ISO/IEC JTC 1/SC 29/WG 03 N1325

**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:** 2024-08-27

**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 N1376**

**November 2024, Kemer, TR**

|  |  |
| --- | --- |
| **Title** | **Technologies under consideration on carriage of V3C data** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **24439** |

1. Introduction

This document contains technologies under consideration for carriage of V3C data as of the 17th 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 1 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 2 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.