

avtcore
Internet-Draft
Intended status: Standards Track
Expires: 18 April 2026

S. Gudumasu
A. Hamza
InterDigital
15 October 2025

Viewport and Region-of-Interest-Dependent Delivery of Visual Volumetric
Media
draft-ietf-avtcore-rtp-volumetric-media-roi-01

Abstract

This document describes RTCP messages and RTP header extensions to enable partial access and support viewport- and region-of-interest-dependent delivery of visual volumetric media such as visual volumetric video-based coding (V3C). Partial access refers to the ability to access retrieve or deliver only a subset of the media content. The RTCP messages and RTP header extensions described in this document are useful for XR services which transport coded visual volumetric content, such as point clouds.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 18 April 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Background on Visual Volumetric Video-based Coding (V3C)	4
2. Conventions	4
3. Definitions, and Abbreviations	4
3.1. Definitions	4
4. Format of RTCP feedback messages	5
4.1. Static 3D regions request	5
4.1.1. Message format	6
4.2. Arbitrary spatial region request	6
4.2.1. Message format	6
4.3. Viewport request	7
4.3.1. Message format	8
5. RTP header extension for signaling transmitted 3D regions information	11
5.1. Response to a static 3D regions request	11
5.1.1. Message format	11
5.2. Response to an arbitrary spatial region request	12
5.2.1. Message format	12
5.3. Response to a 3D viewport request	15
5.3.1. Message format	15
5.4. Dynamic 3D regions information transmission	16
5.4.1. Message format	16
6. SDP signaling for Viewport and Region-of-Interest dependent delivery of V3C data	19
6.1. SDP signaling of static 3D regions	19
6.2. SDP signaling for region-of-interest feedback messages capability	20
6.2.1. Request for static 3D regions	21
6.2.2. Request for arbitrary spatial region	21
6.2.3. Request for a viewport	22
6.3. SDP signaling for 3D regions transported using RTP header extension	22
6.4. SDP signaling for dynamic 3D regions information transported using RTP header extension	23
6.5. Offer/Answer Considerations	23
7. Security Considerations	29
8. IANA Considerations	29
8.1. 3d-regions SDP attribute	29
8.2. Spatial region and 3D viewport FMT types for RTCP PSFB messages	30
8.3. 3D region type URIs for RTP Header Extensions in SDP	30

9. References	31
9.1. Normative References	31
9.2. Informative References	31
Authors' Addresses	33

1. Introduction

Unlike traditional 2D videos, visual volumetric media represent 3D shapes or objects. Examples of such media include point clouds, meshes, and volumetric videos. For example, a point cloud is a set of data points in space which may represent a 3D shape or object. Each point position has its set of Cartesian coordinates (X, Y, Z) and attribute information such as texture/color, reflectance, or transparency.

To enable parallel processing, partial access, as well as a variety of other functionalities, a visual volumetric media frame can be divided into a number of independently decodable tiles. For partial access use cases, these tiles are mapped to three-dimensional (3D) sub-divisions of the space encompassing the volumetric object, referred to here as 3D regions. The 3D regions are axis-aligned cuboids, i.e., with no associated orientation or rotation, defined in Cartesian space using an anchor point and size of the spatial region along the three axes. The position of the anchor point and the size of the spatial region are defined in terms of volumetric pixels relative to the origin of the volumetric content's coordinate system. Each 3D region has bounding box information of that spatial region and an association with one or more tiles present in that spatial region. The 3D regions information can be used by the receiving devices to stream or access only a subset of the coded media content. With the information provided by the 3D spatial regions, a player can access relevant parts of the immersive media content (e.g., by determining which spatial regions and/or objects falls within the boundaries of the user's viewport or region(s)-of-interest and mapping those to tiles).

When the bounding box information of a spatial region and its association with one or more tiles in the visual volumetric frame is not changing over time, those 3D regions are referred as static 3D regions. Otherwise, if the bounding box information of a spatial region or its association with one or more tiles changes over time, then those 3D regions are referred as dynamic 3D regions. An immersive media content provider provides static or dynamic 3D regions information to the immersive media receivers. The media player requests one or more interested 3D regions based on that information. In some cases, the media player can also request for an arbitrary 3D region within the immersive media content.

This document defines RTCP messages and RTP header extensions to enable partial access and support viewport- and region-of-interest-dependent delivery of visual volumetric media such as visual volumetric video-based coding (V3C) [ISO.IEC.23090-5]. The defined RTCP messages and RTP header extensions can be used with the RTP payload format for V3C in [I-D.draft-ietf-avtcore-rtp-v3c].

1.1. Background on Visual Volumetric Video-based Coding (V3C)

A volumetric media content may be coded using the visual volumetric video-based coding standard 23090-5 [ISO.IEC.23090-5]. V3C is generic mechanism for volumetric video coding and it can be used by applications targeting volumetric content, such as point clouds, immersive video with depth, mesh representations of visual volumetric frames, etc. Examples of such applications are Video-based Point Cloud Compression (V-PCC) [ISO.IEC.23090-5], and MPEG Immersive Video (MIV) [ISO.IEC.23090-12]. V3C encoding of a volumetric frame is achieved through a conversion of volumetric frame from its 3D representation to multiple 2D representations and a generation of associated data. V3C supports the concept of tiling where the volumetric frame is encoded in a number of tiles to enable parallel encoding/decoding and for easy access to one or more regions of V3C content, especially in streaming scenarios. The ISO/IEC 23090-5 specification also defines a set of Volumetric Annotation SEI messages providing information on different objects within the V3C content and the spatial regions or V3C atlas tiles associated with those objects. Moreover, the ISO/IEC International Standards 23090-10 [ISO.IEC.23090-10] defines information on the different spatial regions defined for the V3C content, including the bounding box for the spatial region and its association with one or more V3C atlas tiles. The RTP payload format for V3C content is defined in [I-D.draft-ietf-avtcore-rtp-v3c]. This allows for packetization of one or more V3C Network Abstraction Layer (NAL) units in a RTP packet payload as well as fragmentation of a V3C NAL unit into multiple RTP packets.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Definitions, and Abbreviations

3.1. Definitions

The following terms are defined here for convenience:

Coordinate Systems: The reference coordinate system is a right-handed 3D Cartesian coordinate system with 6 degrees of freedoms (DoFs): 3 translations along the 3 x-y-z dimensions, and 3 rotations about the 3 x-y-z dimensions with the right-hand. The following variations can be derived: Cartesian coordinate system: the reference coordinate system with the 3 translations but without the 3 rotations. World coordinate space - referring to scene space, where manipulation is done relative to scene origin: the reference coordinate system with the origin at the scene origin and with the 3 translations and 3 rotations limited to the scene space (or scene viewing space).

cuboid: A volume having six rectangular faces placed at right angles.

field of view: The extent of the observable world in captured/recorded content or in a physical display device.

tile: independently decodable rectangular 2D region of a video frame or cuboid 3D region of a volumetric frame

4. Format of RTCP feedback messages

The 3D regions present in a volumetric media object can be signaled using an SDP extension. This document extends the RTCP feedback messages defined in the RTP/AVPF [RFC4585] RTP profile and in [RFC5104] to define RTCP feedback messages for requesting static 3D regions, an arbitrary spatial region, or a certain viewport. These messages can be transmitted by the receiver to inform the sender of the desired region(s)-of-interest.

These feedback messages follow a similar message format as RTCP Full Intra Request and Temporal-Spatial Trade-off Request messages defined in [RFC5104]. The message may be sent in a regular full compound RTCP packet or in an early RTCP packet, as per the RTP/AVPF profile rules.

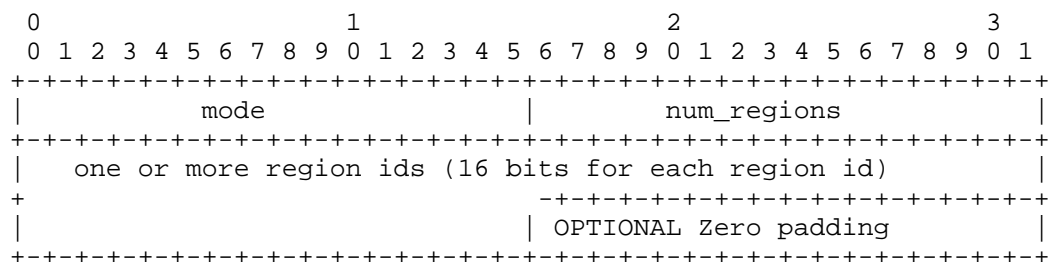
4.1. Static 3D regions request

When the 3D regions available at the sender-side are static, the RTCP feedback message for requesting one or more 3D regions-of-interest contains the required number of 3D regions and a list of region_id parameters. The values of region_id SHALL be acquired from the "a=3d-regions" attributes defined in section 6.1 that are signaled by the sender during SDP negotiation.

4.1.1. Message format

The static 3D regions request feedback message is identified by the RTCP payload type value PT=PSFB, which indicates payload-specific Feedback messages, and message type FMT=18.

The FCI field MUST contain a list of one or more static 3D region ids.



mode (16 bits): This field is uniquely set to all ones for static 3d-regions request.

num_regions (16 bits): indicate the number of interested 3D regions

region_id (16 bits): identifies a pre-defined 3D region

4.2. Arbitrary spatial region request

The RTCP feedback message for a desired spatial region SHALL contain the parameters position_x, position_y, position_z, size_x, size_y and size_z. The values for each of the parameters is indicated using four bytes. The sender SHALL ignore arbitrary spatial region requests describing a region outside the original volumetric content.

4.2.1. Message format

The arbitrary spatial region request feedback message is identified by an RTCP payload type value PT=PSFB and message type FMT=18.

The FCI field for the RTCP feedback message for arbitrary spatial region request is formatted as follows:

4.3.1. Message format

The RTCP feedback message for requesting a viewport is identified by the RTCP payload type value PT=PSFB and message type FMT=19. The FCI SHALL contain exactly one 3D viewport. The FCI format for 3D viewport request feedback message is as follows.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|E|C|I|F|L| CT  | cam_pos_x(h) | cam_pos_x   | cam_pos_x   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| cam_pos_x(l) | cam_pos_y(h) | cam_pos_y   | cam_pos_y   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| cam_pos_y(l) | cam_pos_z(h) | cam_pos_z   | cam_pos_z   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| cam_pos_z(l) | cam_quat_x(h)| cam_quat_x(l)| cam_quat_y(h)|
+-----+-----+-----+-----+-----+-----+-----+-----+
| cam_quat_y(l) | cam_quat_z(h)| cam_quat_z(l)| horizontal_fov
+-----+-----+-----+-----+-----+-----+-----+-----+
| horizontal_fov
+-----+-----+-----+-----+-----+-----+-----+-----+
| vertical_fov
+-----+-----+-----+-----+-----+-----+-----+-----+
| clipping_near_plane
+-----+-----+-----+-----+-----+-----+-----+-----+
| clipping_far_plane | Zero padding |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The desired viewport information in the RTCP feedback viewport message is composed of the following parameters:

ext_camera_flag (E) [1 bit]: This flag value equal to 1 indicates that extrinsic camera parameters information is present in the message. Value 0 indicates that extrinsic camera parameters information is not present in the message.

center_view_flag (C) [1 bit]: This flag indicates whether the signalled viewport position corresponds to the center of the viewport or to one of two stereo positions of the viewport. Value 1 indicates that the signalled viewport position corresponds to the center of the viewport. Value 0 indicates that the signalled viewport position corresponds to one of two stereo positions of the viewport. When ext_camera_flag is set to value 0, this flag value is set to 0.

`int_camera_flag` (I) [1 bit]: Intrinsic camera flag value equal to 1 indicates that intrinsic camera parameters information is present in the message. Value 0 indicates that intrinsic camera parameters information is not present in the message.

`equal_fov_flag` (F) [1 bit]: This flag indicates whether the horizontal FOV and the vertical FOV of the viewport are equal or not. Value 1 indicates the horizontal FOV and vertical FOV are equal. Value 0 indicates horizontal FOV and vertical FOV are not equal. When `int_camera_flag` value is 0, this flag value is set to 1 otherwise set to 0.

`left_view_flag` (L) [1 bit]: This flag indicates whether the signalled viewport information correspond to the left or the right stereo position of the viewport. Value 1 indicates that the signalled viewport information corresponds to the left stereo position of the viewport. Value 0 indicates that the viewport information signalled correspond to the right stereo positions of the viewport.

`camera_type` (CT) [3 bits]: indicates the projection method of the viewport. Value 0 specifies equirectangular projection (ERP). Value 1 specifies a perspective projection. Value 2 specifies an orthographic projection. Values in the range 3 to 7 are reserved for future use.

`cam_pos_x`, `cam_pos_y`, and `cam_pos_z` (32 bits): respectively, indicate the x, y, and z coordinates of the position of the camera in metres in the global reference coordinate system. The value for each field is expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. This information shall be present only when the `ext_camera_flag` (E bit) is set to 1.

`cam_quat_x`, `cam_quat_y`, and `cam_quat_z` (32 bits): indicate the x, y, and z components, respectively, of the rotation of the camera using the quaternion representation. The values are in the range of -2^{14} to 2^{14} , inclusive. When the component of rotation is not present, its value is inferred to be equal to 0. This information shall be present only when the `ext_camera_flag` (E bit) is set to 1.

The value of rotation components may be calculated as follows:

$$qX = \text{cam_quat_x} / 2^{14}, qY = \text{cam_quat_y} / 2^{14}, qZ = \text{cam_quat_z} / 2^{14}$$

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

$$qW = \text{Sqrt}(1 - (qX^2 + qY^2 + qZ^2))$$

The point (w,x,y,z) represents a rotation around the axis directed by the vector (x,y,z) by an angle

$$2 \cdot \cos^{-1}(w) = 2 \cdot \sin^{-1}(\sqrt{x^2 + y^2 + z^2}).$$

`horizontal_fov` (32 bits): indicates the longitude range corresponding to the horizontal size of the viewport region, in units of radians, when `camera_type` is ERP projection. The value is in the range 0 to 2π . When `camera_type` is perspective projection this value specifies the horizontal field of view in radians. The value is in the range of 0 and π . When `camera_type` is orthographic projection, this value specifies the horizontal size of the orthogonal in metres. The value is expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. This information shall be present only when the `int_camera_flag` (I bit) is set to 1.

`vertical_fov` (32 bits): specifies the latitude range corresponding to the vertical size of the viewport region, in units of radians, when `camera_type` is ERP projection. The value is in the range 0 to π . When `camera_type` is perspective projection this value specifies the relative aspect ratio of viewport for perspective projection (horizontal/vertical). The value is expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. When `camera_type` is orthographic projection, this value specifies the relative aspect ratio of viewport for orthogonal projection (horizontal/vertical). The value is expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. This information shall be present only when the `int_camera_flag` (I bit) is set to 1 and `equal_fov_flag` (F) is set to 0. Other cases, vertical FOV information shall not be present.

`clipping_near_plane` and `clipping_far_plane` (32 bits): indicate the near and far depths (or distances) based on the near and far clipping planes of the viewport in meters. The values are expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. This information shall be present only when the `int_camera_flag` (I bit) is set to 1.

5. RTP header extension for signaling transmitted 3D regions information

The sender response may or may not agree with the exact 3D regions of interest requested by the receiver but may contain an extended or reduced version of the requested spatial region(s) depending on the number and size of the 3D regions available in the content that overlap with the requested spatial region(s). This helps the receiver determine when to send subsequent spatial region requests, e.g., in response to head movement sensor information and based on the spatial volume covered by the 3D regions transmitted by the sender. Moreover, signaling the 3D regions sent by the sender also indicates the start of an RTP media flow belonging to a requested 3D region of interest. A response to a request for 3D regions-of-interest involves the sender signaling information of the volumetric media 3D regions that are included in the response.

5.1. Response to a static 3D regions request

If the transmitted 3D regions information response corresponds to a request for one or more of the static 3D regions signaled during SDP negotiation, then the transmitted 3D regions information SHALL be carried using the RTP header extension and includes a `num_regions` field and a list of region ids corresponding to the static 3D regions included in the response. The value for the `num_regions` and list of `region_id` parameters is indicated using two bytes.

5.1.1. Message format

The payload of the transmitted static 3D regions information header extension element can be encoded using the two-byte header defined in [RFC8285].

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
ID								len=xx								num_regions															
one or more region ids (16 bits for each region id)																															
																OPTIONAL Zero padding															

ID (8 bit): is the local identifier.

len (8 bit): is the length of extension data in bytes not including the ID and length fields. The value zero indicates there is no data following.

num_regions (16 bits): indicate the number of transmitted 3D regions.

region_id (16 bit): is a unique identifier for a pre-defined static 3D region in the encoded media.

5.2. Response to an arbitrary spatial region request

If the transmitted 3D region information response corresponds to a request for an arbitrary spatial region, the transmitted 3D regions information SHALL be carried using the RTP header extensions as specified in [RFC8285].

5.2.1. Message format

The payload of the transmitted 3D regions information header extension element can be encoded using the two-byte header defined in [RFC8285].

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
| position_x(h) | position_x   | position_x   | position_x(l)|
+-----+-----+-----+-----+
| position_y(h) | position_y   | position_y   | position_y(l)|
+-----+-----+-----+-----+
| position_z(h) | position_z   | position_z   | position_z(l)|
+-----+-----+-----+-----+
| size_x(h)     | size_x     | size_x     | size_x(l)  |
+-----+-----+-----+-----+
| size_y(h)     | size_y     | size_y     | size_y(l)  |
+-----+-----+-----+-----+
| size_z(h)     | size_z     | size_z     | size_z(l)  |
+-----+-----+-----+-----+
| region_id(h)  | region_id(l) | num_tiles(h) | num_tiles(l) |
+-----+-----+-----+-----+
|               one or more tile ids (16 bits for each tile id)               |
+-----+-----+-----+-----+

```

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|   ID   |   L=xx   | num_regions(h) | num_regions(l) |
+-----+-----+-----+-----+
|               one or more spatial regions information               |
+-----+-----+-----+-----+
|               +-----+-----+-----+-----+               |
|               |   OPTIONAL zero padding   |               |
+-----+-----+-----+-----+

```

ID (8 bit): is the local identifier.

len (8 bit): is the length of extension data in bytes not including the ID and length fields. The value zero indicates there is no data following.

num_regions (16 bit): indicate the number of transmitted 3D regions.

position_x (32 bits): specifies the origin position of the 3D bounding box in the Cartesian coordinates along the x axis.

`position_y` (32 bits): specifies the origin position of the 3D bounding box in the Cartesian coordinates along the y axis.

`position_z` (32 bits): specifies the origin position of the 3D bounding box in the Cartesian coordinates along the z axis.

`size_x` (32 bits): specifies the extension of the 3D bounding box of the volumetric media in the Cartesian coordinates along the x axis relative to the origin position.

`size_y` (32 bits): specifies the extension of the 3D bounding box of the volumetric media in the Cartesian coordinates along the y axis relative to the origin position.

`size_z` (32 bits): specifies the extension of the 3D bounding box of the volumetric media in the Cartesian coordinates along the z axis relative to the origin position.

`region_id` (16 bits): is a unique identifier for a 3D region in the encoded media.

`num_tiles` (16 bits): identifies the number of tile identifiers associated with that spatial region.

`tile_id` (16 bits); identifies a tile identifier associated with that spatial region.

If the requested region-of-interest is an arbitrary spatial region, the sender may choose to send one or more pre-defined 3D regions which were signaled to the receiver during SDP negotiation which overlap with the requested arbitrary spatial region. In this case, the transmitted 3D regions information SHALL be carried using the RTP header extension.

The payload of the transmitted static 3D regions information header extension element can be encoded using two-byte header defined in [RFC8285].

```

0               1               2               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  ID      | len=xx      |          num_regions          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| one or more region ids (16 bits for each region id) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                OPTIONAL Zero padding                                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

ID (8 bit): is the local identifier.

len (8 bit): is the length of extension data in bytes not including the ID and length fields. The value zero indicates there is no data following.

num_regions (16 bits): indicate the number of transmitted 3D regions.

region_id (16 bit): is a unique identifier for a pre-defined static 3D region in the encoded media.

5.3. Response to a 3D viewport request

When an RTCP feedback message for a desired 3D viewport is received by a sender, the sender SHALL respond to receiver with one or more 3D spatial regions information that overlap with the requested viewport. As the transmitted 3D regions correspond to the static 3D regions (indicated via the URN `urn:ietf:params:rtp-hdext:static-3d-regions-` sent in the SDP negotiation), the signaling of the transmitted 3D regions use the RTP header extension.

5.3.1. Message format

The payload of the transmitted static 3D regions information header extension element can be encoded using the two-byte header defined in [RFC8285].

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
+-----+-----+-----+-----+			
ID	len=xx	num_regions	
+-----+-----+-----+-----+			
one or more region ids (16 bits for each region id)			
+-----+-----+-----+-----+			
		OPTIONAL Zero padding	
+-----+-----+-----+-----+			

ID (8 bit): is the local identifier.

len (8 bit): is the length of extension data in bytes not including the ID and length fields. The value zero indicates there is no data following.

num_regions (16 bits): indicate the number of transmitted 3D regions.

region_id (16 bit): is a unique identifier for a pre-defined static 3D region in the encoded media.

5.4. Dynamic 3D regions information transmission

When the 3D regions information in a volumetric media content is changing over time, the transport of the updated 3D regions information SHALL be carried using an RTP header extension. The RTP header extension payload carries the total number of spatial regions present in the volumetric media and each spatial region information.

5.4.1. Message format

The payload of the transmitted dynamic 3D regions information header extension element can be encoded using two-byte header defined in [RFC8285].


```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|  ID           |  L=xx         |  num_regions(h) |  num_regions(l) |
+-----+-----+-----+-----+
|                                     |
+               one or more spatial regions information               +
|                                     |
+                                     |
+               +-----+-----+-----+-----+
|               |  OPTIONAL zero padding  |
+-----+-----+-----+-----+

```

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
| position_x(h) | position_x   | position_x   | position_x(l) |
+-----+-----+-----+-----+
| position_y(h) | position_y   | position_y   | position_y(l) |
+-----+-----+-----+-----+
| position_z(h) | position_z   | position_z   | position_z(l) |
+-----+-----+-----+-----+
| size_x(h)     | size_x       | size_x       | size_x(l)     |
+-----+-----+-----+-----+
| size_y(h)     | size_y       | size_y       | size_y(l)     |
+-----+-----+-----+-----+
| size_z(h)     | size_z       | size_z       | size_z(l)     |
+-----+-----+-----+-----+
| region_id(h)  | region_id(l) | num_tiles(h) | num_tiles(l)  |
+-----+-----+-----+-----+
|               one or more tile ids (16 bits for each tile id)               |
+-----+-----+-----+-----+

```

ID (8 bit): is the local identifier.

len (8 bit): is the length of extension data in bytes not including the ID and length fields. The value zero indicates there is no data following.

num_regions (16 bit): indicates the total number of dynamic 3D regions present in the volumetric media.

position_x (32 bits): specifies the origin position of the 3D bounding box in the Cartesian coordinates along the x axis.

`position_y` (32 bits): specifies the origin position of the 3D bounding box in the Cartesian coordinates along the y axis.

`position_z` (32 bits): specifies the origin position of the 3D bounding box in the Cartesian coordinates along the z axis.

`size_X` (32 bits): specifies the extension of the 3D bounding box of the volumetric media in the Cartesian coordinates along the x axis relative to the origin position.

`size_Y` (32 bits): specifies the extension of the 3D bounding box of the volumetric media in the Cartesian coordinates along the y axis relative to the origin position.

`size_Z` (32 bits): specifies the extension of the 3D bounding box of the volumetric media in the Cartesian coordinates along the z axis relative to the origin position.

`region_id` (16 bits): is an identifier for a 3D region.

`num_tiles` (16 bits): identifies the number of tile identifiers associated with that spatial region.

`tile_id` (16 bits): identifies a tile identifier associated with that spatial region.

When the total number of spatial regions information is large and cannot be accommodated into a single RTP packet due to RTP header extension size limitations or RTP packet size limitations, the information of all updated spatial regions present in an immersive media content is signaled over multiple RTP packets. When the dynamic spatial regions information is sent in multiple RTP packets, the first, and last RTP packets carrying the dynamic spatial regions information in an RTP header extension data is identified using the 'appbits' values.

In the two-byte header form, the 16-bit value required by the RTP specification for a header extension, labeled in the RTP specification [RFC8285], was defined as shown below.

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
  +-----+-----+-----+-----+
  |           0x100           |appbits|
  +-----+-----+-----+-----+
```

The 'appbits' field in the RTP header extension SHALL be defined as below for the transmitted dynamic 3D regions information (indicated via the URN urn:ietf:params:rtp-hdext:dynamic-3d-regions-sent in the SDP negotiation).

```

0
0 1 2 3
+---+---+
|0|0|S|E|
+---+---+

```

S (1 bit): This bit is set to 1 if this is the first RTP packet carrying the dynamic 3d regions information otherwise set to 0.

E (1 bit): This bit is set to 1 if this is the last RTP packet carrying the dynamic 3d regions information otherwise set to 0.

6. SDP signaling for Viewport and Region-of-Interest dependent delivery of V3C data

6.1. SDP signaling of static 3D regions

The 3D regions present in a volumetric media object can be signaled as an SDP extension. A sender MAY offer information on static 3D regions present in the volumetric media in the initial offer-answer negotiation by carrying it in the SDP message. This is done by including the "a=3d-regions" attribute under the relevant media line.

The following parameters are provided in the attribute for each static 3D region:

region_id: identifies a pre-defined 3D region.

position_x: specifies the origin position of the 3D region in the Cartesian coordinate system along the x axis.

position_y: specifies the origin position of the 3D region in the Cartesian coordinate system along the y axis.

position_z: specifies the origin position of the 3D region box in the Cartesian coordinate system along the z axis.

size_x: specifies the extension of the 3D region in the Cartesian coordinates along the x axis relative to the origin position.

size_y: specifies the extension of the 3D region in the Cartesian coordinates along the y axis relative to the origin position.

size_z: specifies the extension of the 3D region in the Cartesian coordinates along the z axis relative to the origin position.

name: specifies the name of the pre-defined 3D region.

The syntax for the "a=3d-regions" attribute conforms to the following ABNF (byte-string defined in [RFC8866] and WSP and DIGIT defined in [RFC5234]):

```
3d-regions = "3d-regions:" PT 1*WSP attr-list
PT = 1*DIGIT / "*"
attr-list = ( set *(1*WSP set) ) / "*"
; WSP and DIGIT defined in [RFC5234]
set= "[" "region_id=" idvalue "," "position_x=" posvalue ","
      "position_y=" posvalue "," "position_z=" posvalue ","
      "size_x=" sizevalue "," "size_y=" sizevalue ","
      "size_z=" sizevalue "," "Name=" namevalue "]"
idvalue= onetone*2DIGIT
; Digit between 1 and 9 that is followed by 0 to 2 other digits
posvalue = sizevalue / "0"
; position may be "0"
sizevalue = onetone *5DIGIT
; Digit between 1 and 9 that is followed by 0 to 5 other digits
onetone = "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"
; Digit between 1 and 9
namevalue = byte-string
; byte-string defined in [RFC8866]
```

An example use of the "a=3d-regions" attribute relative to a media line

```
m=application 40008 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=fmtp:100 v3c-unit-header=08000000; // atlas
a=mid:4
a=3d-regions:99 [region_id=0,position_x=0,position_y=0,position_z=0,
size_x=540,size_y=360,size_z=360,name=Head] [region_id=1,
position_x=0,position_y=360,position_z=0,size_x=1080,size_y=360,
size_z=360,name=Arms] [region_id=2,position_x=0,position_y=720,
position_z=0,size_x=540,size_y=360,size_z=360,name=Body]
[region_id=3,position_x=0,position_y=1080,position_z=0,size_x=540,
size_y=360,size_z=360,name=Legs]
```

6.2. SDP signaling for region-of-interest feedback messages capability

A client supporting region-of-interest-dependent streams SHALL support at least one of the following modes of requesting a desired region-of-interest (signaled from a receiver to a sender):

- * Static 3D regions
- * Arbitrary spatial region
- * Viewport

6.2.1. Request for static 3D regions

A client supporting the static 3D regions mode SHALL include the `a=rtcp-fb` attribute with the static 3D regions feedback type under the relevant media line scope. The static 3D regions type in conjunction with the RTCP feedback method is expressed with the following parameter: `static-3d-regions`. A wildcard payload type ("`*`") may be used to indicate that the RTCP feedback capability attribute for signaling static 3D regions request capability applies to all payload types. If several types of 3D regions signaling is supported and/or the same static 3D regions are specified for a subset of the payload types, several "`a=rtcp-fb`" lines can be used.

Here is an example usage of this attribute to signal static 3D regions relative to a media line based on the RTCP feedback method:

```
a=rtcp-fb:* ack static-3d-regions
```

6.2.2. Request for arbitrary spatial region

A client that supports requests for arbitrary spatial region SHALL indicate this in the SDP offer for the volumetric media where arbitrary spatial region request capabilities are desired. This is done by including the `a=rtcp-fb` attribute line within the scope of the relevant media line in the SDP message with a feedback message type corresponding to the arbitrary spatial region mode. The RTCP feedback message type corresponding to the arbitrary spatial region request is expressed with the parameter: `arbitrary-spatial-region`. A wildcard payload type ("`*`") may be used to indicate that the RTCP feedback capability attribute for signaling arbitrary spatial region request capability applies to all payload types. If the same arbitrary spatial region capability is specified for a subset of the payload types, several "`a=rtcp-fb`" lines can be used.

Here is an example for the usage of this attribute to signal support for arbitrary spatial region requests in an SDP message based on the RTCP feedback method:

```
a=rtcp-fb:* ack arbitrary-spatial-region
```

6.2.3. Request for a viewport

A client (sender or receiver) supporting streaming of immersive media content based on the user's viewport SHALL offer the 'Viewport-dependent streaming (VDS)' capability in SDP for all volumetric media content where viewport-based immersive media streaming is desired. VDS support is offered by including the `a=rtcp-fb` attribute under the relevant media line scope. The VDS support using the RTCP feedback method is expressed with the following parameter: `3d-viewport`. A wildcard payload type ("`*`") may be used to indicate that the RTCP feedback capability attribute for VDS capability applies to all payload types. If the same VDS capability is specified for a subset of the payload types, several "`a=rtcp-fb`" lines can be used. Here is an example usage of this attribute to signal viewport-dependent streaming capability relative to a media line based on the RTCP feedback method:

```
a=rtcp-fb:* ack 3d-viewport
```

6.3. SDP signaling for 3D regions transported using RTP header extension

A client supporting receiving of static 3D regions, arbitrary spatial regions and viewport information feedback messages SHOULD include the transported 3D regions information signaling capability in its SDP offer for all volumetric media streams. The transported 3D regions information is signalled by extending RTP Header extension mechanism defined in [RFC8285].

The transported 3D regions signaling capability is offered by including the `a=extmap` attribute under the relevant media line scope.

The URN corresponding to an arbitrary spatial region is

```
urn:ietf:params:rtp-hdext:arbitrary-3d-regions-sent
```

The URN corresponding to static 3D regions is

```
urn:ietf:params:rtp-hdext:static-3d-regions-sent.
```

Here is an example usage of this URN to signal transmitted 3D regions relative to a media line (e.g., this signaling can be part of the atlas component media line):

```
a=extmap:9 urn:ietf:params:rtp-hdext:static-3d-regions-sent
a=extmap:10 urn:ietf:params:rtp-hdext:arbitrary-3d-regions-sent
```

The numbers 9 and 10 in the example may be replaced with any number in the range 1-254 using the two-byte header extension mechanism.

6.4. SDP signaling for dynamic 3D regions information transported using RTP header extension

When the 3D regions in an immersive media content are changing over time, a sender transmits all the dynamic 3D regions information to the receiver whenever the 3D regions are updated or changed. This information is not sent in response to any RTCP feedback message received from a receiver.

A sender supporting the transmission of dynamic 3D regions information SHOULD offer the dynamic 3D regions signaling capability in the SDP offer for all volumetric media content. The dynamic 3D regions information transmission capability signaling in SDP is offered by including the `a=extmap` attribute under the relevant media line scope.

The URN corresponding to the transmitted dynamic 3D regions information is

`urn:ietf:params:rtp-hdext:dynamic-3d-regions-sent.`

Here is an example usage of this URN to signal transmitted dynamic 3D regions relative to a media line (e.g., this signaling can be part of the atlas component media line):

`a=extmap:255 urn:ietf:params:rtp-hdext:dynamic-3d-regions-sent`

6.5. Offer/Answer Considerations

The following SDP offer/answer examples are provided for V3C content.

An example of offer which supports providing information of static 3D regions present in the volumetric media and providing region-of-interest-dependent streams with the RTCP feedback request modes static 3D regions, arbitrary spatial region and viewport.

```
a=group:v3c 1 2 3 4
a=v3cfmtp:v3c-ptl-level-idc=60;
  sprop-v3c-parameter-set=AQD/AAAP/zwAAAAAADwIAQ5BwAAOADjgQAADkA==
m=video 40000 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=2;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=sendonly
a=mid:1
m=video 40002 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=3;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:2
a=sendonly
m=video 40004 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=4;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:3
a=sendonly
m=application 40006 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=v3cfmtp:sprop-v3c-unit-type=1;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0;
a=mid:4
a=sendonly
a=3d-regions:100 [region_id=0,position_x=0,position_y=0,position_z=0,
  size_x=540,size_y=360,size_z=360,name=Head]
  [region_id=1,position_x=0,position_y=360,position_z=0,size_x=1080,
  size_y=360,size_z=360,name=Arms]
  [region_id=2,position_x=0,position_y=720,position_z=0,size_x=540,
  size_y=360,size_z=360,name=Body]
  [region_id=3,position_x=0,position_y=1080,position_z=0,size_x=540,
  size_y=360,size_z=360,name=Legs]
a=rtcp-fb:* ack static-3d-regions
a=rtcp-fb:* ack arbitrary-spatial-region
a=rtcp-fb:* ack 3d-viewport
```


An example answer which accepts the information of static 3D regions present in the volumetric media and requests region-of-interest, interested viewport content with the RTCP feedback request modes static 3D regions, arbitrary spatial region and viewport.

```
...
a=group:v3c 1 2 3 4
m=video 50000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=recvonly
m=video 50002 RTP/AVP 97
a=rtpmap:97 H265/90000
a=recvonly
m=video 50004 RTP/AVP 98
a=rtpmap:98 H266/90000
a=recvonly
m=application 50006 RTP/AVP 96
a=rtpmap:100 v3c/90000
a=recvonly
a=3d-regions:100 [region_id=0,position_x=0,position_y=0,position_z=0,
  size_x=540,size_y=360,size_z=360,name=Head] [region_id=1,
  position_x=0,position_y=360,position_z=0,size_x=1080,size_y=360,
  size_z=360,name=Arms] [region_id=2,position_x=0,position_y=720,
  position_z=0,size_x=540,size_y=360,size_z=360,name=Body]
  [region_id=3,position_x=0,position_y=1080,position_z=0,size_x=540,
  size_y=360,size_z=360,name=Legs]
a=rtcp-fb:* ack static-3d-regions
a=rtcp-fb:* ack arbitrary-spatial-region
a=rtcp-fb:* ack 3d-viewport
```

An example of offer which supports the transported 3D regions information signaling capability.

```
a=group:v3c 1 2 3 4
a=v3cfmtp:v3c-ptl-level-idc=60;
  sprop-v3c-parameter-set=AQD/AAAP/zwAAAAAADwIAQ5BwAAOADjgQAADkA==
m=video 40000 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=2;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=sendonly
a=mid:1
m=video 40002 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=3;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:2
a=sendonly
m=video 40004 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=4;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:3
a=sendonly
m=application 40006 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=v3cfmtp:sprop-v3c-unit-type=1;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0;
a=mid:4
a=sendonly
a=3d-regions:100 [region_id=0,position_x=0,position_y=0,position_z=0,
  size_x=540,size_y=360,size_z=360,name=Head] [region_id=1,
  position_x=0,position_y=360,position_z=0,size_x=1080,size_y=360,
  size_z=360,name=Arms] [region_id=2,position_x=0,position_y=720,
  position_z=0,size_x=540,size_y=360,size_z=360,name=Body]
  [region_id=3,position_x=0,position_y=1080,position_z=0,size_x=540,
  size_y=360,size_z=360,name=Legs]
a=rtcp-fb:* ack static-3d-regions
a=rtcp-fb:* ack arbitrary-spatial-region
a=rtcp-fb:* ack 3d-viewport
a=extmap:9/sendonly urn:ietf:params:rtp-hdrext:static-3d-regions-sent
a=extmap:10/sendonly
  urn:ietf:params:rtp-hdrext:arbitrary-3d-regions-sent
```

An example answer which supports sending only static region-of-interest RTCP feedback request messages and receiving the transported 3D regions information.

```
...
a=group:v3c 1 2 3 4
m=video 50000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=recvonly
m=video 50002 RTP/AVP 97
a=rtpmap:97 H265/90000
a=recvonly
m=video 50004 RTP/AVP 98
a=rtpmap:98 H266/90000
a=recvonly
m=application 50006 RTP/AVP 96
a=rtpmap:100 v3c/90000
a=recvonly
a=3d-regions:100 [region_id=0,position_x=0,position_y=0,position_z=0,
  size_x=540,size_y=360,size_z=360,name=Head] [region_id=1,
  position_x=0,position_y=360,position_z=0,size_x=1080,size_y=360,
  size_z=360,name=Arms] [region_id=2,position_x=0,position_y=720,
  position_z=0,size_x=540,size_y=360,size_z=360,name=Body]
  [region_id=3,position_x=0,position_y=1080,position_z=0,size_x=540,
  size_y=360,size_z=360,name=Legs]
a=rtcp-fb:* ack static-3d-regions
a=extmap:9/recvonly urn:ietf:params:rtp-hdrext:static-3d-regions-sent
```

An example of offer which supports transmission of dynamic 3D regions information and it's signaling capability.

```
a=group:v3c 1 2 3 4
a=v3cfmtp:v3c-ptl-level-idc=60;
  sprop-v3c-parameter-set=AQD/AAAP/zwAAAAAADwIAQ5BwAAOAdjgQAADkA==
m=video 40000 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=2;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=sendonly
a=mid:1
m=video 40002 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=3;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:2
a=sendonly
m=video 40004 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=4;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:3
a=sendonly
m=application 40006 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=v3cfmtp:sprop-v3c-unit-type=1;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0;
a=mid:4
a=sendonly
a=3d-regions:100 [region_id=0,position_x=0,position_y=0,position_z=0,
  size_x=540,size_y=360,size_z=360,name=Head] [region_id=1,
  position_x=0,position_y=360,position_z=0,size_x=1080,size_y=360,
  size_z=360,name=Arms] [region_id=2,position_x=0,position_y=720,
  position_z=0,size_x=540,size_y=360,size_z=360,name=Body]
  [region_id=3,position_x=0,position_y=1080,position_z=0,size_x=540,
  size_y=360,size_z=360,name=Legs]
a=extmap:255/sendonly
  urn:ietf:params:rtp-hdrext:dynamic-3d-regions-sent
```

An example answer which accepts receiving of dynamic 3D regions information and it's signaling capability.

```
...
a=group:v3c 1 2 3 4
m=video 50000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=recvonly
m=video 50002 RTP/AVP 97
a=rtpmap:97 H265/90000
a=recvonly
m=video 50004 RTP/AVP 98
a=rtpmap:98 H266/90000
a=recvonly
m=application 50006 RTP/AVP 96
a=rtpmap:100 v3c/90000
a=recvonly
a=3d-regions:100 [region_id=0,position_x=0,position_y=0,position_z=0,
  size_x=540,size_y=360,size_z=360,name=Head] [region_id=1,
  position_x=0,position_y=360,position_z=0,size_x=1080,size_y=360,
  size_z=360,name=Arms] [region_id=2,position_x=0,position_y=720,
  position_z=0,size_x=540,size_y=360,size_z=360,name=Body]
  [region_id=3,position_x=0,position_y=1080,position_z=0,size_x=540,
  size_y=360,size_z=360,name=Legs]
a=extmap:255/recvonly
  urn:ietf:params:rtp-hdrext:dynamic-3d-regions-sent
```

7. Security Considerations

RTCP feedback messages and RTP packets using the header extension format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124].

8. IANA Considerations

This document contains three considerations to IANA: new SDP attribute, new format values for RTCP PSFB payload types and new URIs for RTP Header Extensions.

8.1. 3d-regions SDP attribute

This document defines a new session and media level SDP attribute: "3d-regions". This attribute will be registered by IANA under attribute-field names (<attribute-name>) registry in Session Description Protocol (SDP). The "3d-regions" attribute is used to convey 3D regions present in a volumetric media object. Its format is defined in Section 6.1. Further semantics are provided in Table 1.

Type	SDP Name	Usage Level	Mux Category	Reference
attribute	3d-regions	session, media	NORMAL	"this memo"

Table 1: Additional details for <attribute-name> Registry

8.2. Spatial region and 3D viewport FMT types for RTCP PSFB messages

This document defines two new format (FMT) values for RTCP PSFB payload types. These values will be registered by IANA under FMT values for PSFB Payload Types in RTP parameters. The "Spatial Region" RTCP PSFB message is used for requesting a static 3D region or an arbitrary spatial region. It's format is defined in Section 4.1 and 4.2 respectively. The "3D Viewport" RTCP PSFB message is used for requesting a certain 3D viewport. It's format is defined in Section 4.3. Further semantics are provided in Table 2.

Value	Name	Long Name	Reference
18	SR	Spatial Region	"this memo"
19	3DVP	3D Viewport	"this memo"

Table 2: Additional details of new FMT
Values for PSFB Payload Types

8.3. 3D region type URIs for RTP Header Extensions in SDP

This document defines three new RTP Compact Header Extensions. These values will be registered by IANA under RTP Compact Header Extensions. If the transmitted 3D regions information response corresponds to a request for one or more of the static 3D regions or for an arbitrary spatial region, then the transmitted 3D regions information SHALL be carried using the RTP header extension. The RTP HE URI used for the static 3D Regions is "urn:ietf:params:rtp-hdext:static-3d-regions-sent". It's format is defined in Section 5.1. The RTP HE URI used for the arbitrary 3D Regions is "urn:ietf:params:rtp-hdext:arbitrary-3d-regions-sent". It's format is defined in Section 5.2. When the 3D regions in an immersive media content are changing over time, the dynamic 3D regions information is transmitted using an RTP HE. The RTP HE URI used for the dynamic 3D Regions is "urn:ietf:params:rtp-hdext:dynamic-3d-regions-sent". It's format is defined in Section 5.4. The semantics are provided in

Table 3.

Extension URI	Description	Reference
urn:ietf:params:rtp-hdext:static-3d-regions-sent	Signaling of static 3d-regions information for the sent media	"this memo"
urn:ietf:params:rtp-hdext:arbitrary-3d-regions-sent	Signaling of arbitrary 3d-regions information for the sent media	"this memo"
urn:ietf:params:rtp-hdext:dynamic-3d-regions-sent	Signaling of dynamic 3d-regions information for the sent media	"this memo"

Table 3: Additional details for new RTP Compact Header Extensions

9. References

9.1. Normative References

- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/rfc/rfc4585>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/rfc/rfc5234>>.
- [RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <<https://www.rfc-editor.org/rfc/rfc8285>>.
- [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <<https://www.rfc-editor.org/rfc/rfc8866>>.

9.2. Informative References

- [I-D.draft-ietf-avtcore-rtp-v3c]
Ilola, L. and L. Kondrad, "RTP Payload Format for Visual Volumetric Video-based Coding (V3C)", Work in Progress, Internet-Draft, draft-ietf-avtcore-rtp-v3c-12, 15 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-v3c-12>>.
- [ISO.IEC.23090-10]
ISO/IEC, "Information technology - Coded representation of immersive media - Part 10: Carriage of visual volumetric video-based coding data", ISO/IEC FDIS 23090-10, 2022, <<https://www.iso.org/standard/78991.html>>.
- [ISO.IEC.23090-12]
ISO/IEC, "Information technology - Coded representation of immersive media - Part 12: MPEG Immersive video (MIV)", ISO/IEC 23090-12, 2022, <<https://www.iso.org/standard/79113.html>>.
- [ISO.IEC.23090-5]
ISO/IEC, "Information technology - Coded representation of immersive media - Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)", ISO/IEC 23090-5, 2021, <<https://www.iso.org/standard/73025.html>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/rfc/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/rfc/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/rfc/rfc3711>>.

- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/rfc/rfc5104>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<https://www.rfc-editor.org/rfc/rfc5124>>.

Authors' Addresses

Srinivas Gudumasu
InterDigital
Canada
Email: srinivas.gudumasu@interdigital.com

Ahmed Hamza
InterDigital
Canada
Email: ahmed.hamza@interdigital.com