allow for random access.

7.4.5 V3C video component track

7.4.5.1 General

A V3C video component track carries 2D video encoded data of V3C video component. The storage of V3C video component tracks utilizes the existing capabilities of the ISO base media file format and derived specifications, for example, ISO/IEC 14496-15 defines mechanisms for carriage of ISO/IEC 14496-10 and ISO/IEC 23008-2 coded V3C video components.

V3C video component tracks shall be represented in the file as restricted video and shall use a generic restricted sample entry 'resv' with additional requirements:

— SchemeTypeBox is present in RestrictedSchemeInfoBox and scheme_type is set to 'vvvc'

— SchemeInformationBox is present in RestrictedSchemeInfoBox and contain a V3CUnitHeaderBox.

— In track header the track_in_movie flag is set to 0, to indicate that this track should not be presented alone.

NOTE There is no restriction on the video codec used for encoding V3C video components. Each V3C video component can be encoded using different video codecs.

7.4.5.2 Multimap video box

7.4.5.2.1 Definition

Box Type: 'mmvi'

Container: SchemeInformationBox

Mandatory: No

Quantity: Zero or One

The MultiMapVideoBox is used to indicate that decoded video frames contain two or more temporarily interleaved video frames that represent maps. This box shall not be present in the case of single-track encapsulation of V3C content.

When MultiMapVideoBox is present, it indicates that a temporal interleaving map packing arrangement is used. File parsers should implicitly set the composition time for map_count_minus1 + 1 consecutive samples to be equal to that of the first sample in the group of samples in the interleaved map packing arrangement.

When temporal interleaving map packing arrangement is used, each sync sample and each SAP sample with SAP_type in the range of 1 to 3, inclusive, indicated by the stream access point sample group, represent an iterative arrangement of samples representing map 0 to map_count_minus1, in composition time order, up to but excluding the next sync sample or SAP sample with SAP_type in the range of 1 to 3, inclusive, indicated by the stream access point sample group.

7.4.5.2.2 Syntax

```
aligned(8) class MultiMapVideoBox extends FullBox('mmvi', version = 0, 0){
    bit(4) reserved = 0;
    unsigned int(4) map_count_minus1;
}
```

7.4.5.2.3 Semantics

map_count_minus1 plus 1 indicates the number of maps present in the track as consecutive samples.

7.4.6 Track references

7.4.6.1 Referencing V3C atlas tracks

To link a V3C atlas track with sample entry `'v3cb'` to V3C atlas tracks with sample entries `'v3a1'` or `'v3ag'`, the track reference tool defined in ISO/IEC 14496-12 shall be used. The 4CCs of these track reference types shall be `'v3cs'`.

7.4.6.2 Referencing V3C atlas tile tracks

To link a V3C atlas track with sample entries `'v3c1'`, `'v3cg'`, `'v3a1'` or `'v3ag'` to V3C atlas tile tracks with sample entry `'v3t1'`, the track reference tool of ISO/IEC 14496-12 shall be used. The 4CCs of these track reference types shall be `'v3ct'`.

7.4.6.3 Referencing V3C video component tracks

To link a V3C atlas track with sample entries `'v3c1'`, `'v3cg'`, `’v3a1’`, or `’v3ag’`, or a V3C atlas tile track with sample entry `'v3t1'` to video component tracks, the track reference tool of ISO/IEC 14496-12 shall be used. One or more `TrackReferenceTypeBoxes` shall be added to a `TrackReferenceBox` within the `TrackBox` of the V3C atlas track or V3C atlas tile track, one for each component. The `TrackReferenceTypeBox` shall contain an array of track_IDs designating the video tracks which the V3C atlas track or V3C atlas tile track references. The `reference_type` of a `TrackReferenceTypeBox` identifies the type of the video component (i.e., occupancy, geometry, or attribute). The 4CCs of these track reference types shall be:

— `'v3vo'`: the referenced track(s) contain the video-coded occupancy V3C component

— `'v3vg'`: the referenced track(s) contain the video-coded geometry V3C component

— `'v3va'`: the referenced track(s) contain the video-coded attribute V3C component

The type of the V3C component carried by the referenced restricted video track, and signalled in the `RestrictedSchemeInfoBox` of the track, shall match the reference type of the track reference from the V3C atlas track or V3C atlas tile track.

When `'v3ct'` track reference is present in V3C atlas track, `'v3va'`, `'v3vo'`, `'v3vg'` references shall not be used.

7.4.7 Track alternatives and track grouping

7.4.7.1 V3C content alternatives

V3C content may be encoded as different versions in the file format. Different alternatives are indicated by the alternate tracks mechanism defined in ISO/IEC 14496-12 (i.e., `alternate_group` field of the `TrackHeaderBox`). V3C atlas tracks which have the same `alternate_group` value shall be different versions of the same V3C content.

Annex H describes examples of using alternate V3C content.

7.4.7.2 V3C video component alternatives

V3C video component tracks may have alternatives. In such a case, only one of the V3C video component tracks that belong to an alternative group shall be referenced by the V3C atlas track or V3C atlas tile track. V3C video component tracks which are alternatives of each other should use the alternate grouping mechanism, as defined in ISO/IEC 14496-12.

Annex H describes examples of using alternate V3C video components.

7.4.8 Playout groups

7.4.8.1 General

When only some combination of tracks from the alternate versions of the V3C components should be played together, then playout group mechanism shall be used. Playout groups are signalled using `PlayoutTrackGroupBox`.

NOTE A track can be part of more than one playout group.

7.4.8.2 Playout track group

7.4.8.2.1 Definition

Box Types:	`'potg'`
Container:	`TrackGroupBox`
Mandatory:	No
Quantity:	Zero or more

Playout track groups are defined using the track group type `PlayoutTrackGroupBox`, which extends `TrackGroupTypeBox` defined in ISO/IEC 14496-12. A `PlayoutTrackGroupBox` indicates that a track belongs to a set of tracks constituting a playout group. Only tracks within the same playout group can be played out together. For each playout group that a track is a member of, a corresponding instance of `PlayoutTrackGroupBox` with the unique `track_group_id` for that playout group shall be present in the `TrackGroupBox` of that track.

7.4.8.2.2 Syntax

```
aligned(8) class PlayoutTrackGroupBox extends TrackGroupTypeBox('potg') {
    // track_group_id is inherited from TrackGroupTypeBox
}
```

7.4.9 Summary

Table 10 provides a summary of the sample entry types for tracks carrying atlas data defined in this document.

Table 10 — Summary of sample entry types for V3C atlas tracks and atlas tile tracks

Sample entry type		'v3c1'	'v3cg'	'v3cb'	'v3a1'	'v3ag'	'v3t1'
V3C atlas sample entry	VPS	yes	yes	yes	no	no	N/A
	Atlas parameter sets	yes	yes	yes	yes[a]	yes[a]	N/A
	V3C unit header box	yes	yes	yes	yes	yes	N/A
Atlas parameter set sample group		may	may	no	may (partially[b])	may (partially[b])	N/A

[a] It only includes atlas component sub-bitstream parameter sets associated with same vuh_atlas_id indicated in V3C unit header box.

[b] Atlas component sub-bitstream parameter sets associated with same vuh_atlas_id indicated in V3C unit header box are included.

[c] When 'v3t1' track is present, samples of 'v3c1', 'v3cg', 'v3a1' and 'v3ag' track only contains non-ACL NAL units. If 'v3t1' track is not present, samples shall also contain ACL NAL units.

Table 10 *(continued)*

Sample entry type	'v3c1'	'v3cg'	'v3cb'	'v3a1'	'v3ag'	'v3t1'
Track references	'v3vo' 'v3vg' 'v3va' 'v3ct'	'v3vo' 'v3vg' 'v3va' 'v3ct'	'v3cs'	'v3vo' 'v3vg' 'v3va' 'v3ct'	'v3vo' 'v3vg' 'v3va' 'v3ct'	'v3vo' 'v3vg' 'v3va'
Sample	ACL + non-ACL[c]	ACL + non-ACL[c]	non-ACL	ACL + non-ACL[c]	ACL + non-ACL[c]	ACL

[a] It only includes atlas component sub-bitstream parameter sets associated with same vuh_atlas_id indicated in V3C unit header box.

[b] Atlas component sub-bitstream parameter sets associated with same vuh_atlas_id indicated in V3C unit header box are included.

[c] When 'v3t1' track is present, samples of 'v3c1', 'v3cg', 'v3a1' and 'v3ag' track only contains non-ACL NAL units. If 'v3t1' track is not present, samples shall also contain ACL NAL units.

8 Carriage of non-timed visual volumetric video-based coding data

8.1 General

Non-timed V3C content is stored in a file as ISO/IEC 14496-12 items. Three item types called V3C atlas item, V3C atlas tile item and V3C component item are defined for encapsulating non-timed V3C content.

A handler type 4CC code 'volv' stored in the HandlerBox of the MetaBox indicates the presence of V3C atlas items, V3C atlas tile items or V3C component items.

An overview of the structure for encapsulating non-timed V3C data in a single atlas with a single atlas tile is illustrated in Figure 3.

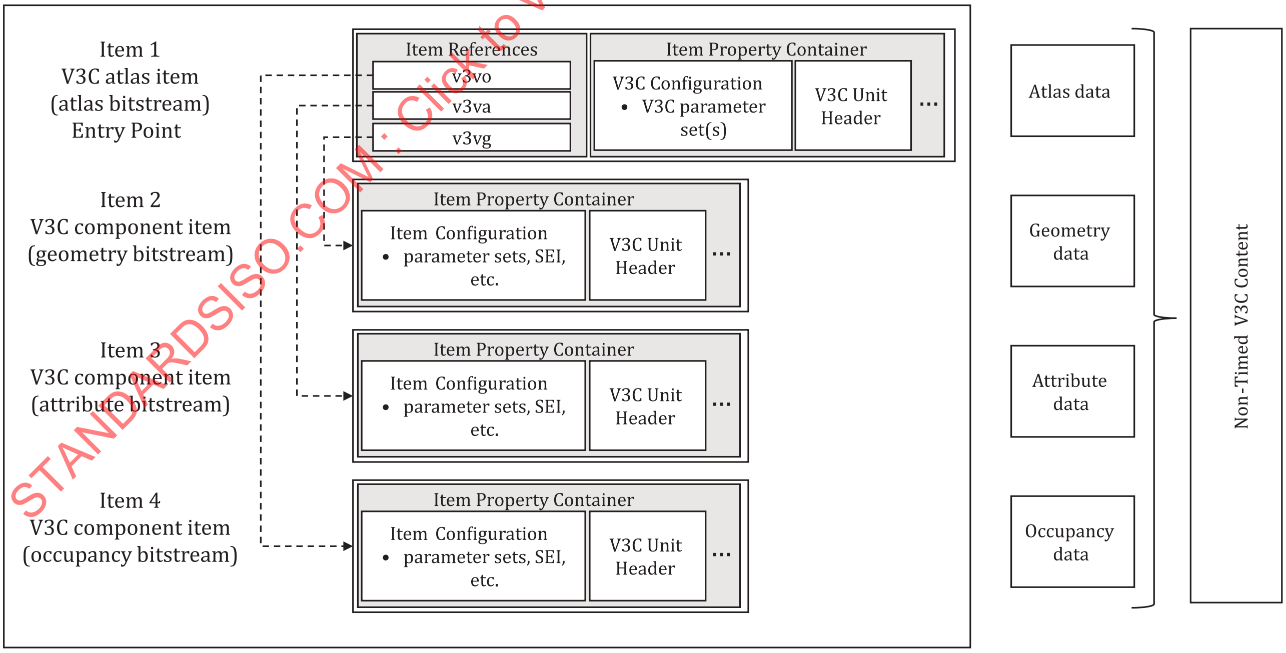

Figure 3 — Overview of structure for encapsulating non-timed V3C data in a single atlas with a single atlas tile

8.2 V3C atlas item

A V3C atlas item is an item which represents an independently decodable coded atlas access unit. Item type 4CC codes 'v3c1', 'v3cb', and 'v3a1'identify V3C atlas items. V3C atlas items store one or more ACL or non-ACL NAL units. Items of type 'v3c1' or 'v3cb' shall be associated with one V3CConfigurationProperty. All atlas items shall be associated with one V3CUnitHeaderProperty.

When the non-timed V3C content contains a single atlas, a V3C atlas item with item type 'v3c1' shall be used. If an item of type 'v3c1' is associated with V3C atlas tile items, the 'v3c1' item stores one or more non-ACL NAL units associated with the V3C atlas tile items. Otherwise, it stores one or more ACL or non-ACL NAL units.

When the non-timed V3C data contains multiple atlases, each atlas component bitstream shall be represented in separate V3C atlas item. One of those items shall be an item of type 'v3cb', while the other items shall be items of type 'v3a1'. In order to indicate the relationship between a 'v3cb' item to 'v3a1' items, an item reference type with 4CC codes 'v3cs' is used.

An 'v3cb' item shall not include any ACL NAL units. If an item of type 'v3a1' is associated with V3C atlas tile items, the 'v3a1' item contains one or more non-ACL NAL units associated with same vuh_atlas_id indicated in the V3CUnitHeaderProperty. Otherwise, it contains one or more ACL or non-ACL NAL units associated with same vuh_atlas_id indicated in the V3CUnitHeaderProperty.

If PrimaryItemBox exists, item_ID in this box shall be set to indicate a V3C atlas item of type 'v3c1' or 'v3cb'.

8.3 V3C atlas tile item

A V3C atlas tile item is an item of type 'v3t1' that contains one or more ACL NAL units which belong to the same atlas.

Each V3C atlas tile item shall be associated with one V3CAtlasTileConfigurationProperty. The V3CAtlasTileConfigurationProperty shall indicate the atlas tile IDs of tiles present in the V3C atlas tile item.

In order to indicate the relationship between a V3C atlas item to V3C atlas tile items, item references with the 4CC code 'v3ct’ are used. These item references are defined “from” a V3C atlas item “to” the related V3C atlas tile items.

8.4 V3C component item

A V3C component item is an item which represents visual V3C component data. V3C component items store V3C unit payload of V3C unit of type V3C_OVD, V3C_GVD, and V3C_AVD. A V3C component item shall store only one access unit of related video component data.

An item type 4CC code for a V3C component item shall be set depending on the codec used to encode corresponding video components. A V3C component item shall be associated with corresponding V3C unit header item property and codec specific configuration item property. V3C component items shall be marked as hidden items.

In order to indicate the association between a V3C atlas item and V3C component items or between V3C atlas tile item and V3C component items, three item reference types with 4CC codes 'v3vo', 'v3vg' and 'v3va' are defined. Item reference is defined “from” either a V3C atlas item or a V3C atlas tile item “to” the related V3C component items. The 4CC codes of item reference types shall be:

— 'v3vo': the referenced V3C component item(s) contain the occupancy data.

— 'v3vg': the referenced V3C component item(s) contain the geometry data.

— 'v3va': the referenced V3C component item(s) contain the attribute data.

8.5 V3C-related item properties

8.5.1 General

Two descriptive item properties are defined to carry the V3C parameter set information and V3C unit header information, respectively.:

8.5.2 V3C configuration item property

8.5.2.1 Definition

Box Types:	'v3cp'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for a V3C item of type 'v3c1' or 'v3cb'
Quantity (per item):	One or more for a V3C item of type 'v3c1' or 'v3cb'

V3C parameter set is stored as descriptive item property and shall be associated with the 'v3c1' or 'v3cb' V3C atlas items. In this version of this document, only one V3C parameter set is stored in V3C configuration item property.

The V3C configuration item property is an essential property. The corresponding essential flag in the ItemProperyAssociationBox shall be set to 1 for a 'v3cp' item property.

8.5.2.2 Syntax

```
aligned(8) class V3CConfigurationProperty
    extends ItemProperty('v3cp', version=0, flags) {
        unsigned int(16) v3c_parameter_set_length;
        // v3c_parameter_set() as defined in ISO/IEC 23090-5
        v3c_parameter_set v3c_parameter_set(v3c_parameter_set_length);
}
```

8.5.2.3 Semantics

v3c_parameter_set_length specifies the size in bytes of the v3c_parameter_set().

v3c_parameter_set contains a V3C parameter set as defined in ISO/IEC 23090-5.

8.5.3 V3C unit header item property

8.5.3.1 Definition

Box Types:	'vutp'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for a V3C atlas item of type 'v3c1' or 'v3a1', or 'v3cb' and for a V3C component item
Quantity (per item):	One

V3C unit header is stored as descriptive item properties and shall be associated with the V3C atlas items and the V3C component items.

The V3C unit header item property is an essential property. The corresponding essential flag in the ItemProperyAssociationBox shall be set to 1 for a 'vutp' item property.

8.5.3.2 Syntax

```
aligned(8) class V3CUnitHeaderProperty() extends ItemFullProperty('vutp', version=0, 0) {
    v3c_unit_header header();  // 4 bytes
}
```

8.5.3.3 Semantics

header contains a v3c_unit_header() syntax structure as defined in ISO/IEC 23090-5.

8.5.4 V3C atlas tile configuration item property

8.5.4.1 Definition

Box Types:	'v3tp'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for a V3C atlas tile item of type 'v3t1'
Quantity (per item):	One

V3CAtlasTileConfigurationProperty is stored as descriptive item properties and shall be associated with the V3C atlas tile items.

The V3C atlas tile configuration item property is an essential property. The corresponding essential flag in the ItemProperyAssociationBox shall be set to 1 for a 'v3tp' item property.

8.5.4.2 Syntax

```
aligned(8) class V3CAtlasTileConfigurationProperty () extends ItemFullProperty(‘v3tp’,
version=0, 0) {
    unsigned int(16) num_tiles;
    for(int i=0; i < num_tiles; i++) {
        unsigned int(16) tile_id;
    }
}
```

8.5.4.3 Semantics

num_tiles indicates the number of tiles contained in related V3C atlas tile item.

tile_id indicates the tile ID of the tile contained in related V3C atlas tile item. The value of tile_id is equal to value of afti_tile_id syntax element in atlas frame tile information, defined in ISO/IEC 23090-5.

8.5.5 Playout groups

8.5.5.1 General

When only some combination of items, from the alternate versions of the V3C component items should be played together, then playout group mechanism shall be used. Playout groups are signalled using PlayoutEntityToGroupBox.

NOTE An item can be part of more than one playout group.

8.5.5.2 Playout entity to group box

8.5.5.2.1 Definition

Box Types: `'eply'`

Container: `GroupsListBox` (`'grpl'`)

Mandatory: No

Quantity: Zero or more

An `EntityToGroupBox` with `grouping_type` equal to `'eply'` specifies items that shall be played out together. An alternate group of entities consists of items that are mapped to the same entity group of type `'altr'`. Only one item within an alternate group should be played or streamed at any one time. A `PlayoutEntityToGroupBox` is used to group non-timed (item) V3C data in the same group.

8.5.5.2.2 Syntax

```
aligned(8) class PlayoutEntityToGroupBox extends EntityToGroupBox('eply') {}
```

9 Partial access of volumetric visual data

9.1 General

Signalling related to partial access functionality is defined in this this clause. Partial access relates to making available only a subset of V3C content. Alternative methods for signalling partial access related information using Volumetric annotation SEI message family, as defined in ISO/IEC 23090-5:2021, Annex F, are discussed in Annex E.

9.2 Common data structures

9.2.1 3D vector

9.2.1.1 Syntax

```
aligned(8) class Vector3(int precision = 32) {
    int reserved_bits = 8 - (precision*3) % 8;
    if (reserved_bits != 8) {
        bit(reserved_bits) reserved = 0;
    }
    unsigned int(precision) x;
    unsigned int(precision) y;
    unsigned int(precision) z;
}
```

9.2.1.2 Semantics

`x`, `y`, and `z` specify the x, y, and z coordinate values, respectively, of a 3D point in the Cartesian coordinate system.

9.2.2 Spatial region bounding box

This data structure defines a bounding box for a 3D region in Cartesian space using an anchor point and a scale along the three axes.

9.2.2.1 Syntax

```
aligned(8) class V3CBoundingBox (anchor_included, scale_included) {
    if (anchor_included) { // anchor is not 0,0,0
```

```
        unsigned int(8) bb_pos_precision;
        Vector3 bb_position(bb_pos_precision);
    }
    if (scale_included) {
        unsigned int(8) bb_scale_precision;
        Vector3 bb_scale(bb_scale_precision);
    }
}
```

9.2.2.2 Semantics

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

`bb_position.x`, `bb_position.y`, and `bb_position.z` indicate the position of the 3D bounding box in the Cartesian coordinates along the x, y, and z axes, respectively.

`bb_scale_precision` indicates the precision of `bb_scale` in number of bits.

`bb_scale.x`, `bb_scale.y`, and `bb_scale.z` indicate the extension of the 3D bounding box of the entire volumetric media in the Cartesian coordinates along the x, y, and z axes, respectively, relative to the origin `(0,0,0)` if `anchor_included` is set to 0 and relative to `bb_position` if `anchor_included` is set to 1.

9.2.3 Tile mapping

This data structure provides the mapping between a spatial region and the set of atlas tiles that contain patches associated with that spatial region. If spatial scalability is enabled, an instance of this data structure provides a separate tile mapping for each level-of-detail.

9.2.3.1 Syntax

```
aligned(8) class TileMapping (spatial_scalability_enabled) {
    if (spatial_scalability_enabled) {
        unsigned int(8) num_lod;
        for (int i=0; i < num_lod; i++) {
            // LoD to tiles mapping
            unsigned int(8) lod_index;
            unsigned int(8) lod_num_tiles;
            for (int j=0; j < lod_num_tiles; j++) {
                bit(2) reserved = 0;
                unsigned int(6) atlas_id;
                unsigned int(16) lod_tile_id;
            }
        }
    } else {
        // spatial regions to tiles mapping
        unsigned int(8) num_tiles;
        for (j=0; j < num_tiles; j++) {
            bit(2) reserved = 0;
            unsigned int(6) atlas_id;
            unsigned int(16) tile_id;
        }
    }
}
```

9.2.3.2 Semantics

`num_lod` indicates the number of LoDs available for an associated 3D spatial region.

`lod_index` indicates the ordering on the LoDs for an associated 3D spatial region. A set of atlas tiles with a certain `lod_index` should be selected with the sets of atlas tiles associated with all lower `lod_index` values. An LoD tile set associated with a lower `lod_index` value should be processed first.

`lod_num_tiles` indicates the number of atlas tiles associated with an LoD of a spatial region.

atlas_id indicates the atlas ID associated with the lod_tile_id or tile_id of a spatial region.

lod_tile_id is an ID for a V3C atlas tile associated with an LoD of the spatial region. The value of lod_tile_id is equal to value of afti_tile_id syntax element in atlas frame tile information, defined in ISO/IEC 23090-5.

num_tiles indicates the number of atlas tiles associated with a spatial region.

tile_id is an ID for a V3C atlas tile that is associated with the spatial region. The value of tile_id is equal to value of afti_tile_id syntax element in atlas frame tile information, defined in ISO/IEC 23090-5.

9.2.4 Object collection

9.2.4.1 Syntax

```
aligned(8) class V3CObject () {
    unsigned int(1) obj_cancel_flag;
    unsigned int(1) obj_priority_present_flag;
    unsigned int(1) obj_dependencies_present_flag;
    unsigned int(1) obj_bounding_box_present_flag;
    unsigned int(1) obj_spatial_scalability_enabled_flag;
    bit(3) reserved = 0;
    unsigned int(8) obj_idx_length;
    unsigned int(obj_idx_length * 8) soi_object_idx;

    if (!obj_cancel_flag) {

        if (obj_priority_present_flag) {
            bit(4) reserved = 0;
            unsigned int(4) obj_priority_value;
        }

        if (obj_dependencies_present_flag) {
            unsigned int(8) obj_num_dependencies;
            unsigned int(8) obj_dep_idx_length;
            for (i=0; i < obj_num_dependencies; i++) {
                unsigned int(obj_dep_idx_length * 8) soi_dep_object_idx[i];
            }
        }

        if (obj_bounding_box_present_flag) {
            V3CBoundingBox obj_bounding_box(1, 1);
        }

        if (obj_spatial_scalability_enabled_flag) {
            unsigned int(8) obj_num_lod;
            for (int j=0; j < obj_num_lod; j++) {
                unsigned int(8) obj_lod_index;
                unsigned int(8) obj_num_tiles;
                for (int k=0; k < obj_num_tiles; k++) {
                    bit(2) reserved = 0;
                    unsigned int(6) obj_atlas_id;
                    unsigned int(16) obj_tile_id;
                }
            }
        } else {
            unsigned int(8) obj_num_tiles;
            for (int j=0; j < obj_num_tiles; j++) {
                bit(2) reserved = 0;
                unsigned int(6) obj_atlas_id;
                unsigned int(16) obj_tile_id;
            }
        }
    }
}
aligned(8) class V3CObjectCollection {
    unsigned int(32) num_objects;
    for (int i=1; i<=num_objects; i++) {
        V3CObject obj;
    }
}
```

9.2.4.2 Semantics

obj_cancel_flag indicates that the object is cancelled.

obj_priority_present_flag indicates whether priority information is available for an object. Value 0 indicates that no object priority information is given. Value 1 indicates that object priority information is present.

obj_dependencies_present_flag indicates whether object dependency information is available for an object. Value 0 indicates that the object does not depend on other objects. Value 1 indicates that the object depends on one or more objects within the V3C content.

obj_bounding_box_present_flag indicates whether 3D bounding boxing information is available for an object. Value 0 indicates that no bounding box information is given. Value 1 indicates that that 3D bounding box information for the object is present.

obj_spatial_scalability_enabled_flag indicates whether the LoD-based scalability is supported by the carried V3C content. Value 1 indicates that LoD-based scalability is supported. Value 0 indicates that the carried V3C content does not include multiple LoDs.

obj_idx_length is the length of the object index, in number of bytes, for an object in the signalled object list.

soi_object_idx indicates the value of an object index, as defined by the object scene information SEI message.

obj_priority_value indicates the priority value of an object in the object update list of the sample. The lower the priority value, the higher the priority.

obj_num_dependencies is the number of objects that an object in the object update list of the sample depends on.

obj_dep_idx_length is the length, in number of bytes, of the index of the dependent object.

soi_dep_object_idx[i] is the index of the i-th object that an object in the object update list of the sample depends on.

obj_bounding_box indicates the bounding box of the object.

obj_num_lod indicates the number of LoDs available for the object.

obj_lod_index indicates the ordering on the LoDs available for the object. A set of atlas tiles with a certain obj_lod_index should be selected with the sets of atlas tiles associated with all lower obj_lod_index values. An LoD tile set associated with a lower obj_lod_index value is processed first.

obj_num_tiles indicates the number of tiles associated with the object.

obj_atlas_id indicates the atlas ID associated with the obj_tile_id.

obj_tile_id indicates the tile ID associated with the object.

num_objects indicates the number of objects in the object collection.

obj contains object-related partial access information.

9.3 Spatial region information structure

9.3.1 Definition

`V3CSpatialRegion` provides information of a spatial region or object based subdivision of the volumetric media and their mapping information to atlas tiles. It may include the x, y, z offset of the spatial region and the width, height, and depth of the region in 3D space, 3D bounding box information of the volumetric media or object based subdivision details.

9.3.2 Syntax

```
aligned(8) class V3CSpatialRegion {
    unsigned int(32) size;
    unsigned int(16) region_id;
    unsigned int(1) bb_anchor_present_flag;
    unsigned int(1) bb_scale_present_flag;
    unsigned int(1) tile_mapping_present_flag;
    unsigned_int(1) tm_spatial_scalability_flag;
    unsigned_int(1) object_collection_present_flag;
    bit(3) reserved = 0;
    if (bb_anchor_present_flag || bb_scale_present_flag) {
        V3CBoundingBox bounding_box(bb_anchor_present_flag, bb_scale_present_flag);
    }
    if (tile_mapping_present_flag) {
        TileMapping tile_map(tm_spatial_scalability_flag);
    }
    if (object_collection_present_flag) {
        ObjectCollection object_collection;
    }
}
```

9.3.3 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 spatial region.

`bb_anchor_present_flag` indicates the presence of the bounding box with an anchor field.

`bb_scale_present_flag` indicates the presence of the bounding box with the scale field.

`tile_mapping_present_flag` indicates the presence of tile mapping.

`tm_spatial_scalability_flag` indicates whether of the signalled tile mapping has multiple levels-of-detail. This flag shall be set to 0 if `tile_mapping_present_flag` is set to 0.

`object_collection_present_flag` indicates the presence of an object collection that lists objects in the spatial region.

9.4 V3C tile video component track grouping

9.4.1 Definition

A V3C tile video component track group is a track group that groups all the tracks carrying V3C video component information associated with a set of atlas tiles. This track group is used when an atlas contains more than one tile and all atlas component tiles for that atlas are carried in the corresponding V3C atlas track. An example of using V3C tile video component track grouping for one atlas track containing tiles 1 and 2 is shown in Figure 4.

The presence of a `TrackGroupTypeBox` with `track_group_type` equal to `'vtcg'` in a track indicates that this track belongs to a group of V3C video component tracks that correspond to a V3C tile video component group.

Tracks belonging to the same V3C tile video component group have the same value of track_group_id for track_group_type 'vtcg', and the track_group_id of tracks from one V3C tile video component track group differs from the track_group_id of tracks from any other V3C tile video component track group.

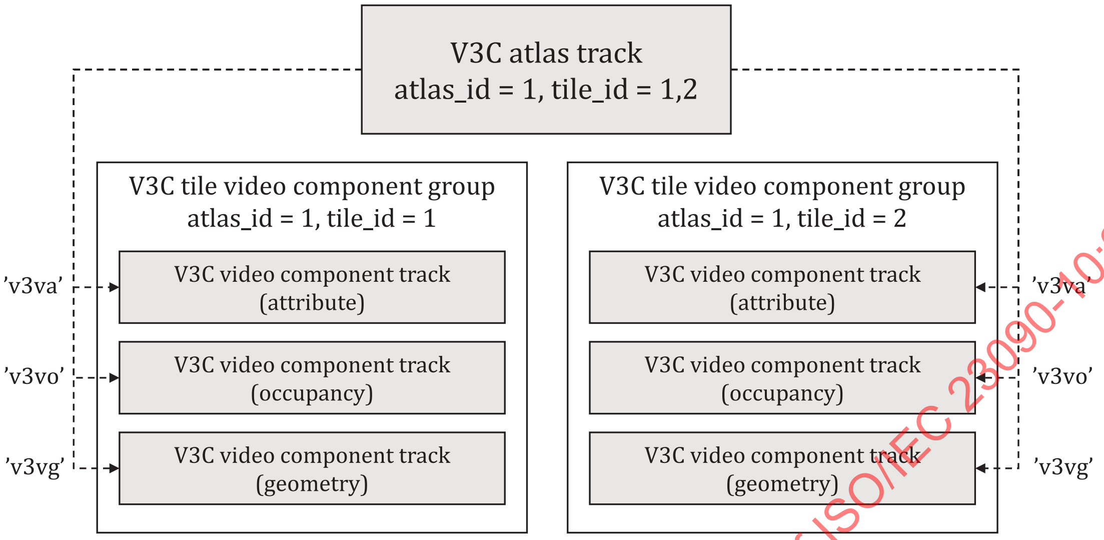
V3C atlas track
atlas_id = 1, tile_id = 1,2
V3C tile video component group
atlas_id = 1, tile_id = 1
V3C tile video component group
atlas_id = 1, tile_id = 2
'v3va'
'v3vo'
'v3vg'
V3C video component track (attribute)
V3C video component track (occupancy)
V3C video component track (geometry)
V3C video component track (attribute)
V3C video component track (occupancy)
V3C video component track (geometry)
'v3va'
'v3vo'
'v3vg'

Figure 4 — Example of V3C tile video component track grouping

9.4.2 Syntax

```
aligned(8) class V3CTileVideoComponentGroupBox extends TrackGroupTypeBox('vtcg') {
    unsigned int(8) num_tiles;
    for (int i=0; i < num_tiles; i++) {
        unsigned int(6) atlas_id;
        bit(2) reserved = 0;
        unsigned int(16) tile_id;
    }
}
```

9.4.3 Semantics

num_tiles is the number of V3C atlas tiles associated with the track group.

atlas_id indicates the atlas ID associated with the tile_id.

tile_id is an id for a V3C atlas tile. The value of tile_id is equal to value of afti_tile_id syntax element in atlas frame tile information, defined in ISO/IEC 23090-5.

9.5 Volumetric media bounding box

9.5.1 Definition

Box Types:	'vpbb'
Container:	V3CAtlasSampleEntry ('v3c1', 'v3cg', 'v3e1', or 'v3eg')
Mandatory:	No
Quantity:	Zero or one

A V3CBoundsBox shall be present in the sample entry of a V3C atlas track when V3CSpatialRegionsBox is present.

When the atlas sequence parameter set of the atlas carried by the V3C atlas track includes VUI parameters, the values of bb_position.x, bb_position.y, and bb_position.z shall be identical to vui_display_box_size[0], vui_display_box_size [1] and vui_display_box_size [2], respectively, in the vui_parameters() syntax structure defined in ISO/IEC 23090-5:2021, Annex G.

9.5.2 Syntax

```
aligned(8) class V3CBoundsBox extends FullBox(‘vpbb’, version=0, 0) {
    V3CBoundingBox bounding_box(0, 1);
}
```

9.6 Static spatial region collection box

9.6.1 Definition

Box Types: 'v3sc'

Container: V3CAtlasSampleEntry (‘v3c1’, 'v3cg', 'v3a1', 'v3ag') or DynamicVolumetricMetadataSampleEntry ('dyvm')

Mandatory: No

Quantity: Zero or one

Static 3D spatial regions in V3C content and their respective associated tracks shall be signalled in a V3CSpatialRegionCollectionBox. 3D spatial regions may overlap each other. Example of partial access for overlapping spatial subdivisions is explained in Annex G.

If scene object information do not change over time, V3CSpatialRegionCollectionBox may be present in the V3CAtlasSampleEntry and include static mapping between partial access regions, i.e., spatial regions or objects, and associated tiles. When a V3CSpatialRegionCollectionBox is present, the information signalled in the box shall have precedence over any information signalled in V3C Volumetric Annotation SEI messages in the bitstream.

For all spatial regions in signalled in a V3CSpatialRegionCollectionBox, the following restrictions shall apply:

— region.bounding_box shall have the bb_scale_present_flag set.

— region.object_collection shall not include any objects with obj_cancel_flag set.

9.6.2 Syntax

```
aligned(8) class V3CSpatialRegionCollectionBox extends FullBox('v3sc', version=0, 0) {
    unsigned int(16) num_regions;
    for(int i=1; i<=num_regions; i++) {
        V3CSpatialRegion region;
    }
}
```

9.6.3 Semantics

num_regions indicates the number of 3D spatial regions in the V3C media.

region describes attributes related to 3D spatial region belonging to the V3C media.

9.7 Dynamic spatial region information

9.7.1 General

If the V3C atlas track has an associated timed-metadata track with a sample entry type 'dyvm', 3D spatial regions defined for the volumetric media stream carried by the V3C atlas track are considered as dynamic regions (i.e., the spatial region information may dynamically change over time) and V3CSpatialRegionCollectionBox shall be present in the sample entry of the timed-metadata track, and not in the sample entry of the V3C track. If objects are added or removed in the middle of the bitstream by scene object information SEI messages, then at least one V3CVolumetricMetadataSample shall be present.

The associated timed-metadata track shall contain a 'cdsc' track reference to the V3C atlas track carrying the atlas component bitstream.

9.7.2 Sample entry

```
aligned(8) class DynamicVolumetricMetadataSampleEntry extends MetaDataSampleEntry('dyvm')
{
    V3CSpatialRegionCollectionBox();
}
```

9.7.3 Sample format

9.7.3.1 General

Each sample is either:

a) Exactly one empty sample with the sample_size=0 (representing a period of non-zero duration in which there is no updates to spatial region information); or

b) One or more V3CSpatialRegion elements that share the same start time and end time.

9.7.3.2 Syntax

```
aligned(8) class V3CVolumetricMetadataSample() {
     unsigned int(16) num_regions;
     for(int i=1; i<=num_regions; i++) {
         V3CSpatialRegion region;
     }
}
```

9.7.3.3 Semantics

num_regions indicates the number of 3D spatial regions signalled in the sample. This may not necessarily be equal to the total number of available regions. Only spatial regions whose position and/or dimensions are being updated are present in the sample.

region describes attributes related to 3D spatial region signalled in the sample.

9.7.4 Sync samples

All V3CSpatialRegion elements signalled in sync samples of the dynamic spatial region information timed-metadata track shall satisfy all the following conditions:

— V3CBoundingBox shall have both the bb_anchor_present_flag and the bb_scale_present_flag set, i.e., each bounding box shall include both the position and scale components.

— ObjectCollection shall not include any objects with obj_cancel_flag set.

9.8 Storage of atlas tiles using NALUMapEntry

NALUMapEntry specified in ISO/IEC 14496-15 shall be present in the V3C atlas track when V3CSpatialRegionCollectionBox is present and no V3C atlas tile tracks are associated with the V3C atlas track. This document uses NALUMapEntry as specified in ISO/IEC 14496-15 with the following additional requirements:

— The NALUMapEntry, when present, is used to assign an identifier, called groupID, to each atlas NAL units.

— For ACL NAL units, groupID shall be equal to ath_id + 1, where ath_id is specified in ISO/IEC 23090-5. For non-ACL NAL units, groupID shall be equal to 0, which implies that the atlas NAL unit is required for decoding any atlas tile in the same atlas frame to which the NAL unit belongs.

— The NALUMapEntry, when present, may or may not be linked to a sample group description setting the grouping_type_parameter of the SampleToGroupBox of type 'nalm'. Consequently, A SampleToGroupBox of type 'nalm' may or may not use version 0 of the box.

10 Viewport information

10.1 General

This clause specifies signalling of viewport information and associated intrinsic and extrinsic camera information for V3C content in container files. Viewport information may be conveyed through the defined viewport information structure that includes an extrinsic camera information structure that specifies the viewport's position and the rotation. In addition, the viewport information structure includes an intrinsic camera information structure. The receiver may render the V3C content based on the signalled viewport information at any point in time.

10.2 Structures

10.2.1 Extrinsic camera information

10.2.1.1 Syntax

```
aligned(8) class ExtCameraInfo () {
    unsigned int(8)[4] cam_pos_x;
    unsigned int(8)[4] cam_pos_y;
    unsigned int(8)[4] cam_pos_z;
    signed int(32) cam_quat_x;
    signed int(32) cam_quat_y;
    signed int(32) cam_quat_z;
}
```

10.2.1.2 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 metres 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:

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

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})).

NOTE As aligned ISO/IEC 23090-5, 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.

10.2.2 Intrinsic camera information

10.2.2.1 Syntax

```
aligned(8) class IntCameraInfo () {
    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;
    }
    if (camera_type == 1) {
        signed int(32) perspective_horizontal_fov;
        unsigned int(8)[4] perspective_aspect_ratio;
    }
    if (camera_type == 2) {
        unsigned int(8)[4] ortho_aspect_ratio;
        unsigned int(8)[4] ortho_horizontal_size;
    }
    unsigned int(8)[4] clipping_near_plane;
    unsigned int(8)[4] clipping_far_plane;
}
```

10.2.2.2 Semantics

`camera_id` is an identifier number that is used to identify 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 metres. 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 metres. 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.

10.2.3 Viewport information

10.2.3.1 Syntax

```
aligned(8) class ViewportInfo (ext_camera_flag, 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;
        }
        ExtCameraInfo extCamInfo();
    }
    if (int_camera_flag == 1) {
        IntCameraInfo intCamInfo();
    }
}
```

10.2.3.2 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. Value 1 indicates that the signalled viewport position corresponds to the centre of the viewport. Value 0 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. Value 1 indicates that the signalled viewport information corresponds to the left stereo position of the viewport. Value 0 indicates that the viewport information signalled correspond to the right stereo positions of the viewport.

`extCamInfo` is an instance of `ExtCameraInfo` defining the extrinsic camera parameters for the viewport.

`intCamInfo` is an instance of `IntCameraInfo` defining the intrinsic camera parameters for the viewport.

10.3 Viewport information timed-metadata track

10.3.1 General

This subclause describes the use of the timed metadata track to signal viewport information in V3C carriage format, composed of intrinsic and extrinsic camera parameters, including viewport position and rotation information as well as viewport camera parameters. To signal viewport information for a V3C content, the viewport information timed metadata track only references to related V3C atlas tracks, not directly to V3C video component tracks.

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 V3C atlas tracks with a '`cdsc`' track reference, it describes each V3C atlas track individually.

NOTE The receiver can render the V3C content based on the signalled viewport information at any point in time, or it can recommend user to consume content based on the viewport information via other means. Viewport information track can also reference other non-V3C related video tracks stored in the same file. When viewport information track references both a V3C content and a video track, the referenced video track may contain the 2D rendering of the referenced V3C content.

10.3.2 Viewport information sample entry

10.3.2.1 Definition

Sample Entry Type: '`6vpt`'

Container: Sample Description Box ('stsd')

Mandatory: No

Quantity: 0 or 1

The sample entry for viewport information associated to the V3C carriage format is defined by the ViewportInfoSampleEntry.

The viewport information sample entry shall contain a ViewportInfoConfigurationBox, describing the viewport type and, if applicable to all the samples of the track, the intrinsic and/or extrinsic camera parameters.

The codecs parameter value for this track as defined in RFC 6381 shall be set to '6vpt'.

10.3.2.2 Syntax

```
aligned(8) class ViewportInfoConfigurationBox
extends FullBox('6vpC', version = 0, 0) {
    unsigned int(7) viewport_type;
    bit(1) reserved = 0;
    string viewport_description;
    unsigned int(1) dynamic_int_camera_flag;
    unsigned int(1) dynamic_ext_camera_flag;
    bit(6) reserved = 0;
    if (dynamic_int_camera_flag == 0) {
        IntCameraInfo();
    }
    if (dynamic_ext_camera_flag == 0) {
        ExtCameraInfo();
    }
}

aligned(8) class ViewportInfoSampleEntry() extends MetadataSampleEntry ('6vpt') {
    ViewportInfoConfigurationBox();
}
```

10.3.2.3 Semantics

viewport_type specifies the type of the viewport as listed in Table 11 for the i-th viewport parameter set for all samples referring to this sample entry.

Table 11 — 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	A recommended viewport suggested for an associated spatial region
5..239	Reserved
240..255	Unspecified (for use by applications or external specifications)

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

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

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

For viewport_type equal to 3, the timed metadata indicates the recommended initial viewport information, composed of the initial viewport positions and rotations, when playing the associated V3C 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, this metadata track shall be present and associated with the media track. In the absence of this type of metadata, 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.

10.3.3 Viewport information sample format

10.3.3.1 General

Each viewport sample carries an array of viewports of the type defined in the associated sample entry. The parameters of each viewport include the extrinsic and intrinsic camera information parameters described by IntCameraInfo and ExtCameraInfo. While extrinsic camera information parameters described by ExtCameraInfo are expected to be in each sample, intrinsic camera parameters described by IntCameraInfo are only present in a sample if the intrinsic camera parameters signalled in the earlier samples are no longer applicable.

Previously defined extrinsic or intrinsic camera parameters for a certain viewport from an earlier sample shall persist if not modified.

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 V3C atlas 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.

10.3.3.2 Syntax

```
aligned(8) class ViewportInfoSample() {
    unsigned int(8) num_viewports;
    for (int i=1; i <= num_viewports; i++){
        unsigned int(7) viewport_id[i];
        unsigned int(1) viewport_cancel_flag[i];
        if (viewport_cancel_flag[i] == 0) {
            unsigned int(1) camera_extrinsic_flag[i];
            unsigned int(1) camera_intrinsic_flag[i];
            bit(6) reserved = 0;
            ViewportInfo (camera_extrinsic_flag[i], camera_intrinsic_flag[i]);
        }
    }
}
```

10.3.3.3 Semantics

If the viewport information timed-metadata track is present, extrinsic camera parameters expressed by ExtCameraInfo() shall be present in either sample entry or sample level. It is prohibited that both of the following concurrently happen; dynamic_ext_camera_flag in ViewportInfoConfigurationBox equals to 0 and camera_extrinsic_flag[i] equals to 0 for all samples.

num_viewports indicates the number of viewports signalled in the sample.

viewport_id[i] is an identifier number that is used to identify the i-th viewport.

viewport_cancel_flag[i] equals 1 indicates that the viewport with the id viewport_id[i] is cancelled. Indicates that viewport information for the i-th viewport follows.

camera_intrinsic_flag[i] equal to 1 indicates that the intrinsic camera parameters are present in the i-th viewport in the current sample. It shall be equal to 0 if dynamic_int_camera_flag[i] equals to 0. Moreover, it shall be set as 0 when camera_extrinsic_flag[i] equals to 0.

camera_extrinsic_flag[i] equal to 1 indicates that the extrinsic camera parameters are present in the i-th viewport in the current sample. It shall be equal to 0 if dynamic_ext_camera_flag[i] equals to 0.

11 Encapsulation and signalling in MPEG-DASH

11.1 Single track mode

The single-track mode in DASH enables streaming of V3C ISOBMFF files where V3C content is stored using single-track encapsulation, as described in subclause 7.3. The single-track mode in DASH should be represented as one Adaptation Set with one or more Representations. Representations within the sole Adaptation Set shall use the same codec for the corresponding video components (e.g., the occupancy shall have the same codec in all Representations) but are not required to use the same codec for every video component (e.g., the occupancy could use one codec, e.g., ISO/IEC 14496-10, and geometry could be encoded by second codec, e.g., ISO/IEC 23008-2).

If a Representation consists of more than one Media Segment, an Initialization Media Segment shall be present. The Initialization Segment shall contain a V3CDecoderConfigurationRecord with the v3c_parameter_set syntax structure, as defined in (ISO/IEC 23090-5:2021, Clause 7) and a Component Codec Mapping SEI Message, as defined in (ISO/IEC FDIS 23090-5:2021, Annex E).

The first sample of a Media Segment shall have a Stream Access Point (SAP) of type 1 or 2. That means each sub-sample of the first sample shall have a Stream Access Point (SAP) of type 1 or 2.

The following restriction on some of the attributes shall be applied:

- The @mimeType parameter shall be ‘application/mp4’
- The @codecs parameter 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 the ‘codecs’ parameter of a MIME type is used, sub-parameters are 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 **AdaptationSet** element, i.e., the value shall not be different for different Representations in one Adaptation Set.
- The @width and @height shall not be signalled for any Representation.

Examples of DASH signalling are provided in Annex D.

11.2 Multi-track mode

11.2.1 General

In the multi-track mode, each V3C video component shall be represented in the DASH manifest (MPD) file as a separate Adaptation Set. These Adaptation Sets are referred to as Video Component Adaptation Sets. An additional Adaptation Set for atlas information serves as the Main Adaptation Set for the V3C content. If a geometry or attribute component has multiple maps, each map may be signalled using a separate **AdaptationSet** element.

The Main Adaptation Set shall have the @codecs attribute set to 'v3c1', 'v3cg' or 'v3cb' while the @codecs attribute for the Video Component Adaptation Sets, or Representations of these Adaptation Sets, if @codecs is not signalled for the **AdaptationSet** element, is set based on the respective codec used for encoding the component. The value of @codecs shall be set to 'resv.vvvc.XXXX', where XXXX corresponds to the four-character code (4CC) of the video codec from the original_format field in RestrictedSchemeInfoBox of Sample Entry (e.g., 'avc1' or 'hvc1').

The Main Adaptation Set shall contain a single Initialization Segment at the adaptation set level. The Initialization Segment shall contain all parameter sets needed to initialize the V3C decoder, including V3C parameter sets as well as other parameter sets for component sub-bitstreams.

Media Segments for the Representation of the Main Adaptation Set shall contain one or more track fragments of the V3C atlas track. Media Segments for the Representations of Video Component Adaptation Sets shall contain one or more track fragments of the corresponding video component track at the file format level. Examples of DASH signalling are explained in Annex D.

11.2.2 V3C preselections

V3C preselection may either be signalled in MPD using a **PreSelection** element within the **Period** element or a Preselection descriptor at the Adaptation Set level. A V3C **PreSelection** element is signalled , as defined in ISO/IEC 23009-1, with an id list for the @preselectionComponents attribute including the id of the Main Adaptation Set for the volumetric media followed by the ids of the Video Component Adaptation Sets. The @codecs attribute for the Preselection shall be set to 'v3c1', 'v3cg' or 'v3cb' indicating that the media represented by the Preselection is visual volumetric video-based coding media.

Figure 5 illustrates an exemplary DASH configuration for grouping V3C components belonging to a single V3C content within an MPEG-DASH MPD file.

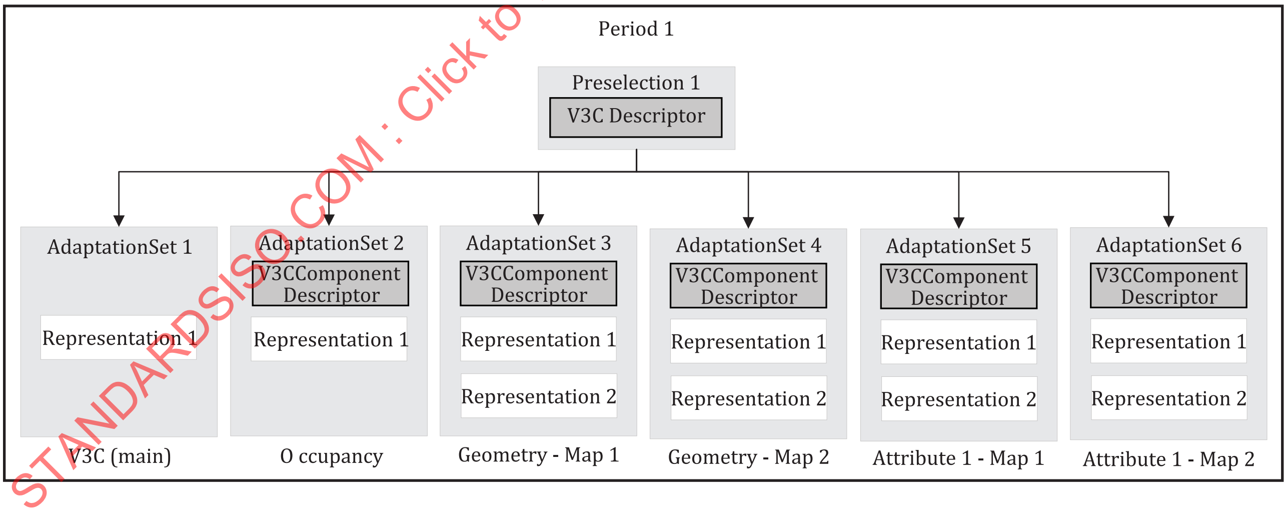

Figure 5 — Grouping V3C components in an MPD using Preselections

When multiple atlases are present in the V3C content, each atlas track shall be represented by a separate Adaptation Set considered as Atlas Adaptation Set. Atlas Adaptation Set shall have the @codecs attribute set to 'v3a1' or 'v3ag'. Representations of Atlas Adaptation Sets shall have a @dependencyId attribute set to the id of the Representation of the Main Adaptation Set. Each Atlas Adaptation Set shall be part of a separate Preselection that includes the Atlas Adaptation Set, as the main Adaptation Set of the Preselection, and Video Component Adaptation Sets for that atlas.

11.2.3 V3C atlas tile preselections

When V3C atlas tiles are carried in separate tracks these shall be represented by separate Adaptation Sets, considered as Atlas Tile Adaptation Sets, with the `@codecs` attribute for the Adaptation Sets set to `'v3t1'`. V3C video component tracks associated with an atlas tile track shall also be carried in separate Adaptation Sets with the `@codecs` attribute set to `'resv.vvvc.`XXXX', where XXXX corresponds to the four-character code (4CC) of the video codec (e.g., `'avc1'` or `'hvc1'`).

Atlas Tile Adaptation Sets and associated Video Component Adaptation Sets shall be part of a single Atlas Tile Preselection in the MPD with the Atlas Tile Adaptation Set being the main Adaptation Set for that Preselection (i.e., the id of the Atlas Tile Adaptation Set is the first id in the list of Adaptation Sets of the `@preselectionComponents` attribute in the **`Preselection`** element or the `@value` attribute of the Preselection descriptor). Representations of the Atlas Tile Adaptation Set of an Atlas Tile Preselection shall have an `@dependencyId` attribute set to the id of a Representation in the corresponding Atlas Adaptation Set.

The concatenation of the Initialization Segment of the Main Adaptation Set, Atlas Adaptation Set and associated Atlas Tile Adaptation Set, in order, followed by subsegments of a Representation of the Main Adaptation Set, Atlas Adaptation Set and the Adaptation Sets associated with the Atlas Tile Preselection, in any order, results in an ISOBMFF file conforming to subclause 7.4.

11.3 DASH MPD descriptors for V3C content

11.3.1 XML namespace and schema

A number of XML elements and attributes are defined in subclause 11.3 and its subclauses. These XML elements are defined in a separate namespace `"urn:mpeg:mpegI:v3c:2020"`. Namespace designator `"v3c:"` is used to refer to this namespace in this document. New XML elements and attributes are defined in XML schema documents in each subclause where a new MPD descriptor is specified. The namespace designator `"xs:"` shall correspond to namespace of XML Schema as defined in W3C Recommendation XML Schema Part 1. Some items in the "Data type" column of the tables of this clause 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.

11.3.2 V3C video component descriptor

To identify the type of Video Component Adaptation Set, a V3CVideoComponent descriptor shall be used. A V3CVideoComponent descriptor is an EssentialProperty descriptor with the `@schemeIdUri` set to "`urn:mpeg:mpegI:v3c:2020:videoComponent`".

At Adaptation Set level, one V3CVideoComponent descriptor shall be signalled for each V3C video component that is present in the Representations of the Video Component Adaptation Set.

The `@value` of the V3CVideoComponent descriptor shall not be present. The V3CVideoComponent descriptor shall include elements and attributes as specified in Table 12.

Table 12 — Elements and attributes for the V3CVideoComponent descriptor

Elements and attributes	Use	Data type	Description
videoComponent	0..N	v3c:VideoComponentType	An element whose attributes specify information for one of the V3C video components present in the Representation(s) of the Adaptation Set.
videoComponent@type	M	xs:string	Indicates the type of the V3C video component. Value 'geom' indicates a geometry component, 'occp' indicates an occupancy component, and 'attr' indicates an attribute component.
videoComponent@is_auxiliary	CM	xs:boolean	A flag indicating whether the V3C video component information represented by the Adaptation Set, within which the V3CVideoComponent descriptor is present, is for auxiliary video. Value true indicates that the video is an auxiliary video and contains RAW and/or EOM patches. Equal to false indicates video may contain RAW and/or EOM patches. If not present, the default value is false.
videoComponent@map_index	CM	xs:integer	Indicates the index of one of the maps of the component represented by the Adaptation Set within which the V3CVideoComponent descriptor is present. Shall only be present if the presentation contains multiple maps which are stored in different Adaptation Sets and **videoComponent**@type has the value 'geom' or 'attr'.
videoComponent@attribute_type	CM	xs:unsignedByte	Indicates the type of the attribute as defined in ISO/IEC 23090-5:2021 Table 3. Only values between 0 and 15, inclusive, are allowed. Shall be present only if the V3C video component is an attribute (i.e. **videoComponent**@type has the value 'attr').
videoComponent@attribute_index	CM	xs:unsignedByte	Indicates the index of the attribute. Shall be a value between 0 and 127, inclusive. Shall be present only if the component is a V3C attribute (i.e. **videoComponent**@type has the value 'attr').

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 @.

Table 12 *(continued)*

Elements and attributes	Use	Data type	Description
videoComponent@attribute_dim_partition_index	CM	xs:unsignedByte	Indicates the index of the dimension partition for the attribute carried in the Adaptation Set within which the V3CVideoComponent descriptor is present. Shall be present only if the component is a V3C attribute (i.e., **videoComponent**@type has the value 'attr'). If not present, the default value is 1.
videoComponent@atlas_id	OD	xs:integer	Indicates the atlas id of the component represented by the Adaptation Set within which the V3CVideoComponent descriptor is present. If not present, the default value is 0.
videoComponent@tile_ids	O	xs:UIntVectorType	Specifies atlas tiles related to data contained in the Adaptation Set by providing a white-space separated list of tile ID values. If not present, the Adaptation Set will contain all tiles of associated with the @atlas_id.

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 types for various elements and attributes shall be as defined in the XML schema. An XML schema for the V3CVideoComponent descriptor shall be as shown below. The schema shall be represented in an XML schema that has namespace urn:mpeg:mpegI:v3c:2020 and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:v3c:2020"
    xmlns:v3c="urn:mpeg:mpegI:v3c:2020"
    elementFormDefault="qualified">
    <xs:element name="videoComponent" type="v3c:VideoComponentType"/>

    <xs:complexType name="VideoComponentType">
        <xs:attribute name="type" type="xs:string" use="required" />
        <xs:attribute name="is_auxiliary" type="xs:boolean" default=”false”/>
        <xs:attribute name="map_index" type="xs:integer" />
        <xs:attribute name="attribute_type" type="xs:unsignedByte" />
        <xs:attribute name="attribute_index" type="xs:unsignedByte" />
        <xs:attribute name="attribute_dim_partition_index" type="xs:unsignedByte" />
        <xs:attribute name="atlas_id" type="xs:integer" use="optional" default="0" />
        <xs:attribute name="tile_ids" type="UIntVectorType" use="optional" />
    </xs:complexType>
</xs:schema>
```

11.3.3 V3C descriptor

A SupplementalProperty element with a `@schemeIdUri` equal to `"urn:mpeg:mpegI:v3c:2020:v3c"` is referred to as a V3C descriptor. At most one V3C descriptor may be present in Main Adaptation Set, Atlas Adaptation Set, Atlas Tile Adaptation Set, V3C Preselection, or Atlas Tile Preselection.

The V3C descriptor shall contain the attributes defined in Table 13.

Table 13 — Attributes for the V3C descriptor

Attributes	Use	Data type	Description
`v3c:@vId`	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 Adaptation Sets in the MPD.
`v3c:@atlas_id`	CM	xs:integer	Indicates the atlas id for the volumetric media information in the track(s) carried by the Adaptation Set. This attribute shall be present if the volumetric media contains more than one atlas which are not alternatives.
`v3c:@tile_ids`	O	xs:UIntVectorType	If present, indicates the atlas tile IDs carried in the Atlas Tile Adaptation Set. The value of the `@tile_ids` attribute is a whitespace separated list of atlas tile IDs. For ISOBMFF, this shall include all tile IDs listed in the `V3CAtlasTileSampleEntry` of the V3C atlas 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 types for the attributes shall be as defined in the XML schema. An XML schema for the V3C descriptor shall be as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:v3c:2020` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
    targetNamespace="urn:mpeg:mpegI:v3c:2020"
    xmlns:v3c="urn:mpeg:mpegI:v3c:2020"
    elementFormDefault="qualified">

    <xs:attribute name="vId" type="xs:string" />
    <xs:attribute name="atlas_id" type="xs:integer" use="optional" />
    <xs:attribute name="tile_ids" type="UIntVectorType" use="optional" />

</xs:schema>
```

When more than one Atlas Preselection for the same volumetric media content are present in the MPD, a V3C descriptor with the `@atlas_id` attribute shall be signalled in each Atlas Preselection.

When more than one Atlas Tile Preselection for the same volumetric media content are present in the MPD, a V3C descriptor with the `@tile_ids` attribute, and the `@atlas_id` attribute if the content contain more than one atlas, shall be signalled in each Atlas Tile Preselection. If the Atlas Tile Preselections belong to different atlases, a V3C descriptor with the @atlas_id attribute shall also be signalled in each atlas Adaptation Set.

When Preselection elements for Atlas Preselections or Atlas Tile Preselections are present in an MPD, the V3C video component descriptor in each of Video Component Adaptation Sets that are part of those Preslections shall not contain the @atlas_id or @tile_ids attributes.

11.4 Supporting multiple versions of a V3C media

Multiple versions of the same volumetric media shall be signalled using separate Preselections. Preselections that represent alternative versions of the same V3C content shall contain a V3C descriptor with the same @vId value. At most one V3C descriptor shall be present at the preselection level. These Preselections are therefore alternatives to each other and the id of the main Adaptation Set of the Preselection, first id in the list of Adaptation Set ids for the @preselectionComponents, may be different (where each version of the visual volumetric media has a separate Main Adaptation Set signalled in the MPD file).

11.5 Switching codecs for V3C video components

If multiple versions of a V3C video component are encoded using a different video codec, each version shall be signalled in a separate Adaptation Set with the value of the @codecs attribute set according to the video codec used. Moreover, each of these Adaptation Sets shall contain a SupplementalProperty descriptor with @schemeIdURI set to urn:mpeg:dash:adaptation-set-switching:2016 and @value is a comma-separated list of Adaptation Set IDs corresponding to the other available versions to indicate that seamless switching between Representations across the Adaptation Sets of these versions is supported. Any additional rules for supporting switching across Adaptation Sets as defined by ISO/IEC 23009-1:2019, subclause 5.3.3.5 shall apply.

11.6 Signalling spatial regions for partial access

11.6.1 Static spatial regions

If the 3D spatial regions are static (i.e., the position and dimensions of each region do not change over the presentation time), the characteristics of the spatial regions and the mappings between those regions and V3C tiles shall be signalled using a V3C3DRegions descriptor. This descriptor is a SupplementalProperty element with a @schemeIdUri equal to "urn:mpeg:mpegI:v3c:2020:v3sr". A single V3C3DRegions descriptor shall be present at the Adaptation Set level or the Representation level in the Main Adaptation Set, or at the Preselection level for the V3C content.

The @value of the V3C3DRegions descriptor shall not be present. The V3C3DRegions descriptor shall include elements and attributes as specified in Table 14.

Table 14 — Elements and attributes for the V3C3DRegions descriptor

Elements and attributes	Use	Data type	Description
v3sr	0..1	v3c:spatialRegionMapType	Container element whose attributes and elements specify a mapping between a 3D spatial region and V3C tiles.
v3sr.spatialRegion	1..N	v3c:spatialRegionType	An element whose attributes define a 3D spatial region and provide a mapping between the defined region and a number of V3C tiles.
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 @.			

Table 14 *(continued)*

Elements and attributes	Use	Data type	Description
v3sr.spatialRegion@id	M	xs:unsignedShort	An identifier for the 3D spatial region. The value of this attribute shall match the value of the region_id field signalled for the corresponding region in the ISOBMFF container.
v3sr.spatialRegion@type	OD	xs:unsignedByte	An attribute whose value indicates the type of the spatial region. Value 0 indicates a cuboid region. Value 1 indicates a region corresponding to viewport. The remaining values are reserved. If not present, the default value is 0.
v3sr.spatialRegion.cuboid	CM	v3c:spatialRegionCuboidType	An element specifying a cuboid extending from the reference point of the spatial region. This element shall be present only when the **spatialRegion**@type attribute is set to 0.
v3sr.spatialRegion.cuboid@anchor	M	UIntVectorType	An attribute containing a triplet of values describing x-, y- and z-components of the anchor point for the bounding box defining the spatial region. The values in the array are in said order and the length of array is three.
v3sr.spatialRegion.cuboid@dimensions	M	UIntVectorType	An attribute containing a triplet of values describing the x-, y- and z-dimensions of the bounding box defining the spatial region. The values in the array are in said order and the length of array is three.
v3sr.spatialRegion.viewport	CM	v3c.spatialRegionViewportType	An element specifying a viewport corresponding to the spatial region. This element shall be present only when the **spatialRegion**@type attribute is set to 1.
v3sr.spatialRegion.viewport@rvIds	M	StringVectorType	A list of space separated identifiers corresponding to the values of the @viewport_id attribute for the RV descriptor indicating viewports corresponding to this region.
v3sr.spatialRegion@tile_ids	CM	xs:UIntVectorType	Indicates the atlas tile IDs mapped to this spatial region. The value of the @tile_ids attribute is a whitespace separated list of atlas tile IDs. This attribute shall be absent in the case of single-track encapsulation of the V3C content or when at least one **lod** element exists.
v3sr.spatialRegion.lod	0..N	v3c:lodType	Container element whose attributes provide a LoD information and corresponding V3C tiles for that LoD.
v3sr.spatialRegion.lod@idx	M	xs:unsignedByte	An identifier that indicates the ordering on the LoDs for an associated 3D spatial region. The value of this attribute shall match the value of the lod_index field signalled for the corresponding LoD in the ISOBMFF container.

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 @.

Table 14 *(continued)*

Elements and attributes	Use	Data type	Description
v3sr.spatialRegion. **lod**@tile_ids	M	xs:UIntVectorType	A list of whitespace separated identifiers corresponding to the values of the atlas tile IDs mapped to this LoD.
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 types for the various elements and attributes of the V3C3DRegions descriptor shall be as defined in the XML schema that has the namespace ’urn:mpeg:mpegI:v3c:2020’ and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
    targetNamespace="urn:mpeg:mpegI:v3c:2020"
    xmlns:v3c="urn:mpeg:mpegI:v3c:2020"
    elementFormDefault="qualified">

    <xs:element name="v3sr" type="v3c:spatialRegionMapType" />

    <xs:complexType name="spatialRegionMapType">
        <xs:element name="spatialRegion" type="v3c:spatialRegionType" minOccurs="1"/>
    </xs:complexType>

    <xs:complexType name="spatialRegionType">
        <xs:attribute name="id" type="xs:unsignedShort" use="required" />
        <xs:attribute name="type" type="xs:unsignedByte" use="optional" default="0" />
        <xs:attribute name="tile_ids" type=”xs:UIntVectorType” />

        <xs:element name="cuboid" type="v3c:spatialRegionCuboidType"
            minOccurs="0" maxOccurs="1"/>
        <xs:element name="viewport" type="v3c:spatialRegionViewportType"
            minOccurs="0" maxOccurs="1"/>
        <xs:element name="lod" type="v3c:lodType" />
    </xs:complexType>

    <xs:complexType name="spatialRegionCuboidType">
        <xs:attribute name="anchor" type="UIntVectorType" use="required"
            minLength="3" maxLength="3" />
        <xs:attribute name="dimensions" type="UIntVectorType" use="required"
            minLength="3" maxLength="3"/>
    </xs:complexType>

    <xs:complexType name="spatialRegionViewportType">
        <xs:attribute name="rvIds" type="StringVectorType" use="required" />
    </xs:complexType>

    <xs:complexType name=”lodType”>
```

```
        <xs:attribute name=”idx” type=”xs:unsignedByte” use=”required” />
        <xs:attribute name="tile_ids" type=”xs:UIntVectorType" use=”required” />

    </xs:complexType>
</xs:schema>
```

11.6.2 Dynamic spatial regions

When the 3D partitions are dynamic, a timed-metadata track for signalling the position and dimensions of each 3D region in the presentation timeline shall be used shall be carried in a separate Adaptation Set with a single Representation that is associated with a Representation in the Main Adaptation Set using the `@associationId` attribute, defined in ISO/IEC 23009-1, and an `@associationType` value that includes the 4CC ‘`cdsc`’.

11.7 Signalling recommended viewports

11.7.1 Static viewports

A **`SupplementalProperty`** with a `@schemeIdUri` equal to `“urn:mpeg:mpegI:v3c:2020:rv”` is defined for the Recommended Viewport (RV) descriptor in order to signal the recommended viewports of the V3C content. This descriptor may be used by content providers to signal a set of viewport position and rotation parameters recommended for rendering the V3C 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 V3C 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	v3c:ViewportInfoType	Container element whose sub-elements and attributes provide information about the viewport.
`ViewportInfo``@vp_pos`	M	v3c:FloatVectorType	Indicates the x-, y- and z-coordinates of the position of the viewport in metres in the global reference coordinate system. The values in the array are in said order and the length of array is three. If the viewport is dynamic, this attribute specifies the initial viewport position. Otherwise, this attribute specifies a static viewport’s position.

Legend:

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 `@`.

Table 15 *(continued)*

Elements and attributes	Use	Data type	Description
ViewportInfo@vp_quat	M	v3c: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 viewport is dynamic, this attribute specifies the initial viewport rotation. Otherwise, this attribute specifies a static viewport's 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 position of the viewport.
ViewportInfo@initialViewport	O	xs:boolean	If equal to "true", this attribute specifies that this viewport is the initial viewport that should be used out of all the recommended viewports in the current Period. If equal to "false", this attribute specifies that this viewport is not the initial viewport in the current Period. In a Period at most one viewport shall have **ViewpointInfo**@initialViewport equal to "true".
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 11.

Legend:

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 types for the various elements and attributes of the RV descriptor shall be as defined in the XML schema that has the namespace 'urn:mpeg:mpegI:v3c:2020' and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
    targetNamespace="urn:mpeg:mpegI:v3c:2020"
    xmlns:omaf="urn:mpeg:mpegI:v3c:2020"
    elementFormDefault="qualified">

    <xs:attribute name="viewport_id" type="xs:integer" use="optional" />
    <xs:element name="ViewportInfo" type="v3c:ViewportInfoType"/>

    <xs:complexType name="ViewportInfoType">
        <xs:attribute name="vp_pos" type="FloatVectorType" use="required"
            minLength="3" maxLength="3"/>
        <xs:attribute name="vp_quat" type="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="initialViewport" 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>

</xs:schema>
```

11.7.2 Dynamic viewports

For dynamic viewports where the viewport position and/or rotation change over time, a timed-metadata track with a sample entry of type '6vpt' as described in subclause 10.3 shall be used for signalling changes to the viewport position and rotation at different times in the presentation timeline. This timed-metadata track shall be carried in a separate Adaptation Set with a single Representation that is associated with a Representation in the Main Adaptation Set using the @associationId attribute, defined in ISO/IEC 23009-1, and an @associationType value that includes the 4CC 'cdsc'.

12 Encapsulation and signalling MMT

12.1 Introduction

For the carriage of V3C content using MMT, two types of signaling information are provided. One is the signaling information provided per MMT Package or MMT Assets by using the general MMT signaling messages, tables, and descriptors which are agnostic to specific content types or media types. Such information is used by the MMT receiving entity to understand the structure of MMT Packages or the properties of MMT Assets. Another is signaling information specific to V3C content which provides the structure of the V3C content and properties specific to the V3C content or its components. As depicted in Figure 6, the general MMT signaling information is processed by the MMT Receiving Entity and the V3C content specific information is processed by the V3C content specific information processing entity. In some cases, feedback may be provided from the V3C content specific information processing entity to the MMT Receiving Entity according to the V3C content specific signaling information.

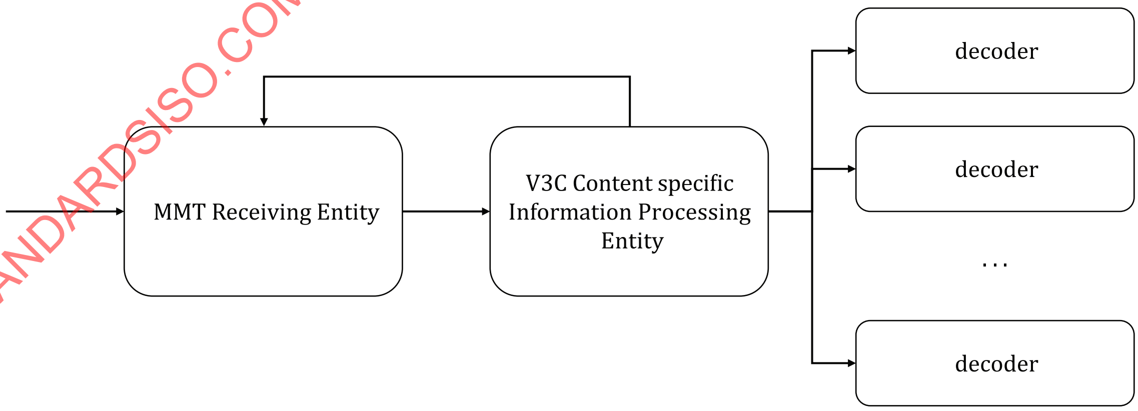

Figure 6 — Conceptual architecture of the client receiving V3C Content with MMT

12.2 MMT signalling descriptors for V3C content

12.2.1 Asset reference descriptor

12.2.1.1 General

This descriptor signals reference relationship information among Assets. The Asset Reference 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 V3C content, an Asset Reference descriptor is carried in the Asset descriptor loop of signalling messages or signalling tables of an Asset for the V3C atlas bitstream. The descriptor includes track references to the Assets carrying V3C video component bitstreams with the track reference types for each V3C video components. When V3C atlas tiles are present in the V3C content, the Asset descriptor loop of the Asset carrying the V3C atlas bitstream includes an Asset Reference descriptor to the Assets carrying the V3C atlas tile bitstream and the Asset descriptor loop of the Assets carrying V3C atlas tile bitstream carries the Asset Reference descriptor providing references to the Assets carrying the V3C video component bitstreams. When a V3C content has multiple V3C atlases, then the Asset descriptor loop of the Asset carrying common information applicable to all V3C atlases carries Asset Reference descriptors to the Assets carrying the V3C atlas bitstreams.

12.2.1.2 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 ; i<N ; i++){			
reference_type		**32**	**char**
asset_id()			
}			
}			

12.2.1.3 Semantics

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 MP4RA (http://www.mp4ra.org).

asset_id() provides the identifier of the Asset referenced by the Asset this descriptor is associated with, as defined in ISO/IEC 23008-1:2017, subclause 10.6.2.

12.2.2 V3C Asset descriptor

12.2.2.1 General

This Asset descriptor is used to inform the receiving entity and the consuming application about the content of an Asset that carries V3C content. 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 V3C content.

12.2.2.2 Syntax

Table 17 shows the syntax of the V3C Asset descriptor.

Table 17 — V3C Asset descriptor

Syntax	Values	No. of bits	Mnemonic
V3C_descriptor () {	'1111111'	**16**	**uimsbf**
descriptor_tag		**16**	**uimsbf**
descriptor_length	**N1**	**8**	**uimsbf**
data_type		**1**	**bslbf**
all_tiles_present_flag		**7**	**bslbf**
reserved			
if (!all_tiles_present_flag) {		**16**	**uimsbf**
num_tiles			
for (i=0; i<N1; i++) {		**16**	**uimsbf**
tile_id			
}			
}			
}			

12.2.2.3 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 V3C data present in this Asset group. Values for this field are listed in Table 18.

Table 18 — Values for data_type

Value	Description
0x00	All V3C components data (i.e., V3C atlas and V3C video components)
0x01	Atlas component data
0x02	Occupancy component data
0x03	Geometry component data
0x04	Attribute component data
0x05	Dynamic volumetric timed-metadata Information
0x06	Viewport timed-metadata information
0x07-0xFF	Reserved

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

num_tiles indicate the number of tiles carried in this Asset.

tile_id indicates a unique identifier for a particular atlas tile.

12.3 MMT signalling messages for V3C Content

12.3.1 General

Several MMT signalling messages are defined for V3C content. The V3C content-specific messages shall have the value of the message_id field set to 0x0000 and the value of the application_identifer() field set to urn:mpeg:mmt:app:v3c:2020 for identification. The type of the message is identified by the value of the v3c_application_message_type field as listed in Table 19.

Table 19 — Values for v3c_application_message_type

Application Message Type	Application Message Name
0x01	V3CAssetGroupMessage
0x02	V3CselectionMessage
0x03	V3CviewChangeFeedbackMessage
0x04-0xFF	Reserved

12.3.2 V3C Asset Group message

12.3.2.1 General

When sending V3C content via MMT, the V3CassetGroupMessage shall be used. This message provides the receiving entity with the information about the Assets associated with the V3C content. A receiving entity may then request a unique sub-set of these V3C Assets using the V3CSelectionMessage message. This message may also be used to inform the receiving entity about which of these Assets are currently being streamed to the receiving entity.

12.3.2.2 Syntax

Table 20 shows the syntax of the V3C Asset Group message.

Table 20 — V3C Asset Group message

Syntax	Values	No. of bits	Mnemonic
V3C_asset_group_message () {	**N1**	**16**	**uimsbf**
message_id		**8**	**uimsbf**
version		**16**	**uimsbf**
length			
message_payload **{**			
application_identifier()		**8**	**uimsbf**
v3c_application_message_type	'111111'	**8**	**uimsbf**
num_v3c_asset_groups		**16**	**uimsbf**
start_time	**N2**		
for (i=0; i<N1; i++) {		**8**	**uimsbf**
v3c_asset_group_id		**1**	**bslbf**
pending_flag		**1**	**bslbf**
3D_spatial_region_info_flag		**6**	**bslbf**
reserved			
if (***3D_spatial_region_info_flag***){		**16**	**uimsbf**
num_regions			
for (j=0; j<N2; j++) {			
V3CSpatialRegion()			
}			
}			
}			
}			
}			

12.3.2.3 Semantics

message_id indicates the identifier of the V3C application message.

version indicates the version of the V3C application message.

length indicates the length of the V3C 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.

v3c_application_message_type indicates the type of the V3C application message.

num_v3c_asset_groups indicates the number of V3C Asset groups, where each group contains the Assets associated with a V3C component.

start_time indicates the presentation time of the V3C content from which the state of the Assets listed in this message are applicable.

v3c_asset_group_id indicates the value of the group identification field of the Asset Reference descriptor carried in the Asset descriptor loop of the Asset for the V3C atlas bitstream or V3C atlas tile bitstream.

`pending_flag` indicates if all the data components for an Asset group are ready for rendering. When set to 1, it indicates that the data is ready, otherwise the flag is 0.

`3D_spatial_region_info_flag` indicates whether 3D spatial region information is present for an Asset group or not. Value 0 indicates this 3D sptial region informaiton is not provided. Value 1 indicates this 3D sptial region informaiton is provided.

`num_regions` indicates the number of 3D spatial region information.

`V3CSpatialRegion()` is an instance of `V3CSpatialRegion` defined in subclause 9.3 of this document 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 23008-1:2017, subclause 10.6.2.

12.3.3 V3C Selection message

12.3.3.1 General

The client uses this feedback message to request the set of Assets to be streamed by the sending entity.

12.3.3.2 Syntax

Table 21 shows the syntax of the V3C Selection message.

Table 21 — V3C Selection Message

Syntax	Values	No. of bits	Mnemonic
V3C_selection_message () {	**N1**	16	**uimsbf**
message_id		8	**uimsbf**
version		16	**uimsbf**
length	‘1111’		
`message_payload` **{**			
`application_identifier()`	**N2**	8	**uimsbf**
v3c_application_message_type			
num_selected_asset_groups			
for (i=0; i<N1; i++) {		8	**uimsbf**
v3c_asset_group_id		4	**bslbf**
reserved		4	**bslbf**
switching_mode		16	**uimsbf**
num_assets			
if (switching_mode == 0x01 || 0x02) {			
for (j=0; j<N2; j++) {			
asset_id()			
}			
}			
}			
}			

12.3.3.3 Semantics

`message_id` indicates the identifier of the V3C application message.

`version` indicates the version of the V3C application message.

`length` indicates the length of the V3C 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.

`v3c_application_message_type` indicates the type of the V3C application message.

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

`v3c_asset_group_id` is the value of the group identification field of the Asset_Reference_Descriptor carried in the asset descriptor loop of the Asset for the V3C Atlas bitstream or V3C Atlas tile bitstream

`switching_mode` indicates the switching mode used for the selection of assets as requested by the receiving entity. Switching modes are described in Table 22.

Table 22 — Switching modes and corresponding definition

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()` provides the asset identifier of the Asset, as defined in ISO/IEC 23008-1:2017, subclause 10.6.2, for the state change according to the switching mode specified.

12.3.4 V3C View Change Feedback message

12.3.4.1 General

For view-dependent delivery of V3C content through MMT, the client may use the V3C 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 client.

12.3.4.2 Syntax

Table 23 shows the syntax of the V3C View Change Feedback message.

Table 23 — V3C View Change Feedback message

Syntax	Values	No. of bits	Mnemonic
V3C_view_change_feedback_message () {		**16**	**uimsbf**
message_id		**8**	**uimsbf**
version		**16**	**uimsbf**
length			
message_payload **{**			
application_identifier()		**8**	**uimsbf**
v3c_application_message_type		**32**	**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			
}			
}			

12.3.4.3 Semantics

message_id indicates the identifier of the V3C application message.

version indicates the version of the V3C application message.

length indicates the length of the V3C 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.

v3c_application_message_type indicates the type of the V3C 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 metres in the global reference coordinate system. The values are in units of 2^{-16} metres.

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.

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 metres.

horizontal_fov specifies the longitude range corresponding to the horizontal size of the viewport region, in 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 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 V3C 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 containing visual volumetric video-based coding content. A specific brand may require some or all of 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 ISO/IEC 14496-12).

A.2 Single-track encapsulation of V3C data

The brand `'v3st'` may be present among the `compatible_brands` list of the `FileTypeBox`. File readers conforming to the `'v3st'` brand shall support single track encapsulation of V3C data specified in subclause 7.3.

A.3 Multi-track encapsulation of V3C data

The brand `'v3mt'` may be present among the `compatible_brands` list of the `FileTypeBox`. File readers conforming to the `'v3mt'` brand shall support multi-track encapsulation of V3C data specified in subclause 7.4.

The brand '`v3mp`' may be present among the `compatible_brands` of the `FileTypeBox`. File readers conforming to the '`v3mp`' brand shall support multi track encapsulation of V3C data specified in subclause 7.4, tools specified in Clauses 9, 10, 11 and 12.

If the brand `'v3mp'` is present among the `compatible_brands` of the `FileTypeBox`, the following constraint is applied:

— If volumetric annotation SEI messages are carried by the atlas sub-bitstream and object information in the scene object information SEI messages do not change over time, scene object information SEI message should be carried in the `setup_unit` arrays in the `V3CDecoderConfigurationRecord`.

A.4 Encapsulation of non-timed V3C data

The brand `'v3nt'` may be present among the `compatible_brands` of the `FileTypeBox`. File readers conforming to the `'v3nt'` brand shall support encapsulation of non-timed V3C data specified in Clause 8. Moreover, the brand `'v3nt'` does not mandate a `MovieBox`(`'moov'`) in the file.

Annex B
(normative)

V3C DASH schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:v3c:2020"
    xmlns:v3c="urn:mpeg:mpegI:v3c:2020"
    elementFormDefault="qualified">

    <xs:element name="videoComponent" type="v3c:VideoComponentType" />
    <xs:complexType name="VideoComponentType">
        <xs:attribute name="type" type="xs:string" use="required" />
        <xs:attribute name="is_auxiliary" type="xs:boolean" default="false" />
        <xs:attribute name="map_index" type="xs:integer" />
        <xs:attribute name="attribute_type" type="xs:unsignedByte" />
        <xs:attribute name="attribute_index" type="xs:unsignedByte" />
        <xs:attribute name="attribute_dim_partition_index" type="xs:unsignedByte" />
        <xs:attribute name="atlas_id" type="xs:integer" use="optional" default="0" />
        <xs:attribute name="tile_ids" type="UIntVectorType" use="optional" />
    </xs:complexType>

    <xs:attribute name="vId" type="xs:string" />
    <xs:attribute name="atlas_id" type="xs:integer" use="optional" />
    <xs:attribute name="tile_ids" type="UIntVectorType" use="optional" />

    <xs:element name="v3sr" type="v3c:spatialRegionMapType" />
    <xs:complexType name="spatialRegionMapType">
        <xs:element name="spatialRegion" type="v3c:spatialRegionType" minOccurs="1"/>
    </xs:complexType>

    <xs:complexType name="spatialRegionType">
        <xs:attribute name="id" type="xs:unsignedShort" use="required" />
        <xs:attribute name="type" type="xs:unsignedByte" use="optional" default="0" />
        <xs:attribute name="tile_ids" type="UIntVectorType" />
        <xs:element name="cuboid" type="v3c:spatialRegionCuboidType" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="lod" type="v3c:lodType" />
        <xs:element name="viewport" type="v3c:spatialRegionViewportType"
            minOccurs="0" maxOccurs="1"/>
    </xs:complexType>

    <xs:complexType name="spatialRegionCuboidType">
        <xs:attribute name="anchor" type="UIntVectorType" use="required"
            minLength="3" maxLength="3" />
        <xs:attribute name="dimensions" type="UIntVectorType" use="required"
            minLength="3" maxLength="3"/>
    </xs:complexType>

<xs:complexType name="spatialRegionViewportType">
        <xs:attribute name="rvIds" type="StringVectorType" use="required" />
    </xs:complexType>

    <xs:complexType name=”lodType”>
        <xs:attribute name="idx" type="xs:unsignedByte" use="required" />
        <xs:attribute name="tile_ids" type="UIntVectorType" use="required" />
    </xs:complexType>

    <!-- Added support for 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:attribute name="viewport_id" type="xs:integer" use="optional" />
    <xs:element name="ViewportInfo" type="v3c:ViewportInfoType"/>

    <xs:complexType name="ViewportInfoType">
        <xs:attribute name="vp_pos" type="FloatVectorType" use="required"
            minLength="3" maxLength="3"/>
        <xs:attribute name="vp_quat" type="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="initialViewport" 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>
</xs:schema>
```

Annex C
(normative)

MIME types and sub-parameters

C.1 MIME types and sub-types

When MIME type is associated with V3C content as described in this document, 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 V3C family

When the first element of a value is a code indicating a codec from ISO/IEC 23090-5, as documented in subclause 7.3 ('`v3e1`' or '`v3eg`') or in subclause 7.4 ('`v3c1`', '`v3cg`', '`v3cb`', '`v3a1`', '`v3ag`', or '`v3t1`') and the respective track can be interpreted as containing an atlas sub-bitstream, the elements following are a series of values from `v3c_parameter_set` syntax structure, as defined in ISO/IEC 23090-5, contained in `v3c_parameter_set` of the V3C decoder configuration record, separated by period characters (“.”). In all numeric encodings, leading zeroes may be omitted.

— the `ptl_tier_flag`, encoded as ‘`L`’ (`ptl_tier_flag==0`) or ‘`H`’ (`ptl_tier_flag==1`), followed by the ptl_level_idc, encoded as a decimal number;

— the `ptl_profile_codec_group_idc` encoded as a hexadecimal number;

— the `ptl_profile_pcc_toolset_idc` encoded as a hexadecimal number

— the `ptl_profile_reconstruction_idc` encoded as a hexadecimal number

EXAMPLE codecs=v3e1.L2.1.0.1

Main tier, Level 2, video components are encoded with ISO/IEC 23008-2, V-PCC Basic toolset profile, Rec1 reconstruction profile.

Annex D
(informative)

DASH MPD examples

D.1 Single track example

An example of MPD with a single-track mode is presented below.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
    type="static"
    mediaPresentationDuration="PT3256S"
    minBufferTime="PT1.2S"
    profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

    <BaseURL>http://cdn1.example.com/</BaseURL>
    <BaseURL>http://cdn2.example.com/</BaseURL>

    <Period>
        <AdaptationSet
            mimeType="video/mp4" codecs="v3e1.L2.0.0.1, resv.vvvc.avc1.4D401E"
frameRate="30">
            <SegmentList>
                <Initialization sourceURL="seg-m-init.mp4"/>
            </SegmentList>
            <Representation bandwidth="512000">
                <BaseURL>vpcc-512k.mp4</BaseURL>
            </Representation>
            <Representation bandwidth="1024000">
                <BaseURL>vpcc-1024k.mp4</BaseURL>
            </Representation>
            <Representation bandwidth="2048000">
                <BaseURL>vpcc-2048k.mp4</BaseURL>
            </Representation>
        </AdaptationSet>
    </Period>
</MPD>
```

D.2 Multi-track example (using Preselection element)

In this example, the video components of a V3C sequence are available in two different resolutions. Note that the resolution of the occupancy component may not necessarily be identical to that of the geometry and attribute components. For each resolution alternative of the geometry and attribute V3C video components, two bitrate alternatives are available.

Let O_A and O_B be the two occupancy resolutions. Similarly, let G_A and G_B be the two available geometry resolutions and A_A and A_B be the two attribute resolutions. Since each geometry and attribute resolution is available at two different bitrates, let us denote these as $G_{A,1}$, $G_{A,2}$, $G_{B,1}$, $G_{B,2}$, $A_{A,1}$, $A_{A,2}$, $A_{B,1}$, and $A_{B,2}$. Occupancy O_A is compatible with $G_{A,1}$, $G_{A,2}$, $A_{A,1}$, and $A_{A,2}$. While occupancy O_B is compatible with $G_{B,1}$, $G_{B,2}$, $A_{B,1}$, and $A_{B,2}$.

Each resolution of the geometry and attribute components can be signalled by a separate Adaptation Set with two Representations, one for each bitrate. Each occupancy resolution is also signalled using a separate Adaptation Set with a single Representation. Each Video Component Adaptation Set includes a V3CVideoComponent descriptor with the `@type` set to the corresponding value. Finally, the V3C track which includes the atlas bitstream is signalled with an Adaptation Set containing a single Representation.

Compatible Video Component Adaptation Sets, along with the Main Adaptation Set, are grouped together in two Preselections in the MPD. To indicate that these Adaptation Sets are referenced in at least one Preselection, a Preselection descriptor without the `@value` is signalled in each Adaptation Set. Each Preselection includes a V3C descriptor that indicates at least the mandatory `@vId`. The values assigned to the `@vId` of the two Preselections are identical, indicating that both Preselections belong to the same point cloud content.

The Main Adaptation Set contains the Initialization Segment for the complete experience. Therefore, in the case of an ISOBMFF container, the Initialization Segment contains `TrackBox`es for the V3C atlas track as well as the V3C video component tracks of all representations of the video components (all resolutions and bitrates).

Figure D.1 illustrates the different Adaptation Sets and their relation to the Preselections that represent the V3C content described in this example.

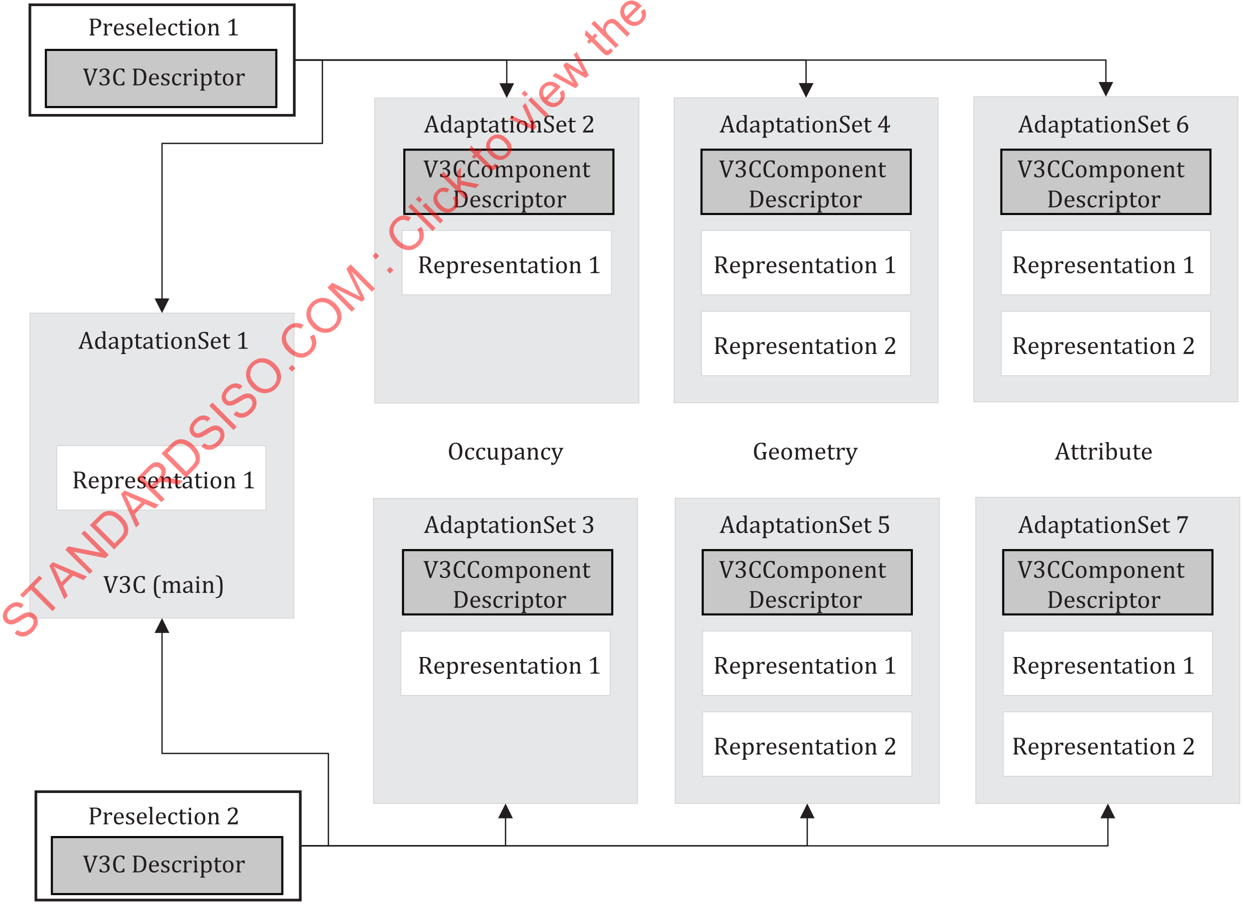

Figure D.1 — MPD layout for the multi-track V3C media example

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xmlns:v3c="urn:mpeg:mpegI:v3c:2020"
    type="static"
    mediaPresentationDuration="PT10S"
    minBufferTime="PT1S"
    profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

<Period>
    <!-- Main V3C AdaptationSet -->
    <AdaptationSet id="1" codecs="v3c1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <!-- Occupancy -->
    <AdaptationSet id="2" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:component">
                <v3c:videoComponent type="occp" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <AdaptationSet id="3" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:component">
                <v3c:videoComponent type="occp" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <!-- Geometry -->
    <AdaptationSet id="4" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:component">
                <v3c:videoComponent type="geom" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>
    <AdaptationSet id="5" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:component">
                <v3c:videoComponent type="geom" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <!-- Attribute -->
    <AdaptationSet id="6" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:component">
                <v3c:videoComponent type="attr" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <AdaptationSet id="7" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:component">
                <v3c:videoComponent type="attr" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <!-- Preselections -->
    <Preselection id="1" tag="1" preselectionComponents="1 2 4 6" codecs="v3c1">
            <!—V3C Descriptor -->
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:vpc" vId="1" />
    </Preselection>

    <Preselection id="2" tag="2" preselectionComponents="1 3 5 7" codecs="v3c1">
            <!—V3C Descriptor -->
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:vpc" vId="1" />
    </Preselection>

</Period>
</MPD>
```

D.3 Multi-track example (using preselection descriptor)

The following MPD example demonstrates how Preselection descriptors can be used for signalling the same V3C content described in subclause D.2.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<MPD
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xmlns:v3c="urn:mpeg:mpegI:v3c:2020"
    type="static"
    mediaPresentationDuration="PT10S"
    minBufferTime="PT1S"
    profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

<Period>
    <!-- Main V3C AdaptationSet -->
    <AdaptationSet id="1" codecs="v3c1">
             <!-- V3C Descriptor -->
             <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:v3c" vId="1" />
            <!-- Preselection Descriptors -->
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" value="1,1 2
4 6" />
             <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" value="2,1
3 5 7" />
             <!-- Representation -->
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <!-- Occupancy -->
    <AdaptationSet id="2" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:videoComponent" >
                <v3c:videoComponent type="occp" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <AdaptationSet id="3" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:videoComponent">
                <v3c:videoComponent type="occp" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <!-- Geometry -->
    <AdaptationSet id="4" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:videoComponent">
                <v3c:videoComponent type="geom" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <AdaptationSet id="5" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:videoComponent">
                <v3c:videoComponent type="geom" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <!-- Attribute -->
    <AdaptationSet id="6" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:videoComponent">
                <v3c:videoComponent type="attr" attribute_type="1" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

    <AdaptationSet id="7" mimeType="video/mp4" codecs="resv.vvvc.hvc1">
            <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" />
            <EssentialProperty schemeIdUri="urn:mpeg:mpegI:v3c:2020:videoComponent">
                <v3c:videoComponent type="attr" attribute_type="1" />
            </EssentialProperty>
            <Representation>
                ...
            </Representation>
    </AdaptationSet>

</Period>
</MPD>
```

D.4 Multi-track example with multiple atlas tile tracks

In this MPD example, the V3C content is encapsulated using multi-track encapsulation and the container includes one atlas track, two atlas tile tracks, and six V3C video component tracks. Where each atlas tile track is associated with three V3C video component tracks carrying occupancy, geometry, and attribute information for the atlas tiles carried by the atlas tile track. The MPD file therefore contains nine Adaptation Sets and two Preselections, each Preselection grouping an Atlas Tile Adaptation Set with associated Video Component Adaptation Sets. And the Representation of each Atlas Tile Adaptation Set depends on the Representation of the Main Adaptation Set. The layout of this MPD is shown in Figure D.2.