ISO/IEC JTC 1/SC 29/WG 03 N1501

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Technologies under consideration on carriage of V3C data

**Status:** Approved

**Date of document:** 2025-06-16

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

**No. of pages:** 119 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N1501**

**April 2025, OnLine**

|  |  |
| --- | --- |
| **Title** | **Technologies under consideration on carriage of V3C data** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **25037** |

1. Introduction

This document contains technologies under consideration for carriage of V3C data as of the 18th MPEG Systems WG meeting.

1. Signaling of instanceable variations for V-DMC encoded content (m69853)

# 1         Introduction

The input and output of V-DMC [1] codec are meshes, which are widely used for modeling and rendering in computer graphics. Due to their popularity, multiple optimizations have been implemented in hardware to improve their rendering performance. One of such optimizations is instanced rendering, instanced drawing, or instancing, which allows reusing mesh information to efficiently draw multiple versions of a mesh. Simply drawing multiple versions of the same mesh itself might not be very useful, so instanced rendering offers limited customization options of each instance to generate visual alternatives from the same mesh model. Instancing offers for example the following methods for customizing individual instances:

       Transformation, to generate instances in different world positions, orientations and sizes (a).

       Texture variants, to attach different textures for instances. All instances must share the same material, but texture variants are supported using texture offsets that can be configured per instance.

       Animation variants via skinning or morphing, to animate instances with different pose or pace.

        Other shader accessible customizations.

Due to the popularity of instancing, it seems useful to be able to provide information for efficiently rendering different variants of V-DMC encoded meshes.

## 1.1        Use cases for instanced drawing

Instanced drawing becomes relevant if there is value for rendering the same mesh multiple times with minimal variations between the objects. For example, the following use cases can be considered:

1.     Scenes with multiple similar objects like (crowds of people).

2.     Scenes where a low LOD mesh is repeated several times for example in the background (ties extremely well with the base-mesh concept from V-CMD).

3.     Particle systems, where each particle is a mesh with small differences.

4.     Procedural generation (like particle systems), where a scene can be populated procedurally while leveraging shared geometry of its components.

5.     Systems where draw calls are expensive, for example on battery-powered VR/AR devices every draw call counts.

# 2         Description

Some of the signaling for instanced rendering may be done in the application logic or in delivery systems protocols. These could include the transformation related signaling as it will not have an impact on the storage and largely depend on the use case of the application. Also, the way V-DMC handles compression of mesh-frames makes it less suitable for animation variants because it doesn’t rely on skinning- or morphing-based animations (the mesh between frames is different, thus not instanceable). That leaves the texture variants, which would benefit from being stored together with the compressed V-DMC bitstreams.

To provide necessary information for instanced rendering of texture variants, track grouping seems like a logical tool considering that in a group one or more tracks may be needed but are not required for display. As such track references seem like a less fitting solution.

Typically, alternate\_groups [2] are used to indicate video track alternatives that contain different versions of the same visual track. However, the logic for handling alternate\_groups is different than what is intended for instanced variants. The intention for alternate groups is that each track in the group describes how it is different from the other tracks, e.g. in video resolution or bitrate. For instanced variants every 2D video related parameter may be the same, making it impossible to distinguish visual alternatives in the group. Furthermore, in an alternate group typically only one track is used for display, for instanced rendering many alternative tracks may be needed. Hence, another way of establishing the relationship is needed.

As such it would be preferred if the signaling of texture variants for instancing would be kept as simple as possible. The bare minimum would be to indicate if and how the V-DMC encoded mesh can be rendered using instanced drawing, and which attribute tracks can be considered as input for instancing.

# 3         Proposed changes as described in m69129

It is proposed to make the following changes to V3C carriage specification [3]:

       Add indication to atlas track for V-DMC that the mesh can be instanced, but not temporally.

       Add new track group for indicating that the tracks in a group contain attribute alternatives for instanced drawing.

       Add new entity group for indicating that the tracks and items in a group contain attribute alternatives for instanced drawing.

## 3.1        V3C instanced rendering box

### 3.1.1       Definition

Box Types: 'vire'  
Container: Sample Entry ('v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag')   
Mandatory: No  
Quantity: one

A single V3CInstancedRenderingBox may present in the sample entry of a V3C atlas track indicating that the mesh representation associated with the atlas track may be used for instanced rendering.

### 3.1.2       Syntax

aligned(8) class V3CInstancedRenderingBox extends FullBox('vire', version = 0, 0){

unsigned int(1) can\_be\_instanced;

unsigned int(1) can\_be\_temporally\_instanced;

unsigned int(6) reserved;

}

### 3.1.3       Semantics

can\_be\_instanced, when equal to 1, indicates that the mesh information associated with the atlas track may be rendered using instanced drawing.

can\_be\_temporally\_instanced, when equal to 1, indicates that the mesh information associated with the atlas track may be used for temporal instancing.

## 3.2        Grouping alternative attributes for instanced rendering

### 3.2.1       General

For instanced rendering, multiple versions of an attribute track may exist that can be indexed using the same set of UV-coordinates and the same mesh. For such applications, instanced attribute track group offers the needed information to identify tracks that can be considered as alternative attributes for instanced drawing. An application may choose to use zero or more tracks from the group and the group and does not mandate any rendering behavior. It merely informs that there is an option for instanced drawing with multiple different attribute variants.

### 3.2.2       Instanced attribute variant track group

#### 3.2.2.1    Definition

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

Instanced attribute variant track groups are defined using InstancedAttributeVariantTrackGroupBox, which extends TrackGroupTypeBox as defined in ISO/IEC 14496-12. The tracks belonging to the track group are considered as alternative attributes for instanced rendering and shall contain information that can be indexed using the same set of UV-coordinates. Zero or more tracks of the group may be used at any given time.

#### 3.2.2.2    Syntax

aligned(8) class InstancedAttributeVariantTrackGroupBox extends TrackGroupTypeBox('iavg') {

// track\_group\_id is inherited from TrackGroupTypeBox

}

## 3.3        Grouping alternative attributes for instanced rendering

### 3.3.1       General

For instanced rendering, multiple versions of an attribute item may exist that can be indexed using the same set of UV-coordinates and the same mesh. For such applications, instanced attribute entity group offers the needed information to identify items that can be considered as alternative attributes for instanced drawing. An application may choose to use zero or more items from the entity group and the entity group does not mandate any rendering behavior. It merely informs that there is an option for instanced drawing with different attribute variants.

### 3.3.2       Instanced attribute variant entity to group box

#### 3.3.2.1    Definition

Box Types: 'eiav'  
Container: GroupsListBox ('grpl')  
Mandatory: No  
Quantity: Zero or more

An EntityToGroupBox with grouping\_type equal to 'eiav' indicates attribute items that shall be considered as variants for instanced rendering. Each item belonging to the entity group shall be indexable with the same UV-coordinates. An InstancedAttributeVariantEntityToGroupBox is used to group non-timed (item) V3C data in the same group. Zero or more items from the group may be needed at any given time.

#### 3.3.2.2    Syntax

aligned(8) class InstancedAttributeVariantEntityToGroupBox extends EntityToGroupBox('eiav') {}

# 4         Proposal

It is proposed to include Clauses 3.1, 3.2 and 3.3 in edition two of ISO/IEC 23090-10.

# 5         References

[1] ISO/IEC CD 23090-29 Video-based dynamic mesh coding.

[2] ISO/IEC 14496-12 ISO base media file format.

[3] ISO/IEC 23090-10 WD Carriage of Visual Volumetric Video-based Coding Data Ed 2.

1. On signaling multiple V3C parameter sets in ISO/IEC 23090-10 (m70325)

# Introduction

The ISO/IEC 23090-10 (Carriage of V3C Data) specification ‎[1] defines how to carry V3C bitstream in ISOBMFF media containers. The ISO/IEC 23090-5 (Visual Volumetric Video-based Coding) specification [2] supports signaling of multiple V3C parameter sets in a V3C bitstream. When the VPS information is changing over time in the V3C bitstream, the V3C unit header information for the following V3C units will also change accordingly to reference the new VPS.

The ISO/IEC 23090-10 Carriage of V3C Data specification [1] defines the storage of V3C parameter sets (VPS) in a V3CDecoderConfigurationRecord. Though the semantics of the V3CDecoderConfigurationRecord provides the flexibility to include multiple V3C parameter sets, the specification restricts the number of V3C parameter sets to one. TMC2 reference software supports generating the V3C bitstreams with multiple V3C parameter sets.

Hence, the ISO/IEC 23090-10 specification does not clearly specify how to handle multiple V3C parameter sets and multiple V3C unit headers information when they are changing over the time in a V3C bitstream.

In this contribution, we propose two different solutions to address the above-mentioned issue:

1. by allowing multiple sample entries
2. by introducing a new sample to group box and the Sample group description entry to signal the V3C parameter sets and the associated V3C unit headers present in the V3C bitstream.

The specification text changes for Solution 1 are provided in a different file. The specification text changes for Solution 2 are provided in section 2.2 of this document.

# Proposed Solutions

## Solution-1: Multiple sample entries

### V3C bitstream track

When the V3C data is carried using a V3C bitstream track and multiple the V3C parameter sets are present in a V3C bitstream, V3C parameter sets information can be stored using multiple sample entries in the V3C bitstream track.

Each V3C parameter set associated with one or more samples in a V3C bitstream track shall be stored in a separate SampleEntry instance. All sample entries present in a V3C bitstream track are signalled in the SampleDescriptionBox of that track. Each SampleEntry in a SampleDescriptionBox is identified using the sample\_description\_index. The group of samples in a V3C bitstream track that refers to a V3C parameter set, shall use the corresponding sample\_description\_index value in the SampleToChunkBox for those samples.

### Multi-track encapsulation of V3C data

#### Atlas track

Each V3C parameter set and the V3C unit header associated with one or more samples in a V3C atlas track shall be stored in a separate SampleEntry instance. The group of samples in a V3C atlas track that refers to a V3C parameter set and a V3C unit header information, shall use the corresponding sample\_description\_index value in the SampleToChunkBox for those samples.

#### Video component track

Each V3C unit header associated with one or more of samples in a video component track shall be stored in a separate SampleEntry instance (in SchemeInformationBox). The group of samples in a video component track that refers a V3C unit header information, shall use the corresponding sample\_description\_index value in the SampleToChunkBox for those samples.

#### Atlas tile tracks

##### Atlas track

when multiple V3C parameter sets are present in a V3C bitstream, the V3CDecoderConfigurationRecord present in the atlas track that references the V3C atlas tile tracks shall store all the V3C parameter sets present in the V3C bitstream. In this case, the num\_of\_v3c\_parameter\_sets in V3CDecoderConfigurationRecord shall be equal to the number of unique V3C parameter sets present in the bitstream.

When multiple V3C parameter sets are available in the V3C bitstream, the V3CUnitHeaderBox present in the atlas track that references the V3C atlas tile tracks, shall include all unique atlas V3C unit headers and the version of the V3CUnitHeaderBox box shall be set to 1.

##### Syntax

aligned(8) class V3CUnitHeaderBox extends FullBox('vunt', version = 1, 0){

if(version == 1) {

unsigned int(8) num\_v3c\_unit\_headers;

for (int i=0; i < num\_v3c\_unit\_headers; i++) {

bit(8) header[4];

}

}

else {

v3c\_unit\_header header(); // 4-bytes as defined in ISO/IEC FDIS 23090-5

}

}

##### Semantics

num\_v3c\_unit\_headers specify the number of atlas V3C unit headers signalled in the V3CUnitHeaderBox.

##### Atlas tile track

When multiple V3C parameter sets are present in the V3C bitstream, V3CAtlasTileConfigurationBox is extended to include v3c\_parameter\_set\_index and v3c\_unit\_header\_index as below and the version of the V3CAtlasTileConfigurationBox box shall be set to 1.

##### Syntax

class V3CAtlasTileConfigurationBox extends FullBox('v3tC', version = 1, 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;

}

if(version == 1)

{

unsigned int(4) v3c\_parameter\_set\_index;

unsigned int(4) v3c\_unit\_header\_index;

}

}

##### Semantics

v3c\_parameter\_set\_index is an integer that gives the index of the v3c parameter set that is referred by the samples. The index ranges from 1 to the number of unique V3C parameter sets present in the V3C bitstream. This value represents the index of the V3C parameter sets present in the V3CconfigurationBox of the atlas track that references the V3C atlas tile tracks.

v3c\_unit\_header\_index is an integer that gives the index of the v3c unit header that is used by the samples. The index ranges from 1 to the number of unique atlas V3C unit headers present in the V3C bitstream. This value represents the index of the V3C unit headers present in the V3CUnitHeaderBox of the atlas track that references the V3C atlas tile tracks.

#### Multiple Atlas tracks

Each V3C unit header information associated with one or more samples in a V3C atlas track shall be stored in a separate SampleEntry instance. The group of samples in a V3C atlas track that refers to a V3C unit header information, shall use the corresponding sample\_description\_index value in the SampleToChunkBox for those samples.

The V3C atlas base track with sample entry type 'v3cb' shall contain only one SampleEntry even though multiple V3C parameter sets are present in the V3C bitstream. The V3CconfigurationBox present in the V3C atlas base track with sample entry type 'v3cb' shall store all the unique V3C parameter sets present in the bitstream.

##### **Syntax**

aligned(8) class V3CAtlasSampleEntry() extends VolumetricVisualSampleEntry (type) {

// type is 'v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag'

V3CConfigurationBox config;

V3CUnitHeaderBox unit\_header;

if (type == 'v3a1' || type == 'v3ag')

{

unsigned int(4) v3c\_parameter\_set\_index;

unsigned int(4) reserved {0};

}

}

##### **Semantics**

v3c\_parameter\_set\_index is an integer that provides the index of the v3c parameter set that is referred by the samples. The index ranges from 1 to the number of unique V3C parameter sets present in the V3C bitstream. For 'v3a1' or 'v3ag' track types, this value represents the index of the V3C parameter set present in the V3CconfigurationBox of the atlas base track that references the V3C atlas tracks. For all other track types this value shall be set to 1.

## Solution-2: V3C parameter set sample group (‘vvps’)

### General

When the V3C data is carried using a single track or multiple tracks and multiple V3C parameter sets are present in a V3C bitstream, the V3CConfigurationBox shall signal all unique V3C parameter sets present in the V3C bitstream. The num\_of\_v3c\_parameter\_sets in a V3CDecoderConfigurationRecord shall be equal to the number of unique V3C parameter sets present in the V3C bitstream. Under the 'v3a1' and 'v3ag' sample entry, no V3C parameter set shall be stored in the v3c\_parameter\_set array.

When multiple V3C parameter sets are present in a V3C bitstream, the V3CUnitHeaderBox shall include all unique V3C unit headers and the version of the V3CUnitHeaderBox box shall be set to 1.

#### Syntax

aligned(8) class V3CUnitHeaderBox extends FullBox('vunt', version = 1, 0){

if (version == 1) {

unsigned int(8) num\_v3c\_unit\_headers;

for (int i=0; i < num\_v3c\_unit\_headers; i++) {

bit(8) header[4];

}

}

else {

v3c\_unit\_header header(); // 4-bytes as defined in ISO/IEC FDIS 23090-5

}

}

#### Semantics

num\_v3c\_unit\_headers specify the number of V3C unit headers signalled in the V3CUnitHeaderBox.

### Sample group and sample group description

The use of 'vvps' for the grouping\_type in sample grouping represents the assignment of samples in a track to the corresponding V3C parameter set and the V3C unit header carried in the SampleGroupDescriptionEntryBox. When a SampleToGroupBox with grouping\_type equal to 'vvps' is present, an accompanying SampleGroupDescriptionBox with the same grouping type shall be present and SampleToGroupBox contains the index of the sample group description entry that the sample belongs to.

* When multiple atlas tracks are present, the atlas base track that references the V3C atlas tracks shall not contain any sample to group box angood sample group description box with a grouping\_type equal to 'vvps'.
* When atlas tile tracks are present, the atlas track that references the V3C atlas tile tracks shall not contain any sample to group box and sample group description box with a grouping\_type equal to 'vvps'.

#### Definition

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

For a V3C bitstream track, a VPS information sample group entry defines the V3C parameter set information for the samples in a V3C bitstream track that use the same V3C parameter set information. For other tracks, a VPS information sample group entry signals the V3C parameter set and V3C Unit header information for the atlas samples that use the same V3C parameter set information.

When multiple V3C parameter sets are present in the V3C bitstream, the sample grouping type 'vvps' shall only be used in tracks with the sample entries 'v3e1', 'v3eg', 'v3c1', 'v3cg', 'v3t1','v3a1' or 'v3ag'.

When multiple V3C parameter sets are present in the V3C bitstream, the sample grouping type 'vvps' shall be present in V3C video component tracks.

When the multiple V3C parameter sets are not available in the V3C bitstream, this sample group and the associated sample group description entries shall not be present in any track.

#### Syntax

The syntax of the V3CParameterSetInfoEntry is as below,

For 'v3c1', 'v3cg', 'v3t1','v3a1', 'v3ag', 'v3e1', 'v3eg', or video track types,

aligned(8) class V3CParameterSetInfoEntry() extends VolumetricVisualSampleGroupEntry ('vv3c')

{  
 unsigned int(4) v3c\_parameter\_set\_index;

unsigned int(4) v3c\_unit\_header\_index;

}

#### Semantics

v3c\_parameter\_set\_index is an integer that gives the index of the v3c parameter set that is referred by the samples. The index ranges from 1 to the number of V3C parameter sets present in the V3CconfigurationBox. For 'v3a1' or 'v3ag' track types, this value represents the index of the V3C parameter sets present in the V3CconfigurationBox of the atlas base track that references the V3C atlas tracks. For 'v3t1' track types, this value represents the index of the V3C parameter sets present in the V3CconfigurationBox of the atlas track that references the V3C atlas tile tracks. For V3C video component tracks, this value shall be set to zero.

v3c\_unit\_header\_index is an integer that gives the index of the v3c unit header that is used by the samples. The index ranges from 1 to the number of V3C unit headers present in the V3CUnitHeaderBox. This value shall be set zero for V3C bitstream tracks with sample entry type 'v3e1', and 'v3eg'. For 'v3t1' track types, this value represents the index of the V3C unit headers present in the V3CUnitHeaderBox of the atlas track that references the V3C atlas tile tracks. For V3C video component tracks, this value represents the index of the V3C header information present in the SchemeInformationBox.

# Extraction process

## Solution-1: Multiple sample entries method

#### Multi-track with single atlas track

Figure 1 below illustrates an informative example of V3C ISOBMFF file structure with one atlas track and 3 video component tracks. Track 1 is an atlas track containing multiple sample entries. The sample entry 1 in track 1 contains a V3CConfigurationBox containing one V3C parameter set referred as VPS1 and a V3CUnitHeaderBox containing one V3C Unit header information referred as VUH1. Similarly sample entry 2 contains VPS2 and VUH2 in the respective boxes.

When multiple V3C parameters are present in a V3C bitstream, the samples in track 1 using the VPS1 and VUH1 (stored in Sample Entry 1 as shown in Figure 1) are identified using the sample\_description\_index value in the SampleToChunkBox which is set to 1. Similarly, the samples in track 1 using the VPS2 and VUH2 (stored in Sample Entry 2 as shown in Figure 1) are identified using the sample\_description\_index value in the SampleToChunkBox which is set to 2.



Figure Informative – Example of using multiple sample entries in multi-track V3C file

## Solution-2: Sample grouping method

### Multi-track with single atlas track

Figure 2 below illustrates an informative example of V3C ISOBMFF file structure with one atlas track and a video component track. Track 1 is an atlas track and track 2 is a video component track. Atlas track contains the sample group description box and sample to group box with grouping type ‘vvps’. The sample group description box with grouping\_type equal to ‘vvps’ present in track 1 signals the different V3C parameter sets used in different samples of that track. The sample to group box with grouping\_type equal to ‘vvps’ contains associated sample group description entry index for each sample present in that track.

When multiple V3C parameters are pre present in a V3C bitstream, the index of the V3C parameter set stored in V3CConfigurationBox and the index of the V3C unit header information stored in V3CUnitHeaderBox are signaled in a sample group description entry and the samples using that V3C parameter set are indicated in the sample to group box with grouping\_type equal to ‘vvps’. In the example shown in Figure 2, samples from 1 to 100 in track 1 fetched the information about V3C parameter set from sample group entry description at index 1. In the example shown in Figure 2, the sample group entry description at index 1 specifies that, the samples from 1 to 100 uses the V3C parameter set at index 1 present in the V3CconfigurationBox (V3cC box shown in Figure 5). Also, the samples from 1 to 100 uses the V3C unit header information at index 1 present in the V3CUnitHeaderBox (VUNT box shown in Figure 5).

Similarly, the sample group entry description at index 2 specifies that, the samples from 101 to 300 uses the V3C parameter set at index 2 present in the V3CconfigurationBox (V3cC box shown in Figure 2) and the V3C unit header information at index 2 present in the V3CUnitHeaderBox (VUNT box shown in Figure 2).

In video component track (track 2), the index of the V3C unit header information stored in V3CUnitHeaderBox are signaled in a sample group description entry and the samples using that V3C unit header information are indicated in the sample to group box with grouping\_type equal to ‘vvps’. In track 2 of the example shown in Figure 2, the sample group entry description at index 1 specifies that, the samples from 1 to 100 uses the V3C unit header information at index 1 present in the V3CUnitHeaderBox (as shown in Figure 2 track 2 VUNT box). Also, the samples from 101 to 200 in track 2 uses the V3C unit header information at index 2 present in the V3CUnitHeaderBox. The samples from 1 to 100 in track 2, fetches the V3C parameter set information from the corresponding atlas sample present in track 1.



Figure Informative – Example of using ‘vvps’ sample group in multi-track V3C file

# Comparison

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Multiple sample entries** | **Sample group** |
| Specification text | Minimal | Addition of new sample grouping type |
| File Size overhead | High | Small |

# Recommendations

We recommend adopting the proposed changes described in sections 2 and 3 of this document and integrating them into the 2nd edition of the ISO/IEC 23090-10 specification. Section 3 of this document describes the extraction process and is recommended to be integrated as an informative Annex in the 2nd edition of the ISO/IEC 23090-10 specification.

# References

1. WG03N00241, “Text of ISO/IEC FDIS 23090-10 Carriage of Visual Volumetric Video-based Coding Data”, MPEG#134, April 2021.
2. WG07N00553, “Text of ISO/IEC FDIS 23090-5 2nd Edition Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)”, MPEG#141, Online, January 2023.

Nokia may have patent claims relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms for such patent claims required to implement the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

1. On submesh sub-samples for V-DMC (m70103)

NOTE: submesh\_id has been already integrated into the specification text of the 23090-10 2nd edition

**1.     Introduction**

At the MPEG#147 meeting in Sapporo, methods to store submeshes in a basemesh and submesh track were proposed in m68922 [1].

This contribution provides a background on some specific parameters for submeshes carried in sub-sample of basemesh and submesh track. Section 2 provides a background on the importance of the specific parameters. Section 3 proposes techniques to indicates the specific parameters for submeshes carried in sub-samples in a basemesh and submesh track.

**2.     Background**

In V-DMC, specified in ISO/IEC 23090-29 [2], the submeshID is used a unique identifier for a submesh. The submesh identifier is specified in the afmi\_submesh\_id. The submesh ID is a useful property of a submesh for submesh identification as well as serves important in determining the association of a submesh to an atlas patch.

A Basemesh bitstream format specified in Annex H of ISO/IEC 23090-29 [2] employs two different sub-codecs to code the mesh sequence, namely an intra mesh codec and an inter mesh codec. Some prefix data in the intra/inter mesh payload may be replicated across different submeshes.

Therefore, when bmsps\_codec\_specific\_parameters\_present\_flag is equal 1, the basemesh sequence parameter set can carry some prefix data of the coded mesh payload. In order to fully decode the mesh payload, the prefix data is concatenated with the submesh data carried as described in H.9.4.5.2.1. of ISO/IEC 23090-29 [2].

**3.     Proposal**

**3.1. Definition of a sub-sample for Basemesh Track**

A sample in a Basemesh Track may include a set of submeshes. Each submesh in a Basemesh Track is a sub-sample. For the use of the SubSampleInformationBox in ISO/IEC 14496-12 [3] in a Basemesh track, a sub-sample is defined based on the value of the flags field of the sub-sample information box as specified below. The presence of this box is mandatory in the case where more than one submesh is carried in the track. If the sub sample information box is present in a Basemesh track containing submesh data for more than one submesh, the codec\_specific\_parameters field in the box shall have the semantics defined here.

flags specifies the type of sub-sample information given in this box. If the value indicated flags field is 0, then a sub-sample contains one Submesh NAL unit in the group of submeshes carried in a sample of the Basemesh track. Values other than 0 are reserved for future use.

The subsample\_priority field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12 [3].

The discardable field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded.

The codec\_specific\_parameters field of the SubSampleInformationBox is defined for Basemesh track as follows:

if (flags == 0) {  
 unsigned int(16) submesh\_id;  
 unsigned int(1) rap\_nal\_unit\_flag;  
 unsigned int(1) submesh\_self\_contained\_flag;  
 bit(14) reserved = 0;  
}

The semantics of the above fields are:

submesh\_id indicates the identifier for a submesh contained in this sub-sample of the basemesh track. submesh\_id shall be equal to the submesh identifier signaled in the syntax element bmsi\_submesh\_id syntax element.

rap\_nal\_unit\_flag when this flag is set to 0, this indicates that none of the NAL units in the sub-sample has nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2]. Value 1 indicates that all NAL units in the sub-sample have nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2].

submesh\_self\_contained\_flag equal to 0 indicates that some data e.g. prefix data for each submesh is present as specified in Basemesh sequence parameter set in ISO/IEC 23090-29 [1]. submesh\_self\_contained\_flag equal to 1 indicates that each sub-sample in the sample of the Basemesh Track is self-contained.

**3.2. Definition of a sub-sample for Submesh Track**

A sample in a submesh track may include a set of submeshes. Each submesh in a submesh track is a sub-sample. For the use of the SubSampleInformationBox in ISO/IEC 14496-12 [3] in a Basemesh submesh track, a sub-sample is defined based on the value of the flags field of the sub-sample information box as specified below. The presence of this box is mandatory in the case where more than one submesh is carried in the track.

If the SubSampleInformationBox is present in a submesh track containing data for more than one submesh, the codec\_specific\_parameters field in the box shall have the semantics defined below.

flags specifies the type of sub-sample information given in this box. If the value of flags is

0, then a sub-sample contains one Submesh NAL unit in a set of submeshes in a sample of a Submesh track. Values other than 0 are reserved for future use.

The subsample\_priority field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12 [3].

The discardable field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded

The codec\_specific\_parameters field of the SubSampleInformationBox is defined for Basemesh bitstream as follows:

if (flags == 0) {  
 unsigned int(1) rap\_nal\_unit\_flag;  
 unsigned int(1) submesh\_self\_contained\_flag;  
 bit(30) reserved = 0;  
}

The semantics of the above fields are:

rap\_nal\_unit\_flag When this flag is set to 0, this indicates that none of the NAL units in the sub-sample has nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2]. Value 1 indicates that all NAL units in the sub-sample have nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2].

submesh\_self\_contained\_flag equal to 0 indicates that some data e.g. prefix data for each submesh is present as specified in Basemesh sequence parameter set in ISO/IEC 23090-29 [2]. submesh\_self\_contained\_flag equal to 1 indicates that each sub-sample in the sample of the Submesh Track is self-contained.

**4.     Recommendations**

It is recommended to adopt the proposal on mesh and submesh sub-samples in section 3 of this contribution and to integrate it into the WD of ISO/IEC 23090-10 2nd edition.

1. (38.2)[VOL-SYS] On V3C single track encapsulation (m71356)

# Introduction

For single-track carriage, the current working draft of ISO/IEC 23090-10 2nd edition [1], proposes different V3CBitstreamSampleEntry (sections 7.2.1.2.2 and 7.3.3.2.2) which contain a variable number of decoder configuration boxes but none of these sample entries are exhaustive.

This contribution then proposes a full V3C decoder configuration record, to provide all the needed decoder configurations in a single box.

We also inform VOL-SYS group about other ISOBMFF tool under consideration that may apply to provide all decoder configurations for single-track carriage.

1. **Proposal for exhaustive V3C decoder configuration record**

The proposed modifications consist in:

* Adding a new section to specify the syntax and semantics of a full V3C decoder configuration record.
* Updating the section 6.2.2, to use new version of V3CConfigurationBox for enabling usage of the full V3C decoder configuration box in case of single-track carriage.
* Updating the sections 7.2.1.2.2 and 7.3.3.2.2 to consider the full V3C decoder configuration box.
  1. ***New section for full V3C decoder configuration record***

It is proposed to add the following new section

6.2.z V3C full decoder configuration record

6.2.z.1 Definition

The V3C full decoder configuration record provides decoder configurations for all the V3C sub-bitstreams of a V3C bitstream. The number of signalled decoder configurations shall be less than or equal to the number of V3C sub-bitstreams of the V3C bitstream.

The number of decoder configurations may be less than the number of V3C sub-bitstreams when has\_decoder\_mapping equals to 1. It enables to signal a decoder configuration record that applies to several V3C sub-bitstreams.

When the number of decoder configuration records is equal to the number of V3C sub-bitstreams of the V3C bitstream, the has\_decoder\_mapping shall be set to 0. The implicit declaration order for the decoder configuration records shall be:

* all non-video Codec decoders for basemesh (if any) in the order of the vuh\_atlas\_id,
* followed by all non-video Codec decoders for arithmetic displacement (if any) in the order of the vuh\_atlas\_id,
* followed by all video codec decoders for occupancy (if any) in the order of the vuh\_atlas\_id and vuh\_map\_index,
* followed by all video codec decoders for geometry (if any) in the order of the vuh\_atlas\_id,
* followed by all video codec decoders for attributes (if any) in the order of the vuh\_atlas\_id and vuh\_attribute\_index,
* followed by all video codec decoders for packed video (if any) in the order of the vuh\_atlas\_id.

6.2.z.2 Syntax

aligned(8) class V3CFullDecoderConfigurationRecord() {

bool has\_specific\_codec = FALSE;

V3CDecoderConfigurationRecord v3c\_config;

unsigned int(1) has\_decoder\_mapping;

unsigned int(8) config\_num;

unsigned int(7) ptl\_profile\_codec\_group\_idc;

unsigned int(8) ptl\_profile\_toolset\_idc;

PtlVideoCodecGroupIdc = ptl\_profile\_codec\_group\_idc & 0x0F;

PtlNonVideoCodecGroupIdc = (ptl\_profile\_codec\_group\_idc&0x70) >> 4;

for (int k=0; k < config\_num; k++) {

unsigned int(1) isVideoCodecGroup;

if (has\_decoder\_mapping) {

unsigned int(8) num\_subbitstreams;

for (i=0; i < num\_subbitstreams; i++) {

unsigned int(32) v3c\_unit\_header\_4bytes;

}

}

if (isVideoCodecGroup) {

if (PtlVideoCodecGroupIdc != 15) {

DecoderConfigurationRecord[PtlVideoCodecGroupIdc] decoderCfg;

} else { // the codec type from CCM SEI

has\_specific\_codec = TRUE;

}

}

else if (PtlNonVideoCodecGroupIdc != 7) {

DecoderConfigurationRecord[PtlNonVideoCodecGroupIdc, ptl\_profile\_toolset\_idc, v3c\_unit\_header\_4bytes] decoderCfg;

} else { // the codec type from CCM SEI

has\_specific\_codec = TRUE;

}

}

if (has\_specific\_codec == TRUE) {

unsigned int (8)[] ccm\_sei\_payload;

}

}

6.2.z.3 Semantics

has\_decoder\_mapping equal to 1 indicates that the mapping of a decoder configuration record to its corresponding V3C sub-bitstream(s) is explicit and provided in the mapping decoder to sub-bitstream mapping record. When equal to 0, it indicates that the mapping of a decoder configuration to its corresponding V3C sub-bitstream(s) is implicit.

config\_num indicates the number of decoder configuration records described in the V3CFullDecoderConfigurationRecord. The value of config\_num may be less than the number of V3C sub-bitstreams when has\_decoder\_mapping equals to 1. Otherwise, when has\_decoder\_mapping equal to 0, config\_num shall be equal to the number of V3C sub-bitstreams.

num\_subbitstreams indicates the number of sub-bitstreams that use the decoder configurations specified by decoderCfg.

v3c\_unit\_header\_4bytes provides the first 4 bytes of the V3C unit containing the V3C sub-bitstream data to which the decoder configuration applies.

decoderCfg contains a decoder configuration record providing setup information for a decoder of one or more sub-bitstreams of the V3C bitstream. The decoder configuration record is determined from V3C specification and its extensions: for video coded sub-bitstreams, from the Table A-1 in Annex A.3 of ISO/IEC 23090-5; for non-video coded sub-bitstreams from Table A-3 – “Available Non-Video CodecGroup profile components” in ISO/IEC 23090-5 or corresponding Tables in derived specifications (Part-29, for example). For video coded sub-bitstreams, the decoder config record is the one corresponding to the entry at index = PtlVideoCodecGroupIdc, according to the following mapping:

|  |  |  |
| --- | --- | --- |
| **PtlVideoCodecGroupIdc** | **4CC code** | DecoderConfigurationRecord |
| 0 | 'avc3' | AVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 1 | 'hev1' | HEVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 2 | 'hev1' | HEVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 3 | 'vvi1' | VVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 4 | 'hev1' | HEVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 5..14 | – | undefined |
| 15 | provided by external means or may be determined by component codec mapping SEI message (F.2.7) | undefined |

For non-video coded sub-bitstreams, the decoder config record is the one corresponding to the entry at index = PtlNonVideoCodecGroupIdc in one of the V3C or V-DMC table, depending on the value of the ptl\_profile\_toolset\_idc:

For ptl\_profile\_toolset\_idc < 128, it is undefined. For ptl\_profile\_toolset\_idc >= 128, the decoder config record is the one corresponding to the entry at index = PtlNonVideoCodecGroupIdc, according to the following mapping:

|  |  |  |
| --- | --- | --- |
| **Non-Video CodecGroup** | **PtlNonVideoCodecGroupIdc** | DecoderConfigurationRecord |
| BaseMesh | 0 | VDMCBaseMeshDecoderConfigurationRecord from ISO/IEC 23090-10 |
| BaseMesh, AC displacement | 1 | VDMCBaseMeshDecoderConfigurationRecord from ISO/IEC 23090-10 (for v3c\_unit\_header\_4bytes of type V3C\_BMD)  VDMCDisplacementDecoderConfigurationRecord from ISO/IEC 23090-10 (for v3c\_unit\_header\_4bytes of type V3C\_ADD) |
| Reserved | 2..6 | Undefined |
| MP4RA | 7 | Undefined |

ccm\_sei\_payload is the payload of the F.3.7 Component codec mapping SEI message semantics as defined in ISO/IEC 23090-5.

* 1. ***Modification of V3CConfigurationBox***

Proposed changes are yellow-highlighted.

6.2.2.1 Definition

A V3C decoder configuration box includes a V3CDecoderConfigurationRecord as defined in

subclause 6.2.1 or a V3CFullDecoderConfigurationRecord as defined in 6.2.z.

In this document, the value of version shall be equal to 0, except for single-track carriage with ‘vdm1’ or ‘vdmg’ sample entries where version=1 shall be used. Single-track carriage with ‘v3e1’ or ‘v3eg’ sample entries may used version=1.

6.2.2.2 Syntax

class V3CConfigurationBox(version) extends FullBox('v3cC', version = 0 or 1, 0) {

if (version == 0) {

V3CDecoderConfigurationRecord v3c\_config~~(version)~~;

}

else if (version == 1) {

V3CFullDecoderConfigurationRecord v3c\_config;

}

}

* 1. ***Modification of section for single-track carriage.***

Proposed changes are yellow- highlighted .

Proposed change 1, in 7.2.1.2.2 Syntax

aligned(8) class V3CBitstreamSampleEntry() extends VolumetricVisualSampleEntry (type) {

// type is 'v3e1' or 'v3eg'

V3CConfigurationBox v3c\_config(version=0 or 1);

//additional boxes

}

Proposed change 2, in 7.3.3.2.2 Syntax

aligned(8) class V3CBitstreamSampleEntry() extends VolumetricVisualSampleEntry ('vdm1' or 'vdmg') {

V3CConfigurationBox v3c\_config(version=1);

~~VDMCBaseMeshConfigurationBox bmesh\_config;~~

~~VDMCDisplacementConfigurationBox displ\_config; //optional~~

//additional boxes

}

1. **Conclusion**

In this contribution, we proposed for V3C single-track carriage, a sample entry containing a single configuration box ‘v3cC’, that exhaustively describes the decoder configurations required to decode a V3C bitstream.

We think our approach is more flexible than the Working Draft’s one [1], and could be used for new V3C extensions of single-track carriage. We then suggest the group to adopt this proposal.

Moreover, we encourage the group to also look at the split-sample approach under consider in File Format group [2].

1. **References**
2. MDS24428-WG03N1365, “*WD of ISO/IEC 23090-10:Carriage of Visual Volumetric Video-based Coding (V3C) Data (2nd edition)*” , MPEG#148, December 2024
3. MDS24528\_WG03\_N01385, “*Technologies under Consideration for ISO/IEC 14496-12 (ISOBMFF)*” , MPEG#148, December 2024
4. [VOL SYS] V-DMC Tile-based Component Track Group (m71212)

# Introduction

Basemesh and submesh components as an extension of V3C bitstream components has been included in part 5[1]. The submesh track group which was proposed in [2] to signal the relation between one or more atlas tile tracks and one or more submesh tracks has also been agreed. However, considering the case that more than one atlas tiles within one or more atlas tracks may be associated with more than one submesh tracks in separate submesh tracks, it is necessary to define the relation among atlases, atlas tiles and submeshes so that the file parser in the receiver can parse only selected atlas tile(s) with the related submesh track(s). In this contribution, we suggest to add atlas id in the SubMeshTrackGroupBox to support this case.

1. **Proposed Specification Text**

**[Replace the texts in the subclause 7.3.4.4.4]**

**7.3.4.4.4 Submesh track group**

**7.3.4.4.4.1 General**

Each submesh track can include one or more submeshes and these submeshes are related to one or more atlas tiles carried in one or more atlas tile tracks. To signal the relation between atlas tile tracks and associated submesh tracks, a track group SubMeshTrackGroupBox extends TrackGroupTypeBox defined in ISO/IEC 14496-12, is defined.

**7.3.4.4.4.2 Syntax**

aligned(8) class SubMeshTrackGroupBox extends TrackGroupTypeBox('smtg') {

// track\_group\_id is inherited from TrackGroupTypeBox

unsigned int(16) num\_tiles;

for(int i=0; i < num\_tiles; i++) {

unsigned int(16) tile\_id;

unsigned int(16) num\_submeshes;

for(int j=0; j < num\_submeshes; j++) {

unsigned int(16) submesh\_id;

}

}

}

**7.3.4.4.4.3 Semantics**

num\_tiles indicates the number of atlas tiles associated with this track group.

tile\_id specifies the atlas tile ID of an atlas tile associated with the submeshes for this track group instance. The value of tile\_id is equal to value of afti\_tile\_id syntax element in atlas frame tile information, defined in ISO/IEC FDIS 23090-5

num\_submeshes indicates the number of submeshes associated with the the atlas tile.

submesh\_id is the identifier for the submesh for this track group instance. The value of submesh\_id is equal to the value of the corresponding bmsi\_submesh\_id syntax element in bmesh\_submesh\_information(), defined in ISO/IEC 23090-29:Annex H.

**[with the following texts – highlighted in yellow for updated texts]**

**7.3.4.4.4 Submesh track group**

**7.3.4.4.4.1 General**

Each submesh track can include one or more submeshes and these submeshes are related to one or more atlas tiles carried in one or more corresponding atlas tracks or atlas tile tracks. To signal the relation between atlas tiles and associated submesh tracks, a track group SubMeshTrackGroupBox extends TrackGroupTypeBox defined in ISO/IEC 14496-12, is defined.

**7.3.4.4.4.2 Syntax**

aligned(8) class SubMeshTrackGroupBox extends TrackGroupTypeBox('smtg') {

// track\_group\_id is inherited from TrackGroupTypeBox

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;

unsigned int(16) num\_submeshes;

for(int j=0; j < num\_submeshes; j++) {

unsigned int(16) submesh\_id;

}

}

}

**7.3.4.4.4.3 Semantics**

num\_tiles indicates the number of atlas tiles associated with this track group.

atlas\_id indicates the atlas ID associated with the tile\_id.

tile\_id specifies the atlas tile ID of an atlas tile associated with the submeshes for this track group instance. The value of tile\_id is equal to value of afti\_tile\_id syntax element in atlas frame tile information, defined in ISO/IEC FDIS 23090-5

num\_submeshes indicates the number of submeshes associated with the the atlas tile.

submesh\_id is the identifier for the submesh for this track group instance. The value of submesh\_id is equal to the value of the corresponding bmsi\_submesh\_id syntax element in bmesh\_submesh\_information(), defined in ISO/IEC 23090-29:Annex H.

1. **Conclusion**

We suggest our proposed text to be included into the 2nd edition of ISO/IEC 23090-10.

1. **References**

[1] Coded representation of immersive media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)

[2] m68741 [VOL SYS]Dynamic Mesh Bitstream Format in ISOBMFF -<https://git.mpeg.expert/MPEG/Systems/PCC-SYS/V-PCC/-/issues/203>

1. [VolSys] On associating submesh to atlas tile in V-DMC carriage (m71422)
2. **Introduction**

This contribution highlights the association of atlas tiles in the atlas bitstream with corresponding submeshes in the basemesh bitstream for reconstruction of 3D meshes. This contribution proposes necessary information to signal this association in atlas tracks and atlas tile tracks and provides information on appropriate grouping of relevant tracks for partial access in V-DMC content carried in ISOBMFF containers.

1. **Background**

In V-DMC, specified in ISO/IEC 23090-29 [1], the atlas sub-bitstream provides the necessary information for the reconstruction of the basemesh or a set of submeshes in the basemesh. A V-DMC decoder requires information from the atlas bitstream to reconstruct the basemesh. A basemesh is sub-divided into submeshes which allows for efficient access and independent decoding capability. An atlas tile in the atlas bitstream stores information for a submesh or a set of submeshes to enable 3D reconstruction of the basemesh.

A new mesh patch type is introduced in ISO/IEC 23090-29 [1] in the atlas bitstream. A mesh patch contains necessary information for mesh reconstruction, such as the submesh identifier to associate the submesh with the mesh patch, displacement information, etc. Each mesh patch is associated with one submesh.

1. **Proposal**

In order to associate the atlas information stored in an atlas track or an atlas tile track and the submesh, the following items are proposed.

1. V3C Atlas Tile Submesh Configuration Box
2. Atlas tile and submesh track grouping example
   1. **V3C Atlas Submesh Configuration Box**

A V3CAtlasSubmeshConfigurationBox provides information on the submeshes associated with each atlas tile in a V3C atlas track or a V3C atlas tile track in the case of V-DMC content. The V3CAtlasSubmeshConfigurationBox is defined as follows:

Box Type: 'v3SC'  
Container: V3CAtlasSampleEntry or V3CAltasSampleEntry  
Mandatory: Yes (in the case of V-DMC)  
Quantity: Zero or One

aligned(8) class V3CAtlasSubmeshConfigurationBox extends FullBox(`v3SC`, version = 0, 0) {

unsigned int(16) num\_tiles;

for(int i=0; i < num\_tiles; i++){  
 unsigned int(16) tile\_id;  
 unsigned int(16) num\_submeshes\_minus1;  
 for (j = 0; j < num\_submeshes\_minus1 + 1; j++) {  
 unsigned int(16) submesh\_id;  
 }

}

}

The semantics of V3CAtlasSubmeshConfigurationBox are as follows:

num\_tiles indicates the number of atlas tiles contained in this track.

tile\_id specifies the atlas tile ID of the atlas tile contained in the track. The value of tile\_id is equal to value of afti\_tile\_id syntax element in the atlas frame tile information, as defined in ISO/IEC 23090-5 [2].

num\_submeshes\_minus1 plus 1 specifies the number of submeshes corresponding to an atlas tile with atlas ID tile\_id.

submesh\_id indicates the identifier for the submesh carried in the atlas tile track.

V3CAtlasSubmeshConfigurationBox shall be present for when the profile toolset indicator in the V3C parameter sets corresponds to a value reserved for V-DMC.

The usage of the V3CAtlasSubmeshConfigurationBox is illustrated in the following two examples (highlighted in yellow). The first example shows the use of V3CAtlasSubmeshConfigBox in V3CAtlasSampleEntry().

aligned(8) class V3CAtlasSampleEntry() extends VolumetricVisualSampleEntry (type) {

// type is 'v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag'

V3CConfigurationBox config;

V3CUnitHeaderBox unit\_header;

V3CAtlasSubmeshConfigurationBox submesh\_info; // optional

}

The second example shows the use of V3CAtlasSubmeshConfigurationBox in V3CAtlasTileSampleEntry().

aligned(8) class V3CAtlasTileSampleEntry() extends VolumetricVisualSampleEntry ('v3t1') {

V3CAtlasTileConfigurationBox tile\_info;  
 V3CAtlasSubmeshConfigurationBox submesh\_info; // optional

}

* 1. **Atlas tile and submesh track grouping example**

This section is informative to illustrate an example an atlas tile track which contains information for a submesh or a set of submeshes contained in the submesh track(s) with the respective submeshID(s) as shown in Figure 1. One important aspect to consider if that a submesh track shall not be grouped with more than one Atlas tile track.

**A screenshot of a black and white diagram

Description automatically generated**

Figure 1 Atlas tile and submesh track grouping

1. **Recommendations**

It is recommended to adopt the proposal in Section 3 to the WD of ISO/IEC 23090-10 2nd edition.

**References**

1. Text of ISO/IEC DIS 23090-29 Video-based mesh coding, Online, <https://dms.mpeg.expert/doc_end_user/documents/148_Kemer/wg11/MDS24469_WG07_N01027.zip>
2. ISO/IEC 23090-5:2023, Information technology — Coded representation of immersive media, Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC), Online, https://www.iso.org/standard/83535.html
3. ISO/IEC, "ISO/IEC 14496-12:2022 Information technology - Coding of audio-visual objects Part 12: ISO bases media file format," 2022. [Online]. Available: https://www.iso.org/standard/83102.html
4. [VolSys] On sub-sample usage for submesh in V-DMC carriage (m71420)
5. **Introduction**

At the MPEG#148 meeting in Antalya, method to store submeshes in sub-samples of a basemesh and/or submesh tracks was proposed in m70103 [1].

This contribution provides a background on the need to indicate specific parameter for submeshes carried in sub-sample of basemesh. The use of sub-sample for carrying submesh in submesh track is also described. Section 2 provides a background on the importance of the specific parameters. Section 3 proposes techniques to indicates the specific parameters for submeshes carried in sub-samples in a basemesh and submesh track.

1. **Background**

A Basemesh bitstream format specified in Annex H of ISO/IEC 23090-29 [2] employs two different sub-codecs to code the mesh sequence, namely an intra mesh codec and an inter mesh codec. Some prefix data in the intra/inter mesh payload may be replicated across different submeshes.

A syntax element is specified in ISO/IEC 23090-29 [2] bmsps\_codec\_specific\_parameters\_present\_flag. When bmsps\_codec\_specific\_parameters\_present\_flag is equal 1, the basemesh sequence parameter set can carry some prefix data of the coded mesh payload. In order to fully decode the mesh payload, the prefix data is concatenated with the submesh data carried as described in H.9.4.5.2.1. of ISO/IEC 23090-29 [2]. In case, the submesh payload is sent to the decoder for decoding with the prefix data, the decoder will crash.

Therefore, it is important to determine whether the subsample is self-contained or an additional procedure to concatenate the prefix data with the sub-sample data is needed.

1. **Proposal**
   1. **Adding specific parameters in SubSampleInformationBox in Basemesh Track**

A sample in a Basemesh Track may include a set of submeshes. Each submesh in a Basemesh Track is a sub-sample. For the use of the SubSampleInformationBox in ISO/IEC 14496-12 [3] in a Basemesh track, a sub-sample is defined based on the value of the flags field of the sub-sample information box as specified below. The presence of this box is mandatory in the case where more than one submesh is carried in the track. If the sub sample information box is present in a Basemesh track containing submesh data for more than one submesh, the codec\_specific\_parameters field in the box shall have the semantics defined here.

1. 7.3.4.4.3.5. Basemesh track and submesh track sub-sample

The codec\_specific\_parameters field of the SubSampleInformationBox is defined for Basemesh track as follows:

if (flags == 0) {  
 unsigned int(16) submesh\_id;  
 unsigned int(1) rap\_nal\_unit\_flag;  
 unsigned int(1) submesh\_self\_contained\_flag;  
 bit(14) reserved = 0;  
}

The semantics of the above fields are:

submesh\_id indicates the identifier for a submesh contained in this sub-sample of the basemesh track. submesh\_id shall be equal to the submesh identifier signaled in the syntax element bmsi\_submesh\_id syntax element.

rap\_nal\_unit\_flag when this flag is set to 0, this indicates that none of the NAL units in the sub-sample has nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2]. Value 1 indicates that all NAL units in the sub-sample have nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2].

submesh\_self\_contained\_flag equal to 0 indicates that some data e.g. prefix data for each submesh is present as specified in Basemesh sequence parameter set in ISO/IEC 23090-29 [1]. submesh\_self\_contained\_flag equal to 1 indicates that each sub-sample in the sample of the Basemesh Track is self-contained.

* 1. **Codec specific parameters for Submesh track**

A sample in a submesh track may include a set of submeshes. Each submesh in a submesh track is a sub-sample. For the use of the SubSampleInformationBox in ISO/IEC 14496-12 [3] in a Basemesh submesh track, a sub-sample is defined based on the value of the flags field of the sub-sample information box as specified below. The presence of this box is mandatory in the case where more than one submesh is carried in the track.

If the SubSampleInformationBox is present in a submesh track containing data for more than one submesh, the codec\_specific\_parameters field in the box shall have the semantics defined below.

flags specifies the type of sub-sample information given in this box. If the value of flags is

0, then a sub-sample contains one Submesh NAL unit in a set of submeshes in a sample of a Submesh track. Values other than 0 are reserved for future use.

The subsample\_priority field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12 [3].

The discardable field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded

The codec\_specific\_parameters field of the SubSampleInformationBox is defined for Basemesh bitstream as follows:

if (flags == 0) {  
 unsigned int(1) rap\_nal\_unit\_flag;  
 unsigned int(1) submesh\_self\_contained\_flag;  
 bit(30) reserved = 0;  
}

The semantics of the above fields are:

rap\_nal\_unit\_flag When this flag is set to 0, this indicates that none of the NAL units in the sub-sample has nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2]. Value 1 indicates that all NAL units in the sub-sample have nal\_unit\_type equal to BNAL\_IDR\_W\_RADL, BNAL\_IDR\_N\_LP, or BNAL\_CRA as specified in ISO/IEC 23090-29 Annex H [2].

submesh\_self\_contained\_flag equal to 0 indicates that some data e.g. prefix data for each submesh is present as specified in Basemesh sequence parameter set in ISO/IEC 23090-29 [2]. submesh\_self\_contained\_flag equal to 1 indicates that each sub-sample in the sample of the Submesh Track is self-contained.

1. **Recommendations**

It is recommended to adopt the proposal in section 3 to integrate it into the WD of ISO/IEC 23090-10 3rd edition.

**References**

1. [VolSys] On submesh sub-samples for V-DMC, Online, <https://dms.mpeg.expert/doc_end_user/documents/148_Kemer/wg11/m70103-v1-m70103_submesh_carriage_v1.zip>
2. Text of ISO/IEC DIS 23090-29 Video-based mesh coding, Online, https://dms.mpeg.expert/doc\_end\_user/documents/148\_Kemer/wg11/MDS24469\_WG07\_N01027.zip
3. ISO/IEC, "ISO/IEC 14496-12:2022 Information technology - Coding of audio-visual objects Part 12: ISO bases media file format," 2022. [Online]. Available: https://www.iso.org/standard/83102.html
4. [VolSys] On sub-samples for submesh track (m72199)

|  |  |
| --- | --- |
|  |  |

1. **Introduction**

This contribution recommends considering sub-sample encapsulation scheme for submeshes in submesh track.

1. **Background**

At MPEG meeting 149, contribution m71420 [1] proposed use of sub-samples to store submeshes in submesh track. Since, an atlas tile track may be associated with multiple sub-meshes as described in contribution m72198[2] it is desirable to encapsulate those sub-meshes into one submesh track as sub-samples.

A basemesh is a coarse representation of the original mesh sequence. The basemesh is further partitioned into submeshes. In our understanding, the effective bitrate for each submesh is effective low to be considered as an independent track. Additionally in the case of viewport-based streaming scenario, if a submesh track is carrying only one submesh, then this may cause request of multiple submesh track(s) by the client for an atlas tile track. Thus, causing an overload of requests and potentially causing track management issue.

1. **Proposed changes to WD**

Changes to the working draft of ISO/IEC 23090-10 [3] are highlighted with yellow.

1. 7.3.4.4.3.5. Basemesh track and submesh track sub-sample

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

— If the value of the flags field is 0, each sub-sample contains one Submesh NAL unit in the group of submeshes carried in a sample of the Basemesh track or the Submesh track.

— The values other than 0 are reserved for future use.

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

The discardable field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded.

When the SubSampleInformationBox is present in a basemesh track, the codec\_specific\_parameters field in the box shall have the semantics as follows:

if (flags == 0) {  
 unsigned int(16) submesh\_id;

bit(16) reserved = 0;

}

The semantics of the above fields are:

submesh\_id indicates the identifier for a submesh contained in this sub-sample of the basemesh track or submesh track. submesh\_id shall be equal to the submesh identifier signaled in the syntax element bmsi\_submesh\_id syntax element.

Some common parameters for submeshes in sub-samples can be signaled in the SubSampleInformationBox in the basemesh track or submesh track. Alternative efficient representation of SubSampleInformationBox as specified in ISO/IEC 14496-12 Amd1 [4] can also be considered. Potential candidate for the common parameters can be submesh\_self\_contained\_flag flag which indicates that some data e.g. prefix data for each submesh is present as specified in Basemesh Sequence Parameter Set in ISO/IEC 23090-29[5].

1. **Conclusion**

It is recommended to adopt the changes proposed in Section 3 in the next release of the working draft of ISO/IEC 23090-10[3].

1. **References**
2. m71420, [VolSys] On sub-sample usage for submesh in V-DMC carriage”, MPEG#149, Geneva, Türkiye, Online, <https://dms.mpeg.expert/doc_end_user/documents/149_Geneva/wg11/m71420-v1-m71420.zip>
3. m72198, [VolSys] Information on sub-meshes and atlas tiles”, MPEG#150, Online
4. WD of ISO/IEC 23090-10 2nd edition Carriage of visual volumetric video-based coding data , MPEG149, https://dms.mpeg.expert/doc\_end\_user/documents/149\_Geneva/wg11/MDS24734\_WG03\_N01434.zip
5. Text of ISO/IEC 14496-12 8th edition FDAM 1 Support for T.35, original sample duration and other improvements, Online, <https://dms.mpeg.expert/doc_end_user/documents/148_Kemer/wg11/MDS24526_WG03_N01383.zip>
6. WG07N01099, “Technologies for Video-based mesh coding “, MPEG#149, Geneva, Switzerland, Online, <https://dms.mpeg.expert/doc_end_user/documents/149_Geneva/wg11/MDS24820_WG07_N01099.zip>
7. [VolSys] Information on submeshes and atlas tiles (m72198)

|  |  |
| --- | --- |
|  |  |

1. **Introduction**

This contribution shares information on the relationship between sub-meshes and atlas tiles as specified in ISO/IEC 23090-29 [1]. At the MPEG#149 meeting in Geneva, a method to express the association of atlas tile and sub-meshes was proposed in m71420 [2].

1. **Background**

This section provides some background on V-DMC and the relationship between a sub-mesh and an atlas tile. This section also draws inspiration from the relationship between the basemesh and atlas bitstreams in an encapsulation scheme.

* 1. **ISO/IEC 23090-29**

A V-DMC bitstream includes components such as the atlas component, the basemesh component, the displacement component, and, optionally, a number of attribute components. The components are indicated by different V3C unit types, namely: V3C\_AD, V3C\_BMD, V3C\_GVD/V3C\_ADD, and V3C\_AVD, respectively. A basemesh is a mesh structure containing possibly a reduced number of mesh vertices which is enhanced using displacement coefficients coded in the displacement bitstream.

As specified in ISO/IEC 23090-5 [3]an atlas frame is partitioned into patches. A collection of patches can be combined into an atlas tile.

ISO/IEC 23090-29 [1] altered the definition of an atlas frame as follows:

*An atlas frame 2D rectangular array of atlas samples onto which patches (ISO/IEC 23090-5(4E):2025:3.91) are projected and additional information related to the patches (ISO/IEC 23090-5(4E):2025:3.91), corresponding to a volumetric frame (ISO/IEC 23090-5(4E):2025:3.142) and a list of meshpatches (3.9) and additional information related to the meshpatches (3.9), corresponding to a volumetric frame (ISO/IEC 23090-5(4E):2025:3.142)*

A meshpatch is an element of an atlas associated with basemesh information. The mesh patch codes data in an atlas which can enable the conversion of the basemesh into the reconstructed mesh. The mesh patch data indicates the associated submeshID in mdu\_submesh\_id specified in a meshpatch in an atlas tile. The value of mdu\_submesh\_id shall be one of the values indicated by afmi\_submesh\_id and is in the range 0 to 65535, inclusive. The value of mdu\_submesh\_id is unique as per the bitstream conformance requirements specified in sub-clause 8.4.6.2.5. in ISO/IEC 23090-29 [1].

*It is a requirement of atlas bitstream conformance that afmi\_submesh\_id[ i ] shall not be equal to afmi\_submesh\_id[ j ] for all i != j.*

Additionally, the specification mentions the following bitstream conformance statements.

*In bitstreams conforming to this version of this document a coded atlas frame shall not contain two or more meshpatch data units, within geometry tiles with ath\_type equal to P\_TILE or I\_TILE, that indicate the same LOD index and the same submesh ID.*

*In bitstreams conforming to this version of this document a coded atlas frame shall not contain two or more meshpatch data units, within tiles with ath\_type equal to P\_TILE\_ATTR or I\_TILE\_ATTR, that indicate the same LOD index and the same submesh ID.*

According to the statements, a meshpatch data unit in an atlas tile with ath\_type equal to P\_TILE or I\_TILE, P\_TILE\_ATTR, or I\_TILE\_ATTR indicate a unique submeshID. Therefore, an atlas tile carrying meshpatch data units corresponds to a unique set of submeshIDs.

**2.2. ISO/IEC 23090-10**

In ISO/IEC 23090-10 [4] two methods of the storage for V3C bitstream 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. In multi-track encapsulation, three types of tracks are defined: V3C atlas track, V3C atlas tile track, and V3C video component track. A V3C atlas track is a volumetric visual track containing V3C atlas bitstream. A V3C atlas tile track is a volumetric visual track containing portion of V3C atlas bitstream corresponding to one or more tiles in case of multi-track container. A V3C atlas track references zero or more V3C atlas tile tracks or zero or more V3C video component tracks.

To encapsulate a V-DMC bitstream, the track definition for the basemesh component and displacement components are necessary to be defined. As described in Section 2.1, an atlas tile is associated with a unique set of sub-meshes in a basemesh bitstream. Therefore, it is desirable to exhibit this relationship when encapsulating a V-DMC bitstream in an ISOBMFF container using the multi-track encapsulation scheme.

For cases such as partial delivery, a client may request content which corresponds to the region-of-interest of the user. The content may be covered by an atlas tile which is further related to a unique set of submeshes in the basemesh bitstream. The information about the association of an atlas tile in an atlas track/atlas tile track to submeshes in a basemesh/submesh track is useful for requesting the appropriate data to cover the region-of-interest of the user.

1. **Proposal**

For V-DMC, it is important to consider the association between the atlas tile in the atlas bitstream and the sub-meshes in the basemesh bitstream as specified in ISO/IEC 23090-29 [1]. Since the atlas track in ISO/IEC 23090-10[4] is the entry point in the file, the association information related to the atlas tile and sub-meshes should therefore be expressed.

In order to associate the information stored in mesh patch data unit carried in an atlas track or an atlas tile track and the submesh, a V3CAtlasSubmeshConfigurationBox is proposed.

* 1. **V3C Atlas Submesh Configuration Box**

A V3CAtlasSubmeshConfigurationBox provides information on the submeshes associated with each atlas tile in a V3C atlas track or a V3C atlas tile track in the case of V-DMC content. The V3CAtlasSubmeshConfigurationBox is defined as follows:

Box Type: 'v3SC'  
Container: V3CAtlasSampleEntry or V3CAltasTileSampleEntry  
Mandatory: Yes (in the case of V-DMC)  
Quantity: Zero or One

aligned(8) class V3CAtlasSubmeshConfigurationBox extends FullBox('v3SC', version = 0, 0) {

unsigned int(16) num\_tiles;

for(int i=0; i < num\_tiles; i++){  
 unsigned int(16) tile\_id;  
 unsigned int(16) num\_submeshes\_minus1;  
 for (j = 0; j < num\_submeshes\_minus1 + 1; j++) {  
 unsigned int(16) submesh\_id;  
 }

}

}

The semantics of V3CAtlasSubmeshConfigurationBox are as follows:

num\_tiles indicates the number of atlas tiles contained in this track.

tile\_id specifies the atlas tile ID of the atlas tile contained in the track. The value of tile\_id is equal to value of afti\_tile\_id syntax element in the atlas frame tile information, as defined in ISO/IEC 23090-5 [2].

num\_submeshes\_minus1 plus 1 specifies the number of submeshes corresponding to an atlas tile with atlas ID tile\_id.

submesh\_id indicates the identifier for the submesh carried in the atlas tile track.

V3CAtlasSubmeshConfigurationBox shall be present for when the profile toolset indicator in the V3C parameter sets corresponds to a value reserved for V-DMC.

* 1. **Examples**

The usage of the V3CAtlasSubmeshConfigurationBox is illustrated in the following two examples (highlighted in yellow). The first example shows the use of V3CAtlasSubmeshConfigBox in V3CAtlasSampleEntry().

aligned(8) class V3CAtlasSampleEntry() extends VolumetricVisualSampleEntry (type) {

// type is 'v3c1', 'v3cg', 'v3cb', 'v3a1', or 'v3ag'

V3CConfigurationBox config;

V3CUnitHeaderBox unit\_header;

V3CAtlasSubmeshConfigurationBox submesh\_info; // optional

}

The second example shows the use of V3CAtlasSubmeshConfigurationBox in V3CAtlasTileSampleEntry().

aligned(8) class V3CAtlasTileSampleEntry() extends VolumetricVisualSampleEntry ('v3t1') {

V3CAtlasTileConfigurationBox tile\_info;  
 V3CAtlasSubmeshConfigurationBox submesh\_info; // optional

}

1. **Recommendations**

It is recommended to adopt the proposal in Section 3 for the next release of the Working Draft of ISO/IEC 23090-10 2nd edition[5].

**References**

1. WG07N01099, “Technologies for Video-based mesh coding “, MPEG#149, Geneva, Switzerland, Online, <https://dms.mpeg.expert/doc_end_user/documents/149_Geneva/wg11/MDS24820_WG07_N01099.zip>
2. m71422, On associating atlas /atlas tile track to submesh in V-DMC carriage, MPEG#148, Geneva, <https://dms.mpeg.expert/doc_end_user/documents/149_Geneva/wg11/m71422-v2-m71422_v2.zip>
3. Information technology — Coded representation of immersive media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC), Online, <https://www.iso.org/standard/73025.html>
4. ISO/IEC 23090-10:2022 Information technology — Coded representation of immersive media Part 10: Carriage of visual volumetric video-based coding data, Online, <https://www.iso.org/standard/78991.html>
5. WD of ISO/IEC 23090-10 2nd edition Carriage of visual volumetric video-based coding data , MPEG149, https://dms.mpeg.expert/doc\_end\_user/documents/149\_Geneva/wg11/MDS24734\_WG03\_N01434.zip
6. (37.1)[VOL-SYS] On V3C single track encapsulation (m72403)

# Introduction

This contribution deals with V3C bitstream sample entry, currently defined in sections 7.2.1.2.2 and 7.3.2.2.2 of ISO/IEC 23090-10 2nd edition [1].

We propose a simplification of the exhaustive decoder configuration information in V3C single track initially proposed in [2].

Second version of this document proposes to reconsider the definition of V3C Bitstream sample for single track encapsulation. This is based on the clarifications of V3C Composition Unit obtained from WG7 experts related to the input contribution m72342 [4]. The new section 4 proposes a new sample definition, for discussion with VOL-SYS group.

1. **Reminder about initial proposal for exhaustive V3C decoder configuration record**

In [2], we proposed a V3C configuration box describing the decoder configuration records for all sub-bitstreams with a mapping of these configuration records to the sub-bitstream(s) it applies. The mapping included all kinds of configuration records except the one of the atlas (i.e video and non-video coded sub-bitstreams).

When discussed during MPEG#149 meeting, some parameters were questioned about their usefulness. (See [Issue#218](https://git.mpeg.expert/MPEG/Systems/PCC-SYS/V-PCC/-/issues/218#note_115826))

The next section proposes two versions for a simplified version of the initial proposal from [2].

1. **Updated proposal**

We propose 2 options depending on whether we describe decoder configuration information per sub-bitstream type (multiple configuration boxes) or not (single configuration box).

We think option 1 is generic and extensible while option 2 is closer to the current Working Draft, mandating standardization of new configuration boxes each time V3C is extended with new V3C unit types. Both can apply to V3C and V-DMC.

* 1. ***Single configuration box approach***

It is proposed to extend the definition of V3CDecoderConfigurationRecord with a new version providing all the required decoder configuration records, with their mapping to sub-bitstream(s), as follows:

Proposed Change 1.1 => Extend 6.2.1.2 – Syntax with the yellow-highlighted text below

aligned(8) class V3CDecoderConfigurationRecord(unsigned int version) {

if(version == 0 || version == 1){ // unchanged from Part-5 section 6.2.1.2

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;

bit(8) 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;

bit(8) setup\_unit[setup\_unit\_length];

}

}

}

if(version == 1){ // exhaustive list of decoder config record with mapping

unsigned int(7) additional\_config\_num;

unsigned int(1) has\_decoder\_mapping;

for (int k=0; k < additional\_config\_num; k++) {

unsigned int(1) isVideoCodecGroup;

unsigned int(5) vuh\_unit\_type; // the value to navigate in Tables below; coming from the V3C unit containing one of the xPS stored in the dec Config.

bit(2) reserved = 0;

if (isVideoCodecGroup) {

if (PtlVideoCodecGroupIdc != 15) {

DecoderConfigurationRecord[PtlVideoCodecGroupIdc] decoderCfg;

}

}

else {

if (PtlNonVideoCodecGroupIdc != 7) {

DecoderConfigurationRecord[PtlNonVideoCodecGroupIdc, ptl\_profile\_toolset\_idc, vuh\_unit\_type] decoderCfg;

}

}

if (has\_decoder\_mapping) {

unsigned int(8) num\_subbitstreams;

for (i=0; i < num\_subbitstreams; i++) {

unsigned int(32) v3c\_unit\_header\_4bytes;

}

}

}

}

Proposed Change 1.2 => Add the following semantics to the ones in section 6.2.1.3 - Semantics:

additional\_config\_num indicates the number of decoder configuration records for sub-bitstreams described in the V3CDecoderConfigurationRecord, except the one for the atlas sub-bitstream. The value of additional\_config\_num may be less than the number of V3C sub-bitstreams when has\_decoder\_mapping equals to 1. Otherwise, when has\_decoder\_mapping equal to 0, additional\_config\_num shall be equal to the number of V3C sub-bitstreams.

has\_decoder\_mapping equal to 1 indicates that the mapping of a decoder configuration record to its corresponding V3C sub-bitstream(s) is explicit and is provided in the loop of additional configuration records for sub-bitstreams. When equal to 0, it indicates that the mapping of a decoder configuration is unknown.

num\_subbitstreams indicates the number of sub-bitstreams that use the decoder configurations specified by decoderCfg.

v3c\_unit\_header\_4bytes provides the first 4 bytes of the V3C unit containing the V3C sub-bitstream data to which the decoder configuration record applies.

decoderCfg contains a decoder configuration record providing setup information for a decoder of one or more sub-bitstreams of the V3C bitstream. The decoder configuration record is determined from V3C specification and its extensions: for video coded sub-bitstreams, from the Table A-1 in Annex A.3 of ISO/IEC 23090-5; for non-video coded sub-bitstreams from Table A-3 – “Available Non-Video CodecGroup profile components” in ISO/IEC 23090-5 or corresponding Tables in derived specifications (Part-29, for example). For video coded sub-bitstreams, the decoder config record is the one corresponding to the entry at index = PtlVideoCodecGroupIdc, according to the following mapping:

|  |  |  |
| --- | --- | --- |
| **PtlVideoCodecGroupIdc** | **4CC code** | DecoderConfigurationRecord |
| 0 | 'avc3' | AVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 1 | 'hev1' | HEVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 2 | 'hev1' | HEVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 3 | 'vvi1' | VVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 4 | 'hev1' | HEVCDecoderConfigurationRecord from ISO/IEC 14496-15 |
| 5..14 | – | undefined |
| 15 | provided by external means or may be determined by component codec mapping SEI message (F.2.7) | undefined |

For non-video coded sub-bitstreams, the decoder config record is the one corresponding to the entry at index = PtlNonVideoCodecGroupIdc in one of the V3C or V-DMC table, depending on the value of the ptl\_profile\_toolset\_idc:

For ptl\_profile\_toolset\_idc < 128, it is undefined. For ptl\_profile\_toolset\_idc >= 128, the decoder config record is the one corresponding to the entry at index = PtlNonVideoCodecGroupIdc, according to the following mapping:

|  |  |  |
| --- | --- | --- |
| **Non-Video CodecGroup** | **PtlNonVideoCodecGroupIdc** | DecoderConfigurationRecord |
| BaseMesh | 0 | VDMCBaseMeshDecoderConfigurationRecord from ISO/IEC 23090-10 |
| BaseMesh, AC displacement | 1 | VDMCBaseMeshDecoderConfigurationRecord from ISO/IEC 23090-10 (for vuh\_unit\_type = V3C\_BMD)  VDMCDisplacementDecoderConfigurationRecord from ISO/IEC 23090-10 (for vuh\_unit\_type = V3C\_ADD) |
| Reserved | 2..6 | Undefined |
| MP4RA | 7 | Undefined |

* 1. ***Multiple configuration boxes approach***

It is proposed to extend the definition of V3CBitstreamSampleEntry for V3C single track to list all the required decoder configuration boxes with mapping information of these configurations to their corresponding sub-bitstreams.

Proposed Change 2.1 => Add a new section for the definition of V3CVideoConfigurationBox:

6.x.x Syntax

class V3CVideoConfigurationBox(codec\_type) extends FullBox (codec\_type, version=0, 0) {

switch (codec\_type) {

case ‘avcC’:

AVCDecoderConfigurationRecord video\_config;

break;

case ‘hvcC’:

HEVCDecoderConfigurationRecord video\_config;

break;

case ‘vvcC’:

VVCDecoderConfigurationRecord video\_config;

break;

default:

break;

}

}

6.x.x Semantics

video\_config contains a decoder configuration record providing setup information for a video decoder of one or more sub-bitstreams of the V3C bitstream. The type of the video decoder configuration record is determined from V3C specification and its extensions: from the Table A-1 in Annex A.3 of ISO/IEC 23090-5.

Proposed Change 2.2 => Update the Section 7.2.1.2.2 Syntax (for 'v3e1', 'v3eg' sample entry types) with the yellow-highlighted text

aligned(8) class V3CBitstreamSampleEntry() extends VolumetricVisualSampleEntry (type) {

// type is 'v3e1' or 'v3eg'

V3CConfigurationBox v3c\_config;

// zero or more configuration for video sub-bitstreams

V3CVideoConfigurationBox video\_config1(coding\_type1=’avcC’);

V3CVideoConfigurationBox video\_config2(coding\_type2=’hvcC’);

DecoderConfigurationMappingBox mapping; //optional

//additional boxes

}

Proposed Change 2.3 => Update the Section 7.3.2.2.2 Syntax with the yellow-highlighted text

aligned(8) class V3CBitstreamSampleEntry()

extends VolumetricVisualSampleEntry ('vdm1' or 'vdmg') {

V3CConfigurationBox v3c\_config; // ‘v3cC’

VDMCBaseMeshConfigurationBox bmesh\_config; // ‘vbmC’

VDMCDisplacementConfigurationBox displ\_config; // optional ‘vdcC’

// zero or more configuration boxes for video sub-bitstreams

V3CVideoConfigurationBox video\_config1(coding\_type2=’hvcC’);

V3CVideoConfigurationBox video\_config2(coding\_type2=’hvcC’);

DecoderConfigurationMappingBox mapping; //optional

//additional boxes

}

The number of signalled decoder configurations in a V3CBistreamSampleEntry shall be less than or equal to the number of V3C sub-bitstreams of the V3C bitstream. When it is less than the number of V3C sub-bitstreams, the DecoderConfigurationMappingBox shall be present to indicate which decoder configuration applies to which sub-bitstream(s).

Proposed Change 2.x => Add a new section 6.x.x to define the decoder configuration to sub-bitstream mapping

6.x.x.1 Definition

Box Types: 'dssm'

Container: V3CBitstreamSampleEntry

Mandatory: No

Quantity: Zero or one (per sample entry)

The DecoderConfigurationMappingBox provides associations or mapping between decoder configuration information and sub-bitstreams to wich a decoder configuration applies

Proposed Change 2.x => Add a new section 6.x.x.2 to define the syntax for decoder configuration to sub-bitstream mapping

aligned(8) class DecoderConfigurationMappingBox extends FullBox(‘dssm’, version=0, flags=0) {

unsigned int(8) nb\_associations;

for (i=0; i < nb\_associations; i++) {

unsigned int(8) decoder\_cfg\_index;

unsigned int(8) num\_subbitstreams;

for (j=0; j < num\_subbitstreams; j++)

unsigned int(32) v3c\_unit\_header;

}

}

Proposed Change 2.x => Add a new section 6.x.x.3 to define the semantics for decoder configuration to sub-bitstream mapping

nb\_associations indicates the number of associations declared in this box.

num\_subbitstreams indicates the number of sub-bitstreams that use the decoder configurations specified by decoder\_cfg\_index.

v3c\_unit\_header is the V3C unit header as per ISO/IEC 23090-5 of the V3C unit applying to the sub-bitstream that is mapped to a decoder configuration information.

decoder\_cfg\_index provides an index of a declared list of decoder configuration boxes for the V3C bitstream. It is a 1-based index with values between 1 to the number of configuration boxes declared in the containing sample entry, minus 1 (the V3CConfigurationBox is not mapped).

1. **Update of V3C Bitstream Sample definition**

During the presentation of m72342 [4], WG7 experts clarified the notion of V3C Composition Unit (c.f. https://git.mpeg.expert/MPEG/3dgh/v-dmc/contributions/-/issues/1286#note\_120736).

We then suggest to change the definition of sample format for V-DMC single track encapsulation (section 7.3.2.3 in [1]) :

*A V3C bitstream sample shall contain one V3C composition unit which is a set of all sub-bitstream composition units that share the same presentation time and each sub-bitstream composition unit shall contain one or more V3C units which belong to a particular presentation time. A sample may be self-contained (e.g., a sync sample) or decoding-wise dependent on other samples of V3C bitstream track*.

With the following definition :

*A V3C bitstream sample shall contain one or more V3C units which belong to a particular presentation time. Each sample contains one or more V3C composition units which is a set of all sub-bitstream composition units that share the same composition time. Presentation time of each V3C bitstream sample shall correspond to the earliest composition time of the V3C composition units present in the V3C units of the sample. A sample may be self-contained (e.g., a sync sample) or decoding-wise dependent on other samples of V3C bitstream track*.

1. **Conclusion**

In this contribution, we proposed two options for indicating the decoder configurations and their mapping to sub-bitstreams in V3C single-track carriage.

We encourage the group to discuss and select one option for inclusion in current Working Draft of ISO/IEC 23090-10 2nd edition.

1. **References**
2. MDS24734\_WG03\_N01434, “*WD of ISO/IEC 23090-10 2nd edition Carriage of visual volumetric video-based coding data*” , MPEG#149, January 2025
3. m71356, “(38.2)[VOL-SYS] On V3C single track encapsulation”, MPEG#149, January 2025
4. MDS24528\_WG03\_N01385, “*Technologies under Consideration for ISO/IEC 14496-12 (ISOBMFF)*” , MPEG#148, December 2024
5. m72342, “*[V3C] On V3C Composition units”,* MPEG#150, April 2025
6. (37.1)[VOL-SYS] On V3C multi track encapsulation (m72405)

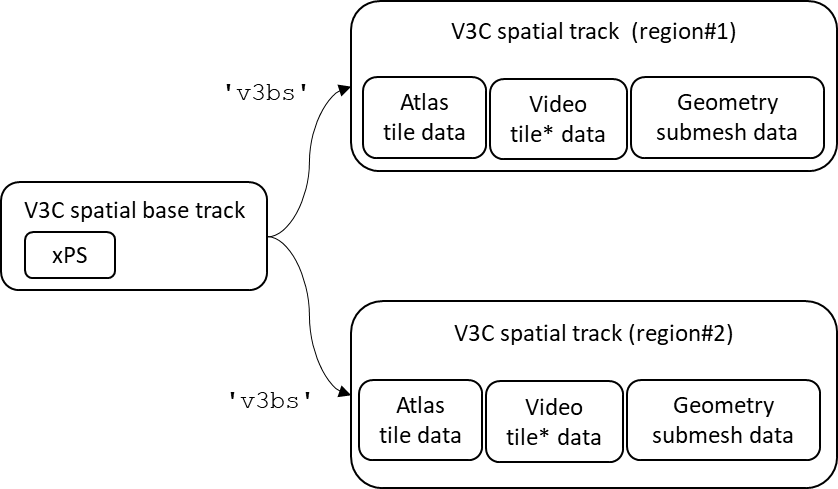
# Introduction

This contribution deals with V3C carriage in ISOBMFF as multiple tracks with tracks dedicated to partial access. We propose V3C spatial tracks that multiplex data corresponding to independently decodable and accessible regions within a 3D scene.

1. **Tracks for spatial access**

ISO/IEC 23090-10 allows the sample entries of an atlas track to declare a collection of 3D regions, using the V3CSpatialRegionCollectionBox. This box provides a mapping between 3D regions and atlas tiles. Then, for region-based access, it makes sense to organize tracks according to these regions:

* The atlas can be split in atlas tiles, with one or more tiles per region, as indicated in the TileMapping structure of the V3CSpatialRegionCollectionBox.
* Along with each set of atlas tiles for a region, the corresponding data for the other components can be multiplexed in a single track as illustrated below.

****

**Figure 1: V3C spatial tracks**

1. **Proposal**

To support spatial tracks for V3C, we propose adding the following sections.

**Add a section** 7.2.2.Y V3C spatial base track

A V3C spatial base track references one or more spatial tracks using track reference with a track reference type equal to ‘v3bs’ (**V3**C **b**ase track to **s**patial track). It contains the parameter sets or SEI messages for one or more referenced spatial tracks. It shall not contain any coded-layer data units (e.g. ACL NAL units, VCL NAL units for video sub-bitstreams, BMC NAL units or DCL NAL units). V3C spatial base track uses V3CSpatialBaseSampleEntry which extends VolumetricVisualSampleEntry with a sample entry type of 'v3b1’.

**Add a section** 7.2.2.Y.1 V3C spatial base sample entry

7.2.2.X.1.1 Definition

|  |  |
| --- | --- |
| Sample Entry Type: | 'v3b1' |
| Container: | SampleDescriptionBox |
| Mandatory: | Yes, when spatial tracks are present |
| Quantity: | Zero or more (in a file) |

7.2.2.Y.1.2 Syntax

aligned class V3CSpatialBaseSampleEntry extends VolumetricVisualSampleEntry('v3b1') {

V3CConfigurationBox config; // v3cC with full decoder configurations

V3CSpatialRegionCollectionBox regions; // optional

}

7.2.2.Y.1.3 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.

config contains a single instance of V3CConfigurationBox providing the exhaustive list of decoder configurations.

regions describes the regions covered by the spatial tracks referenced from the spatial base track with this sample entry. When the number of regions declared in the V3CSpatialRegionCollectionBox is equal to 0, it means that spatial tracks correspond to one or more atlas tiles and not to an identified region.

**Add a section** 7.2.2.Y.2 V3C spatial base track sync sample

A sync sample in a V3C spatial base track is a sample that provides random access point for the spatial tracks it references.

**Add a section** 7.2.2.X V3C spatial track

**Add a section** 7.2.2.X.1 V3C spatial sample entry

7.2.2.X.1.1 Definition

|  |  |
| --- | --- |
| Sample Entry Type: | 'vss1' |
| Container: | SampleDescriptionBox |
| Mandatory: | Yes, in spatial tracks |
| Quantity: | One or more |

A spatial track contains data units corresponding to one or more 3D regions that are independently decodable. A sample of a spatial track sample shall contain at least one V3C unit.

The parent spatial base track is indicated by a track reference of type 'v3bs' from the spatial base track to the spatial track.

For V-DMC, V-PCC and MIV, the following statements shall be verified:

* each sample shall comprise at least one atlas V3C unit;
* all the data units of a spatial track belong to a same set of atlas tiles. The set of atlas tiles may be described in the sample entry of the spatial track.

7.2.2.X.1.2 Syntax

aligned class V3CSpatialSampleEntry extends VolumetricVisualSampleEntry('vss1') {

V3CSpatialConfigurationBox config; // optional

}

class V3CSpatialConfigurationBox extends FullBox('vssC', version = 0, 0) {

V3CSpatialRegionCollectionBox regions; // optional

}

7.2.2.X.1.3 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.

config contains a single instance of V3CSpatialConfigurationBox. When present, it contains at least a V3CSpatialConfigurationBox that describes the set of atlas tiles contained in the spatial track, for example as a V3CSpatialRegionCollectionBox.

regions describe the regions covered by the spatial track. These regions shall be a subset of the regions declared in a V3CSpatialRegionCollectionBox in the parent spatial base track.

**Add a section** 7.2.2.X.2 V3C spatial track sample format

7.2.2.X.2.1 Definition

7.2.2.X.2.2 Syntax

aligned(8) class V3CSpatialSample {

// sample\_size size of sample from SampleSizeBox

for (int i=0; i < sample\_size; ) {

unsigned int(v3c\_config.unit\_size\_precision\_bytes\_minus1 + 1)\*8) v3c\_unit\_size;

bit(8) ss\_v3c\_unit[v3c\_unit\_size];

i += v3c\_unit\_size + v3c\_config.unit\_size\_precision\_bytes\_minus1 + 1;

}

}

7.2.2.X.2.3 Semantics

v3c\_unit\_size specifies the size, in bytes, of the ss\_v3c\_unit array. The size is equivalent to the sample stream v3c unit size ssvu\_v3c\_unit\_size as defined in ISO/IEC 23090-5, Annex C.

ss\_v3c\_unit contains a single V3C unit in V3C unit sample stream format as defined in ISO/IEC 23090-5:2021 in Annex C.

**Add a section** 7.2.2.X.3 V3C spatial track sync sample

A sync sample in a V3C spatial track is a sample for which all sub-bitstream composition units are sub-bitstream IRAP composition units as defined in ISO/IEC 23090-5.

**Add a section** 7.2.2.X.4 V3C spatial track subsample

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

A V3C spatial track may contain one SubSampleInformationBox in its SampleTableBox, or in the TrackFragmentBox of each of its MovieFragmentBoxes, which lists the V3C spatial 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.

1. **Conclusion**

In this contribution, we proposed to organize multi-track encapsulation for spatial access.

We encourage the group to discuss and consider including V3C spatial tracks in current Working Draft of ISO/IEC 23090-10 2nd edition.

1. **References**
2. MDS24734\_WG03\_N01434, “*WD of ISO/IEC 23090-10 2nd edition Carriage of visual volumetric video-based coding data*” , MPEG#149, January 2025