ICS 35.040

CCS L 71

团体标准

T/AI 109.1—XXXX

|  |
| --- |
|  |

信息技术 智能媒体编码 第1部分：

系统

Information technology – Intelligent media coding –

Part 1: System

点击此处添加与国际标准一致性程度的标识

|  |
| --- |
| （征求意见稿） |
| （在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上） |

XXXX - XX - XX实施

**中关村视听产业技术创新联盟** 发布

XXXX - XX - XX发布

1. 目 次

[前言 II](#_Toc176078112)

[引言 III](#_Toc176078113)

[1 范围 1](#_Toc176078115)

[2 规范性引用文件 1](#_Toc176078116)

[3 术语和定义 2](#_Toc176078117)

[4 缩略语 3](#_Toc176078118)

[5 约定 3](#_Toc176078119)

[6 架构 3](#_Toc176078120)

[6.1概述 3](#_Toc176078121)

[6.2 沉浸媒体系统架构 5](#_Toc176078122)

[6.3 代码类型索引 6](#_Toc176078123)

[7 沉浸媒体处理 7](#_Toc176078124)

[7.1基于全景视频的处理 7](#_Toc176078125)

[7.2 基于容积视频的处理 14](#_Toc176078126)

[7.3 基于自由视角视频的处理 16](#_Toc176078127)

[8 沉浸媒体文件格式 16](#_Toc176078128)

[8.1 基本结构和语义 16](#_Toc176078129)

[8.2 基于沉浸媒体的数据盒扩展 42](#_Toc176078130)

[8.3沉浸媒体定时元数据 72](#_Toc176078131)

[8.4非时序沉浸式媒体封装 87](#_Toc176078132)

[9 沉浸媒体DASH传输信令 95](#_Toc176078133)

[9.1 沉浸媒体DASH传输架构 95](#_Toc176078134)

[9.2 沉浸媒体的DASH MPD描述子 100](#_Toc176078135)

[10 沉浸媒体SMT传输信令 115](#_Toc176078136)

[10.1 概述 115](#_Toc176078137)

[10.2 全景视频的SMT传输信令 116](#_Toc176078138)

[10.3 容积视频的SMT传输信令 118](#_Toc176078139)

[10.4 自由视角视频的SMT传输信令 123](#_Toc176078140)

[10.5 沉浸媒体交互反馈信息 123](#_Toc176078141)

[附　录　A （资料性） 沉浸媒体介绍 129](#_Toc176078142)

[附　录　B （资料性） AVS-I客户端参考模型 131](#_Toc176078143)

1. 前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是T/AI 109《信息技术 智能媒体编码》的第1部分。T/AI 109已经发布了如下部分：

——第2部分：视频；

——第3部分：沉浸式音频；

——第4部分：符合性测试；

——第6部分：智能媒体格式。

本文件由数字音视频编解码技术标准工作组提出。

本文件由中关村视听产业技术创新联盟归口。

本部分起草单位：上海交通大学、中兴通讯股份有限公司、腾讯科技有限公司、广东博华超高清创新中心有限公司、OPPO广东移动通信有限公司、咪咕文化科技有限公司、北京工业大学、北京三星通信技术研究有限公司、华为技术有限公司、鹏城实验室、北京大学。

本部分主要起草人： 徐异凌、黄成、李秋婷、胡颖、管云峰、张伟民、许晓中、刘杉、龙仕强、杨智尧、李琳、贝悦、王琦、徐嵩、郑建铧、牟伦田、冯亚楠、陈丽丽、金晶、王一帆、杨开发、侯朴玥、殷骄阳、王恒超、侯礼志、高粼遥、刘畅、张钰杰、白雅贤、吴越、邸佩云、黄铁军、赵海英、高文。

1. 引 言

T/AI 109.1规定了沉浸媒体系统中的虚拟现实视频和音频数据的封装格式、传输格式和信令格式。适用于沉浸媒体系统中的网络流媒体、视频直播和点播等应用。

T/AI 109拟由以下部分组成：

——第1部分：系统。目的在于确立沉浸媒体虚拟现实数据的系统信令；

——第2部分：视频。目的在于确立智能媒体高效视频压缩方法的解码过程；

——第3部分：沉浸式音频。目的在于确立适用沉浸式音频高效压缩方法的解码过程；

——第4部分：符合性测试。目的在于确立测试验证编码位流和解码器是否满足T/AI 109所规定的要求；

——第5部分：参考软件。目的在于确立对智能媒体高效视频压缩方法和沉浸式音频高效压缩方法的参考实现；

——第6部分：智能媒体格式。目的在于确立异构网络中智能媒体编码数据的存储格式和传输信令；

——第7部分：音频封装与传输。目的在于确立异构网络中的沉浸式音频数据提供封装和传输；

——第10部分：实时语音。目的在于确立面向下一代实时语音通信的解码过程。

本文件的发布机构提请注意，声明符合本文件时，可能涉及7.3，7.3.1，7.3.2，8.1，8.1.1，8.1.3，8.2，8.2.1，8.2.2，8.2.3，8.2.5，8.3.1.5，8.3.2，8.3.4，8.3.5，9.1，9.1.1，9.1.2，9.2.2.3，9.2.3.2，9.2.3，9.2.4，9.2.6，10.2.1，10.3.3，10.3.4，10.5.3，10.5.4中如下24项与媒体编解码、封装传输技术相关的专利的使用。专利申请号及名称如下：

| 序号 | 专利申请号 | 专利名称 | 涉及专利的章、条 |
| --- | --- | --- | --- |
| 1 | 201810760047.6 | 基于视频内容消费的反馈信息标识方法、系统及存储介质 | 10.5.3，10.5.4 |
| 2 | 201811108139.2 | 沉浸媒体内容随用户移动变化的方法及系统 | 10.5.4 |
| 3 | 201810806586.9 | 基于视频内容的指示信息标识方法、系统及存储介质 | 10.2.1 |
| 4 | 202110642375.8 | 基于视频内容的指示信息标识方法、系统及存储介质 | 10.2.1 |
| 5 | 202010301699.0 | 多媒体数据收发方法、系统、处理器和播放器 | 8.1 |
| 6 | 202110441806.4 | 一种三维点云的封装方法、装置及介质 | 8.1，8.2 |
| 7 | 202210975614.6 | 三维点云的传输方法、接收方法、传输装置、接收装置及存储介质 | 10.3.4 |
| 8 | 202010026089.4 | 点云数据封装方法及传输方法 | 10.3.3 |
| 9 | 202011418994.0 | 数据传输方法、装置、设备和存储介质 | 10.3.3 |
| 10 | 201710557241.X | 基于用户消费信息的视频处理方法及装置 | 8.3.1.5 |
| 11 | 201711484881.9 | 视频传输方法、客户端、服务器 | 9.1.1,9.2.2.3 |
| 12 | 201810993340.7 | 视频数据的处理，传输方法及装置，视频数据的处理系统 | 8.1.1 ，8.2.5，8.3.2，9.1，9.2.6 |
| 13 | PCT/CN2019/109490 | FILE FORMAT FOR POINT CLOUD DATA | 7.3.1，7.3.2，8.2.2，8.3.5 |
| 14 | PCT/CN2020/070854 | POINT CLOUD DATA PROCESSING | 7.3.1 ，9.1.2 |
| 15 | PCT/CN2020/098009 | VOLUMETRIC MEDIA PROCESS METHODS AND APPARATUS | 9.1.2，9.2.3.2 |
| 16 | 202110098943.2 | 一种容积媒体处理方法、装置、存储介质及电子装置 | 7.3.1，8.1.3，8.2.2，8.3.5 |
| 17 | PCT/CN2020/084827 | VOLUMETRIC VISUAL MEDIA PROCESS METHODS AND APPARATUS | 7.3.1，8.1.3，8.2.2 ，8.3.4 |
| 18 | 201811459740.6 | 视频数据的处理方法、装置、相关设备及存储介质 | 8.2.1 |
| 19 | 201911223329.3 | 媒体资源播放及文本渲染方法、装置、设备和存储介质 | 8.2.1 |
| 20 | 202010581346.0 | 观看沉浸式媒体的媒体的信息表示和处理的方法及装置 | 8.2.2，9.2.3 |
| 21 | PCT/CN2019/122095 | MULTI-VIEW VIDEO PROCESSING METHOD AND APPARATUS | 8.2.2 |
| 22 | PCT/CN2021/072214 | MULTI-TRACK BASED IMMERSIVE MEDIA PLAYOUT | 8.2.2 |
| 23 | 202210249027.9 | 数据处理方法、装置、设备、存储介质及程序产品 | 7.3，8.2.3 |
| 24 | 202210651448.4 | 媒体信息处理方法、媒体信息播放方法及装置、存储介质 | 7.3，9.2.4 |

本文件的发布机构对于该专利的真实性、有效性和范围无任何立场。该专利持有人已向本文件的发布机构保证，他愿意同任何申请人在合理且无歧视的条款和条件下，就专利授权许可进行谈判。该专利持有人的声明已在本文件的发布机构备案，相关信息可以通过以下联系方式获得：

专利持有人：上海交通大学

地址：上海市闵行区东川路800号

专利持有人：中兴通讯股份有限公司

地址：广东省深圳市南山区科技南路55号

联 系 人：黄铁军（数字音视频编解码技术标准工作组秘书长）

通讯地址：北京大学理科2号楼2641室

邮政编码：100871

电子邮件：tjhuang@pku.edu.cn

电话：+8610-62756172

传真：+8610-62751638

网址：http://www.avs.org.cn

请注意除上述专利外，本文件的某些内容仍可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

信息技术 智能媒体编码 第1部分：系统

* 1. 范围

本文件规定了沉浸媒体系统中的虚拟现实视频和音频数据的封装格式、传输格式和信令格式。

本文件适用于沉浸媒体系统中的网络流媒体、视频直播和点播等应用。

* 1. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 33475.1-2019 信息技术 高效多媒体编码 第1部分：系统（Information technology—High efficiency multimedia coding—Part 1: System）

GB/T 33475.6-2024 信息技术 高效多媒体编码 第6部分：智能媒体传输（Information technology—High efficiency multimedia coding—Part 6: Smart media transport）

GB/T 20214118-T-469 信息技术 可关闭字幕（Information technology - Closed Captioning ）

ISO/IEC 14496-12 信息技术 音视频对象的编码 第12部分：ISO基本媒体文件格式（Information technology – Coding of audio-visual objects – Part 12: ISO base media file format）

ISO/IEC 23008-1 信息技术 异构环境下的高效编码和媒体传送 第1部分：MPEG媒体传输（Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 1: MPEG media transport (MMT)）

IETF RFC 1738 统一资源定位符（Uniform Resource Locators (URL)）

IETF RFC 3406 统一资源命名空间定义机制（Uniform Resource Names (URN) Namespace Definition Mechanisms）

IETF RFC 3986 统一资源标识符：通用语法（Uniform Resource Identifier (URI): Generic Syntax）

IETF RFC 4122 通用唯一标识符URN命名空间（A Universally Unique Identifier (UUID) URN Namespace）

W3C XML 可扩展置标语言（Extensible Markup Language (XML)）

ISO/IEC 14496-15 信息技术 – 音视频对象编码 第15部分：基于ISO媒体文件格式的网络抽象层单元结构化视频承载（Information technology — Coding of audio-visual objects — Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO based media file format）

ISO/IEC 23090-5 信息技术 沉浸媒体编码表示 第5部分：基于视觉的容积视频编码和基于视频的点云压缩（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 23008-12:2017 信息技术 异构环境下的高效编码和媒体传送 第12部分：图像文件格式 （Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format）

ISO/IEC 23009-1:2019 信息技术 基于HTTP的动态自适应流 第1部分：媒体表示描述和分片格式（Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats）

* 1. 术语和定义

下列术语和定义适用于本文件。

3.1

视场 field of view

虚拟现实视频格式中同一时刻可以观察到的视频区域。

3.2

对等映射 equal sphere projection

在球面视频向平面映射的过程中，平面视频均匀地保留了球面视频的所有内容。

3.3

非对等映射 unequal sphere projection

在球面视频向平面映射的过程中，平面视频非均匀地保留了球面视频的内容，具体地，在主视场区域内视频质量高，在其他区域视频质量低。

3.4

智能媒体传输协议 smart media transport protocol

用于在IP网络上传输有效载荷的应用层传送协议。

3.5

媒体资源 asset

任何与唯一标识符联系的用作构建一个多媒体演示的多媒体数据实体。

3.6

媒体内容 media content

媒体数据中包含的具有相同时间基准的音频、视频、字幕等信息。

3.7

媒体播放列表 media playlist

一个规范化描述媒体呈现的文件，包含了媒体内容描述。

3.8

媒体表示 media representation

在媒体呈现描述文件中描述媒体片段信息的元素

3.9

视点 viewpoint

一个全景摄像机对应的全景媒体内容。

3.10

视窗 viewport

全景视频中，视窗为用户观看的球面区域。

容积视频中，视窗为属性信息投影到全景或三维的图像或视频对应视场的平面，适用于用户特定观看朝向和旋转角度的播放或观看。

3.11

容积媒体 volumetric media

指捕获自三维空间的视觉媒体内容，包含但不限于多视图视频、视频编码点云、几何编码点云。

3.12

容积视频 volumetric video

采用视频编码方式并提供3DoF+、6DoF观看体验的容积媒体，在文件封装中包含容积视频类型轨道的沉浸式媒体，包括多视图视频、视频编码点云。

3.13

公共参考坐标系 common reference coordinate system

中心等于(0, 0, 0)的三维笛卡尔坐标系，用作视点群组内所有视点的参考坐标系。

3.14

全局坐标系 global coordinate system

表示相同采集位置的音频、全景视频和图像内容的共同坐标轴，全局坐标系的原点通常与全景视频采集设备的中心点以及观察者头部在三维空间中的位置相同。

3.15

本地坐标系 local coordinate system

将全局坐标系旋转后获得的坐标系

* 1. 缩略语

|  |  |  |
| --- | --- | --- |
| ALC  DASH  HLS  HTTP  ISOBMFF  NAL  ROI  SMT  SMTP  TCP  URI  URL  URN  XML  SEI  NAL  MPD  3DoF  3DoF+  6DoF  V3C  SAP | 图集编码层  动态自适应流媒体传输  动态码率自适应传输  超文本传送协议  ISO基媒体文件格式  网络抽象层  感兴趣区域  智能媒体传输  智能媒体传输协议  传输控制协议  统一资源标识符  统一资源定位器  统一资源名称  可扩展置标语言  补充增强信息  网络抽象层  媒体表示描述  三自由度  增强的三自由度  六自由度  容积视频  媒体流接入点 | （Atlas Coding Layer）  （Dynamic Adaptive Streaming over HTTP）  （HTTP Live Streaming）  （Hypertext Transfer Protocol）  （ISO Base Media File Format）  （Network Abstraction Layer）  （Region of Interest）  （Smart Media Transport）  （SMT Protocol）  （Transmission Control Protocol）  （Uniform Resource Identifier）  （Uniform Resource Locator）  （Uniform Resource Name）  （Extensible Mark-up Language）  （Supplemental Enhancement Information）  （network abstraction layer）  （Media Presentation Description）  （Three Degree of Freedom）  （Three Degree of Freedom Flus）  （Six Degree of Freedom）  （Visual Volumetric Video）  （Stream Access Point） |

* 1. 约定

下列约定适用于本文件。

本文件使用了大端模式的字节序表示方案。

架构

6.1概述

如图 1所示，沉浸媒体系统端到端处理流程包括：内容获取与制作、沉浸媒体编码/文件封装、沉浸媒体传输、沉浸媒体解码/文件解封装、沉浸媒体渲染等主要技术环节。



图 1 沉浸媒体系统处理流程

其中，沉浸媒体内容处理、沉浸媒体文 件格式，以及沉浸媒体传输信令属于本标准的规范性范畴，对应的章节组织如下：

——沉浸媒体内容处理定义在第7章，包括：

·全景视频处理过程

·全景视频坐标系统

·全景视频投影

·全局与本地坐标轴转换

·全景视频封装格式

·容积视频处理过程

·容积视频的部分访问

——沉浸媒体文件格式定义在第8章，包括：

·基本结构和语义，基于沉浸媒体处理的全景视频的投影格式、旋转格式、内容覆盖、球面区域以及区域封装等结构表示，以及容积视频的解码配置、空间区域信息、视图及其分组、相机信息等结构表示

·基于沉浸媒体的数据盒扩展，参考ISOBMFF的数据盒格式，包括全景视频的投影数据盒、旋转数据盒、覆盖信息数据盒、区域封装数据盒等，以及容积视频单轨封装和多轨数据盒、视图相关数据 静态区信息数据盒等

·沉浸媒体定时元数据，定义了定时元数据的样本入口及样本格式，以及基于沉浸媒体的球面区域定时元数据、多视点定时元数据、视窗信息定时元数据、视图定时元数据等

·非时序沉浸式媒体封装

——沉浸媒体DASH传输信令定义在第9章，包括：

·沉浸媒体DASH传输架构

·沉浸媒体DASH MPD描述子，分别定义了支持全景视频描述子（包括投影类型信息的描述子、内容覆盖信息的描述子、推荐视窗信息的描述子等）、和支持容积视频描述子（包括容积视信息的描述子、容积视频组件信息的描述子、空间区域信息的描述子等）

——沉浸媒体SMT传输信令和交互反馈信令定义在第10章，包括：

·沉浸媒体虚拟现实信息资源描述符，对当前携带虚拟现实内容资源的具体内容进行了描述

·沉浸媒体ROI引导信息表

·沉浸媒体容积视频视角依赖媒体数据包消息

·沉浸媒体容积视频媒体资源索引描述符

·沉浸媒体容积视频媒体资源组消息

·沉浸媒体交互反馈消息结构体

·沉浸媒体交互反馈信息

·沉浸媒体交互反馈信令表

### 6.2 沉浸媒体系统架构

本节定义了沉浸媒体系统的架构，见图2。主要规定了从服务器到客户端的具体的沉浸媒体内容的处理及表述、文件格式和传输信令。



图 2 沉浸媒体系统架构

现实世界的听觉-视觉场景通过音频传感器、摄像设备（普通摄像头、立体摄像头、光场摄像头）以及传感设备（激光雷达）采集，转化为一系列的数据信号后制作成沉浸式媒体内容呈现给用户观看。摄像设备部署在特定的位置获取一定空间内视频/图像内容，音频可以通过不同的麦克风配置来获取，视频/图像和音频在时间和空间内保持同步。沉浸媒体系统中视频/图像内容制作可分为全景视频制作和容积视频制作。

1. 全景视频制作：由一组摄像机或一个带有多个摄像头和传感器的摄像设备录制而成。摄像头通常可以获取在设备中心周围所有方向的内容；
2. 容积视频制作：由部署在多个位置的相机或相机阵列拍摄得到的多视图视频、点云数据、光场等形式的内容制作而成。容积视频包含几何信息、属性信息、占位图信息以及图集数据等，容积视频在编码前一般需要进行特定处理，例如点云数据在编码前需要切割、映射等过程。

全景视频的像素信息和容积视频的几何信息、属性信息以及占位图信息可以采用传统的视频编码方式，容积视频的图集数据需要采用熵编码方式，然后，按一定格式（如ISOBMFF）将编码的媒体封装在文件容器中并结合描述媒体内容属性的元数据，组成一个媒体文件或者组成一个初始化片段和媒体片段。

在服务器中，存储了媒体呈现描述/信令信息和媒体文件资源。媒体呈现描述/信令信息给客户端提供了足够的通知信息，使得对应的媒体内容在一种传输机制下被交付到播放器并进行消费。客户端可以根据终端状态，例如头部/眼部/位置追踪、网络吞吐量等，采用质量/视点自适应方式动态请求媒体文件资源。

媒体文件通过传输机制（DASH、SMT）传输给用户终端。用户终端接收到媒体文件后，对文件进行解封装、解码、拼接/合成/重构、渲染等一系列处理后再显示沉浸媒体内容。

### 6.3 代码类型索引

6.3.1 样本入口类型

本文档涉及的样本入口类型如表1所示。

表 1 样本入口类型表

|  |  |  |
| --- | --- | --- |
| **样本入口类型** | **章节** | **描述** |
| dyvp | 8.3.2.1.2 | 应用于全景视频动态视点定时元数据轨道。 |
| rcvp | 8.3.1.5.2 | 应用于全景视频推荐视窗元数据轨道。 |
| ttsl | 8.3.1.6.2 | 应用于全景视频定时文本球体位置元数据轨道。 |
| srzm | 8.3.1.7.2 | 应用于全景视频球面区域缩放定时元数据轨道。 |
| v3e1 | 8.2.2.7.2 | 应用于单轨道封装模式，所有的图集参数集和SEI消息都存放解码器配置信息中。 |
| v3eg | 8.2.2.7.2 | 应用于单轨道封装模式，图集参数集和SEI消息存放于解码器配置信息和轨道样本中。 |
| v3c1 | ‎8.2.2.8.2 | 应用于容积视频多轨道封装模式，且容积视频码流包含唯一图集轨道。样本入口类型为'v3c1'的容积视频图集轨道下，所有图集子码流的参数集和SEI信息应存放在setupUnit数组中。 |
| v3cg | ‎8.2.2.8.2 | 应用于容积视频多轨道封装模式，且容积视频码流包含唯一图集轨道。样本入口类型为'v3cg'的容积视频轨道下，图集子码流的参数集和SEI信息可以存放在setupUnit数组中，或者也可以存放在图集的子码流中。 |
| v3cb | ‎8.2.2.8.2 | 应用于容积视频多轨道封装模式，且容积视频码流包含多个图集轨道。样本入口类型为'v3cb'的容积视频图集轨道作为基础轨道。 |
| v3a1 | 8.2.2.8.2 | 应用于容积视频多轨道封装模式，且容积视频码流包含多个图集轨道。样本入口类型为'v3al'的容积视频图集轨道下，所有图集子码流的参数集和SEI信息应存放在setupUnit数组中。 |
| v3ag | 8.2.2.8.2 | 应用于容积视频多轨道封装模式，且容积视频码流包含多个图集轨道。样本入口类型为'v3ag'的容积视频图集轨道下，图集子码流的参数集和SEI信息可以存放在setupUnit数组中，或者也可以存放在图集的子码流中。 |
| v3t1 | 8.2.2.8.3 | 应用于容积视频多轨道封装模式，为容积视频图集分片轨道的样本入口类型。 |
| dyvm | ‎8.3.5.2 | 应用于容积视频动态空间信息的定时元数据轨道。 |

6.3.2 轨道参考类型

本文档涉及的轨道索引类型如表2所示。

表 2 轨道索引类型表

| **轨道索引类型** | **章节** | **描述** |
| --- | --- | --- |
| v3cs | 8.2.2.8 | 被索引的轨道为容积视频图集轨道，用于从样本入口类型为'v3cb'的轨道链接至样本入口类型为'v3a1'或'v3ag'的轨道。 |
| v3ct | ‎8.2.2.8 | 被索引的轨道为容积视频图集分片轨道，用于从样本入口类型为'v3c1', 'v3cg', 'v3a1'或'v3ag'的轨道链接至样本入口类型为'v3ct'的轨道。 |
| v3vo | 8.2.2.8 | 被索引的轨道为携带占位图数据的容积视频组件轨道，用于从样本入口类型为'v3c1', 'v3cg', ‘v3a1’, ’v3ag’或'v3t1'的轨道链接至视频组件轨道。 |
| v3vg | ‎8.2.2.8 | 被索引的轨道为携带几何数据的容积视频组件轨道，用于从样本入口类型为'v3c1', 'v3cg', ’v3a1’, ’v3ag’或'v3t1'的轨道链接至视频组件轨道。 |
| v3va | ‎8.2.2.8 | 被索引的轨道为携带属性数据的容积视频组件轨道，用于从样本入口类型为'v3c1', 'v3cg', ’v3a1’, ’v3ag’或'v3t1'的轨道链接至视频组件轨道。 |
| v3vp | 8.2.2.8 | 被索引的轨道为携带封装数据的容积视频组件轨道，用于从入口类型为'v3c1', 'v3cg', ’v3a1’, ’v3ag’或'v3t1'的轨道链接至视频组件轨道。 |
| vdep | 8.2.3.5.3 | 被索引的轨道为自由视角视频的纹理图轨道，用于从深度图轨道链接至纹理图轨道，且深度图和纹理图之间存在编解码依赖。 | |
| vdpi | 8.2.3.5.3 | 被索引的轨道为自由视角视频的纹理图轨道，用于从深度图轨道链接至纹理图轨道，且深度图和纹理图之间不存在编解码依赖。 | |
| cvtr | 8.2.3.5.3 | 被索引的轨道为原始相机视频图像所在的媒体轨道，用于从虚拟相机视频图像所在的媒体轨道链接至原始相机视频图像所在的媒体轨道。 | |

6.3.3 轨道组类型

本文档涉及的轨道组类型如表3所示。

表 3 轨道组类型表

|  |  |  |
| --- | --- | --- |
| **轨道组类型** | **章节** | **描述** |
| vwpt | 8.2.1.5 | 用于关联全景视频视点群组中的各轨道。 |
| icrg | 8.2.1.11 | 用于关联全景视频独立编解码区域组中的各轨道。 |
| vtcg | ‎8.2.2.9 | 用于关联容积视频图集分片组件轨道组中的各轨道 |
| vwtg | 8.2.2.14 | 用于关联容积视频多视图轨道组中的各轨道。 |
| afvg | 8.2.3.6 | 用于关联自由视角轨道组中的各轨道。 | |

沉浸媒体处理

7.1基于全景视频的处理

7.1.1基于全景视频处理的基本过程

本小节描述了全景视频处理的基本过程。图3展示了内容制作过程中从球面图像到封装图像的转换，以及终端播放器渲染过程中从封装图像到球面图像的对应转换。该过程也可适用于对非时序的全景图像的处理。

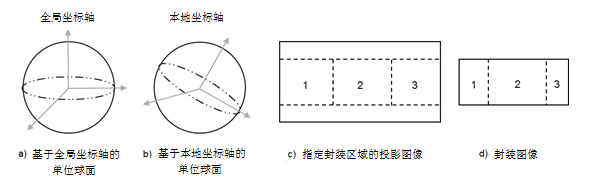


图 3 球面图像到封装图像及反向处理过程示例

1. 内容制作包含如下步骤：
2. 采集输入的多幅源图像被拼接成一幅全局坐标轴单位球面上的球面图像，如图3a所示；
3. 单位球面在全局坐标轴上进行旋转，如图3b所示。通过旋转角度来表示本地坐标轴到全局坐标轴转换的旋转量，旋转角度在RotationBox中指定。单位球面的本地坐标轴是经过旋转后的坐标系统的轴。如果本地坐标轴和全局坐标轴相同时，则不需要RotationBox。
4. 如图3c所示，将旋转后的单位球面上的球面图像转换成一幅二维的投影图像，如使用章节7.1.4.1的经纬图投影。当采用立体内容的空间封装时，双目视图对应的两幅球面图像被转换成两幅构成图像，随后，使用帧封装将两幅构成图像封装成一幅投影图像。
5. 使用矩形区域封装从投影图像中得到一幅封装图像。在图3c和3d中描述了一个封装示例，图3c的虚线矩形指示在一幅投影图像中的多个投影区域，图3d中的各区域指示对应的各封装区域。本示例中，对投影区域1和3进行下采样，而投影区域2保持其原始分辨率。
6. 终端播放器将封装图像的多个采样位置（如图3d）映射到一个单位球面上（如图3a），进行渲染播放，可执行如下步骤：
7. 从一个视频轨道或图形数据项中解码图像获取如图3d所示的一幅封装图像；
8. 如果指示采用区域封装，封装图像的多个采样位置被转换为各自投影图像的采样位置，如图3c所示。否则，投影图像等同于封装图像；
9. 如果指示投影图像采用空间的帧封装，投影图像的多个样本位置被转换为投影图像的各自构成图像的样本位置。否则，投影图像对应的构成图像即是投影图像本身；
10. 投影图像的采样位置被转换为相对本地坐标轴的球面坐标。这个最终的样本位置对应于一个球面图像，如图3b所示。

如果指示了坐标轴旋转，本地坐标轴的球面坐标被转换为全局坐标轴的球面坐标。否则，全局坐标轴与本地坐标轴相同。

7.1.2全景视频坐标系统

全景视频坐标系统由一个单位球面和三条坐标轴组成，三条坐标轴即是X轴（从后往前）、Y轴（横向，从左往右）、和Z轴（垂直，从下往上），在球中心交汇。

在球面上的任意一点可通过一对球面坐标方位角 φ和俯仰角θ表示。图4指出了球面坐标方位角φ和俯仰角θ与X、Y和Z坐标轴的关系。

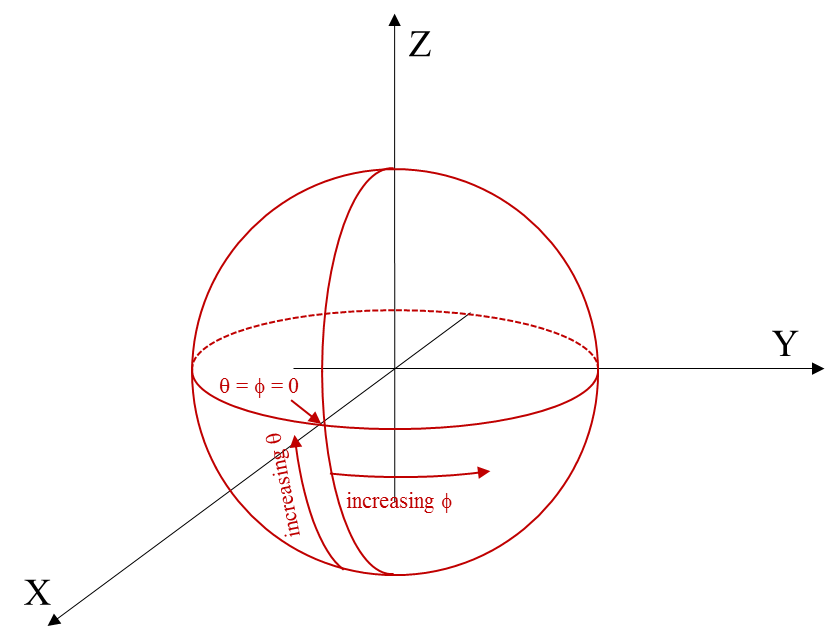


图 4 坐标轴及与球面坐标的关系

方位角φ的取值范围[-180°,180°)，俯仰角θ的取值范围[-90°,90°]。

7.1.3全局与本地坐标轴转换

本小节规定了全景视频坐标系统（7.1.2节规定）中，本地坐标轴及相应球面坐标向全局坐标轴及相应球面坐标的转化过程。

坐标轴转换过程的参数输入：

——3D\_rotation\_type，指示表达三维旋转的方式。

——rotation\_quat\_x，rotation\_quat\_y，rotation\_quat\_z字段取值范围为[-230, 230]。

——rotation\_yaw (αd), rotation\_pitch (βd), rotation\_roll (γd), 以度为单位，其中rotation\_yaw (αd) 和rotation\_roll (γd)的范围为[−180°,180°)，rotation\_pitch (βd) 的范围为[−90°,90°]；

——本地坐标轴的球面坐标(ϕd, θd)，以度为单位。

坐标轴转换过程的参数输出：

——全局坐标轴的球面坐标(ϕ′, θ′)，以度为单位。

如图5所示，坐标轴转换过程规定了沿着全景视频坐标系统的X、Y、Z轴的旋转，从坐标轴的原点朝着正方向端看时，旋转角度沿着顺时针增加，roll (γd)表示沿着X轴的旋转角度，pitch (βd) 表示沿着Y轴的旋转角度，yaw (αd)表示沿着Z轴的旋转角度。



图 5 坐标轴旋转示例图

在内容制作时，不同类型的媒体需要完成在全局坐标系下的统一对齐。当任何yaw (αd), pitch (βd) 和roll (γd) 旋转角度不等于0或者四元数中任何x,y,z,w分量取值不等于0时，终端播放器需要按照本章节描述的方法将本地坐标轴转换成全局坐标轴。转换过程和结果如下：

ϕ = ϕd × ÷ 180  
θ = θd × ÷ 180  
α = αd × ÷ 180  
β = βd × ÷ 180  
γ = γd × ÷ 180

qX = rotation\_quat\_x ÷ 230,

qY = rotation\_quat\_y ÷ 230

qZ = rotation\_quat\_z ÷ 230

qW =

x1 = ×   
y1 = ×   
z1 =

if(3D\_rotation\_type == 0){

x2 = × × x1 − × × y1 + × z1   
y2 = ( × + × × ) × x1 +  
 ( × − × × ) × y1 −  
 ×× z1  
z2 = ( × − × × ) × x1 +  
 ( × + × × ) × y1 +  
 × × z1

}

if(3D\_rotation\_type == 1){

x2 = (1-2(qY2+qZ2) × x1) + 2(qX × qY- qW × qZ) × y1+ 2(qX × qZ+qW × qY) × z1

y2 =2(qX × qY + qW × qZ) × x1 + (1-2(qX2+qZ2) × y1+ 2(qY × qZ - qW × qX) × z1

z2 =2(qX × qZ - qW × qY) × x1 + 2(qY × qZ + qW × qX) × y1+(1-2(qX2+qY2) × z1

}  
ϕ′ = Atan2( y2, x2 ) × 180 ÷ π  
θ′ = × 180 ÷ π

7.1.4投影

7.1.4.1经纬图投影

本节定义了将平面视频帧投影到球面的经纬图投影过程，并给出了映射表达式。像素在球面视频帧中的坐标记为（θ, φ），其中θ是方位角，φ是俯仰角。θ的取值范围是[-π, π]，φ的取值范围是[-π, π]

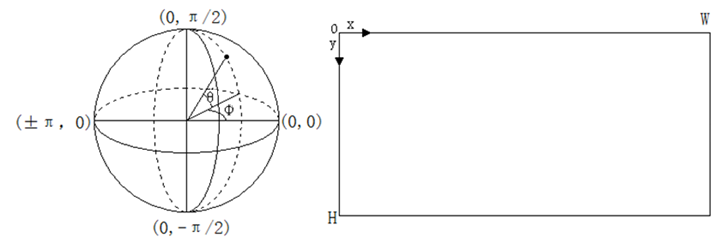


图 6 球面视频与平面视频的坐标关系

平面视频帧是从一个完整的球面视频帧经过对方位角φ和俯仰角θ进行均匀采样得到的。对于一个完整的球面，φ的取值范围是[-π,π)，θ的取值范围是[-π/2,π/2]，如图6所示。

经纬图投影过程的输入：

——平面视频帧的宽度W和高度H

——像素在平面视频帧中的坐标(x,y)

经纬图投影过程的输出：

——像素对应的球面坐标(φ, θ)

平面视频帧中像素的平面坐标(x,y)与球面坐标(ϕ, θ)的对应关系为：

φ = (0.5 - x ÷ W) × 2π

θ = (0.5 - y ÷ H) × π

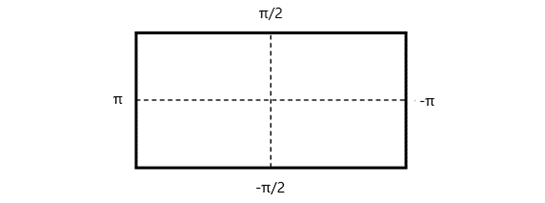


图 7 经纬图投影的平面投影图像的方位角和俯仰角范围

图7展示了经纬图投影的平面投影图像的方位角和俯仰角范围。

注1：输入参数中采样位置的中心点以及样本图片的宽度和高度均为非零值。因此，输出参数φ不能取到-π或π，θ不能取到-π/2或π/2。

注2：平面投影图像表示从坐标系的原点观察到的单位球体的内表面。因此，方位角从左向右逐渐减小。

7.1.4.2 正六面体投影

本小节定义了正六面体映射的球面和平面间的映射关系，通过将一个完整的球面映射为正六面体，并将正六面体的六个面按某种方式排列，建立了像素的局部坐标（xLocal，yLocal）和球面坐标（θ, φ）的之间的关系，并给出了映射表达式。

要将一个完整的球面映射为正六面体，首先需确定球面的外切正六面体，如图8所示，六个面与球面切点的球面坐标分别为（-π/2，0）、（0，0）、（π/2，0）、（π，0）、（0，π/2）和（0，-π/2），将这六个切点所在的正方形切面分别记为左面、 前面、 右面、 背面、 顶面和底面。

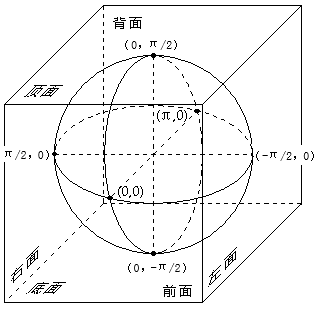


图 8 正六面体采样示意图

连接某个切面内的所有点和球面中心，与球面的所有交点构成的区域是该切面对应的球面区域。将某个切面内像素的局部坐标记为（xLocal，yLocal）。将每个正方形切面的边长用nCubic表示，定义中间变量xNormal和yNormal的值为：

xNormal = (2×xLocal+1-nCubic)÷nCubic

yNormal = (2×yLocal+1-nCubic)÷nCubic

各切面中像素的局部坐标（xLocal，yLocal）和球面坐标（θ, φ）的对应关系如下：

1. 左面：

φ = arctg(xNormal) -π÷2

θ = arctg(yNormal÷sqrt(1+xNormal2))

1. 前面：

φ = arctg(xNormal)

θ = arctg(yNormal÷sqrt(1+xNormal2))

1. 右面：

φ = arctg(xNormal)+π÷2

θ = arctg(yNormal÷sqrt(1+xNormal2))

1. 背面：

θ = arctg(yNormal÷sqrt(1+xNormal2))

当xNormal < 0时，

φ = arctg(xNormal)+π

当xNormal >= 0时，

φ = arctg(xNormal)-π

1. 顶面：

θ = π÷2-arctg(sqrt(xNormal2+yNormal2))

当xNormal >= 0时，

φ = arccos(yNormal÷sqrt(xNormal2+yNormal2))

当xNormal<0时，

φ = -arccos(yNormal÷sqrt(xNormal2+yNormal2))

1. 底面：

θ = arctg(sqrt(xNormal2+yNormal2))-π÷2

当xNormal >= 0时，

φ = arccos(-yNormal÷sqrt(xNormal2+yNormal2))

当xNormal<0时，

φ = -arccos(-yNormal÷sqrt(xNormal2+yNormal2))

正六面体六个面的排列类型如表4所示，各种排列类型的形状及各个面的排列方式如图9所示。

表 4 正六面体面的排列类型

|  |  |  |
| --- | --- | --- |
| **排列类型的值** | **排列类型** | **边长** |
| 0 | 1×6 | W |
| 1 | 2×3 | W/2 |
| 2 | 3×2 | W/3 |
| 3 | 6×1 | W/6 |

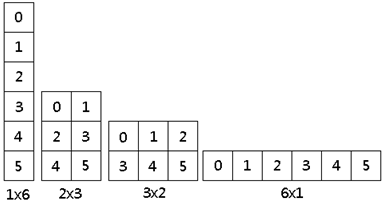


图 9正六面体排列类型及位置索引

7.1.5全景视频封装格式

##### 7.1.5.1 概述

本章节定义了全景视频的区域封装过程。全景视频的区域封装过程将采样点在映射区域的位置信息对应到采样点在封装区域的位置信息，解封装时再进行反向对应。

区域封装格式的标识及参考章节如表5所示。

表 5 区域封装格式

|  |  |  |  |
| --- | --- | --- | --- |
| **标识** | | **区域封装格式** | **章节** |
| 0 | | 矩形区域封装格式 | 7.1.5.2 |
| 1..15 | 保留 | N/A |

7.1.5.2 矩形区域封装的采样位置转换

本小节规定了采样点在封装区域的位置信息反向转换对应到映射区域的过程。

输入参数：

——采样点在封装区域的位置信息(x, y)

——映射区域的宽度和高度(projRegWidth, projRegHeight)

——封装区域的宽度和高度(packedRegWidth, packedRegHeight)

——转换类型(transformType)，具体定义参见8.1.1.5.3节

——采用点在封装图像中的相对偏移量(offsetX, offsetY)，取值范围为[0,1)

注: offsetX 和 offsetY 都等于0.5 时，表示采样点在封装图像的正中心

输出结果：

——映射区域的采样位置中心点(hPos, vPos)

转换过程：

if( transformType  = =  0  | |  transformType  = =  1  | |  transformType  = =  2  | |  transformType  = =  3 ) {  
 horRatio = projRegWidth ÷ packedRegWidth  
 verRatio = projRegHeight ÷ packedRegHeight  
} else if ( transformType  = =  4  | |  transformType  = =  5  | |  transformType  = =  6  | |  
 transformType  = =  7 ) {  
 horRatio = projRegWidth ÷ packedRegHeight  
 verRatio = projRegHeight ÷ packedRegWidth  
}  
if( transformType = = 0 ) {  
 hPos = horRatio ×( x + offsetX )  
 vPos = verRatio × ( y + offsetY )  
} else if ( transformType = = 1 ) {  
 hPos = horRatio × ( packedRegWidth − x − offsetX )   
 vPos = verRatio × ( y + offsetY )  
} else if ( transformType = = 2 ) {  
 hPos = horRatio × ( packedRegWidth − x − offsetX )  
 vPos = verRatio × ( packedRegHeight − y − offsetY )  
} else if ( transformType = = 3 ) {  
 hPos = horRatio × ( x + offsetX )  
 vPos = verRatio × ( packedRegHeight − y − offsetY )  
} else if ( transformType = = 4 ) {  
 hPos = horRatio × ( y + offsetY )  
 vPos = verRatio × ( x + offsetX )  
} else if ( transformType = = 5 ) {  
 hPos = horRatio × ( y + offsetY )  
 vPos = verRatio × ( packedRegWidth − x − offsetX )  
} else if ( transformType = = 6 ) {  
 hPos = horRatio × ( packedRegHeight − y − offsetY )  
 vPos = verRatio × ( packedRegWidth − x − offsetX )  
} else if ( transformType = = 7 ) {  
 hPos = horRatio × ( packedRegHeight − y − offsetY )  
 vPos = verRatio ×( x+ offsetX )  
}

7.2 基于容积视频的处理

7.2.1 基于容积视频的处理过程

容积视频的处理过程涉及容积视频的内容制作和终端播放器的处理，内容制作端将容积视频从三维的表示转换成二维的表示，终端播放器的处理将二维表示的容积视频重构到三维空间。图10展示了容积视频的二维表示。

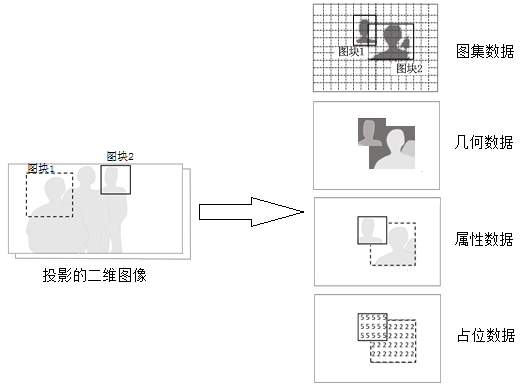


图 10容积视频的二维表示的示例

容积视频内容制作包含如下步骤：

1. 将采集输入的容积视频的三维表示数据投影到二维平面，通常采用正交投影、透视投影、ERP投影方式，投影到二维平面的容积视频通过几何组件、占位组件和属性组件数据来表示，其中，几何组件数据提供容积视频每个点在三维空间中的位置信息，属性组件数据提供容积视频每个点的额外属性（如纹理或材质信息），占位组件数据指示其他组件中的数据是否与容积视频关联；
2. 对容积视频的二维表示的组件数据进行处理生成图块，根据几何组件数据中表示的容积视频的位置，将容积视频的二维表示所在的二维平面区域分割成多个不同大小的矩形区域，一个矩形区域为一个图块，图块包含将该矩形区域反投影到三维空间的必要信息；
3. 打包图块生成图集，将图块放入一个二维网格中，并保证各个图块中的有效部分是没有重叠的。一个容积视频生成的图块可以打包成一个或多个图集；
4. 基于图集数据生成对应的几何数据、属性数据和占位数据，将图集数据、几何数据、属性数据、占位数据组合形成容积视频在二维平面的最终表示。

注：几何组件为必选，占位组件为条件必选，属性组件为可选。

终端播放器将容积视频重构到三维空间，可执行如下步骤：

1. 解封装/解码容积视频，从一个或多个容积视频的媒体轨道提取码流，包括图集/图集分片码流和关联的视频编码组件码流，并解码媒体轨道中提取的码流获得图集数据以及关联的容积视频组件的数据，即属性数据、几何数据、占位数据，或封装视频数据；
2. 重构容积视频，根据图集数据（描述了图块信息和图块与容积视频三维空间对应关系），以及图集关联的容积视频组件的数据，将容积视频的二维表示重构生成容积视频的三维空间场景。

7.2.2 容积视频的部分访问

容积视频的部分访问是对三维空间中容积视频的某一部分或某一对象的访问，需要对容积视频所在的三维空间区域进行划分，根据划分部分的三维空间每个点与二维平面映射关系，获取划分部分的三维空间区域对应的容积视频的二维组件数据。观看容积视频的某一空间区域或某一对象时，通过确定容积视频的一个三维空间区域以及该三维空间区域中的点投影在图集帧上的一个二维区域，根据二维区域中包含的图集数据以及相应容积视频组件数据(即视频编码几何组件、占位组件和属性组件数据，或者封装视频数据)，重建容积视频的三维区域。

对于容积视频的三维空间及其划分表示通过边界框和空间区域来表示容积视频所在的位置、大小等信息。其中，边界框为一个立方体，表示容积视频所在的空间大小。空间区域通过立方体的位置坐标和大小表示，表示一个容积视频的某一部分或某一对象所在的三维空间。空间区域包含在三维边界框中，一个三维边界框中可以存在多个空间区域。

7.3 基于自由视角视频的处理

自由视角视频源是以多相机阵列采集或合成虚拟相机得到的空间不同视角的视频组合。为了便于自由视角视频的表达，存储，压缩和处理，本标准将自由视角视频数据表达为多相机采集的纹理图，多相机纹理图所对应的深度图，以及相应的自由视角视频内容描述元数据的组合，其中，元数据中包含了多相机的参数，以及自由视角视频的拼接布局和边缘保护等描述信息。自由视角视频数据表达如图 11所示。

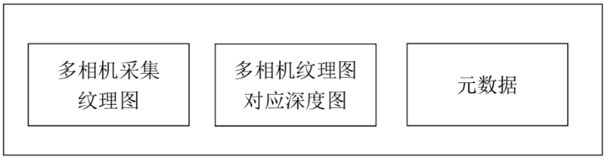


图 11 自由视角视频数据表达

1. 通过多相机矩阵阵列采集得到多个原始相机的视频图像，包括纹理图和深度图。
2. 通过合成得到多个虚拟相机的视频图像，即根据原始相机采集的视频图像预先合成虚拟相机视频图像，通常一个虚拟相机视频图像至少需要两个原始相机图像合成。（可选）
3. 将多个原始相机的纹理图信息和对应的深度图信息进行拼接处理。（可选）
4. 通过平面视频压缩方式对多个相机（原始相机、虚拟相机）的图像信息进行编码。
5. 将上述过程的自由视角视频数据和元数据描述信息封装生成自由视角视频的媒体文件。

终端播放器将自由视角视频重构到虚拟视角，可执行如下步骤：

1. 获取自由视角视频的媒体文件，根据目标视角及元数据信息，从视频文件中解码获得一个或多个相机的视频图像信息。
2. 处理解码的自由视角视频数据，根据自由视角视频算法（可采用应用自身的算法），结合深度图和纹理图的信息，合成用户所切换的目标视角对应的虚拟视点。如果内容制作时已预先合成虚拟相机的视频图像，目标视角对应自由视角视频中的原始相机或虚拟相机，可直接使用解码后的自由视角视频数据。

沉浸媒体文件格式

8.1 基本结构和语义

8.1.1基于全景视频的结构和语义

8.1.1.1 投影格式结构

###### 8.1.1.1.1语法

aligned(8) class ProjectionFormatStruct() {  
 unsigned int(5) projection\_type;

bit(3) reserved = 0;  
}

8.1.1.1.2语义

projection\_type指示投影图像到球面坐标系统的映射类型，projection\_type的取值如表6所示：

表 6 全景视频投影格式

|  |  |  |
| --- | --- | --- |
| **表示值** | **投影方式** | **章节** |
| 0 | 经纬图投影 | 7.1.4.1 |
| 1 | 正六面体投影 | 7.1.4.2 |
| 2..31 | 保留 | N/A |

8.1.1.2 旋转结构

8.1.1.2.1定义

该结构提供本地坐标轴转换到全局坐标轴所需的旋转信息，旋转信息以欧拉角或者四元数的方式表示。在立体的全景视频情况下，该旋转结构适用于每一幅双目视图。

8.1.1.2.2语法

aligned(8) class RotationStruct() {

unsigned int(8) 3D\_rotation\_type;

if(3D\_rotation\_type == 0){  
 signed int(32) rotation\_yaw;  
 signed int(32) rotation\_pitch;  
 signed int(32) rotation\_roll;  
 }  
 if(3D\_rotation\_type == 1){  
 signed int(32) rotation\_quat\_x;  
 signed int(32) rotation\_quat\_y;  
 signed int(32) rotation\_quat\_z;  
 }

}

8.1.1.2.3语义

3D\_rotation\_type 指示旋转信息的表示类型。该字段取值为0表示旋转信息以欧拉角的形式给出；该字段取值为1表示旋转信息以四元数的形式给出。其余取值保留。

rotation\_yaw, rotation\_pitch 和 rotation\_roll分别指示沿着X轴、Y轴和Z轴旋转的偏航角度，俯仰角度和翻滚角度，以2−16度为单位。 其中rotation\_yaw字段取值范围为[−180\*216 , 180\*216-1]， rotation\_pitch字段取值范围为[−90\*216 , 90\*216]，rotation\_roll字段取值范围为[−180\*216 , 180\*216–1]。

rotation\_quat\_x, rotation\_quat\_y和rotation\_quat\_z用于计算坐标轴旋转四元数中的qX， qY， qZ分量，取值范围为[-230, 230]。各分量的计算规则如下：

——qX = rotation\_quat\_x ÷ 230, qY = rotation\_quat\_y ÷ 230, qZ = rotation\_quat\_z ÷ 230

——第4个分量qW可按照四元数的数学特性推导得到:

qW = Sqrt( 1 – ( qX2+ qY2 + qZ2 ) )

——旋转四元数各分量缺省值为0。

——四元数(w, x, y, z) 代表围绕向量(x, y, z) 以角度 进行旋转。

8.1.1.3 内容覆盖结构

8.1.1.3.1定义

内容覆盖结构提供了一个或多个基于全局坐标的球面区域信息，该球面区域为相应媒体内容所覆盖的区域。

8.1.1.3.2语法

aligned(8) class ContentCoverageStruct() {  
 unsigned int(8) coverage\_shape\_type;  
 unsigned int(8) num\_regions;  
 unsigned int(1) view\_idc\_presence\_flag;  
 if (view\_idc\_presence\_flag == 0) {  
 unsigned int(2) default\_view\_idc;  
 bit(5) reserved = 0;  
 } else {  
 bit(7) reserved = 0;  
 }  
 for ( i = 0; i < num\_regions; i++) {  
 if (view\_idc\_presence\_flag == 1) {  
 unsigned int(2) view\_idc[i];  
 bit(6) reserved = 0;  
 }  
 SphereRegionStruct(1);  
 }  
}

8.1.1.3.3语义

coverage\_shape\_type，指出表示内容覆盖的球面区域形状。coverage\_shape\_type和shape\_type有相同的语义。当区域封装结构适用于ContentCoverageStruct中的语义时，coverage\_shape\_type的值表示形状类型值。

num\_regions，指出球面区域的数量，数值0保留。

view\_idc\_presence\_flag取值为0时表示view\_idc[i]不存在。view\_idc\_presence\_flag取值为1时表示view\_idc[i]存在，并指明该球面区域与其所指的具体(左眼, 右眼, 或双眼)视图的关联。

default\_view\_idc取值为0时表示每个球面区域为单视场；取值为1时表示球面区域是左眼视图的立体内容；取值为2时表示球面区域是右眼视图的立体内容；取值为3时表示球面区域包含左眼和右眼的视图。

view\_idc[i]取值为1时表示第i个球面区域在一个立体内容的左眼视图上；取值为2时表示第i个球面区域在一个立体内容的右眼视图上；取值为3时表示第i个球面区域同时在左眼和右眼视图上。view\_idc[i]取值为0保留。

注：view\_idc\_presence\_flag 为1可指明是非对称的立体覆盖。例如，一个非对称立体覆盖， num\_regions可设置为2，指出一个球面区域在左边覆盖是从−π/2 到π/2的方位角，而指出另一个球面区域在右边覆盖是从-π/3 到π/3的俯仰角。

当ContentCoverageStruct()中包含SphereRegionStruct(1)时，需要使用球面区域的结构，同时interpolate应取值为0。

内容覆盖区域通过num\_regions 及其相应的SphereRegionStruct(1) 的合集来指出，当num\_regions大于1时，内容覆盖可能是非连续的。

8.1.1.4 球面区域结构

8.1.1.4.1定义

球面区域结构（SphereRegionStruct）指定了全景视频单位球面上的一个球面区域。

当centre\_tilt为0时，球面区域通过如下结构指定：

——如果azimuth\_range 和 elevation\_range为0，该结构指定的球面区域是球形表面的一个点

——否则，球面区域通过变量centreAzimuth, centreElevation, cAzimuth1, cAzimuth2, cElevation1和cElevation2来定义：

·centreAzimuth = centre\_azimuth ÷ 65536

·centreElevation = centre\_elevation ÷ 65536

·cAzimuth1 = ( centre\_azimuth – azimuth\_range ÷ 2 ) ÷ 65536

·cAzimuth2 = ( centre\_azimuth + azimuth\_range ÷ 2 ) ÷ 65536

·cElevation1 = ( centre\_elevation – elevation\_range ÷ 2 ) ÷ 65536

·cElevation2 = ( centre\_elevation + elevation\_range ÷ 2 ) ÷ 65536

参考在SphereRegionStruct实例的结构的语义中指定的形状类型值（coverage\_shape\_type），球面区域定义如下：

——当形状类型值为0时，球面区域通过cAzimuth1, cAzimuth2, cElevation1, cElevation2定义的四个大圆，和由centreAzimuth 和centreElevation定义的中心点来指定，如图 12所示

——当形状类型值为1时，球面区域通过由cAzimuth1, cAzimuth2, cElevation1, cElevation2四点定义的两个方位角圆和两个俯仰角圆，和通过centreAzimuth和centreElevation定义的中心点来指定，如图 13所示

——当centre\_tilt不为0时，首先由上述定义得到球面区域，然后沿着从球心穿过球面区域中心点的轴对该球面区域倾斜旋转，得到最终位置的球面区域。当从轴的起始端朝着正方向端看时，倾斜旋转的角度值顺时针方向增加。

——形状类型值为0时指定的由四个大圆确定的球形区域如图 12所示。

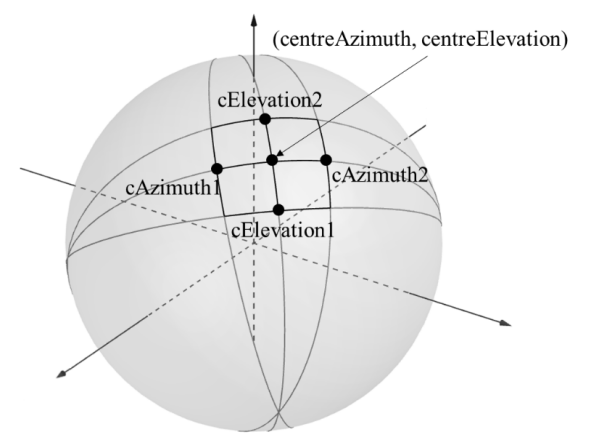


图 12由四个大圆确定的球面区域

——形状类型值为1时指定由两个方位角圆和两个俯仰角圆确定的球面区域如图 13所示：

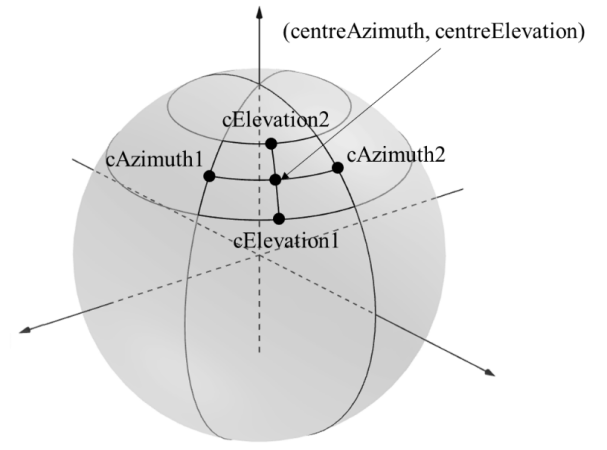


图 13 由两个方位角圆和两个俯仰角圆确定的球面区域

8.1.1.4.2语法

aligned(8) SphereRegionStruct(range\_included\_flag) {  
 signed int(32) centre\_azimuth;  
 signed int(32) centre\_elevation;  
 singed int(32) centre\_tilt;  
 if (range\_included\_flag) {  
 unsigned int(32) azimuth\_range;  
 unsigned int(32) elevation\_range;  
 }  
 unsigned int(1) interpolate;  
 bit(7) reserved = 0;  
}

8.1.1.4.3语义

centre\_azimuth 和centre\_elevation，分别指示以2−16为精度的球面区域中心点的方位角和俯仰角的值。centre\_azimuth的范围是[−π\* 216,π\* 216−1]。centre\_elevation的范围是[−π/2\*216,π/2 \* 216-1]

centre\_tilt指示以2−16为精度的球面区域的倾斜角角度，范围是[−180°\*216,180°\*216−1]。

azimuth\_range 和elevation\_range，如果存在，分别指示以2−16为精度的球面区域中方位角和俯仰角范围。azimuth\_range 和elevation\_range指示通过球面区域中心的范围，如图11和12所示。当azimuth\_range 和elevation\_range不存在于SphereRegionStruct的实例中，则在包含SphereRegionStruct实例的结构语义中指定。azimuth\_range的范围是[0,2π\*216]，elevation\_range的范围是[0,π\*216]。

interpolate在包含SphereRegionStruct实例的结构语义中指示。

8.1.1.5 区域封装结构

8.1.1.5.1定义

区域封装结构(RegionWisePackingStruct) 规定了封装区域与对应投影区域之间的映射关系和保护带的位置和大小。

本节中解码图像根据不同场景在语义上分别表示为：

——视频场景下表示解码视频轨中其中一幅采样图像；

——图片场景下表示一个解码重构的图片。

区域封装结构(RegionWisePackingStruct)的内容包括:

——投影图像的宽度proj\_picture\_width 和高度proj\_picture\_height；

——封装图像的宽度packed\_picture\_width 和高度 packed\_picture\_height；

——对于立体投影图像（即有左目和右目两个图像成分），当其帧封装方式确定为上下或左右并排时, constituent\_picture\_matching\_flag 等于1，表示在语法上对两个图像成分的投影区域信息和封装区域信息以及保护带区域信息单独应用：

·封装图像和投影图像有相同的立体封装格式；

·投影区域和封装区域的数量之和为语法结构中num\_regions值的两倍；

封装格式目前只定义了矩形区域封装结构RectRegionPacking(i)，其内容包括:

——proj\_reg\_width[i], proj\_reg\_height[i], proj\_reg\_top[i], 和 proj\_reg\_left[i] 分别表示第i个投影区域的宽度,高度,区域顶端的偏移量和左端的偏移量；

——transform\_type[i] 表示第i个封装区域重新映射到第i个投影区域的旋转和镜像等映射类型；

——packed\_reg\_width[i], packed\_reg\_height[i], packed\_reg\_top[i], 和 packed\_reg\_left[i] 分别表示第i个封装区域的宽度,高度,区域顶端的偏移量和左端的偏移量；

保护带格式定义为GuardBand(i)，其内容包括:

——left\_gb\_width[i], right\_gb\_width[i], top\_gb\_height[i]和 bottom\_gb\_height[i] 分别表示第i个封装区域周围左侧,右侧,上侧和下侧保护带的尺寸大小；

——gb\_not\_used\_for\_pred\_flag[i]表示第i个封装区域在图像编码时在预测流程中未使用保护带信息；

——gb\_type[i][j]表示了第i个封装区域的保护带类型；

图 14中左图举例了一个投影图像中一个投影区域的位置和大小的描述方法, 右图举例了一个封装图像中一个封装区域和其保护带的位置和大小的描述方法。

****

图 14 投影区域与对应封装区域及保护带示意图

8.1.1.5.2语法

aligned(8) class RegionWisePackingStruct() {  
 unsigned int(1) constituent\_picture\_matching\_flag;  
 bit(7) reserved = 0;  
 unsigned int(8) num\_regions;  
 unsigned int(32) proj\_picture\_width;  
 unsigned int(32) proj\_picture\_height;  
 unsigned int(16) packed\_picture\_width;  
 unsigned int(16) packed\_picture\_height;  
 for (i = 0; i < num\_regions; i++) {  
 bit(3) reserved = 0;  
 unsigned int(1) guard\_band\_flag[i];  
 unsigned int(4) packing\_type[i];  
 if (packing\_type[i] == 0) {  
 RectRegionPacking(i);  
 if (guard\_band\_flag[i])  
 GuardBand(i);  
 }  
 }  
}

其中类RectRegionPacking的语法为:

aligned(8) class RectRegionPacking(i) {  
 unsigned int(32) proj\_reg\_width[i];  
 unsigned int(32) proj\_reg\_height[i];  
 unsigned int(32) proj\_reg\_top[i];  
 unsigned int(32) proj\_reg\_left[i];  
 unsigned int(3) transform\_type[i];  
 bit(5) reserved = 0;  
 unsigned int(16) packed\_reg\_width[i];  
 unsigned int(16) packed\_reg\_height[i];  
 unsigned int(16) packed\_reg\_top[i];  
 unsigned int(16) packed\_reg\_left[i];  
}

类GuardBand的语法为:

aligned(8) class GuardBand(i) {  
 unsigned int(8) left\_gb\_width[i];  
 unsigned int(8) right\_gb\_width[i];  
 unsigned int(8) top\_gb\_height[i];  
 unsigned int(8) bottom\_gb\_height[i];  
 unsigned int(1) gb\_not\_used\_for\_pred\_flag[i];  
 for (j = 0; j < 4; j++)  
 unsigned int(3) gb\_type[i][j];  
 bit(3) reserved = 0;  
}

8.1.1.5.3语义

constituent\_picture\_matching\_flag 等于1表示对投影图像的两个图像成分其投影区域信息和封装区域信息在语法上单独应用而且封装图像和投影图像有相同的立体封装格式, 且投影区域和封装区域的数量之和为语法结构中num\_regions值的两倍。 constituent\_picture\_matching\_flag 等于 0 表示投影区域信息和封装区域信息对应于整个投影图像。当SpatiallyPackedStereoFlag为0时, constituent\_picture\_matching\_flag应等于0。

注1：当对两个图像成分使用区域封装立体内容时，此标识设置为1可以允许区域封装信息信令更加简洁。

constituent\_picture\_matching\_flag 为0时num\_regions标识封装区域的数量，此时num\_regions的0值意义保留。当constituent\_picture\_matching\_flag为1时，封装区域的总数量等于2 \* num\_regions，每一个RectRegionPacking(i) 描述一个图像成分对应的投影图像和封装图像的信息。

proj\_picture\_width 和 proj\_picture\_height 表示相应投影图像的宽度和高度， proj\_picture\_width 和 proj\_picture\_height 的值都应大于 0， 单位为图像样本的采样单元。

注2：对投影图像亮度和色度的采样阵列也会使用这里采样单元。

packed\_picture\_width和packed\_picture\_height表示相应封装图像的宽度和高度， packed\_picture\_width和packed\_picture\_height的值都应大于 0。

packing\_type[i] 表示区域封装类型, 取值遵循表5中的定义。

RectRegionPacking(i) 表示了第i个投影区域与对应第i个封装区域之间的区域封装映射。

proj\_reg\_width[i], proj\_reg\_height[i], proj\_reg\_top[i], 和 proj\_reg\_left[i]分别表示第i个投影区域的宽度,高度,区域顶端的偏移量和左端的偏移量。

注3：两个投影区域可能会部分或者完全重合，比如此时两个投影区域标识了质量等级不同， 而对于重合部分应采用相对高质量投影对应的封装区域用来做最后的渲染。

transform\_type[i] 表示第i个封装区域重新映射到第i个投影区域的旋转和镜像等映射类型，当transform\_type[i] 中既包括旋转也包括镜像时，先做旋转，后做镜像：

0: 无转换

1: 水平镜像

2: 顺时针旋转180度

3: 先顺时针旋转180度再水平镜像

4: 先顺时针旋转90度再水平镜像

5: 顺时针旋转90度

6: 先顺时针旋转270度再水平镜像

7: 顺时针旋转270度

注4：第7.1.5.2 节中定义了采样点在封装区域的位置信息如何反向转换对应到映射区域的过程。

packed\_reg\_width[i], packed\_reg\_height[i], packed\_reg\_top[i], 和 packed\_reg\_left[i] 分别表示第i个封装区域的宽度,高度,区域顶端的偏移量和左端的偏移量。packed\_reg\_width[i], packed\_reg\_height[i], packed\_reg\_top[i], 和 packed\_reg\_left[i] 应表示解码图像中亮度采样单元中的水平和垂直坐标整数值。

注5：两个封装区域可能部分或完全重合。

GuardBand(i) 表示了第i个封装区域的保护带 。

left\_gb\_width[i], right\_gb\_width[i] top\_gb\_height[i]和bottom\_gb\_height[i] 分别表示封装图像的第i个封装区域左侧, 右侧, 上侧和下侧保护带的宽度, 当GuardBand(i)存在时, 这4个值至少有1个大于0。

gb\_not\_used\_for\_pred\_flag[i] 等于1表示第i个封装区域在图像编码时在帧间预测流程中未使用保护带信息, 等于0表示不确定i个封装区域在图像编码时在帧间预测流程中是否使用保护带信息。

gb\_type[i][j]表示了第i个封装区域的保护带类型, j等于 0, 1, 2, 或 3时分别表示封装区域左侧, 右侧, 上侧和下侧保护带有如下定义的类型:

gb\_type[i][j] 等于0表示保护带的内容和封装区域内容间的关系不确定, 当gb\_not\_used\_for\_pred\_flag[i]等于0时gb\_type[i][j]不应等于0。

gb\_type[i][j] 等于 1表示保护带的内容满足在封装区域边界外侧和保护带像素之间插入子像素的条件，比如保护带可以是封装区域边界上像素的拷贝值。

gb\_type[i][j]等于2表示保护带的内容为封装区域实际相邻图像的实际值，但图像质量为相邻图像质量向封装区域图像质量的过渡。

gb\_type[i][j] 等于3表示保护带的内容为封装区域实际相邻图像的实际值，而且图像质量也为封装区域图像质量。

gb\_type[i][j]大于3的值为保留值。

相应数据变量NumRegions, PackedRegLeft[n], PackedRegTop[n], PackedRegWidth[n], PackedRegHeight[n], ProjRegLeft[n], ProjRegTop[n], ProjRegWidth[n], ProjRegHeight[n], TrasnformType[n], PackingType[n] 的定义如下：

n 取值从0到num\_regions – 1时：

PackedRegLeft[n] 等于 packed\_reg\_left[n].

PackedRegTop[n] 等于packed\_reg\_top[n].

PackedRegWidth[n] 等于packed\_reg\_width[n].

PackedRegHeight[n] 等于 packed\_reg\_height[n].

ProjRegLeft[n] 等于 proj\_reg\_left[n].

ProjRegTop[n] 等于 proj\_reg\_top[n].

ProjRegWidth[n] 等于 proj\_reg\_width[n].

ProjRegHeight[n] 等于 proj\_reg\_height[n].

TransformType[n] 等于 transform\_type[n].

PackingType[n] 等于 packing\_type[n].

如果constituent\_picture\_matching\_flag 等于 0：

NumRegions 等于 num\_regions.

否则 constituent\_picture\_matching\_flag 等于 1：

NumRegions 等于 2 \* num\_regions.

当TopBottomFlag 等于1时：

projLeftOffset 和 packedLeftOffset 都等于 0

projTopOffset等于proj\_picture\_height / 2， packedTopOffset 等于 packed\_picture\_height / 2

当 SideBySideFlag 等于 1时：

projLeftOffset 等于 proj\_picture\_width / 2 ，packedLeftOffset 等于 packed\_picture\_width / 2

projTopOffset 和 packedTopOffset 都等于0

n取值从NumRegions / 2 到 NumRegions – 1时：

nIdx 等于 n – NumRegions / 2.

PackedRegLeft[n] 等于 packed\_reg\_left[nIdx] + packedLeftOffset.

PackedRegTop[n] 等于 packed\_reg\_top[nIdx] + packedTopOffset.

PackedRegWidth[n] 等于 packed\_reg\_width[nIdx].

PackedRegHeight[n] 等于 packed\_reg\_height[nIdx].

ProjRegLeft[n] 等于 proj\_reg\_left[nIdx] + projLeftOffset.

ProjRegTop[n] 等于 proj\_reg\_top[nIdx] + projTopOffset.

ProjRegWidth[n] 等于 proj\_reg\_width[nIdx].

ProjRegHeight[n] 等于 proj\_reg\_height[nIdx].

TransformType[n] 等于 transform\_type[nIdx].

PackingType[n] 等于 packing\_type[nIdx].

n 取值从 0 到NumRegions – 1时，ProjRegWidth[n], ProjRegHeight[n], ProjRegTop[n], 和 ProjRegLeft[n] 的取值范围为：

ProjRegWidth[n] 取值从 1 到proj\_picture\_width，

ProjRegHeight[n] 取值从1 到 proj\_picture\_height，

ProjRegLeft[n] 取值从0 到 proj\_picture\_width – 1，

ProjRegTop[n] 取值从0 到 proj\_picture\_height – 1，

当ProjRegLeft[n] 的值比proj\_picture\_width / HorDiv1 小时, ProjRegLeft[n] 与 ProjRegWidth[n] 之和应小于等于proj\_picture\_width / HorDiv1 ，否则ProjRegLeft[n] 与 ProjRegWidth[n] 之和应小于等于proj\_picture\_width / HorDiv1  \* 2，

当ProjRegTop[n] 的值比proj\_picture\_height / VerDiv1小时, ProjRegTop[n] 与 ProjRegHeight[n] 之和应小于等于proj\_picture\_height / VerDiv1，否则ProjRegTop[n] 与 ProjRegHeight[n] 之和应小于等于proj\_picture\_height / VerDiv1 \* 2。

n 取值从0到NumRegions – 1时，PackedRegWidth[n], PackedRegHeight[n], PackedRegTop[n], 和 PackedRegLeft[n] 的取值范围为：

PackedRegWidth[n] 取值从 1 到packed\_picture\_width,

PackedRegHeight[n] 取值从 1 到packed\_picture\_height,

PackedRegLeft[n] 取值从 0 到 packed\_picture\_width - 1,

PackedRegTop[n] 取值从 0 到packed\_picture\_height - 1,

当PackedRegLeft[n] 的值比 packed\_picture\_width / HorDiv1小时, PackedRegLeft[n] 与PackedRegWidth[n] 之和应小于等于 packed\_picture\_width / HorDiv1 ，否则PackedRegLeft[n] 与PackedRegWidth[n] 之和应小于等于 packed\_picture\_width / HorDiv1 \* 2，

当PackedRegTop[n] 的值比packed\_picture\_height / VerDiv1小时, PackedRegTop[n] 与 PackedRegHeight[n] 之和应小于等于packed\_picture\_height / VerDiv1，否则PackedRegTop[n] 与PackedRegHeight[n] 之和应小于等于packed\_picture\_height / VerDiv1 \* 2。

当解码图像色度格式为4:2:0 或 4:2:2时，PackedRegLeft[n] 取值应为亮度采样单位坐标系下一个偶数横坐标值，PackedRegWidth[n] 应对应偶数个亮度采样点。

当解码图像色度格式为 4:2:0 时，PackedRegTop[n] 取值应为亮度采样单位坐标系下一个偶数纵坐标值，ProjRegHeight[n] 应对应偶数个亮度采样点。

8.1.1.6 视点信息结构

8.1.1.6.1定义

视点信息结构(ViewpointInfoStruct)指出了全景视频轨道的视点信息。该结构通过三个子结构描述具体的视点信息，包括：

视点位置结构（ViewpointPosStruct）指示视点的三维空间位置和视点的GPS位置；

旋转信息结构(RotationStruct)指示视点的全局坐标系相对于公共参考坐标系的旋转方向；

视点群组结构（ViewpointGroupStruct）指示视点所属的视点群组。

注：上述子结构可在其他数据结构中独立使用。

8.1.1.6.2语法

aligned(8) ViewpointInfoStruct(gcs\_rotation\_flag) {  
 ViewpointPosStruct();  
 if(gcs\_rotation\_flag == 1)

RotationStruct vpGlobalCoordinateSys();

unsigned int(1) group\_alignment\_flag;  
 bit(7) reserved = 0;  
 if(group\_alignment\_flag == 1){  
 ViewpointGroupStruct();  
}

aligned(8) ViewpointPosStruct() {

signed int(32) pos\_x;

signed int(32) pos\_y;

signed int(32) pos\_z;

unsigned int(1) gpspos\_present\_flag;

bit(31) reserved = 0;

if(gpspos\_present\_flag == 1) {

signed int(32) gpspos\_longitude;

signed int(32) gpspos\_latitude;

signed int(32) gpspos\_altitude;

}

}

aligned(8) class ViewpointGroupStruct() {

unsigned int(8) group\_id;

string group\_description;

}

8.1.1.6.3语义

pos\_x， pos\_y和pos\_z 指示在以（0,0,0）为中心的公共参考坐标系中视点的三维空间位置的笛卡尔坐标值。对于动态视点，则指定动态视点的初始视点位置；

gpspos\_present\_flag取值为1时表示存在视点GPS位置，取值为0表示不存在视点GPS位置；

gpspos\_longitude，gpspos\_latitude和gpspos\_altitude 分别指示视点GPS位置的经度坐标，纬度坐标和高度坐标；

gcs\_rotation\_flag取值为1时表示存在视点的全局坐标系的X，Y和Z坐标轴相对于公共参考坐标系的旋转角度，即存在gcs\_yaw， gcs\_pitch和gcs\_roll；取值为0时表示不存在gcs\_yaw， gcs\_pitch和gcs\_roll。

vpGlobalCoordinateSys指示视点的全局坐标系旋转信息。视点切换时，视点的全局坐标系旋转方向用于确定该视点的全局坐标系中播放全景视频轨道时用户的观看方向。

3D\_rotation\_type 指示旋转信息的表示类型。该字段取值为0表示旋转信息以欧拉角的形式给出；该字段取值为1表示旋转信息以四元数的形式给出。其余取值保留。

rotation\_yaw， rotation\_pitch和rotaion\_roll 分别指示视点的全局坐标系的X，Y，Z轴相对于公共参考坐标系坐标轴的旋转方向的偏转角，俯仰角和翻滚角，以欧拉角的形式给出。

rotation\_quat\_x , rotation\_quat\_y和rotation\_quat\_z分别指示四元数表示中的x，y和z分量的取值，上述字段的取值范围为[-230，230]。 当旋转信息各分量缺失时，各分量默认取值为0。

group\_alignment\_flag 取值为1时表示存在视点组结构，并且视点属于独立的坐标系(及坐标原点)；取值为0时表示不存在视点组结构，则视点属于公共参考坐标系。

注：当两个视点具有不同的group\_id，它们的位置坐标不可对比，因为它们属于不同的坐标系。

group\_id指示视点群组的标识符，该视点群组中的所有视点共用一个公共参考坐标系；

group\_description提供视点群组的文本描述，以空值结尾的UTF-8字符串。

8.1.1.7 缩放区域结构

8.1.1.7.1定义

缩放区域结构指示全景视频的投影图像上的缩放区域信息。对应于全景视频的球面区域或二维区域缩放操作，终端解码并播放球面区域或二维区域在全景视频的投影图像上的一个或者多个缩放区域的视频数据。

8.1.1.7.2语法

aligned(8) class RegionWiseZoomingStruct() {  
 unsigned int(8) num\_regions;  
 for (i = 0; i < num\_regions; i++) {  
 unsigned int(32) zoom\_reg\_width[i];   
 unsigned int(32) zoom\_reg\_height[i];   
 unsigned int(32) zoom\_reg\_top[i];   
 unsigned int(32) zoom\_reg\_left[i];   
 unsigned int(8) zoom\_ratio;  
 unsigned int(8) zoom\_algorithm\_type;  
 unsigned int(8) zoom\_symbolization\_type;  
 unsigned int(8) zoom\_area\_type;   
 string zoom\_description;  
 }  
}

8.1.1.7.3语义

num\_regions 指示对应于同一个全景视频的球面区域或投影图像上的二维区域的缩放区域的数量。可能存在一个或者多个缩放区域，其中，不同缩放区域的视频数据具有不同的分辨率或质量；

zoom\_reg\_width[i]，zoom\_reg\_height[i]，zoom\_reg\_top[i]和zoom\_reg\_left[i]分别指示第i个缩放区域的宽、高、垂直偏移和水平偏移；

zoom\_ratio 指示缩放区域的缩放比例，以2-3为单位。该字段取值为0说明该区域未经过缩放，该字段取值为非0值则指示该区域缩放后的尺寸相较于原始尺寸的实际比例或近似比例。

zoom\_algorithm\_type 指示缩放区域在渲染呈现时所使用的缩放算法类型，如表7所示：

表 7 缩放算法类型

|  |  |
| --- | --- |
| **取值** | **描述** |
| 0 | 凸起变焦 |
| 1 | 球形变焦（保证中心失真最小） |
| 2 | 盘状均匀变焦 |
| 3..255 | 未定义 |

zoom\_symbolization\_type 指示缩放区域的边界符号化类型；

zoom\_area\_type 指示缩放区域的类型，如表8所示：

表 8 缩放区域类型

|  |  |
| --- | --- |
| **取值** | **描述** |
| 0 | 导演剪辑的缩放区域，即根据内容提供者的创作意图缩放视频 |
| 1 | 根据观看统计数据的测量结果选择的缩放区域 |
| 2..239 | 保留 |
| 240..255 | 未定义 |

zoom\_description以空字符结尾的UTF-8字符串，提供缩放区域的文本描述。

8.1.1.8感兴趣区域结构

8.1.1.8.1定义

感兴趣区域结构指示全景视频轨道中感兴趣区域相关信息。感兴趣区域为全景视频的球面区域，播放终端根据用户需求播放与感兴趣区域关联的媒体资源。

8.1.1.8.2语法

aligned(8) ROIRegionStruct() {  
 unsigned int(8) roi\_shape\_type;  
 unsigned int(8) roi\_id;  
 unsigned int(8) roi\_type;  
 string roi\_description;  
 SphereRegionStruct(1);  
}

8.1.1.8.3语义

roi\_shape\_type指示感兴趣区域空间区域形状类型；

roi\_id指示感兴趣区域的标识符，在当前轨道中唯一；

roi\_type指示感兴趣区域的类型，如表9所示；

表 9 感兴趣区域类型

|  |  |
| --- | --- |
| **取值** | **描述** |
| 0 | 创作者推荐，根据创作者意图为用户推荐观看方向 |
| 1 | 画面增强，如精彩区域特写、提供高清画面等 |
| 2 | 实时热点，提供当前视频中最受关注的区域信息 |
| 3 | 方位指引 |
| 4 | 多屏互动，同时提供多个方向视频画面 |
| 5..255 | 自定义 |

roi\_description，以空字符结尾的UTF-8字符串，提供感兴趣区域的文本描述；

SphereRegionStruct()指示感兴趣区域空间区域，相对于全局坐标轴，感兴趣区域的形状类型值（shape\_type）由roi\_shape\_type指示，SphereRegionStruct(1)中的插值（interpolate）应等于0；

8.1.2基于容积视频的结构和语义

8.1.2.1容积视频的媒体封装结构

8.1.2.1.1定义

容积视频的码流中包括图集数据、几何数据、占位数据和属性数据，可以采用单轨道方式或多轨道方式实现系统层的媒体封装。

8.1.2.1.2单轨道的媒体封装

单轨道的媒体封装用一个媒体轨道封装一个容积视频的所有码流，如下图 15所示，轨道中的每一个样本可包含图集数据、几何数据、属性数据、占位数据等。采用单轨道方式对容积视频数据封装将不会对码流做进一步的处理。在媒体传输时需要传输媒体轨道中的整个容积视频采用单轨道方式进行封装的容积视频数据可以在多媒体系统中根据需求进一步处理（比如进一步生成按照多轨道方式进行封装的容积视频数据、转码、DASH分片等）。



图 15 单轨道的媒体封装结构示例图

8.1.2.1.3多轨道的媒体封装

多轨道的媒体封装采用多个媒体轨道封装一个容积视频的码流，多轨道的媒体封装结构如下图 16所示，根据容积视频的码流中每个单元的类型，每一个媒体轨道的样本只存放数据类型相同的单元。基于多轨道封装的容积视频媒体轨道分为两类，一类是容积视频图集/图集分片轨道，包含容积视频的参数信息和图集/图集分片数据；一类是容积视频组件轨道，包含占位组件（可选）、几何组件和/或属性组件的二维视频编码数据。容积视频图集/图集分片轨道通过轨道参考关联到对应容积视频组件轨道。

多轨道的媒体封装中，属于同一个容积视频编码序列的媒体轨道是时间对齐的。与同一个容积视频帧相关但属于不同媒体轨道中的所有样本都具有相同的呈现时间。容积视频的图集数据所对应的样本解码时间应该等于或先于容积视频帧的合成时间。



图 16 多轨道的媒体封装结构示例图

多轨道封装容积视频的基础结构的特征如下：

一个容积视频图集轨道，包含容积视频参数集和图集参数集，其样本承载图集数据。容积视频图集轨道包含多个轨道参考，分别参考到存放容积视频二维组件的媒体轨道；

——零个、一个或多个容积视频组件轨道，其样本承载基于视频编码的占位数据；

——零个、一个或多个容积视频组件轨道，其样本承载基于视频编码的几何数据；

——零个、一个或多个容积视频组件轨道，其样本承载基于视频编码的属性数据；

——零个、一个或多个容积视频组件轨道，其样本承载基于视频编码的封装视频数据，包括占位数据、几何数据和属性数据。

为了便于访问容积视频的部分区域或者某一对象，可以将一个容积视频按多个图集或多个图集分片的方式来存储。

一个容积视频按多个图集的方式进行存储，基于多图集的多轨道封装容积视频的特征如下：

——多个容积视频图集轨道，一个容积视频图集轨道只包含一个图集，包含容积视频的基于该图集的参数集，其样本承载对应该图集的图集数据，则需要多个容积视频图集轨道来包含多个图集及其数据，其中一个容积视频图集轨道只包含容积视频参数集和图集参数集，其样本不承载图集数据，并与其它容积视频图集轨道关联；

——多个容积视频组件轨道，一个图集对应一组或一个容积视频组件轨道，每个对应的容积视频组件轨道中的样本承载对应该图集的几何数据、占位图数据、属性数据或封装视频数据。

一个容积视频按多个图集分片的方式进行存储，一个图集划分为多个分片，每一个或多个图集分片数据可以存储到一个单独的容积视频图集分片轨道中，基于图集分片的多轨道封装容积视频的特征如下：

——一个容积视频图集轨道，包含容积视频的通用参数集和基于图集的参数集，基于图集的参数集将被所有存储图集分片的容积视频图集分片轨道共享；

——多个容积视频图集分片轨道，其样本承载一个或一组图集分片对应的图集数据，该容积视频图集分片轨道不包含任何参数集，其与容积视频图集轨道关联；

——多个容积视频组件轨道，一个容积视频图集分片轨道关联一组或一个容积视频组件轨道，每个的容积视频组件轨道中的样本分别承载对应图集分片的几何数据、占位图数据、属性数据或封装视频数据。

8.1.2.2容积视频解码配置信息结构

8.1.2.2.1定义

容积视频解码配置信息提供容积视频码流的具体解码信息（比如参数集和SEI信息），用于解码器的进一步配置和初始化。其中，V3CparameterSet数组应包含一个v3c\_parameter\_set()，该函数在ISO/IEC 23090-5定义。

如果终端侧不支持或识别configurationVersion，则不应该解码该配置信息或其适用的码流。SetupUnit数组应包含图集子码流的参数集，该参数集对于解码器配置信息所在的样本入口和图集子码流SEI信息所参考的编码容积视频序列是固定的。

8.1.2.2.2语法

aligned(8) class V3CDecoderConfigurationRecord {  
 unsigned int(8) configurationVersion = 1;  
 unsigned int(2) lengthSizeMinusOne;  
 bit(1) reserved = 0;  
 unsigned int(5) numOfV3CParameterSets;  
 for (i=0; i < numOfV3CParameterSets; i++) {  
 unsigned int(16) V3CParameterSetLength;  
 v3c\_unit(V3CParameterSetLength) v3cParameterSet;   
 }  
 unsigned int(8) numOfSetupUnitArrays;  
 for (j=0; j < numOfSetupUnitArrays; j++) {  
 bit(1) array\_completeness;  
 bit(1) reserved = 0;  
 unsigned int(6) NAL\_unit\_type;  
 unsigned int(8) numNALUnits;  
 for (i=0; i < numNALUnits; i++) {  
 unsigned int(16) SetupUnitLength;  
 nal\_unit(SetupUnitLength) setupUnit;   
 }  
 }  
}

8.1.2.2.3语义

configurationVersion指示版本域。通过版本号的变化指示样本入口的可兼容的变换。

lengthSizeMinusOne 加1，指示使用该配置信息的媒体流中一个容积视频样本里NALUnitLength域的长度值（以字节为单位）。该域值应等于图集子码流的sample\_stream\_nal\_header()中ssnh\_unit\_size\_precision\_bytes\_minus1的值，该值在ISO/IEC 23090-5定义。

numOfV3CParameterSets指示在解码配置信息中容积视频参数集单元的数量。

V3CParameterSetLength 指示v3cParameterSet 域的大小，以字节为单位。

v3cParameterSet指示一个V3C\_VPS类型的容积视频单元，该单元在ISO/IEC 23090-5定义。

numOfSetupUnitArrays指示指定类型的图集NAL单元数组的数量。

array\_completeness取值为1时，指示已知类型的所有图集NAL单元在随后的数组中，不存在于媒体流中；取值为0时，指示其他的指定类型的图集NAL单元存在于媒体流中。缺省值和允许值受限于样本入口类型。

NAL\_unit\_type指示随后的数组中图集NAL单元的类型。采用ISO/IEC23090-5中的定义指示图集NAL单元的类型，包括NAL\_ASPS, NAL\_AAPS, NAL\_AFPS, NAL\_PREFIX\_ESEI, NAL\_PREFIX\_NSEI, NAL\_SUFFIX\_ESEI, 或 NAL\_SUFFIX\_NSEI。

numNALUnits指示指明类型的图集NAL单元数量，该图集NAL单元包含在配置信息中。SEI数组应只包含说明性的SEI信息（如用户数据SEI信息），提供关于媒体流的信息。

SetupUnitLength指示setupUnit域的大小，以字节为单位。该域的长度包含一个NAL单元头和NAL单元载荷的大小，但不包含域本身的大小。

setupUnit指示包含的NAL单元，NAL单元类型为NAL\_ASPS, NAL\_AAPS, NAL\_AFPS, NAL\_PREFIX\_ESEI, NAL\_PREFIX\_NSEI, NAL\_SUFFIX\_ESEI, 或 NAL\_SUFFIX\_NSEI，在ISO/IEC 23090-5中定义，当NAL单元在setupUnit中存在时，NAL\_PREFIX\_SEI 或 NAL\_SUFFIX\_SEI包含一个说明性的SEI信息，用于提供媒体流的信息。

8.1.2.3 空间区域信息结构

8.1.2.3.1 定义

三维空间区域结构（3DSpatialRegionStruct）和三维边框结构（3DBoundingBoxStruct）分别提供了容积视频媒体的空间区域信息（包括空间区域x，y，z轴的偏移，三维空间区域的宽度、高度和深度）和边框信息。

8.1.2.3.2 语法

aligned(8) class 3DPoint() {  
 unsigned int(16) x;  
 unsigned int(16) y;  
 unsigned int(16) z;  
}

aligned(8) class CuboidRegionStruct() {  
 unsigned int(16) cuboid\_dx;  
 unsigned int(16) cuboid\_dy;  
 unsigned int(16) cuboid\_dz;  
}

aligned(8) class 3DSpatialRegionStruct(dimensions\_included\_flag) {  
 unsigned int(16) 3d\_region\_id;  
 3DPoint anchor;  
 if (dimensions\_included\_flag) {  
 CuboidRegionStruct();  
 }  
}

aligned(8) class 3DBoundingBoxStruct() {  
 unsigned int(16) bb\_dx;  
 unsigned int(16) bb\_dy;  
 unsigned int(16) bb\_dz;  
}

8.1.2.3.3 语义

3d\_region\_id指示空间区域的标识符。

x, y, z指示笛卡尔坐标系下三维点的x，z，y坐标值。

cuboid\_dx, cuboid\_dy, cuboid\_dz指示笛卡尔坐标系下的长方体子区域相对于锚点的沿x，y，z轴的尺寸。

anchor指示笛卡尔坐标系下作为三维空间区域锚的一个三维点。

bb\_dx, bb\_dy, bb\_dz指示笛卡尔坐标系下整个容积媒体三维边框的延伸相对于原点（0，0，0）的沿x，y，z轴的尺寸。

dimensions\_included\_flag取值为0时表示空间区域的维度与具有相同 3d\_region\_id 的第一个 3DSpatialRegionStruct 实例的维度相同（例如，在前一个样本中或在 V3CSpatialRegionsBox 中），并且该空间区域的维度已经在至少一个 3DSpatialRegionStruct 实例中明确指出；取值为1时表示空间区域维度被指出。

8.1.2.4容积视频的视图信息结构

8.1.2.4.1定义

多视图的容积视频包含基础视图和辅助视图，在重构三维场景过程中需要根据用户的观看位置、方向等选择一个或多个视图的容积视频合成/渲染目标视图。其中，辅助视图对应的容积视频需要根据基础视图的容积视频数据合成。

ViewInfoStruct提供容积视频相应的视图信息。

8.1.2.4.2语法

aligned(8) class ViewInfoStruct(camera\_parameters\_included\_flag) {  
 unsigned int(16) view\_id;  
 unsigned int(16) view\_group\_id;  
 string view\_description;  
 unsigned int(1) basic\_view\_flag;  
 bit(7) reserved = 0;  
 if (camera\_parameters\_included\_flag) {

ExtCameraInfoStruct();  
IntCameraInfoStruct();

}

}

8.1.2.4.3语义

view\_id 指示视图的标识。

view\_group\_id指示视图所属的视图组的标识。

view\_descritption以空值结尾的UTF-8字符串，提供视图的文字描述。

basic\_view\_flag取值为1表示当前视图为基础视图；取值为0表示当前视图为辅助视图。

camera\_parameters\_included\_flag取值为1时表示存在当前视图关联的相机参数信息；取值为0时表示不存在当前视图关联的相机参数信息。

8.1.2.5容积视频的视图分组信息结构

8.1.2.5.1定义

容积视频的视图分组是根据视图之间的依赖关系对容积视频的多个视图进行分组，一个视图组包含一个基础视图和任意个辅助视图。基于视图分组的信息从视图组中选择一个或多个视图的容积视频数据可合成/渲染出目标视图，其中，基础视图为必选。

ViewGroupInfoStruct提供容积视频的视图分组信息，描述一个或多个视图。

8.1.2.5.2语法

aligned(8) class ViewGroupInfoStruct() {  
unsigned int(16) view\_group\_id;  
string view\_group\_description;  
unsigned int(8) num\_views;  
for (i=0; i < num\_views; i++) {

unsigned int(16) view\_id;

unsigned int(1) basic\_view\_flag;

bit(7) reserved = 0;

if (camera\_parameters\_included\_flag) {

ExtCameraInfoStruct();

IntCameraInfoStruct();  
}

}  
}

8.1.2.5.3语义

view\_group\_id指示视图组的标识。

view\_group\_descritption以空值结尾的UTF-8字符串，提供视图组的文字描述。

num\_views 指示视图组中的视图数量。

view\_id指示视图组中视图的标识。

basic\_view\_flag 取值为1 时表示该视图是基础视图；取值为0时表示该视图是辅助视图。

camera\_parameters\_included\_flag取值为 1时表示存在相机参数结构 ；取值为 0时不存在相机参数结构。

8.1.2.6容积视频可替换信息结构

8.1.2.6.1定义

当容积视频的组件轨道存在可替换轨道时，容积视频可替换信息结构V3CAlternativeInfoStruct用于指示可替换组内多个组件轨道之间的差异。

8.1.2.6.2语法

aligned(8) class V3CAlternativeInfoStruct() {

unsigned int(1) quality\_ranking\_flag;

unsigned int(1) codec\_type\_flag;

bit(6) reserved=0;

if(quality\_ranking\_flag == 1){

unsigned int(8) quality\_ranking;

}

if(codec\_type\_flag == 1){

unsigned int(32) codec\_type;

}

}

8.1.2.6.3语义

quality\_ranking\_flag取值为1时表示可替换组内的轨道之间存在质量上的可替换关系；取值为0时表示可替换组内的轨道之间不存在质量上的可替换关系。

codec\_type\_flag取值为1时表示可替换组内的轨道之间存在编码类型上的可替换关系；取值为0时表示可替换组内的轨道之间不存在编码类型上的可替换关系。

quality\_ranking指示质量等级信息，该字段取值越小，表明对应组件轨道的质量越高。

codec\_type指示对应组件轨道的编码类型。

8.1.2.7相机信息结构

8.1.2.7.1相机外参信息结构

8.1.2.7.1.1定义

相机外参信息结构ExtCameraInfoStruct用于指示相机的外参信息，容积视频视窗的位置和旋转信息以及自由视角视频的位置和旋转信息。

8.1.2.7.1.2语法

aligned(8) class ExtCameraInfoStruct() {  
unsigned\_int(1) camera\_pos\_present;  
unsigned int(1) camera\_ori\_present;

bit(6) reserved=0;  
if (camera\_pos\_present == 1){

signed int(32) camera\_pos\_x;

signed int(32) camera\_pos\_y;

signed int(32) camera\_pos\_z;  
}  
if (camera\_ori\_present == 1){

signed int(32) camera\_quat\_x;

signed int(32) camera\_quat\_y;

signed int(32) camera\_quat\_z;  
}

}

8.1.2.7.1.3语义

camera\_pos\_present取值为1时表示存在相机位置参数；取值为 0时表示不存在相机位置参数。

camera\_ori\_present取值为1时表示存在相机方向参数；取值为0时表示不存在相机方向参数。

camera\_pos\_x, camera\_pos\_y ,camera\_pos\_z 分别表示全局参考坐标系中相机位置的x、y和z坐标。该值应以2−16米为单位。

cam\_quat\_x, cam\_quat\_y和cam\_quat\_z用于计算相机旋转四元数中的qX， qY， qZ分量，取值范围为[-230, 230]。各分量的计算规则如下：

——qX = cam\_quat\_x ÷ 230, qY = cam\_quat\_y ÷ 230, qZ = cam\_quat\_z ÷ 230

——第4个分量qW可按照四元数的数学特性推导得到:

qW = Sqrt( 1 – ( qX2 + qY2 + qZ2 ) )

——相机旋转四元数各分量缺省值为0。

——四元数(w, x, y, z) 代表围绕向量(x, y, z) 以角度进行旋转。

8.1.2.7.2相机内参信息结构

8.1.2.7.2.1定义

相机内参信息结构IntCameraInfoStruct用于指示相机的内参信息，容积视频视窗信息包含相机的内参信息以及自由视角视频视窗的相机类型和相关参数信息。

8.1.2.7.2.2语法

aligned(8) class IntCameraInfoStruct() {

unsigned int(10) camera\_id;

unsigned int(1) camera\_depth\_present;

unsigned int(3) camera\_type;

bit(2) reserved = 0;

if (camera\_type == 0) {

signed int(32) erp\_horizontal\_fov;

signed int(32) erp\_vertical\_fov;

}

if (camera\_type == 1) {

signed int(32) perspective\_horizontal\_fov;

unsigned int(8)[4] perspective\_aspect\_ratio;

}

if (camera\_type == 2) {

unsigned int(8)[4] ortho\_aspect\_ratio;

unsigned int(8)[4] ortho\_horizontal\_size;

}

if (camera\_type == 3) {

unsigned int(8)[4] camera\_focal\_length\_x;

unsigned int(8)[4] camera\_focal\_length\_y;

unsigned int(8)[4] camera\_principle\_point\_x;

unsigned int(8)[4] camera\_principle\_point\_y;

}

if (camera\_type == 4) {

unsigned int(8)[4] camera\_principle\_point\_x;

unsigned int(8)[4] camera\_principle\_point\_y;

}

if (camera\_depth\_present == 1){

unsigned int(32) camera\_near\_depth;

unsigned int(32) camera\_far\_depth;

}

}

8.1.2.7.2.3语义

camera\_id指示对应视窗的相机标识符。

camera\_depth\_present取值为1时表示存在相机深度参数；取值为 0时表示不存在相机深度参数。

camera\_type指示视窗相机的投影算法类型。取值为0表示ERP投影；取值为1表示透视投影；取值为2表示正交投影；取值为3表示自由视角针孔相机模型；取值为4表示自由视角鱼眼相机模型；其余取值保留。

erp\_horizontal\_fov指示ERP投影时，视窗区域水平方向的经度范围，以弧度为单位。该字段取值范围为(0,2π)。

erp\_vertical\_fov指示ERP投影时，视窗区域垂直方向的维度范围，以弧度为单位。该字段取值范围为(0,π)。

perspective\_horizontal\_fov指示透视投影时，视窗区域的水平范围，以弧度为单位。该字段取值范围为(0,π)。

perspective\_aspect\_ratio指示透视投影时，视窗的纵横比例(水平/垂直)。 该字段取值为32位浮点数，解析过程遵循IEEE 754标准。

ortho\_aspect\_ratio 指示正交投影时，视窗的纵横比例(水平/垂直)。 该字段取值为32位浮点数，解析过程遵循IEEE 754标准。

ortho\_horizontal\_size指示正交投影时，视窗水平方向的尺寸，以米为单位。 该字段取值为32位浮点数，解析过程遵循IEEE 754标准。

camera\_focal\_length\_x、camera\_focal\_length\_y 分别指示相机焦距的x，y分量取值。该字段取值为32位浮点数，解析过程遵循IEEE 754标准。

camera\_principle\_point\_x，camera\_principle\_point\_y分别指示相机主光轴在图像坐标系中偏移的x与y分量。该字段取值为32位浮点数，解析过程遵循IEEE 754标准。

camera\_near\_depth、camera\_far\_depth 分别指示与相机相关联的视锥体的近和远平面指示近和远深度（或距离），值应以 2-16 米为单位。

8.1.2.8视窗信息结构

8.1.2.8.1定义

视窗信息结构ViewportInfoStruct用于指示视窗信息，该结构包含相机的内参和外参信息。

8.1.2.8.2语法

aligned(8) class ViewportInfoStruct(ext\_camera\_flag, int\_camera\_flag) {

if (ext\_camera\_flag == 1) {

unsigned int(1) center\_view\_flag;

bit(6) reserved = 0;

if (center\_view\_flag == 0) {

unsigned int(1) left\_view\_flag;

} else {

bit(1) reserved = 0;

}

ExtCameraInfoStruct extCamInfo();

}

if (int\_camera\_flag == 1) {

IntCameraInfoStruct intCamInfo();

}

}

8.1.2.8.3语义

center\_view\_flag取值为1时表示指示的视窗位置对应视窗中心；取值为0时表示指示的视窗位置对应视窗双目位置中的一个。

left\_view\_flag取值为1时表示指示的视窗位置信息对应视窗双目位置中的左目；取值为0时表示指示的视窗位置信息对应视窗双目位置中的右目。

extCamInfo()指示视窗的外参信息。

intCamInfo()指示视窗的内参信息。

8.1.2.9观看空间结构

8.1.2.9.1定义

三维观看空间的结构由ViewingSpaceStruct表示，是由一个或多个复杂几何体SpaceShapeStruct组合构造，通常采用CSG（Constructive Solid Geometry，构造几何体）方式组合。而复杂几何体是由基础几何体（立方体、圆柱体、半球体）有序的相组合，通常采用CSG方式组合或按顺序插值方式得到。

8.1.2.9.2 语法

aligned(8) class point() {  
 signed int(32) center\_x;  
 signed int(32) center\_y;  
 signed int(32) center\_z;  
}

aligned(8) class Cuboidstruct() {  
 if(camera\_inferred\_flag==0)  
 point();  
 signed int(32) size\_x;  
 signed int(32) size\_y;  
 signed int(32) size\_z;  
}

aligned(8) class SpheroidStruct() {  
 if(camera\_inferred\_flag==0)   
 point();  
 unsigned int (32) radius\_x;  
 unsigned int (32) radius\_y;  
 unsigned int (32) radius\_z;   
}  
}

aligned(8) class HalfspaceStruct() {  
 signed int(32) normal\_x;  
 signed int(32) normal\_y;  
 signed int(32) normal\_z;  
 unsigned int (32) distance;  
}

aligned(8) class ViewingDirectionConstrainStruct (camera\_inferred\_flag) {  
 if（camera\_inferred\_flag == 0）{  
 signed int(32)viewing\_direction\_center\_x;  
 signed int(32)viewing\_direction\_center\_y;  
 signed int(32)viewing\_direction\_center\_z;  
 }  
 signed int(32) viewing\_direction\_yaw\_range;  
 signed int(32) viewing\_direction\_pitch\_range;  
 if( guard\_band\_present\_flag)  
 signed int(32) guard\_band\_direction\_size；  
}

aligned(8) class SpaceShapeStruct() {  
 unsigned int(16) num\_primitive\_shape;  
 unsigned int(8) primitive\_shape\_operation;  
 unsigned int(8) viewing\_space\_shape\_type;  
 unsigned int(16) distance\_scale;  
 unsigned int(8) camera\_inferred\_flag;  
 unsigned int(8) guard\_band\_present\_flag;  
 for(int i=0；i< num\_primitive\_shape; i++){  
 if(camera\_inferred\_flag==1)  
 unsigned int(16) view\_id;  
 if(viewing\_space\_shape\_type==0)  
 Cuboidstruct();  
 else if(viewing\_space\_shape\_type==1)  
 SpheroidStruct();  
 else if(viewing\_space\_shape\_type==2)  
 HalfspaceStruct ();   
 if( guard\_band\_present\_flag)  
 unsigned int(16) guard\_band\_size;  
 if(viewing\_direction\_constraint\_ flag == 0)  
 ViewingDirectionConstrainStruct(camera\_inferred\_flag);   
 }  
}

aligned(8) class ViewingSpaceStruct () {  
 signed int（8） num\_shape\_space;  
 for(int j=0;j<=num\_shape\_space; j++) {   
 signed int(8) operation\_type;  
 SpaceShapeStruct();   
 }  
}

8.1.2.9.3 语义

center\_x、center\_y、center\_z，分别指示几何结构体的中心点在坐标系中的位置；

size\_x、size\_y、size\_z，分别指示立方体在x、y、z方向上的边长；

radius\_x，radius\_y， radius\_z，分别指示球体在x，y，z维度上的半径；

normal\_x，normal\_y，normal\_z，分别指示定义半球体的平面法向；

distance指示从原点沿着法向量到半球体平面方向的距离。

复杂几何体通过形状空间结构（ShapeSpaceStruct）来描述，由一个或多个基础几何体构造得到。

num\_primitive\_shape指示组成三维观看空间的基础几何体的数量；

primitive\_shape\_operation指示组成三维观看空间对基础几何体形状的操作方式；为0时，指示使用CSG模式对基础几何体进行相加形成复杂几何体；为1时，指示将沿着基础几何体中心形成的路径对基础几何体进行插值操作，以形成复杂几何体；

viewing\_direction\_constraint\_ flag取值为1时表示基础几何体中的观看方向受限制；取值为0时表示基础几何体中的观看方向不受限制；

viewing\_direction\_center\_x，viewing\_direction\_center\_y， viewing\_direction\_center\_z，分别指示在基础几何体中建议观看方向的中心的四元数分量x，y，z；

viewing\_direction\_yaw\_range，viewing\_direction\_pitch\_range，分别指示在基础几何体中建议观看方向的偏航范围和倾斜范围的二分之一；

guard\_band\_present\_flag取值为1时表示基础几何体有保护带；取值为0时表示基础几何体没有保护带；

guard\_band\_direction\_size指示基础几何体内观看方向的保护带的大小，以度数为单位来表示。

camera\_inferred\_flag取值为1时表示基础几何体的对位和方向是与采集设备对应，其中，采集设备与视点索引号对应；取值为0时表示基础几何体的位置和方向不与采集设备对应，需要自行定义中心点位置；

viewing\_space\_shape\_type指示三维观看空间的基础几何体形状，具体的形状类型描述如表10所示：

表 10 基础几何体形状

|  |  |
| --- | --- |
| **viewing\_space\_shape\_type** | **形状** |
| 0 | 立方体 |
| 1 | 球形体 |
| 2 | 半球体 |
| 3 | 保留 |

distance\_scale指示基础几何体的边框距离尺寸的刻度；

view\_id指示基础几何体所对应的摄像机对应视点的标识，通过该标识可以定位到该视点对应媒体数据所在的媒体轨道。

num\_shape\_space指示组成一个三维观看空间结构所需要的复杂几何体的数量；

operation\_type指示该几何体组合成为三维观看空间的CSG操作方式，具体类型如表11所示：

表 11 三维观看空间的CSG操作方式

|  |  |
| --- | --- |
| **operation\_type** | **说明** |
| 0 | Addition，几何体相并的组合方式 |
| 1 | Substraction，几何体相差的组合方式 |
| 2 | Intersection，几何体相交的组合方式 |

8.1.3基于自由视角视频的结构和语义

8.1.3.1纹理图信息结构

8.1.3.1.1定义

纹理图信息结构指示自由视角视频数据采集相机对应的纹理图的编码元数据信息。

8.1.3.1.2语法

aligned(8) TextureInfostruct(){

unsigned int(8) texture\_padding\_size；

unsigned int(16) texture\_top\_left\_x；  
unsigned int(16) texture\_top\_left\_y；  
unsigned int(16) texture\_bottom\_right\_x；  
unsigned int(16) texture\_bottom\_right\_y;

}

8.1.3.1.3语义

texture\_padding\_size指示纹理图的保护带宽度；

texture\_top\_left\_x指示相应原始相机对应的纹理图在视频帧平面中的左上角x坐标；

texture\_top\_left\_y指示相应原始相机对应的纹理图在视频帧中的左上角y坐标；

texture\_bottom\_right\_x指示相应原始相机对应的纹理图在视频帧中的右下角x坐标；

texture\_bottom\_right\_y指示相应原始相机对应的纹理图在视频帧中的右下角y坐标。

8.1.3.2深度图信息结构

8.1.3.2.1定义

深度图信息结构指示自由视角视频数据采集相机对应的深度图的编码元数据信息。

8.1.3.2.2语法

aligned(8) DepthInfoStruct(){

unsigned int(8) depth\_padding\_size；  
unsigned int(16) depth\_top\_left\_x；  
unsigned int(16) depth\_top\_left\_y；  
unsigned int(16) depth\_bottom\_right\_x；  
unsigned int(16) depth\_bottom\_right\_y;  
unsigned int(8) depth\_downsample\_factor;

}

8.1.3.2.3语义

depth\_padding\_size指示深度图的保护带宽度；

depth\_top\_left\_x指示相应原始相机对应的深度图在视频帧平面中的左上角x坐标；

depth\_top\_left\_y指示相应原始相机对应的深度图在视频帧平面中的左上角y坐标；

depth\_bottom\_right\_x指示相应原始相机对应的深度图在视频帧平面中的右下角x坐标；

depth\_bottom\_right\_y指示相应原始相机对应的深度图在视频帧平面中的右下角y坐标；

depth\_downsample\_factor指示深度图降采样的倍数因子，深度图的实际分辨率宽高为相机采集分辨率宽高的1/2 depth\_downsample\_factor。

8.2 基于沉浸媒体的数据盒扩展

8.2.1基于全景视频的数据盒

8.2.1.1投影的全景视频数据盒

8.2.1.1.1定义

数据盒类型（Box Type）:  'povd'

数据盒容器（Container）:  SchemeInformationBox

强制性（Mandatory）: 当scheme\_type 为 'povd'时为强制

数量（Quantity）: 零个或一个

该数据盒为可包含多个子数据盒的容器类数据盒，其所包含的子数据盒用于指出如下信息：

——投影图像的投影格式（C对应轨道中的单视场视频，CL 和CR分别指出立体视窗视频的左右眼视图）

——区域的封装（如果使用）

——本地坐标轴与全局坐标轴之间转换所需的旋转（如果使用）

——轨道的内容覆盖（可选）

变量HorDiv1 和 VerDiv1的值可按如下所示来设置：

如果StereoVideoBox不存在于SchemeInformationBox中，HorDiv1 和 VerDiv1分别设为1；

否则，采用如下设置：

如果指明是并排方式的帧封装，HorDiv1设为2， VerDiv1设为1；

否则，如果指明是上下方式的帧封装，HorDiv1 设为1 ，VerDiv1设为2；

否则，按空间交错方式的帧封装，HorDiv1 和VerDiv1都设为1。

如果RotationBox不存在于ProjectedOmniVideoBox中，RotationFlag设为0，否则，设为1。

如果StereoVideoBox不存在于SchemeInformationBox中，SpatiallyPackedStereoFlag, TopBottomFlag, 和 SideBySideFlag都为0。否则，采用如下设置：

当StereoVideoBox指明上下帧封装，SpatiallyPackedStereoFlag 设为 1, TopBottomFlag 设为 1, SideBySideFlag 设为 0；

当StereoVideoBox指明并排帧封装，SpatiallyPackedStereoFlag 设为 1, TopBottomFlag 设为 0, SideBySideFlag 设为 1；

当StereoVideoBox 指明空间交错帧封装，SpatiallyPackedStereoFlag, TopBottomFlag和SideBySideFlag 都设为 0。

如下适用：

一个单视场的投影亮度图像的宽和高（ConstituentPicWidth 和ConstituentPicHeight）通过如下设置得到：

如果RegionWisePackingBox不存在于 ProjectedOmniVideoBox中，ConstituentPicWidth 和ConstituentPicHeight 分别设置等于 width / HorDiv1和height / VerDiv1, 其中，width和height是VisualSampleEntry的语法元素；

否则，ConstituentPicWidth 和ConstituentPicHeight 分别设置等于 proj\_picture\_width / HorDiv1和proj\_picture\_height / VerDiv1。

8.2.1.1.2语法

aligned(8) class ProjectedOmniVideoBox extends Box('povd') {  
 ProjectionFormatBox();  
}

aligned(8) class ProjectionFormatBox() extends FullBox('prfr', 0, 0) {  
 ProjectionFormatStruct();

}

8.2.1.2旋转数据盒

8.2.1.2.1定义

数据盒类型（Box Type）: ' rotn '  
数据盒容器（Container）: ProjectedOmniVideoBox  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

提供沿着X轴、Y轴和Z轴分别旋转的偏航（yaw）角度，俯仰（pitch）角度和翻滚（roll）角度或者四元数旋转信息，用于本地坐标轴到全局坐标轴的转换。在立体全景视频情况下，分别适用于每一个双目视图景象。当RotationBox不存在时，rotation\_yaw，rotation\_pitch 和rotation\_roll的缺省值为0或者rotation\_x, rotation\_y, rotation\_z, rotation\_w的缺省值为0。

8.2.1.2.2语法

aligned(8) class RotationBox extends FullBox('rotn', 0, 0) {  
 RotationStruct();  
}

8.2.1.3覆盖信息数据盒

8.2.1.3.1定义

数据盒类型（Box Type）: 'covi'  
数据盒容器（Container）: ProjectedOmniVideoBox  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

该数据盒提供了所属媒体轨道内容覆盖区域的信息, 用于指明封装图像覆盖的球面区域。

注：当渲染全景视频内容时，由终端播放器决定如何处理没有被内容覆盖的区域。

8.2.1.3.2语法

aligned(8) class CoverageInformationBox extends FullBox('covi', 0, 0) {  
 ContentCoverageStruct()  
}

8.2.1.4区域封装数据盒

8.2.1.4.1定义

数据盒类型（Box Type）: 'rwpk '  
数据盒容器（Container）: ProjectedOmniVideoBox  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

RegionWisePackingBox规定了封装区域与对应投影区域之间的映射关系。

注：在其他信息中RegionWisePackingBox也提供了在二维笛卡尔图域中的内容覆盖信息。

8.2.1.4.2语法

aligned(8) class RegionWisePackingBox extends FullBox('rwpk', 0, 0) {  
 RegionWisePackingStruct();  
}

8.2.1.5视点群组数据盒

8.2.1.5.1定义

数据盒类型（Box Type）: 'vwpt'  
数据盒容器（Container）: TrackBox  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

视点群组数据盒指示其所在全景视频轨道对应视点的视点信息。

视点群组数据盒为扩展轨道群组类型数据盒（TrackGroupTypeBox），通过轨道群组类型（track\_group\_type）等于'vwpt'的来识别视点群组数据盒。其中，视点群组数据盒中具有相同轨道群组标识（track\_group\_id）的全景视频轨道对应相同视点。

注：如果文件中没有任何轨道携带此视点群组数据盒，则认为该文件内容对应同一个视点。

8.2.1.5.2语法

aligned(8) class ViewpointGroupBox extends TrackGroupTypeBox('vwpt') {

ViewpointInfoStruct();

string viewpoint\_label;

unsigned int(16) viewpoint\_id;

unsigned int(8) viewpoint\_type;

}

8.2.1.5.3语义

viewpoint\_label指示视点的标签，提供轨道群组所对应视点的文本描述，以空字符结尾的UTF-8字符串；

viewpoint\_id指示轨道群组所对应视点的标识；

viewpoint\_type指示轨道群组所对应视点的类型,如表12所示：

表 12 视点类型

|  |  |
| --- | --- |
| **取值** | **描述** |
| 0 | 利用物理摄像机实际拍摄形成的真实视点 |
| 1 | 通过其他真实视点视频数据以及深度等辅助信息合成的虚拟视点 |
| 2..239 | 保留 |
| 240..255 | 未定义 |

8.2.1.6定时文本配置数据盒

8.2.1.6.1定义

数据盒类型（Box Type）: 'ttcf'  
数据盒容器（Container）: XMLSubtitleSampleEntry, WVTTSampleEntry 或 CCSampleEntry  
强制性（Mandatory）: 当scheme\_type 为‘ttcf’时强制  
数量（Quantity）: 一个

8.2.1.6.2语法

class TimedTextConfigBox extends FullBox('ttcf', 0, 0) {

unsigned int(2) relative\_to\_viewport\_type;

unsigned int(1) relative\_disparity\_flag;

unsigned int(1) depth\_included\_flag;

bit(4) reserved = 0;

unsigned int(8) region\_count;

for (i=0;i< region\_count; i++) {

string region\_id;

if (relative\_to\_viewport\_type == 1) {

if (relative\_disparity\_flag)

signed int(16) disparity\_in\_percent;

else

signed int(16) disparity\_in\_pixels;

} else if (relative\_to\_viewport\_type == 0) {

SphereRegionStruct(0);

if (depth\_included\_flag)

unsigned int(16) region\_depth;

} else if(relativerelative\_to\_viewport\_type == 2) {

if (depth\_included\_flag)

unsigned int(16) region\_depth;

unsigned int(8) roi\_id;

}

}

}

}

8.2.1.6.3语义

relative\_to\_viewport\_type指示定时文本的渲染方式。该字段取值为0时，表示将定时文本渲染在球体上的某个固定位置，即仅当用户在渲染文本提示的方向上查看时，该文本提示才可见。该字段取值为1时，表示定时文本应始终显示在显示屏上，即定时文本的渲染与用户的观看方向无关。该字段取值为2时，表示定时文本渲染在感兴趣区域上，仅当用户在于定时文本相关联的感兴趣区域对应的方向上观看时，该文本提示才可见。

注1：当relative\_to\_viewport\_type取值为1时，可显示定时文本的活动区域由定时文本轨道作为矩形区域提供。

relativeRelative\_disparity\_flag取值为1时表示视差是作为一个视图的显示窗口宽度的百分比值提供；取值为0时表示视差是作为像素数提供。

depth\_included\_flag取值为1表示要呈现定时文本的区域的深度（z值），等于0时表示要呈现定时文本的区域的深度（z值）不存在。

regionRegion\_count指示在球体内提供放置位置的定时文本的文本区域数量。每个区域都由一个标识符标识。当存在包含定时文本球面位置的定时元数据轨道并通过“cdsc”类型的轨道参考关联或参考到此定时文本轨道时，区域计数的值应为0。

regionRegion\_id指示定时文本显示的区域ID，region\_id应该与定时文本流的相应区域ID相同。

disparityDisparity\_in\_percent指示视差百分比，以216为单位，是一个视窗宽度的一部分。此值用于在左眼视图中将区域向左移动，在右眼视图中将区域向右移动。该值可能为负，在这种情况下，位移方向是相反的。

disparityDisparity\_in\_pixels指示像素表示的视差。此值用于在左眼视图中将区域向左移动，在右眼视图中将区域向右移动。该值可能为负，在这种情况下，位移方向是相反的。

SphereRegionStruct()指示一个球体位置，该位置与其他信息一起用于确定定时文本在三维空间中的放置和显示位置。球体中心和该球体位置之间的向量是要在其上渲染定时文本提示的渲染三维平面的法向量。该信息和三维平面的深度用于确定渲染三维平面在要在其上渲染定时文本提示的三维空间中的位置。

注2：当SphereRegionStruct()被包含在TimedTextConfigBox()中时，shape\_type、dynamic\_range\_flag、static\_azimuth\_rangee和static\_elevation\_range的值都为0。centreCentre\_azimuth 和centre\_elevation指示球体位置，该位置与其他信息一起用于确定定时文本在三维空间中的放置和显示位置。Centre\_azimuth的范围是[−π\* 216,π\* 216−1]。Entre\_elevation的范围是[−π/2× 216,π/2× 216]，同时centre\_tilt应等于0。

regionRegion\_depth指示渲染定时文本的文本区域的深度信息。当相对于固定球面区域或感兴趣区域渲染定时文本时，需构建相对于单位球的三维平面,在此三维平面上渲染定时文本的一个或者多个文本区域。其中，根据固定球面区域或感兴趣区域确定三维平面对应的球面区域位置，根据深度信息确定三维平面与单位球球心的距离。

roi\_id指示与定时文本关联的感兴趣区域标识符，与全景视频轨道中的感兴趣区域对应。

8.2.1.7感兴趣区域描述数据盒

8.2.1.7.1定义

数据盒类型（Box Type）: 'roid'  
数据盒容器（Container）: VisualSampleEntry or SchemeInformationBox  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

感兴趣区域描述数据盒中的元素指示全景视频的感兴趣区域以及与感兴趣区域相关联的一个或多个多媒体资源。使用该数据盒时，可在用户当前视窗播放与该感兴趣区域相关联的媒体资源，也可以在感兴趣区域中直接播放与其相关的媒体资源。

8.2.1.7.2语法

class RegionOfInterestDescriptionBox extends FullBox('roid', 0, 0) {

unsigned int(8) roi\_count;

for (i=0;i<roi\_count;i++) {

ROIRegionStruct();

unsigned int(8) track\_count;

for (i=0;i<track\_count;i++) {

unsigned int(32) reference\_type;

unsigned int(32) track\_ref\_idx;

}

}

}

8.2.1.7.3语义

roi\_count 指示感兴趣区域的数量；

ROIRegionStruct() 指示感兴趣区域相关信息，包括空间位置、感兴趣区域标识符、关联信息等;

trackTrack\_count指示与感兴趣区域关联的媒体轨道数量；

reference\_type指示与感兴趣区域存在关联关系的媒体资源轨道在视频轨道引用数据盒中的引用类型；

track\_ref\_idx指示与感兴趣区域相关联的媒体轨道标识符在轨道引用数据盒中的索引，轨道引用数据盒类型由reference\_type提供。

8.2.1.8感兴趣区域样本群组入口

8.2.1.8.1概述

感兴趣区域样本群组数据盒表示包含相同感兴趣区域的视频样本属于感兴趣区域样本群组，感兴趣区域样本群组入口的样本群组类型（grouping\_type）等于'roig'，用以描述感兴趣区域空间位置信息。

感兴趣区域样本群组入口中的元素用于指示全景视频的感兴趣区域以及与之关联的媒体资源。

8.2.1.8.2语法

class ROISampleGroupEntry() extends SampleGroupDescriptionEntry (‘roig’) {

unsigned int(8) roi\_count;

for (i=0;i<roi\_count;i++) {

ROIRegionStruct();

unsigned int(8) track\_count;

for (i=0;i<track\_count;i++) {

unsigned int(32) reference\_type;

unsigned int(32) track\_ref\_idx;

}

}

}

8.2.1.8.3语义

roi\_count 指示感兴趣区域的数量；

ROIRegionStruct() 指示感兴趣区域相关信息，包括空间位置、感兴趣区域标识符、关联信息等。

trackTrack\_count指示与感兴趣区域关联的媒体轨道数量；

reference\_type指示与感兴趣区域存在关联关系的媒体资源轨道在视频轨道引用数据盒中的引用类型；

track\_ref\_idx指示与感兴趣区域相关联的媒体轨道标识符在轨道引用数据盒中的索引，轨道引用数据盒类型由reference\_type提供。

8.2.1.9 球面区域缩放数据盒

8.2.1.9.1定义

数据盒类型（Box Type）: 'srwz'  
数据盒容器（Container）: ProjectedOmniVideoBox 或 VisualSampleEntry  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

视频轨道样本入口中的球面区域缩放数据盒用于确定全景视频的球面区域信息，以及球面区域在全景视频的投影图像上的一个或者多个缩放区域信息。

8.2.1.9.2语法

aligned(8) class SphereRegionZoomingBox extends FullBox('srwz', 0, 0) {

unsigned int(8) region\_definition\_type;

unsigned int(8) num\_regions;

unsigned int(1) remaining\_area\_flag;

unsigned int(1) view\_idc\_presence\_flag;

bit(6) reserved = 0;

if (view\_idc\_presence\_flag == 0) {

unsigned int(2) default\_view\_idc;

bit(6) reserved = 0;

}

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

if (view\_idc\_presence\_flag == 1) {

unsigned int(2) view\_idc;

bit(6) reserved = 0;

}

if ((i < (num\_regions - 1)) || (remaining\_area\_flag == 0)){

SphereRegionStruct(1);

unsigned int(1) region\_zoom\_flag;

bit(7) reserved = 0;

}

if(region\_zoom\_flag){

RegionWiseZoomingStruct();

}

}

}

8.2.1.9.3 语义

region\_definition\_type 指示球面区域形状。该字段取值0表示球面区域由四个大圆圈指定，取值1表示球面区域由两个方位圆圈和两个仰俯圆圈指定；

num\_regions 指示支持缩放的球面区域的数量，该字段取值大于等于1，表示确定的球面区域至少为一个；

remaining\_area\_flag 取值0表示通过SphereRegionStruct(1)结构定义球面区域，取值1表示除了由num\_regions-1个SphereRegionStruct(1)结构定义球面区域之外剩余的球面区域；

view\_idc\_presence\_flag 取值0表示不存在view\_idc，取值1表示存在view\_idc，指示球面区域与特定视图（左、右或两者）或单目图像关系；

default\_view\_idc取值0表示球面区域是单目图像，取值1表示球面区域是立体图像的左视图，取值2表示球面区域是立体图像的右视图，取值3表示球面区域包含立体图像的左视图和右视图；

view\_idc 取值0表示球面区域是单目图像，取值1表示球面区域是立体图像的左视图，取值2表示球面区域是立体图像的右视图，取值3表示球面区域包含立体图像的左视图和右视图；

region\_zoom\_flag取值0表示全景视频的投影图像上不存在与球面区域对应的缩放区域，取值1表示全景视频的投影图像上存在与球面区域对应的一个或多个缩放区域。

8.2.1.10 二维区域缩放数据盒

8.2.1.10.1定义

数据盒类型（Box Type）: '2dwz'  
数据盒容器（Container）: ProjectedOmniVideoBox or VisualSampleEntry  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

视频轨道样本入口中的二维区域缩放数据盒用于确定全景视频的投影图像上的二维区域信息，以及二维区域在全景视频的投影图像上的一个或者多个缩放区域信息。

8.2.1.10.2语法

aligned(8) class 2dRegionZoomingBox extends FullBox（'2dwz', 0, 0) {  
 unsigned int(8) num\_regions;  
 unsigned int(1) remaining\_area\_flag;  
 unsigned int(1) view\_idc\_presence\_flag;  
 bit(66) reserved = 0;  
 if (view\_idc\_presence\_flag == 0) {  
 unsigned int(2) default\_view\_idc;  
 bit(6) reserved = 0;  
 }  
 for (i = 0; i < num\_regions; i++) {  
 if (view\_idc\_presence\_flag == 1) {  
 unsigned int(2) view\_idc;  
 bit(6) reserved = 0;  
 }  
 if ((i < (num\_regions –1)) || (remaining\_area\_flag == 0)) {  
 unsigned int(16) left\_offset;  
 unsigned int(16) top\_offset;  
 unsigned int(16) region\_width;  
 unsigned int(16) region\_height;  
 }  
 unsigned int(1) region\_zoom\_flag;  
 bit(7) reserved = 0;  
 if(region\_zoom\_flag)  
 RegionWiseZoomingStruct();  
 }  
}

8.2.1.10.3 语义

num\_regions 指示支持缩放的二维区域的数量，该字段取值大于等于1，表示确定的二维区域至少为一个；

remaining\_area\_flag 取值为0时表示通过left\_offset，top\_offset，region\_width和 region\_height定义二维平面区域，取值为1时表示除了由num\_regions-1个上述定义二维平面区域之外剩余的二维平面区域；

view\_idc\_presence\_flag 取值为0时表示不存在view\_idc，取值为1时表示存在view\_idc，指示球面区域与特定视图（左、右或两者）或单目图像关系；

default\_view\_idc指示缺省的球面区域对应的图像。该字段取值为0时表示球面区域是单目图像，该字段取值为1时表示球面区域是立体图像的左视图，该字段取值为2时表示球面区域是立体图像的右视图，该字段取值为3时表示球面区域包含立体图像的左视图和右视图；

view\_idc指示球面区域对应的图像。该字段取值为0时表示球面区域是单目图像，该字段取值为1表示球面区域是立体图像的左视图，该字段取值为2时表示球面区域是立体图像的右视图，该字段取值为3时表示球面区域包含立体图像的左视图和右视图；

left\_offset和top\_offset分别指示投影图像上的二维平面区域位置的左上角的水平和垂直坐标。

region\_width和region\_height分别指示投影图像上的二维平面区域的宽度和高度；

region\_zoom\_flag取值为0时表示全景视频的投影图像上不存在与二维区域对应的缩放区域，取值为1时表示全景视频的投影图像上存在与二维区域对应的一个或多个缩放区域。

8.2.1.11 独立编解码区域数据盒

8.2.1.11.1 独立编解码区域关联关系

8.2.1.11.1.1 定义

独立编解码区域描述数据盒IndependentlyCodedRegionDescriptionBox（扩展自TrackGroupTypeBox）表明包含该数据盒的轨道属于某个轨道组，该轨道组内的各个轨道之间存在独立编解码区域关联关系（比如一个源视频在平面帧上划分为多个子视频，则各子视频间存在独立编解码区域关联关系）。每个独立编解码区域描述数据盒IndependentlyCodedRegionDescriptionBox都包含一个标识该轨道组的track\_group\_id，属于同一个轨道组的轨道均源自于相同的视频内容且通常在客户端联合呈现。在同一个轨道组内，包含相同composition\_id的轨道在重构时参考同样的坐标系，该坐标系以视频帧的左上顶点为绝对原点（0,0）且重构时的最大宽高由total\_width和total\_height字段定义；该坐标系的x轴由左自右，y轴从上至下。

IndependentlyCodedRegionBox是可选数据盒，若存在，则：

1. 包含于IndependentlyCodedRegionDescriptionBox 且对应的轨道内不再包含 IndependentlyCodedRegionDescriptionEntry（该轨道的视频大小、位置不变）; 或者不包含于IndependentlyCodedRegionDescriptionBox 但对应的轨道内存在一个或多个 IndependentlyCodedRegionDescriptionEntry （该轨道的样本可能对应多个独立编解码区域，且这些独立编解码区域一定属于同一个composition\_id）。
2. 当重构独立编解码区域时，不同独立编解码区域之间允许存在间隙，也允许存在重叠。若存在重叠，此时相互重叠的独立编解码区域对应的轨道，其TrackHeaderBox中的layer字段取值必须互不相同。若多个独立编解码区域对应的轨道中，TrackHeaderBox中的layer字段取值相同，则这些独立编解码区域不允许相互重叠。
3. 当独立编解码区域对应的轨道中包含CoverageInformationBox时，该独立编解码区域的球面覆盖信息由CoverageInformationBox指示。
4. 独立编解码区域的空间信息受限于对应轨道的色度二次采样格式；total\_width、total\_height、region\_vertex\_x、region\_vertex\_y、region\_width以及region\_height字段取值应为整数且满足以下限制：
5. 当采样格式为4:4:4时，取值无限制；
6. 当采样格式为4:2:2时，total\_width、region\_vertex\_x、region\_width取值需为偶数值；
7. 当采样格式为4:2:0时，所有字段取值均需为偶数值。

8.2.1.11.1.2 语法

aligned(8) class CompositionInfoBox extends FullBox('cifo', 0, 0) {  
 unsigned int(1) multi\_resolution\_flag;  
 bit(7) reserved =0;  
 if(multi\_resolution\_flag == 1){  
 unsigned int(8) composition\_id;  
 }   
 unsigned int(32) total\_width;  
 unsigned int(32) total\_height;  
}

aligned(8) class IndependentlyCodedRegionBox extends FullBox('icrg',0,0) {  
 unsigned int(16) region\_vertex\_x;  
 unsigned int(16) region\_vertex\_y;  
 unsigned int(16) region\_width;  
 unsigned int(16) region\_height;  
 unsigned int(1) track\_not\_alone\_flag;  
 unsigned int(1) track\_not\_mergeable\_flag;

unsigned int(1) track\_priority\_info\_flag;

bit(5) reserved = 0;

if(track\_priority\_info\_flag == 1){

unsigned int(8) track\_priority;

}

}

aligned(8) class IndependentlyCodedRegionDescriptionBox extends TrackGroupTypeBox('icrr') {  
 // track\_group\_id 继承自TrackGroupTypeBox;  
 CompositionInfoBox(); // 必选  
 IndependentlyCodedRegionBox(); // 可选  
}

8.2.1.11.1.3 语义

multi\_resolution\_flag取值为0时表示当前轨道组内所有轨道来自于同一个分辨率的视频；取值为1时表示当前轨道组内所有轨道来自于多个分辨率的视频。

composition\_id 指示同一个内容源下不同重构坐标系的标识符，若该字段缺失，则说明当前轨道组内所有轨道均属于同一个重构坐标系。

total\_width 指示同一个独立编解码区域轨道组内且同一个重构坐标系下，各个独立编解码区域重构时允许的最大宽度，以像素为单位。对于track\_group\_id取值相同，且composition\_id取值也相同的轨道，其IndependentlyCodedRegionDescriptionBox中total\_width字段的取值应全部相等。

total\_height指示同一个独立编解码区域轨道组内且同一个重构坐标系下，各个独立编解码区域重构时允许的最大高度，以像素为单位。对于track\_group\_id取值相同，且composition\_id取值也相同的轨道，其IndependentlyCodedRegionDescriptionBox中total\_height字段的取值应全部相等。

region\_vertex\_x，region\_vertex\_y，region\_width，region\_height的语义见8.2.1.11.2章节。

track\_not\_alone\_flag取值为1时表示当前独立编解码区域轨道必须与同轨道组（轨道组类型为'icrr'）内的一个或多个独立编解码区域轨道同时呈现。取值为0时表示当前独立编解码区域轨道可以单独呈现，也可以与同轨道组（轨道组类型为'icrr'）内的一个或多个独立编解码区域轨道同时呈现。

track\_not\_mergeable\_flag取值为1时表示当前独立编解码区域轨道中携带的视频比特流无法与同轨道组（轨道组类型为'icrr'）内composition\_id取值相同的轨道携带的视频比特流，通过改写比特流头部数据的方式直接合并，以构成一个单独的，不存在解码不匹配的视频比特流。解码不匹配指解码当前轨道后的任一像素值不等于解码合并轨道后的对应像素值。比特流合并的一个用例为轨道样本入口类型等等于'hvc2'的HEVC比特流重构。track\_not\_mergeable\_flag取值为0时表示0当前独立编解码区域轨道中携带的视频比特流不可以与同轨道组（轨道组类型为'icrr'）内composition\_id取值相同的轨道携带的视频比特流，通过改写比特流头部数据的方式直接合并，以构成一个单独的，不存在解码不匹配的视频比特流。

track\_priority\_info\_flag取值为1时表示当前独立编解码区域轨道组内各轨道的优先级不同，此时每个独立编解码区域轨道的优先级由track\_priority字段给出。track\_priority\_info\_flag取值为0时表示当前独立编解码区域轨道组内各轨道的优先级相同。

track\_priority指示当前独立编解码区域在存储、传输、呈现过程中的优先级，该字段取值越小，说明对应轨道的优先级越高。

8.2.1.11.2 独立编解码区域样本群组

8.2.1.11.2.1 定义

当样本群组类型字段grouping\_type取值为'icrr'时，该群组对应的位置和尺寸信息属于一个独立编解码区域轨道组中的某个轨道的样本。当SampleToGroupBox类型字段grouping\_type取值为'icrr'时，其版本应取值为1，且grouping\_type\_parameter字段取值应等于对应独立编解码区域轨道组的track\_group\_id。

当重构各个独立编解码区域时，应按照如下步骤：

对属于同一个'icrr'轨道组的所有轨道，按照composition\_id将这些轨道分为对应的子组。

对于被选择轨道的按照合成时间对齐后的每个样本，将其按照TrackHeaderBox 中layer字段的信息从前到后排序，并遵循以下原则：

对于从0到region\_width – 1（包括region\_width – 1）的每个i值，以及从0到region\_height – 1（包括region\_height – 1）的每个j值，重构后的图像帧在( ( i + region\_vertex\_x ) % total\_width, ( j + region\_vertex\_y ) % total\_height ) 位置的像素值等于当前轨道的样本在位置(i, j)的像素值。（其中total\_width、total\_height的取值对应为各子组中的字段）

该样本群组在呈现以及字段取值上的限制，同8.2.11.1.1节。

8.2.1.11.2.2 语法

class IndependentlyCodedRegionDescriptionEntry extends VisualSampleGroupEntry('icrr') {  
 unsigned int(16) region\_vertex\_x;  
 unsigned int(16) region\_vertex\_y;  
 unsigned int(16) region\_width;  
 unsigned int(16) region\_height;  
}

8.2.1.11.2.3 语义

region\_vertex\_x 指示本群组的样本在当前composition\_id对应的重构坐标系下，左上顶点的水平坐标值。若因轨道宽度和高度需要重采样，则该坐标值为重采样之前的坐标值。该坐标值取值范围为0到total\_width – 1（包含total\_width − 1），total\_width字段包含于对应的IndependentlyCodedRegionDescriptionBox。

region\_vertex\_y 指示本群组的样本在当前composition\_id对应的重构坐标系下，左上顶点的垂直坐标值。若因轨道宽度和高度需要重采样，则该坐标值为重采样之前的坐标值。该坐标值取值范围为0到total\_height – 1（包含total\_height − 1），total\_height字段包含于对应的IndependentlyCodedRegionDescriptionBox。

region\_width 指示本群组的样本在当前composition\_id对应的重构坐标系下的宽度值。若因轨道宽度和高度需要重采样，则该宽度值为重采样之前的宽度值。该宽度值取值范围为1到total\_width（包含total\_width）。

region\_height 指示本群组的样本在当前composition\_id对应的重构坐标系下的高度值。若因轨道宽度和高度需要重采样，则该高度值为重采样之前的高度值。该高度值取值范围为1到total\_height（包含total\_height）。

8.2.2基于容积视频的数据盒

8.2.2.1 概述

本章节定义了基于容积视频的数据盒，可适用于多视图视频、视频编码点云等类型的容积视频。对于不同类型的容积视频，基于其数据特性对应不同的容积视频媒体轨道样本、数据盒等。

对于容积视频的媒体轨道，应使用ISO/IEC 14496-12标准中定义的MediaBox数据盒的HandlerBox数据盒中的处理程序类型（handler\_type）来标识，容积视频的handler\_type为'volv'。

8.2.2.2 容积视频媒体头

8.2.2.2.1 定义

数据盒类型（Box Type）： 'vvhd’

数据盒容器（Container）： MediaInformationBox

强制性（Mandatory）： 是

数量（Quantity）： 有且仅有1个

容积视频媒体轨道的MediaInformationBox数据盒（ISO/IEC 14496-12标准定义）中应包含VolumetricVisualMediaHeaderBox 数据盒。

8.2.2.2.2 语法

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

unsigned int(8) application\_media\_type;  
}

8.2.2.2.3 语义

version指示本数据盒的版本类型。

application\_media\_type指示应用场景使用的媒体数据类型，具体取值如表13所示：

表 13 媒体数据类型

|  |  |
| --- | --- |
| **取值** | **语义** |
| 0 | 多视图视频 |
| 1 | 基于视频编码的点云媒体 |
| 2~255 | 保留 |

8.2.2.3 容积视频样本入口

8.2.2.3.1 定义

容积视频轨道的样本入口应使用VolumetricVisualSampleEntry。

8.2.2.3.2 语法

class VolumetricVisualSampleEntry(codingname)   
 extends SampleEntry (codingname){  
 unsigned int(8)[32] compressorname;  
}

8.2.2.3.3 语义

compressorname指示容积视频编解码相关信息。该字段固定为32字节，其首字节为可展示信息的长度（假设为N），首字节后为以UTF-8编码的N个字节的可展示信息，可展示信息后为填充至最大长度的填充字节。该字段取值可被置0。

8.2.2.4 容积视频样本

容积视频样本格式由该容积视频媒体轨道对应的应用场景和编码标准定义。

8.2.2.5容积视频解码配置数据盒

8.2.2.5.1语法

class V3cConfigurationBox extends Box（'v3cc') {  
 V3cDecoderConfigurationRecord() V3cConfig;  
}

8.2.2.5.2语义

V3cDecoderConfigurationRecord定义在章节8.1.3.2中。

8.2.2.6容积视频单元头数据盒

8.2.2.6.1 定义

容积视频单元头数据盒包含于容积视频图集轨道的样本入口中，并且同时包含于所有的容积视频组件轨道的方案信息数据盒SchemeInformationBox的scheme\_specific\_data数据盒数组中。容积视频单元头数据盒包含容积视频单元头，用于描述各自轨道中承载的数据。

8.2.2.6.2 语法

aligned(8) class V3cUnitHeaderBox extends FullBox('vunt', version = 0, 0) {  
 v3c\_unit\_header unit\_header;   
}

8.2.2.6.3 语义

unit\_header指示一个v3c\_unit\_header()，该函数定义在ISO/IEC 23090-5中。

8.2.2.7 单轨道封装的数据盒

8.2.2.7.1 概述

单轨道的容积视频封装由一个单轨道声明（即样本入口类型）来表示容积视频码流，被称作容积视频码流轨道。容积视频数据的单轨道封装采用简单的方式直接封装容积视频码流，容积视频单元头数据结构按原样保留在码流中。

8.2.2.7.2 容积视频码流轨道的样本入口

8.2.2.7.2.1 定义

样本入口类型（Sample Entry Type）: 'v3e1', 'v3eg'  
数据盒容器（Container）: SampleDescriptionBox  
强制性（Mandatory）: 是，类型为'v3e1' 或'v3eh' 的样本入口  
数量（Quantity）: 一个或多个

一个容积视频码流轨道的样本入口应包含一个V3CConfigurationBox，该数据盒定义在章节8.2.2.5。

类型'v3e1'的样本入口下，所有的图集参数集和SEI信息应存在于setupUnit数组中。类型'v3eg'的样本入口下，图集参数集和SEI信息可以存在于setupUnit数组中，也可以存在于容积视频的码流中。

8.2.2.7.2.2 语法

class V3CConfigurationBox extends Box('v3cc') {   
 V3CDecoderConfigurationRecord() V3CConfig;   
}  
  
aligned(8) class V3CBitstreamSampleEntry() extends VolumetricVisualSampleEntry ('v3e1') {   
 V3CConfigurationBox config;  
}

8.2.2.7.2.3 语义

compressorname 指示压缩工具的名称，该参数在基类VolumetricVisualSampleEntry中定义。

8.2.2.7.3 容积视频码流的样本格式

一个容积视频码流的样本应包含一个或多个容积视频单元，该样本中的所有容积视频单元属于同一的表示时间，比如容积视频合成单元。一个样本可能是独立的（如同步样本）或者依赖其他样本进行解码。

8.2.2.8多轨道封装的数据盒

8.2.2.8.1 概述

容积视频数据的多轨道封装将容积视频码流中的容积视频单元按照类型映射到各自的媒体轨道的容器中，并且由多种类型的样本入口类型识别容积视频图集相关的轨道及容积视频组件轨道。同时，通过容积视频图集相关轨道的轨道索引关联并识别容积视频组件轨道，具体轨道索引类型参见章节6.3.2。

8.2.2.8.2 容积视频图集轨道样本入口

8.2.2.8.2.1定义

样本入口类型（Sample Entry Type）: 'v3c1', 'v3cg', 'v3cb', 'v3a1'或 'v3ag'  
数据盒容器（Container）: SampleDescriptionBox  
强制性（Mandatory）: 是，类型为 'v3c1' , 'v3cg' 或 'v3cb'的样本入口   
数量（Quantity）: 一个或多个

容积视频图集轨道应使用样本类型为'v3c1', 'v3cg', 'v3cb', 'v3a1', 或'v3ag'的V3CSampleEntry。容积视频图集轨道样本入口使用的具体规则如下：

一个容积视频图集轨道不应承载两个以上的图集的ACL NAL单元。

当容积视频码流包含唯一的图集时，应使用样本入口为'v3c1' 或 'v3cg'的容积视频图集轨道。

当容积视频码流包含多个图集时，每个图集子码流应封装在一个单独的容积视频图集轨道。其中存在一个容积视频图集轨道的样本入口类型应使用'v3cb'，承载容积视频参数集和通用的图集参数集。其他包含图集子码流的容积视频图集轨道的样本入口类型应使用'v3a1'或者 'v3ag'。

注：仅当VolumetricVisualMediaHeaderBox数据盒中application\_media\_type 字段取值为1时，容积视频文件中可以包含多个图集轨道。

类型为'v3cb'的容积视频图集轨道不应包含任何ACL NAL单元。

容积视频图集轨道样本入口应包含一个V3CConfigurationBox和一个V3CUnitHeaderBox。

样本入口类型为'v3a1' 和'v3ag'的容积视频图集轨道下，容积视频的参数集不应存放在v3cParameterSets数组中。

样本入口类型为'v3c1'的容积视频图集轨道下，所有图集子码流的参数集和SEI信息应存放在setupUnit数组中。

样本入口类型为'v3cg' 和 'v3cb'的容积视频图集轨道下，图集子码流的参数集和SEI信息可以存放在setupUnit数组中，或者也可以存放在图集的子码流中。

样本入口类型为'v3a1'的容积视频图集轨道下，具有相同图集标识的图集子码流的参数集应在setupUnit数组中。

样本入口类型为v3ag'的容积视频图集轨道下，具有相同图集标识的图集子码流的参数集可以在setupUnit数组中，或者也可以在图集子码流中。

一个容积视频的封装文件中，可同时存在多个样本入口类型的容积视频图集轨道，并且所有的图集数据及参数集信息只能包含在容积视频图集轨道。

8.2.2.8.2.2语法

class V3CConfigurationBox extends Box('v3cc') {  
 V3CDecoderConfigurationRecord() V3CConfig;  
}

aligned(8) class V3CSampleEntry() extends VolumetricVisualSampleEntry (type) {  
 V3CConfigurationBox config;  
 V3CUnitHeaderBox unit\_header;  
}

8.2.2.8.2.3语义

compressorname 指示压缩工具的名称，该参数在基类VolumetricVisualSampleEntry中定义。

8.2.2.8.3 容积视频图集分片轨道样本入口

8.2.2.8.3.1定义

样本入口类型（Sample Entry Type）: 'v3t1'  
数据盒容器（Container）: SampleDescriptionBox  
强制性（Mandatory）: 是  
数量（Quantity）: 一个或多个

容积视频图集分片轨道应使用样本入口类型为'v3t1'的AtlasTileSampleEntry。

一个容积视频图集分片轨道应只包含属于相同图集ACL NAL单元，并且包含至少一个分片的ACL NAL单元。当存在容积视频图集分片轨道时，容积视频图集轨道中不应包含图集的ACL NAL单元。

AtlasTileSampleEntry不应包含V3CConfigurationBox 或V3CUnitHeaderBox的信息，该信息由与容积视频图集分片轨道关联的容积视频图集轨道提供。AtlasTileSampleEntry可包含其他的可选数据盒。

8.2.2.8.3.2语法

aligned(8) class AtlasTileSampleEntry() extends VolumetricVisualSampleEntry ('v3t1') {  
 unsigned int(8) configurationVersion = 1;   
 unsigned int(2) lengthSizeMinusOne;  
 unsigned int(6) reserved = 0;  
 unsigned int(16) num\_tiles;   
 for(i=0; i < num\_tiles; i++){  
 unsigned int(16) tile\_id;   
 }  
}

8.2.2.8.3.3语义

configurationVersion指示一个版本域。通过版本号的变化指示样本入口的可兼容变换。

lengthSizeMinusOne 加1，指示该样本入口对应的容积视频样本中nalUnit.ssnu\_nal\_unit\_size域的长度，以字节为单位。该域的值应等于图集子码流的sample\_stream\_nal\_header()中的ssnh\_unit\_size\_precision\_bytes\_minus1的值。

num\_tiles指示该媒体轨道中包含的图集分片数量

tile\_id指示该媒体轨道中图集分片的标识，该标识在ISO/IEC 23090-5中定义。

compressorname 指示压缩工具的名称，该参数在基类VolumetricVisualSampleEntry中定义。

8.2.2.8.4容积视频图集的样本格式

8.2.2.8.4.1定义

容积视频图集轨道或容积视频图集分片轨道中每一个样本对应一个唯一的解码图集的访问单元。

在各个容积视频组件轨道中相应的当前帧样本应与容积视频图集/图集分片轨道中的当前帧样本具有相同的合成时间。每一个容积视频图集样本应只包含一个类型为V3C\_AD（在ISO/IEC 23090-5中定义）的容积视频单元荷载，但可能包含一个或多个图集NAL单元。

当使用类型为'v3cb'的样本入口时，容积视频图集轨道中每个样本对应一个或多个非ACL NAL单元。

当使用类型为'v3c1', 'v3cg', 'v3a1' 或者'v3ag'的样本入口，并且存在一个或多个容积视频图集分片轨道的时候，容积视频图集轨道中的每个样本对应相同图集的一个或多个非ACL NAL单元，对应图集的标识在样本入口的容积视频单元头数据盒中指示。

8.2.2.8.4.2语法

aligned(8) class V3CSample {  
 unsigned int Length = sample\_size;   
 for (i=0; i < Length; ) {  
 sample\_stream\_nal\_unit nalUnit;   
 i += (lengthSizeMinusOne+1) + nalUnit.ssnu\_nal\_unit\_size;  
 }  
}

8.2.2.8.4.3语义

lengthSizeMinusOne加1指示nalUnit.ssnu\_nal\_unit\_size域的长度，以字节为单位。该值在类型为'v3c1', 'v3cg', 'v3cb', 'v3a1'和 'v3ag'的样本入口下的V3CDecoderConfigurationRecord中指示，或者在类型为'v3t1'的样本入口AtlasTileSampleEntry中指示。

nalUnit 指示的NAL单元的样本流格式中单个图集NAL单元，在ISO/IEC 23090-5中定义。

8.2.2.8.5容积视频组件轨道数据盒

8.2.2.8.5.1概述

容积视频组件轨道采用传统的视频封装方式封装二维视频编码数据，容积视频组件轨道的存储采用现存的媒体封装格式和衍生的规范，如使用ISO/IEC 14492-15。

封装文件中的容积视频组件轨道应作为受约束的视频存在，并且使用一个通用类型识别的受约束样本入口。

8.2.2.8.5.2受约束的视频方案

容积视频组件视频轨道应作为受约束的视频在媒体文件中表示，并该类型轨道的样本入口下的RestrictedSchemeInfoBox 中SchemeTypeBox 的scheme\_type设置为'vvvc'。

属性、几何和占位组件数据的编码对视频编解码器没有限制，而且，这些组件数据可能采用不同的视频编解码器进行编码。

8.2.2.8.5.3方案信息

8.2.2.8.5.3.1概述

容积视频组件轨道中应存在SchemeInformationBox，并且该数据盒中包含一个V3cUnitHeaderBox，指示该容积视频组件轨道中的组件信息。

若容积视频组件轨道为封装视频组件轨道，当一个图集的所有图集分片在一个容积视频图集轨道传输时，则该组件轨道中可携带组件分片信息数据盒。

8.2.2.8.5.3.2定义

数据盒类型（Box Type）: 'rcti'  
数据盒容器（Container）: SchemeInformationBox  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

组件分片信息数据盒指示对应组件轨道所关联的图集分片信息。

8.2.2.8.5.3.3语法

aligned(8) class ComponentTileInfoBox () {

unsigned int(8) num\_tiles;

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

unsigned int(16) tile\_id;

}

}

8.2.2.8.5.3.4语义

num\_tiles是与当前组件轨道相关联的容积视频图集片的数量。

tile\_id是图集分片的标识。应与ISO/IEC 23090-5中的ath\_id相同。

8.2.2.8.6容积视频可替换组

8.2.2.8.6.1容积视频内容可替换组

容积视频内容可被编码为不同版本的内容。不同的可替换内容由ISO/IEC 14496-12中定义的可替换组机制（TrackHeaderBox中的alternate\_group字段）指示。若不同的容积视频图集轨道拥有相同alternate\_group取值，则说明容积视频图集轨道对应的容积视频内容之间互为可替换的内容。

8.2.2.8.6.2容积视频组件轨道可替换组

数据盒类型（Box Type）: 'valg'

数据盒容器（Container）: TrackGroupBox  
强制性（Mandatory）: 否  
数量（Quantity）: 0个多个

容积视频组件轨道可以存在可替换关系，相互之间可替换的组件轨道同属于一个容积视频可替换轨道组。可替换轨道组中的组件轨道，仅有一个能被相应的图集轨道或图集分片轨道索引。

8.2.2.8.6.3语法

aligned(8) class V3cAlternativeTrackGroupBox extends TrackGroupTypeBox('potg') {

// track\_group\_id 继承自TrackGroupTypeBox

V3cAlternativeInfoStruct()；

}

8.2.2.8.7容积视频播放轨道组

8.2.2.8.7.1定义

数据盒类型（Box Type）: 'potg'

数据盒容器（Container）: TrackGroupBox  
强制性（Mandatory）: 否  
数量（Quantity）: 0个或多个

当容积视频组件的多个可替换版本之间，仅有某些特定组合应被联合播放时，应使用容积视频播放轨道组来指示这些联合播放的组合。

播放轨道的轨道组类型为PlayoutTrackGroupBox（扩展自ISO/IEC 14496-12中的TrackGroupTypeBox）。PlayoutTrackGroupBox数据盒表示对应轨道属于构成一个播放组的一组轨道。对于播放组中的每一个轨道，该轨道的TrackGroupBox中需包含一个携带独特track\_group\_id的PlayoutTrackGroupBox数据盒。对于一个播放轨道组的所有组件轨道，可以有选择地定义其联合质量等级，用以指示不同质量的媒体内容。

8.2.2.8.7.2语法

aligned(8) class PlayoutTrackGroupBox extends TrackGroupTypeBox('potg') {

unsigned(1) int quality\_ranking\_flag;

bit(7) reserved = 0;

if(quality\_ranking\_flag == 1){

unsigned(8) int quality\_ranking;

}

}

8.2.2.8.7.3语义

quality\_ranking\_flag取值为1时表示容积视频播放轨道组的所有组件轨道具有联合质量等级；取值为0时表示容积视频播放轨道组的所有组件轨道不具有联合质量等级。

quality\_ranking 指示一个容积视频播放轨道组内所有组件轨道的联合质量等级，该字段取值越小，表明联合质量等级越高。

8.2.2.9 容积视频图集分片组件轨道分组

8.2.2.9.1 定义

图集分片（atlas tile）组件轨道组是对携带与图集片组（由一个或多个图集分片所组成）相关联的视频编码分量信息的所有组件轨道进行分组得到的轨道组。当一个图集的所有图集分片在一个容积视频图集轨道传输时，使用基于图集片分组的组件轨道组。

组件轨道中出现Track\_group\_type等于‘vtcg’的TrackGroupTypeBox表示该轨道属于与一个图集分片组相对应的一组V3C组件轨道。

属于相同图集分片组的组件轨道具有相同的Track\_group\_id值。对应不同图集分片组的组件轨道具有不同的Track\_group\_id值。

8.2.2.9.2 语法

aligned(8) class V3CAtlasTileComponentGroupBox extends TrackGroupTypeBox('vtcg') {  
 unsigned int(6) atlas\_id;   
 unsigned int(1) 3D\_spatial\_region\_flag;   
 bit(1) reserved = 0;  
 unsigned int(8) num\_tiles;  
 for (i=0; i < num\_tiles; i++) {  
 unsigned int(16) tile\_id;  
 }  
 if(3D\_spatial\_region\_flag)  
 3DSpatialRegionStruct(1);  
}

8.2.2.9.3 语义

TrackGroupTypeBox内具有相同Track\_group\_id值且Track\_group\_type等于‘vtcg’的轨道属于相同的图集分片组。TrackGroupTypeBox中在Track\_group\_type等于‘vtcg’时，Track\_group\_id被用作图集片组的标识符。

atlas\_id指示图集分片轨道组表示的图集分片所属的图集标识。

num\_tiles指示与图集分片轨道组相关联的容积视频图集片的数量。

tile\_id指示图集分片的标识。应与ISO/IEC 23090-5中的ath\_id相同。

3D\_spatial\_region\_flag取值为1表示图集分片轨道组对应一个三维空间区域；取值为0表示图集分片轨道组不对应一个三维空间区域。

3DSpatialRegionStruct(1)指示图集分片轨道组对应的三维空间区域，属于相同图集分片组的组件轨道对应同一个三维空间区域。

8.2.2.10 容积视频边框信息数据盒

8.2.2.10.1 定义

V3CSpatialRegionsBox存在时，V3cBoundingBox应该在容积视频图集轨道的样本入口中给出。

8.2.2.10.2 语法

aligned(8) class V3CBoundingInformationBox extends FullBox('vpbb',0,0) {  
 3DBoundingBoxStruct();  
}

8.2.2.10.3 语义

3dBoundingBoxStruct指示容积视频包围盒的具体信息。

8.2.2.11 静态区域信息数据盒

8.2.2.11.1 定义

数据盒类型（Box Type）: 'vpsr'  
数据盒容器（Container）: V3CSampleEntry  
强制性（Mandatory）: 否  
数量（Quantity）: 零个或一个

容积视频中的静态三维空间区域和该区域对应的图集/图集分片，以及其对应的媒体轨道应该在V3CspatialRegionsBox中指明。

注：三维空间区域可能相互重叠。

如果容积视频的码流包含描述一个场景中不同对象的信息并且此信息不随时间变化而变化，则定义V3CSpatialRegionsBox数据盒，提供对象与其关联的轨道组的映射关系。

当V3CspatialRegionsBox存在，但没有容积视频图集分片轨道与容积视频图集轨道关联时，容积视频图集轨道中应存在NALUMapEntry（在ISO/IEC 14496-15中定义），指示容积视频图集轨道中图集的所有图集分片，以支持对应图集的一个或多个图集分片的部分访问。在本规范中使用NALUMapEntry具体方式如下：

当其存在时，用于分配groupID作为每个图集NAL单元的标识

对于ACL NAL单元，groupID应该为0，意味着图集NAL单元需要用于解码其对应的相同图集帧的任何图集分片

当其存在时，可以关联或不关联一个样本组的描述，设定类型为‘nalm’的SampleToGroupBox中的grouping\_type\_parameter。因此，一个类型为‘nalm’的SampleToGroupBox可以使用也可以不使用版本号为0的数据盒。

8.2.2.11.2 语法

aligned(8) class V3CSpatialRegionsBox extends FullBox('vpsr', 0 ,0) {  
 bit(1) all\_tiles\_in\_single\_track\_flag;

bit(1) packed\_video\_track\_flag;

bit(6) reserved = 0;  
 unsigned int(16) num\_regions;  
 for (i=0; i < num\_regions; i++) {  
 3dSpatialRegionStruct(1);  
 if (all\_tiles\_in\_single\_track\_flag) {  
 if(packed\_video\_track\_flag){  
 unsigned int(8) num\_packed\_video\_track;  
 for(n=0; n< num\_packed\_video\_track; n++){  
 unsigned int(32) packed\_video\_track\_id;  
 unsigned int(8) num\_tiles;  
 for (k=0; k < num\_tiles; k++) {  
 unsigned int(16) tile\_id;  
 }  
 }  
 }else {

unsigned int(8) num\_ component\_track\_groups;

for (j=0; j < num\_ component\_track\_groups; j++) {  
 unsigned int(32) component\_track\_group\_id;  
 unsigned int(8) num\_tiles;  
 for (k=0; k < num\_tiles; k++) {  
 unsigned int(16) tile\_id;  
 }  
 }  
}

} else {  
 unsigned int(8) num\_tile\_tracks;  
 for (j=0; j < num\_tile\_tracks; j++) {  
 unsigned int(32) tile\_track\_id;  
 }  
 }  
}

}

8.2.2.11.3 语义

all\_tiles\_in\_single\_track\_flag指示图集对应的所有的图集分片承载在容积视频图集轨道，或者分别承载在容积视频图集分片轨道中。值为1时，支持所有的图集分片承载在容积视频图集轨道中，值为0时，指示分别承载在图集分片轨道中。

packed\_video\_track\_flag 指示容积视频图集/图集分片轨道是否引用封装视频组件轨道。值为1时，存在与容积视频图集/图集分片轨道相关的封装视频组件轨道，值为0时，不存在与容积视频图集/图集分片轨道相关的封装视频组件轨道。

num\_regions 指示容积视频中三维空间区域的数量。

num\_tile指示与三维空间区域相对应的图集分片的数量。

num\_packed\_video\_track指示与三维空间区域相关联的封装视频组件轨道数量。

packed\_video\_track\_id 指示与三维空间区域相关联的封装视频组件轨道标识。

num\_component\_track\_groups指示关联一个三维空间区域的容积视频组件轨道组的数量。

component\_track\_group\_id指示承载三维空间区域相关的分片的容积视频组件轨道对应的分片组件轨道组标识。

tile\_id指示与容积视频三维空间对应的图集分片的图集NAL单元的标识。当存在容积视频图集分片组件轨道组时，图集分片与图集分片组件轨道组中的图标识对应。tile\_id的值加1应该等于在容积视频轨道中NALUMapEntry（定义在ISO/IEC 14496-15） 包含的一个条目中的groupID。

num\_tile\_tracks指示与该三维空间区域对应的图集分片媒体轨道。

tile\_track\_id指示与该三维空间对应的图集分片轨道媒体轨道的标识。

8.2.2.12 容积视频的视图信息数据盒

8.2.2.12.1 定义

数据盒类型（Box Type）: 'vpvw'  
数据盒容器（Container）: V3CSampleEntry   
强制性（Mandatory）: 不  
数量（Quantity）: 零个或一个

容积视频的视图和对应的图集信息在V3cViewsBox中指示。一个容积视频图集轨道中的图集对应一个或多个视图，通常一个基础视图对应一个单独的图集。

8.2.2.12.2 语法

aligned(8) class V3CViewsBox extends FullBox('vpvw',0,0) {

unsigned int(16) num\_views;  
for (i=0; i < num\_views; i++) {  
 ViewInfoStruct(1);  
 unsigned int(8) num\_v3c\_tracks;  
 for (j=0; j < num\_v3c\_tracks; j++)   
 unsigned int(32) v3c\_track\_id;  
}

}

8.2.2.12.3 语义

num\_views 指示容积视频中视图的数量。

num\_v3c\_tracks指示与当前视图相关的容积视频图集轨道的数量。

v3c\_track\_id指示承载与当前视图对应的图集数据所在的容积视频图集轨道的标识。

8.2.2.13 容积视频的视图组数据盒

8.2.2.13.1 定义

数据盒类型（Box Type）: 'vwvg'  
数据盒容器（Container）: V3CSampleEntry   
强制性（Mandatory）: 不  
数量（Quantity）: 零个或一个

容积视频的多视图组包括对视图对应的图集的图集数据所在的容积视频图集轨道进行分组。

容积视频的多视图分组信息由V3CViewGroupsBox指示。一个视图组对应一个图集组，通过由图集数据所在的容积视频图集轨道组成的轨道组表示图集组与视图组之间的对应关系。目标视图只能从同一个视图组中选取一个或多个视图来合成/渲染。

8.2.2.13.2 语法

aligned(8) class V3CViewGroupsBox extends FullBox('vwvg',0,0) {

unsigned int(16) num\_view\_groups;  
for (i=0; i < num\_view\_groups; i++) {  
 ViewGroupInfoStruct();  
 unsigned int(32) v3c\_track\_group\_id;  
}

}

8.2.2.13.3 语义

num\_view\_groups 指示容积视频中视图分组的数量。

v3c\_track\_group\_id指示容积视频图集轨道组的标识，该容积视频图集轨道组承载与容积视频的视图组对应的所有图集数据。

8.2.2.14 基于多视图的轨道分组数据盒

8.2.2.14.1 定义

根据容积视频的多视图分组对相应图集数据所在的容积视频图集轨道进行分组，通过轨道组类型为'vwtg'的TrackGroupTypeBox表示该容积视频图集轨道属于一个基于图集组进行分组的容积视频图集轨道组，属于同一个图集组的容积视频图集轨道具有相同的track\_group\_id。

8.2.2.14.2 语法

aligned(8) class V3CTrackGroupBox extends trackGroupTypeBox('vwtg'){

}

8.2.2.15 封装视频子样本数据盒

8.2.2.15.1概述

封装视频组件轨道中封装对应一个或多个图集分片的封装视频码流，每一个样本被划分为多个子样本，子样本包含对应于图集分片的一个或多个容积视频单元类型。封装视频组件轨道中的容积视频单元包括几何数据，属性数据，占用数据类型的视频组件。通过解码封装视频组件轨道中的一个或多个子样本部分访问其对应于图集分片的封装视频码流。

使用封装视频组件轨道的样本表数据盒（SampleTableBox）或者每个电影片段数据盒（MovieFragmentBoxes）的轨道片段数据盒（TrackFragmentBox）中包含的子样本信息数据盒（SubSampleInformationBox）描述封装视频组件轨道中子样本信息。

每个子样本的容积视频单元的类型通过子样本信息数据盒中子样本入口的“编解码特定参数” （codec\_specific\_parameters）字段来指示，字段具体定义如下：

if (flags == 0) {

unsigned int(8) tile\_id;

unsigned int(2) type\_id;

if (type\_id == 4) {

unsigned int(7) attrIdx;

bit(15) reserved = 0;

}

else

bit(22) reserved = 0;

} else if (flags == 1) {

unsigned int(8) tile\_id;

bit(24) reserved = 0;

}

flag指示数据盒中描述的子样本信息的类型，flags值为0时，一个子样本中仅包含对应于一个图集分片的一个特定容积视频单元类型数据，flags值为1时，一个子样本包含对应于同一个图集分片的一个或多个封装类型的容积视频单元数据。

tile\_id指示与子样本相关联的图集瓦片标识（ID）

type\_id指示子样本中容积视频单元数据的类型。

attrIdx指示子样本中包含属性数据容积视频单元属性索引。

8.2.2.16三维观看空间数据盒

8.2.2.16.1定义

数据盒类型（Box Type）: 'vwsp'

数据盒容器（Container）: V3CSample Entry或trackbox

强制性（Mandatory）: 否

数量（Quantity）: 零个或一个

三维观看空间数据盒描述用户在沉浸媒体的场景中被允许的可移动范围，用于限制用户的观看位姿、位置、朝向等及其对应的观看内容。

8.2.2.16.2语法

aligned(8) class ViewingSpaceBox extends FullBox('vwsp', 0, 0) {

signed int（8） num\_viewing\_space;  
for(int j=0;j<= num\_viewing\_space;j++) {   
 unsigned int(16) viewing\_space\_id;  
 ViewingSpaceStruct ();   
 }

}

8.2.2.16.3语义

num\_viewing\_space指示观看空间的数量；

viewing\_space\_id指示观看空间的表示，可使用与容积视频的视点组对应的容积视频轨道组的标识。

8.2.2.17观看空间处理数据盒

8.2.2.17.1定义

数据盒类型（Box Type）: 'vshd'

数据盒容器（Container）: V3CSample Entry

强制性（Mandatory）: 否

数量（Quantity）: 零个或一个

观看空间处理数据盒描述观看空间的处理方式信息，当用户的位姿变化（移动到三维观看空间内部、边缘或外面时）时，根据观看空间的处理方式对用户位姿对应的媒体内容进行获取、重构、渲染等处理。

8.2.2.17.2语法

aligned(8) class ViewingSpaceHandlingBox() extends fullBox(‘vshd’,0,0){

unsigned int(16) num\_viewing\_space;   
for(i=0; i<= num\_viewing\_space; i++){  
 unsigned int(16) viewing\_space\_id;  
 unsigned int(8) num\_handling\_options;  
 for (j=0; j<= num\_handling\_options; j++) {  
 unsigned int(8) handling\_device\_class;  
 unsigned int(8) handling\_application\_class;  
 unsigned int(8) handling\_method;  
 }  
}

}

8.2.2.17.3语义

num\_viewing\_space，指示场景中的三维观看空间数量；

viewing\_space\_id，指示该处理方式适用的三维观看空间的标识；

num\_handling\_options，指示三维观看空间处理方式的选项数量；为0时，表示没有提供三维观看空间处理的方式，则目标设备可根据三维观看空间信息选择合适的处理方式；

handling\_device\_class，指示三维观看空间处理的设备类型，具体描述如下表所示；在同一个三维观看空间中不应该有重复的值，如果值为0时，j+1的值应该为num\_handling\_options；

表 14 三维观看空间处理的设备类型

|  |  |
| --- | --- |
| **值** | **描述** |
| 0 | 匹配所有设备，支持6DOF的位置跟踪 |
| 1 | 支持 6DOF播放头戴设备 |
| 2 | 依赖惯性传感器（IMU）渲染的手机或平板 |
| 3 | 自由立体播放（光场）设备的设备 |
| 4...63 | 保留 |

handling\_application\_class，指示三维观看空间处理的应用类型，具体描述如下表所示；在同一个三维观看空间中不应该有重复的值，如果值为0时，j+1的值应该为 num\_handling\_options；

表 15 三维观看空间处理的应用类型

|  |  |
| --- | --- |
| **值** | **描述** |
| 0 | 匹配所有应用 |
| 1 | 编码的沉浸媒体用于增强现实 |
| 2 | 编码的沉浸媒体用于虚拟现实 |
| 3 | 编码的沉浸媒体用于website |
| 4 | 编码的沉浸媒体用于大屏描述中的一个元素 |
| 5...63 | 保留 |

handling\_method，指示三维观看空间处理方式，具体描述如下表所示；同一个三维观看空间中不应该有重复的值，如果类型值为0时，j+1的值应该为num\_handling\_options；

表 16 三维观看空间处理方式

|  |  |
| --- | --- |
| **值** | **描述** |
| 0 | 缺省的客户端行为 |
| 1 | 在三维观看空间外面时，总是渲染 |
| 2 | 当移动到三维观看空间外面时，屏幕淡出为缺省颜色 |
| 3 | 内容推断，用抽象低频法推断以避免渲染失真，但是保持场景的基本的色调 |
| 4 | 当观看者到达三维观看空间的极限时，其位置和/或方向重置 |
| 5 | 屏幕根据观看者旋转和平移以防止观看者到达三维观看空间的极限 |
| 6 | 屏幕随着观看者旋转以保持观看者在观看的区域内 |
| 7...63 | 保留 |

8.2.3基于自由视角视频的数据盒

8.2.3.1 自由视角视频的受限方案

自由视角视频轨道应作为受限的视频在媒体文件中表示，并将对应视频轨道的样本入口下的RestrictedSchemeInfoBox 中SchemeTypeBox 的scheme\_type设置为'as3f'。自由视角视频的编解码对视频编解码器没有额外限制。

8.2.3.2 自由视角信息数据盒

8.2.3.2.1 定义

数据盒类型（Box Type）: ' afvi '

数据盒容器（Container）: SchemeInformationBox或SampleEntry

强制性（Mandatory）: 不

数量（Quantity）: 零个或一个

自由视角信息数据盒指示对应轨道内包含的一个或多个自由视角及其对应的相机元数据信息。自由视角信息数据盒应包含于包含纹理信息的视频轨道中。

8.2.3.2.2 语法

aligned(8) class AvsFreeViewInfoBox extends FullBox('afvi'){

unsigned int(1) video\_with\_depth\_info\_flag;

bit(7) reserved = 0;

unsigned int(8) camera\_count;  
 for(i=0; i<camera\_count; i++){

unsigned int(1) virtual\_flag;

if(virtual\_flag == 0){

unsigned int(1) depth\_codec\_dependency;

unsigned int(1) camera\_codec\_independency;

}

else{

bit(2) reserved = 0;

}

bit(5) reserved = 0;

IntCameraInfoStruct();  
 ExtCameraInfoStruct();

}

}

8.2.3.1.3 语义

video\_with\_depth\_info\_flag取值为1时表示当前视频轨道同时包含纹理信息以及纹理信息对应的深度信息；取值为0时表示当前视频轨道仅包含纹理信息。

camera\_count：采集视频的所有相机的个数。

virtual\_flag，指示该视角对应的相机是否为虚拟相机；取值为1时，表示该视角对应的相机为虚拟相机；取值为0时，表示该视角对应的相机为原始相机。

depth\_codec\_dependency取值为1时表示当前相机的纹理信息和深度信息之间存在解码依赖关系。取值为0时表示当前相机的纹理信息和深度信息之间不存在解码依赖关系。

camera\_codec\_dependency取值为0时表示当前相机拍摄的纹理和深度信息可以独立解码；取值为1时表示当前相机拍摄的纹理和深度信息不可以独立解码。

IntCameraInfoStruct()指示相机的内参信息；

ExtCameraInfoStruct()指示相机的外参信息。

8.2.3.3 荷载信息数据盒

8.2.3.3.1 定义

数据盒类型（Box Type）: ' plin '  
数据盒容器（Container）: SampleEntry  
强制性（Mandatory）: 不  
数量（Quantity）: 零个或一个

荷载信息数据盒描述相应轨道对应的一个或多个相机的纹理图和深度图的元数据信息。

8.2.3.3.2 语法

aligned(8) class PayloadInfoBox extends FullBox(' plin', 0, 0){

unsigned int(1) texture\_info\_flag;  
unsigned int(1) depth\_info\_flag;

bit(6) reserved = 0;  
unsigned int(16) num\_cameras;  
for (i=0; i<num\_cameras; i++) {  
 unsigned int(16) camera\_id;  
 if(texture\_info\_flag == 1)  
 TextureInfostruct();  
 if(depth\_info\_flag == 1)  
 DepthInfostruct();  
}

}

8.2.3.3.3 语义

texture\_info\_flag指示是否表述描述纹理图的元数据信息；值为1时，表述元数据信息，值为0时，不表述元数据信息；

depth\_info\_flag指示是否表述描述深度图的元数据信息；值为1时，表述元数据信息，值为0时，不表述元数据信息；

num\_cameras指示该媒体轨道对应的相机数量；

camera\_id指示相机的标识；

TextureInfostruct()描述纹理图的元数据信息；

DepthInfostruct()描述深度图的元数据信息。

8.2.3.4 自由视角视频的单轨道封装

8.2.3.4.1 概述

自由视角视频的单轨道封装使用一个轨道来表示自由视角视频码流。采用单轨道封装方式将自由视角视频的数据直接封装到媒体轨道容器中，不对纹理数据和深度数据做额外处理。

对于单轨封装的自由视角视频轨道，其MediaBox数据盒的HandlerBox数据盒中的处理程序类型（handler\_type）应取值为'vide'。

8.2.3.4.2 样本入口

单轨道封装的自由视角视频轨道使用VisualSampleEntry样本入口类型，VisualSampleEntry在ISO/IEC 14496-12中定义。该样本入口应包含一个AvsFreeViewInfoBox数据盒和PayloadInfoBox数据盒，分别指示该自由视角视频轨道中的视角对应的相机信息，和相机对应的纹理图和深度图信息。

8.2.3.4.3自由视角子样本

单轨道封装的每个样本包含多个子样本来描述视角对应的多个相机的深度图和纹理图。若自由视角视频轨道中存在SubSampleInformationBox数据盒，则子样本的使用需依照子样本信息数据盒中flags字段的取值。flags字段取值的含义如下：

* 1. 0：基于相机划分的子样本。一个子样本中包含的数据对应一个相机；
  2. 1：基于自由视角数据类型划分的子样本。一个子样本可能包含：
     1. 相机的纹理信息；
     2. 相机的深度信息；
  3. 其余取值保留。

SubSampleInformationBox数据盒中codec\_specific\_parameters字段的取值如下：

if（flags == 0）{

unsigned int(16) camera\_id;

bit(16) reserved = 0;

}

else if (flags == 1) {

unsigned int(8) payloadType;

bit(24) reserved = 0;

}

payloadType字段取值如下表所示：

表 17 payloadType字段取值

|  |  |
| --- | --- |
| **payloadType取值** | **含义** |
| 0 | 子样本中包含纹理图信息 |
| 1 | 子样本中包含深度图信息 |
| 其他 | 保留 |

8.2.3.5 自由视角视频的多轨道封装

8.2.3.5.1概述

自由视角视频的多轨道封装使用多个媒体轨道封装自由视角视频码流，其中，自由视角视频的多轨道封装包含两种类型的媒体轨道，纹理图轨道和深度图轨道。采用多轨道封装方式将自由视角视频按照数据类型映射到各自的媒体轨道的容器中。属于同一相机的轨道之间通过轨道参考相互关联，并且时间是对齐的。

对于多轨封装的自由视角视频，包含纹理信息的视频轨道的HandlerBox数据盒中的处理程序类型（handler\_type）应取值为'vide'。包含深度信息的视频轨道的HandlerBox数据盒中的处理程序类型（handler\_type）应取值为'auxv'。

8.2.3.5.2 样本入口

多轨道封装的自由视角视频的纹理图轨道和深度图轨道都使用VisualSampleEntry样本入口类型，VisualSampleEntry在ISO/IEC 19946-12中定义。

纹理图轨道和深度图轨道的样本入口中包含AvsFreeViewInfoBox数据盒，其中，该数据盒中描述的相机参数信息不应重复出现在纹理图轨道和其对应的深度图轨道。

当纹理图轨道包含多个相机的纹理图时，其样本入口中应包含PayloadInfoBox数据盒，用于描述纹理图信息；当深度图轨道包含多个相机的的深度图时，其样本入口包含PayloadInfoBox数据盒，用于描述深度图信息。

8.2.3.5.3 样本格式

纹理图轨道或深度图轨道的样本分别对应一个或多个相机图或深度图。轨道中样本对应的相机信息，通过样本入口中的AvsFreeViewInfoBox进行描述。

当样本中包含多个相机的纹理图或深度图时，该样本可以划分为多个子样本，每个子样本对应一个相机的纹理图或深度图，使用SubSampleInformationBox数据盒进行描述。SubSampleInformationBox数据盒的使用应遵循8.2.3.2.3中的定义。

8.2.3.5.3 轨道参考

当自由视角视频中包含相同相机对应的纹理图轨道和深度图轨道时，深度图轨道通过reference\_type为‘vdep’ 或‘vdpi’的轨道参考关联到相同相机对应的纹理图轨道。当深度图轨道中包含的轨道参考类型为‘vdep’时，表示深度图和纹理图之间存在编解码依赖；当深度图轨道中包含的轨道参考类型为‘vdpi’时，表示深度图和纹理图之间不存在编解码依赖。

当自由视角视频中包含原始相机视频图像和预先合成的虚拟相机视频图像时，虚拟相机视频图像所在的媒体轨道通过reference\_type为“cvtr”轨道参考关联到原始相机视频图像所在的媒体轨道，表示虚拟相机和原始相机的关联关系。

8.2.3.6 自由视角轨道组

8.2.3.6.1 定义

若一个自由视角视频被封装为多个视频轨道，且一个视频轨道中仅包含部分相机对应的纹理和/或深度信息，则其中包含纹理信息的视频轨道应通过自由视角轨道组进行关联，此时这些包含纹理信息的视频轨道之间应具备编解码独立性。包含部分相机且对应同一个自由视角视频的不同视频轨道应具备相同的track\_group\_id。

8.2.3.6.2 语法

aligned(8) class AvsFreeViewGroupBox extends TrackGroupTypeBox('afvg') {  
 // track\_group\_id is inherited from TrackGroupTypeBox;

unsigned int(8) camera\_count;  
 for(i=0; i<camera\_count; i++){  
 unsigned int(16) camera\_id;

}  
}

8.2.3.6.3 语义

camera\_count：指示该轨道中包含的自由视角纹理信息（和深度信息）对应的相机数目。

camera\_id：指示每个相机对应的相机标识符。

8.3沉浸媒体定时元数据

8.3.1 基于球面区域的定时元数据

8.3.1.1 概述

本章定义了沉浸媒体特定球面区域的定时元数据。定时元数据轨道描述的使用目的由样本入口的类型所指示。每一个样本对应一个球面区域，对所有样本的指示信息由描述各样本通用属性的常规内容以及描述各样本特殊信息的扩展内容共同组成。

当一个球面区域定时元数据轨道通过一个‘cdsc’轨道参考，关联或参考到一个或多个媒体轨道时，将分别描述每个媒体轨道。

注：语法允许一个样本去指定多个球面区域。然而，也可以通过语义来限制多个样本只有一个球面区域。

8.3.1.2 样本入口

8.3.1.2.1 定义

每一个样本入口必须包含一个SphereRegionConfigBox，其指示了样本所覆盖的球面区域的基本信息。若各样本的球面覆盖区域一致，则在SphereRegionConfigBox中指示各样本的球面覆盖范围信息。

8.3.1.2.2 语法

class SphereRegionSampleEntry(type) extends MetaDataSampleEntry(type) {

SphereRegionConfigBox(); // 强制的

Box[] other\_boxes; // 可选的

}

class SphereRegionConfigBox extends FullBox('rosc', 0, 0) {

unsigned int(8) shape\_type;

bit(7) reserved = 0;

unsigned int(1) dynamic\_range\_flag;

if (dynamic\_range\_flag == 0) {

unsigned int(32) static\_azimuth\_range;

unsigned int(32) static\_elevation\_range;

}

unsigned int(8) num\_regions;

}

8.3.1.2.3 语义

shape\_type 指示球面区域的形状类型。该字段取值为1时表示球面区域通过8.1.4中定义的四个大圆和中心点来指定，如图11所示；该字段取值为1时表示球面区域通过8.1.4定义的两个方位角圆和两个俯仰角圆、和中心点来指定，如图12所示。该字段保留大于1的取值。

dynamic\_range\_flag 取值为0时，表示该样本入口中所有样本的球面区域的范围固定；取值为1时，表示该样本入口各样本的球面区域范围在样本格式中指示。

static\_azimuth\_range ， static\_elevation\_range 分别指示该样本入口中每个样本的球面区域中以球面区域中心点为中心的方位角范围和俯仰角范围，以2-16为精度，如图11或图12所示。static\_azimuth\_range 的取值范围是[0，360 \* 216]。static\_elevation\_range的取值范围是[0，180 \* 216]。当static\_azimuth\_range 与 static\_elevation\_range出现，并且取值同时为0时,该样本入口的各样本的球面区域即是球面上的一点。当static\_azimuth\_range 与static\_elevation\_range出现，azimuth\_range与elevation\_range的取值需要分别与static\_azimuth\_range与static\_elevation\_range一致。

num\_regions 指示该样本入口中各样本的球面区域数量。num\_regions应设置为1~~。~~，保留其他取值（SphereRegionZoomSampleEntry 中的num\_regions字段除外）。

8.3.1.3 样本格式

8.3.1.3.1 定义

每一个样本对应一个特定球面区域，各样本的球面区域覆盖信息由SphereRegionSample提供。在描述样本的扩展信息时，需要以此为基础进行进一步扩展。

8.3.1.3.2 语法

aligned(8) SphereRegionSample() {

for (i = 0; i < num\_regions; i++)

SphereRegionStruct(dynamic\_range\_flag)

}

8.3.1.3.3 语义

SphereRegionStruct()中，interpolate 取值为0时指示该样本中的centre\_azimuth, centre\_elevation, centre\_tilt, azimuth\_range (如果出现)，与 elevation\_range (如果出现)直接应用于目标媒体样本的描述。interpolate 取值为1时指示该样本中的centre\_azimuth, centre\_elevation, centre\_tilt, azimuth\_range (如果出现)，与 elevation\_range (如果出现)基于当前样本以及之前样本进行线性插值以描述目标媒体样本。轨道的第一个样本，或部分轨道的第一个样本，其interpolate必须取值为0。

8.3.1.4 初始观看方向

8.3.1.4.1 定义

该信息指示在播放相应媒体内容时初始观看方向。当缺少这一信息时，按照默认初始观看方向播放，其中centre\_Azimuth、centre\_Elevation和centre\_tilt分别为0。

样本入口类型应设置为‘invo’。

8.3.1.4.2 语法

class InitialViewingOrientationSample() extends SphereRegionSample() {

unsigned int(1) refresh\_flag;

bit(7) reserved = 0;

}

8.3.1.4.3 语义

centre\_azimuth, centre\_elevation, 与 centre\_tilt 指示观看方向， 以2−16 为精度。 centre\_azimuth与centre\_elevation指示视窗的中心点，centre\_tilt指示视窗的倾斜角。

interpolate必须取值为0。

refresh\_flag 取值为0时指示在开始播放相应媒体轨道的时间并行样本时，需要基于该初始视点信息进行播放；取值为1时指示在连续播放每一个相应媒体轨道的时间并行样本时都需要基于该初始视点信息进行播放（例如包括在连续播放以及从时间并行样本开始播放的场景下）。

8.3.1.5 推荐视窗

8.3.1.5.1 定义

该信息指示播放全景视频的视窗。在用户端不具有观看方向自主切换功能或用户放弃（释放）对观看方向切换的控制时，播放推荐视窗的媒体内容。同时，当媒体文件中包含多个视点时，推荐视窗信息中关联的推荐视点信息可以指导对应的视点切换。

注：推荐视窗的定时元数据轨道可以基于导演剪辑或观看统计推测来指明一个推荐视窗。

样本入口类型应设置为‘rcvp’。

8.3.1.5.2 样本入口语法

class RcvpSampleEntry() extends SphereRegionSampleEntry('rcvp') {

RcvpInfoBox(); // 强制的

AssoRcViewpointInfoBox();

}

class RcvpInfoBox extends FullBox('rvif', 0, 0) {

unsigned int(8) viewport\_type;

string viewport\_description;

}

class AssoRcViewpointInfoBox extends FullBox('avpo', 0, 0) {

unsigned int(1) independent\_viewport\_flag;

unsigned int(1) viewpoint\_switch\_flag;

bit(6) reserved=0;

if(independent\_viewport\_flag){

unsigned int(32) viewpoint\_id;

}

8.3.1.5.3 样本入口语义

viewport\_type指示的推荐视窗类型如下表17所示：

表 18 推荐视窗类型

|  |  |
| --- | --- |
| **取值** | **描述** |
| 0 | 导演剪辑的推荐视窗，如：按内容作者或内容提供者的创作意图建议的视窗 |
| 1 | 基于观看统计计算选择的推荐视窗 |
| 2..239 | 保留 |
| 240..255 | 未指定（应用或外部标准使用） |

viewport\_description是以空字符结尾的字符串，提供推荐视窗的文字描述。

在样本入口的SphereRegionConfigBox中，shape\_type取值应该为0。如果static\_azimuth\_range和static\_elevation\_range存在，或azimuth\_range 和elevation\_range存在时，分别指定推荐视窗的方位角和仰俯角的范围。centre\_azimuth和 centre\_elevation 分别指示全局坐标轴的球面区域的中心点，即推荐视窗的中心点。centre\_tilt指示推荐视窗的倾斜角。

independent\_viewport\_flag取值为1时，表示当前推荐视窗轨道关联一个独立的视点，该视点的标识符由当前推荐视窗轨道中的viewpoint\_id字段指示。取值为0时，表示当前媒体文件仅存在一个推荐视窗轨道，该推荐视窗轨道关联所有的视点。

viewpoint\_switch\_flag取值为1时，表示在某个推荐视窗呈现后，需要切换到另一个视点的推荐视窗；取值为0时，表示不需要切换呈现的推荐视窗。

viewpoint\_id指示当前样本入口对应的视点标识符。

8.3.1.5.4 样本语法

class RecommendedViewportSample() extends SphereRegionSample() {

if (viewpoint\_switch\_flag){

unsigned int(32) des\_viewpoint\_id;

unsigned int(1) des\_viewport\_info;

unsigned int(1) timeline\_switch\_flag;

bit(6) reserved=0;

if(des\_viewport\_info){

SphereRegionStruct(1);

}

if(timeline\_switch\_flag){

int(32) relative\_t\_offset;

}

}

}

8.3.1.5.5 样本语义

des\_viewpoint\_id指示该推荐视窗呈现后，需要切换的目标视点的标识符。

des\_viewport\_info取值为1时，表示切换至目标视点后的视窗区域由SphereRegionStruct指定；取值为0时，表示切换至目标视点后的视窗区域维持当前视窗区域不变。

timeline\_switch\_flag取值为1时，表示切换至目标视点后，媒体播放时间线会发生偏移；取值为0时，表示切换至目标视点后，媒体播放时间线保持不变。

relative\_t\_offset指示切换至目标视点后，相对于当前播放时间的偏移量。

8.3.1.6 定时文本球体位置元数据

8.3.1.6.1 概述

定时文本球体位置元数据指示了定时文本在球体三维空间内的位置和显示信息。由于定时文本要在三维空间内一定的位置显示,也就是只有该位置在球体内视窗区域才会被显示出来。

8.3.1.6.2 样本入口格式

样本入口类型应设置为‘ttsl’。此样本类型的样本入口定义为：

class TTSphereLocationSampleEntry() extends SphereRegionSampleEntry('ttsl') {

unsigned int(1) depth\_included\_flag;

bit(7) reserved = 0;

}

depth\_included\_flag 等于1代表样本中定时文本的深度值(z值)存在，0表示深度值不存在。

当SphereRegionSampleEntry() 类对象定义为TTSphereLocationSampleEntry()时， shape\_type，dynamic\_range\_flag，static\_azimuth\_range，和 static\_elevation\_range的值都为0。num\_regions的值可以大于1。

8.3.1.6.3 样本格式

定时文本球体位置元数据的样本格式定义为：

aligned(8) TTSphereLocationSample() extends SphereRegionSample() {  
 for (i=0; i<num\_regions; i++) {  
 string region\_id;  
 if (depth\_included\_flag)  
 unsigned int(16) region\_depth;  
 }  
}

region\_id 定义了定时文本显示的区域ID，region\_id应该与IMSC1或WebVTT中定时文本流的相应区域ID相同。

region\_depth定义了定时文本显示区域的深度值(z值),规定为定时文本区域里的向量,取值为针对单位球的相对值2-16。

当SphereRegionStruct包含在TTSphereLocationSample()结构中时，就表明使用了定时文本球体位置, 并与其他信息一起确定了定时文本在三维空间的显示位置。

8.3.1.7 球面区域缩放定时元数据

8.3.1.7.1 定义

球面区域缩放定时元数据指示了对应球面区域的视频内容在进行自动缩放时的具体缩放比例。当存在球面区域缩放定时元数据时，文件中一定存在SphereRegionZoomingBox且region\_zoom\_flag一定取值为1。

当一个球面区域缩放定时元数据轨道通过一个‘cdsc’轨道参考，参考到一个或多个媒体轨道时，将分别描述每个媒体轨道。

8.3.1.7.2 样本入口

球面区域缩放定时元数据样本入口类型应设置为‘srzm’。此样本类型的样本入口定义为：

class SphereRegionZoomSampleEntry() extends SphereRegionSampleEntry ('srzm') {

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

unsigned int(8) zoom\_ratio;

}

}

球面区域缩放定时元数据的样本入口中，dynamic\_range\_flag应置为1。shape\_type字段取值应与SphereRegionZoomingBox中的region\_definition\_type字段取值相同。num\_regions字段取值应等于SphereRegionZoomingBox中的num\_regions字段取值。

zoom\_ratio指示球面区域的初始缩放比例，以2-3为单位。该字段取值为0说明该区域保持原始尺寸，该字段取值为非0值则指示该区域缩放后的尺寸相较于原始尺寸的实际比例或近似比例。通过该字段，可以索引到SphereRegionZoomingBox中相同SphereRegionStruct对应的相同zoom\_ratio取值的缩放区域。

8.3.1.7.3 样本格式

球面区域缩放定时元数据的样本格式定义为：

aligned(8) SphereRegionZoomSample() extends SphereRegionSample() {  
 for (i=0; i<num\_regions; i++) {

SphereRegionStruct(dynamic\_range\_flag);

unsigned bit(1) zoom\_change\_flag;

bit(7) reserved=0;

if(zoom\_change\_flag == 1){

unsigned int(8) zoom\_ratio;

}

}

}

SphereRegionStruct()语义见8.1.1.4章节。

SphereRegionZoomSample中的SphereRegionStruct对应的球面区域应与SphereRegionZoomingBox中的SphereRegionStruct对应的球面区域相同且一一对应。

zoom\_change\_flag指示当前SphereRegionStruct的缩放比例是否变化。该字段取值为1时，说明对当前SphereRegionStruct的缩放比例发生变化，新的缩放比例由zoom\_ratio指定。

zoom\_ratio指示每一个缩放步骤实际执行的缩放比例，以2-3为单位。该字段取值为0说明该区域保持原始尺寸，该字段取值为非0值则指示该区域缩放后的尺寸相较于原始尺寸的实际比例或近似比例。通过该字段，可以索引到SphereRegionZoomingBox中相同SphereRegionStruct对应的相同zoom\_ratio取值的缩放区域。

8.3.2多视点定时元数据

8.3.2.1动态视点定时元数据

8.3.2.1.1概述

本章定义了视点动态变化的定时元数据。该定时元数据轨道用于指示全景视频轨道对应的动态变化的视点信息。

视点动态变化过程中，根据动态视点定时元数据轨道确定切换后视点信息，即当用户选择从一个视点切换至另一个视点，播放器播放切换后的视点对应的全景视频轨道。其中，播放视点切换后的全景视频轨道时，根据切换后视点的视点信息和/或视点切换前用户的观看方向确定的用户初始观看方向。

注1：如果存在推荐的初始观看方向，依照推荐的初始观看方向播放对应的全景视频轨道；否则，按照视点切换前用户的观看方向播放全景视频轨道。

在同一公共参考坐标系中播放切换后的全景视频轨道时，用户的初始观看方向与切换前用户的观看方向保持一致。

注2：在同一公共参考坐标系中播放切换后的全景视频轨道时，用户的初始观看方向与切换前用户的观看方向保持一致。

动态视点定时元数据轨道通过一个‘cdsc’cdsc’轨道参考，关联或参考到一个或多个全景视频的媒体轨道，并且每个媒体轨道可对应一个轨道群组标识。

8.3.2.1.2样本入口

每一个样本入口提供对应的媒体轨道视点的基本信息。动态视点定时元数据轨道的样本入口类型’'dyv’'，样本入口语法如下所示：

class DynamicViewpointSampleEntry extends MetaDataSampleEntry‘'dyv’') {  
 ViewpointPosStruct();  
 unsigned int(16) viewpoint\_id;  
 string viewpoint\_label;  
 unsigned int(8) viewpoint\_type;  
 unsigned int(1) dynamic\_gcs\_rotation\_flag;  
 bit(7) reserved = 0;  
 if(dynamic\_gcs\_rotation\_flag == 0) {  
 RotationStruct vpGlobalCoordinateSys();  
 }  
}

viewpoint\_id 指示与引用该样本入口的所有样本关联的视点标识；

viewpoint\_lable指示视点的标签，提供视点的文本描述，以空字符结尾的UTF-8字符串；

viewpoint\_type指示动态视点的类型,如8.2.1.5.3章节的表12所示：

dynamic\_gcs\_rotation\_flag 取值为0时，表示视点的全局坐标系的X，Y，Z轴相对于公共参考坐标系的偏转，俯仰和翻滚的旋转角度在引用该样本入口的所有样本中保持不变；取值为1时，表示视点的全局坐标系的X，Y，Z轴相对于公共参考坐标系的偏转，俯仰和翻滚旋转角度在引用该样本入口的样本中定义。

8.3.2.1.3样本格式

每个样本对应一定时间内的切换后的视点的信息，引用该样本入口的样本的语法如下所示：

aligned(8) DynamicViewpointSample() {  
 ViewpointInfoStruct(dynamic\_gcs\_rotation\_flag);  
}

第一个样本的视点信息结构（ViewpointInfoStruct）中group\_alignment\_flag应取值1。

如果后续样本的视点群组信息未改变，样本中可以不包含ViewpointGroupStruct()，推断它与前一个样本的ViewpointGroupStruct()相同。

8.3.2.2初始视点定时元数据

8.3.2.2.1概述

初始视点定时元数据指示选择首次播放的视点时或视点群组切换后应使用的初始视点。在缺省情况下，所有视点中具有最小视点标识值的视点作为播放的初始视点。

注：切换视点群组时，如果存在初始视点，播放器按照初始视点动态元数据中初始视点对应的全景视频轨道来播放。否则，播放器按照视点群组中视点标识值最小的视点对应的全景视频轨道来播放。

8.3.2.2.2样本入口

初始视点定时元数据轨道的样本入口类型为’invp’，样本入口语法如下所示：

class InitialViewpointSampleEntry extends MetaDataSampleEntry(‘invp’) {  
 unsigned int(16) initial\_viewpoint\_id;  
}

initial\_viewpoint\_id 指示引用该样本入口的第一个样本的初始视点的视点标识值。

8.3.2.2.3 样本格式

样本入口（’invp’）的样本的语法如下所示：

aligned(8) InitialViewpointSample() {  
 unsigned int(16) initial\_viewpoint\_id;  
}

initial\_viewpoint\_id 指示该定时元数据轨道样本的初始视点的视点标识值。

8.3.3 视窗信息定时元数据

8.3.3.1 概述

视窗信息定时元数据用于指示容积视频中的视窗信息，以及与视窗相关的相机内参和外参信息，其中相机外参信息包含了视窗的位置和旋转信息。

视窗信息定时元数据仅关联参考至相应的容积视频图集轨道，不直接关联参考容积视频组件轨道。视窗信息定时元数据轨道使用’cdtg’轨道参考关联或参考到相应的轨道或轨道组时，定时元数据轨道描述被关联或参考的所有的轨道。当视窗信息定时元数据轨道使用’cdsc’轨道参考关联或参考至一个或多个图集轨道时，定时元数据轨道分别描述每个图集轨道。

视窗信息定时元数据轨道中的每个样本均可被标记为同步样本。若视窗信息定时元数据轨道中某个样本关联或参考的容积视频图集轨道中与之相同解码时间的样本中存在至少一个同步样本，则该定时元数据样本需被标记为同步样本。

8.3.3.2 样本入口

8.3.3.2.1 定义

数据盒类型（Box Type）: ' 6vpt'  
数据盒容器（Container）: Sample Description Box (‘stsd’)  
强制性（Mandatory）: 否  
数量（Quantity）: 0个或1个

容积视频的视窗信息样本入口为ViewportInfoSampleEntry，类型为’6vpt’。

视窗信息样本入口中需包含ViewportInfoConfigurationBox数据盒，该数据盒中定义了视窗类型和可能的相机内参或相机外参信息，该数据盒中的元数据对轨道中的所有样本生效。

8.3.3.2.1 语法

aligned(8) class ViewportInfoConfigurationBox extends FullBox(‘6vpc’, version=0, 0) {

unsigned int(7) viewport\_type;

string viewport\_description;

unsigned int(1) camera\_info\_flag;

if (camera\_info\_flag == 0)

unsigned int(8) camera\_id;

else {

unsigned int(1) dynamic\_int\_camera\_flag;

unsigned int(1) dynamic\_ext\_camera\_flag;

bit(6) reserved = 0;

if (dynamic\_int\_camera\_flag == 0) {

IntCameraInfoStruct();

}

if (dynamic\_ext\_camera\_flag == 0) {

ExtCameraInfoStruct();

}

}

}

aligned(8) class ViewportInfoSampleEntry() extends MetadataSampleEntry (‘6vpt’) {

ViewportInfoConfigurationBox();

}

8.3.3.2.3 语义

viewport\_type 指示当前样本入口对应的所有样本的视窗类型，其取值含义如下表所示。

表 19 视窗类型

|  |  |
| --- | --- |
| **取值** | **描述** |
| 0 | 根据导演或者内容提供者意图生成的推荐视窗 |
| 1 | 根据用户观看数据统计的推荐视窗 |
| 2 | 根据其余用户选择的视窗生成的推荐视窗 |
| 3 | 推荐的初始视窗 |
| 4 | 与特定空间区域关联的推荐视窗 |
| 5..239 | 保留 |
| 240..255 | 自定义 |

viewport\_description 以空字符结尾的字符串，提供推荐视窗的文本描述信息。

camera\_info\_flag，指示样本入口中是否包含相机的参数信息，取值为0，指示相机的标识信息，通常情况下，该相机参数信息已在其他地方描述；取值为1，指示相机的内外参信息。

camera\_id，指示相机的标识信息。

dynamic\_int\_camera\_flag 取值为0时表示当前样本入口对应的所有样本的相机内参固定不变。如果 dynamic\_ext\_camera\_flag取值为0时，则 dynamic\_int\_camera\_flag 取值也必须为0。取值为1时表示当前样本入口对应的所有样本的相机内参动态变化。

dynamic\_ext\_camera\_flag 取值为0时表示当前样本入口对应的所有样本的相机外参固定不变。 取值为1时表示当前样本入口对应的所有样本的相机外参动态变化。当dynamic\_ext\_camera\_flag为1时，则所有样本中dynamic\_ext\_camera\_flag至少有一个为1。

当viewport\_type字段取值为3时，视窗定时元数据指示推荐的初始视窗信息，包括初始视窗位置和旋转信息。当容积视频媒体需要以特定初始视窗（非默认初始视窗）开始播放时，视窗定时元数据轨道应索引至该容积视频媒体轨道。当该类型的元数据缺失时，则初始视窗的cam\_pos\_x, cam\_pos\_y, cam\_pos\_z, cam\_quat\_x, cam\_quat\_y,和cam\_quat\_z缺省值均为0。

8.3.3.3 样本格式

8.3.3.3.1 概述

每个视窗样本携带一个或多个在样本入口中指定视窗类型的视窗。每个视窗包含相机外参信息和相机内参信息，由IntCameraInfoStruct和ExtCameraInfoStruct定义。相机外参信息应存在于每个样本中，但相机内参信息仅存在于相机内参发生变化的样本中。前一个样本中定义的某个视窗的相机外参和相机内参信息将持续至该视窗的对应信息发生变化时。

8.3.3.3.1 语法

aligned(8) class ViewportInfoSample() {

unsigned int(8) num\_viewports;

for (i=1; i <= num\_viewports; i++){

unsigned int(7) viewport\_id[i];

unsigned int(1) viewport\_cancel\_flag[i];

if (viewport\_cancel\_flag[i] == 0) {

unsigned int(1) camera\_extrinsic\_flag[i];

unsigned int(1) camera\_intrinsic\_flag[i];

unsigned int(1) view\_info\_flag[i];

unsigned int(1) freeview\_camera\_info\_flag[i];

bit(4) reserved = 0;

ViewportInfoStruct(camera\_extrinsic\_flag[i], camera\_intrinsic\_flag[i]);

if(view\_info\_flag[i] == 1){

unsigned int(8) num\_views;

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

unsigned int(16) view\_id;

}

}

if(freeview\_camera\_info\_flag[i] == 1){

unsigned int(8) num\_freeview\_cameras;

for (j=0; j < num\_freeview\_cameras; j++) {

unsigned int(16) freeview\_camera\_id;

}

}

}

}

}

8.3.3.3.3 语义

如果视窗信息元数据轨道存在，则相机外参信息ExtCameraInfoStruct()应当出现于样本入口中或者样本中。以下情况不得出现：dynamic\_ext\_camera\_flag取值为0且所有样本中的camera\_extrinsic\_flag[i]取值均为0。

num\_viewports指示样本中指示的视窗数目。

viewport\_id[i]指示对应视窗的标识符。

viewport\_cancel\_flag[i]取值为1时表示视窗标识符取值为viewport\_id[i]的视窗被取消了。取值为0时表示视窗标识符取值为viewport\_id[i]的视窗相关信息在样本中指示。

camera\_intrinsic\_flag[i]取值为1时表示当前样本中第i个视窗存在相机内参。如果dynamic\_int\_camera\_flag取值为0，则该字段必须取值为0。同时，当camera\_extrinsic\_flag[i]取值为0时，该字段必须取值为0。取值为0时表示当前样本中第i个视窗不指示相机内参。如果媒体文件中存在预先合成的虚拟相机视频图像，该字段必须取值为0。

camera\_extrinsic\_flag[i] 取值为1时表示当前样本中的第i个视窗存在相机外参。如果dynamic\_ext\_camera\_flag取值为0，则该字段必须取值为0。取值为0时表示当前样本中第i个视窗不指示相机外参。如果媒体文件中存在预先合成的虚拟相机视频图像，该字段必须取值为0。

view\_info\_flag[i] 取值为1时表示当前样本中的第i个视窗关联了相应的视图信息；取值为0时表示当前样本中第i个视窗没有关联相应的视图信息。

freeview\_camera\_info\_flag[i]取值为1表示当前样本中的第i个视窗关联了相应的自由视角相机信息。取值为0表示当前样本中的第i个视窗没有关联相应的自由视角相机信息。

num\_views 指示当前视窗关联的视图数量。

view\_id指示视图的标识符。

num\_freeview\_cameras指示视窗关联的自由视角相机的数目。如果媒体文件中存在预先合成的虚拟相机视频图像，该字段必须取值为1。

freeview\_camera\_id指示每个自由视角相机的标识符。

8.3.4 容积视频的视图定时元数据

8.3.4.1 容积视频动态视图定时元数据

8.3.4.1.1 概述

如果容积视频图集轨道具有相关联的定时元数据轨道，其样本入口类型为‘dyvw’，则容积视频图集轨道中承载的媒体数据对应的视图是动态的，即视图信息会随时间动态变化。

视图定时元数据轨道应包含一个“cdsc”轨道参考，关联或参考到携带图集相关参数集的容积视频图集轨道。

8.3.4.1.2 样本入口

aligned(8) class DynamicViewSampleEntry extends MetaDataSampleEntry('dyvw') {  
 V3CViewsBox();  
}

8.3.4.1.3 样本格式

aligned(8) DynamicViewSample() {  
 unsigned int(16) num\_views;  
 for (i=0; i < num\_views; i++){  
 ViewInfoStruct(camera\_parameters\_included\_flag);  
 }  
}

num\_views指示样本中视图的数量。该值等于或小于视图的总数，仅更新视图信息变化 这可能不一定等于可用视图的总数。 示例中仅存在其视图信息正在更新的视图。

ViewInfoStruct()在章节8.1.2.4中定义。其中，如果camera\_parameters\_included\_flag取值为0表示视图对应的相机参数使用具有相同view\_id的ViewInfoStruct中定义的参数，或使用上一个样本中的相机参数。

8.3.4.2 容积视频视图组定时元数据

8.3.4.2.1 概述

如果容积视频图集轨道具有相关联的定时元数据轨道，其样本入口类型为‘dyvg’，则容积视频图集轨道组中承载的媒体数据所对应的视图组是动态的，即视图组信息会随时间动态变化。

视图组定时元数据轨道应包含一个“cdsc”轨道参考，关联或参考到承载视图组对应图集的相关参数的容积视频图集轨道。

8.3.4.2.2 样本入口

aligned(8) class DynamicViewGroupSampleEntry extends MetaDataSampleEntry('dyvg') {  
 V3CViewGroupsBox();  
}

8.3.4.2.3 样本格式

aligned(8) DynamicViewGroupSample() {  
 unsigned int(16) num\_view\_groups;  
 for (i=0; i < num\_view\_groups; i++) {  
 ViewGroupInfoStruct(camera\_parameters\_included\_flag);  
 }  
}

num\_view\_groups指示样本中视图组的数量，该字段取值小于或等于视图组的总数，仅更新视点变化的视图组。

ViewGroupInfoStruct() 在章节8.1.2.5中定义，其中，如果camera\_parameters\_included\_flag为0，视图组中的相机参数为具有相同view\_group\_id的ViewGroupInfoStruct中定义的参数，或使用上一个样本中的相机参数。

8.3.5 容积视频动态空间区域定时元数据

8.3.5.1 概述

如果容积视频图集轨道具有关联的定时元数据轨道，且定时元数据轨道的样本入口类型为‘dyvm’，则将容积视频媒体轨道承载的容积媒体流对应的三维空间区域视为动态区域（即，空间区域信息会随着时间动态变化），并且V3CSpatialRegionsBox应存在于容积视频动态空间区域定时元数据轨道的样本入口中，而不存在于容积视频图集轨道的样本入口中。

容积视频动态空间区域定时元数据轨道应该包含一个引用类型为“cdsc’轨道引用，关联或识别对应的携带图集的容积视频图集轨道或图集分片的容积视频图集分片轨道。

8.3.5.2 样本入口

aligned(8) class DynamicVolumetricMetadataSampleEntry extends MetaDataSampleEntry('dyvm') {  
 V3CSpatialRegionsBox();

}

8.3.5.3 样本格式

8.3.5.3.1 语法

aligned(8) class V3cVolumetricMetadataSample() {  
 bit(1) region\_updates\_flag;

bit(1) all\_tiles\_in\_single\_track\_flag;  
 bit(1) packed\_video\_track\_flag;  
 bit(5) reserved = 0;  
 if (region\_updates\_flag) {  
 unsigned int(16) num\_regions;  
 for (i=0; i < num\_regions; i++) {  
 3DSpatialRegionStruct(dimensions\_included\_flag);  
 bit(1) update\_mapping\_flag;  
 bit(7) reserved = 0;  
 if (update\_mapping\_flag) {

if (all\_tiles\_in\_single\_track\_flag) {

if(packed\_video\_track\_flag){

unsigned int(8) num\_packed\_video\_track;

for(n=0; n< num\_packed\_video\_track; n++){

unsigned int(32) packed\_video\_track\_id;

unsigned int(8) num\_tiles;

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

unsigned int(16) tile\_id;

}

}

}else {

unsigned int(8) num\_ component\_track\_groups;

for (j=0; j < num\_component\_track\_groups; j++) {  
 unsigned int(32) component\_track\_group\_id;   
 unsigned int(8) num\_tiles;  
 for (k=0; k < num\_tiles; k++) {  
 unsigned int(16) tile\_id;  
 }  
 }  
 }   
 }

} else {  
 unsigned int(8) num\_tile\_tracks;  
 for (int j=0; j < num\_tile\_tracks; j++) {  
 unsigned int(32) tile\_track\_id;  
 }  
 }  
 }  
}

}

}

8.3.5.3.2 语义

region\_updates\_flag 指示样本中包括对三维空间区域是否更新。

num\_regions 指示样本中定义的三维空间区域的数量。 其数量可以不等于可用区域的总数，并且只有位置和尺寸正在更新的空间区域才会出现在样本中。

packed\_video\_track\_flag 指示容积视频图集轨道是否关联封装视频组件轨道。值为1时，存在与容积视频图集轨道相关的封装视频组件轨道，值为0时，不存在与容积视频图集轨道相关的封装视频组件轨道。

update\_mapping\_flag 指示是否更新空间区域和地图的图集图块之间的映射。 值1表示存在更新的映射。 值0表示没有更新。

num\_component\_track\_groups 指示与三维空间区域关联的图集片组件轨道组的数量。

num\_packed\_video\_track指示与三维空间区域相关联的封装视频组件轨道数量。

packed\_video\_track\_id 指示与三维空间区域相关联的封装视频组件轨道标识。

num\_tile指示与三维空间区域相关关联的图集分片的数量。

component\_track\_group\_id指示 图集片组件轨道组标识，携带与三维空间区域相关联的容积视频组件，用于关联三维空间区域与图集片组件。

num\_tiles 指示与图集片组件轨道组关联的图集片的数量。

tile\_id 指示在关联的图集片组件轨道组中标识对应的包图集片信息的NAL单元。 tile\_id的值应等于承载图集信息的容积视频图集轨道中NALUMapEntry中的groupID。

num\_tile\_tracks 指示与三维空间区域关联的容积视频图集分片轨道的数量。

tile\_track\_id 指示与三维空间区域相关联的容积视频图集分片轨道的标识。

8.3.6观看空间及处理定时元数据

8.3.6.1概述

本章定义了沉浸媒体观看空间及其对应处理方式的定时元数据。通过样本入口类型‘vshd’指示三维观看空间及处理定时元数据轨道。

定时元数据轨道应包含一个’cdsc’轨道引用，参考或关联到对应携带图集数据的容积视频轨道。

8.3.6.2样本入口

aligned(8) class ViewingSpaceAndHandlingSampleEntry extends MetaDataSampleEntry('vshd') {  
 signed int(8) num\_viewing\_space;  
 for(int i=0;i<=num\_viewing\_space;i++) {  
 unsigned int(8) static\_viewing\_space\_flag[i];  
 unsigned int(8) space\_handling\_flag[i];  
 unsigned int(8) handling\_present[i];  
 unsigned int(16) viewing\_space\_id[i];   
 if(static\_viewing\_space\_flag[i])  
 ViewingSpaceStruct();  
 if(space\_handling\_flag[i]==0 && handling\_present[i] ==1 )

unsigned int(8) num\_handling\_options;  
 for (j=0; j<= num\_handling\_options; j++) {  
 unsigned int(8) handling\_device\_class;  
 unsigned int(8) handling\_application\_class;  
 unsigned int(8) handling\_method;  
 }  
 }  
 }

num\_viewing\_space指示三维观看空间数量。

static\_viewing\_space\_flag取值为1时表示样本入口中所有的样本的三维观看空间保持不变；取值为0时表示该样本入口中各样本的三维观看空间在样本格式中指示。

space\_handling\_flag取值为0时表示样本入口中所有的样本的三维观看空间的处理方式保持不变；取值为1时表示该样本入口中各样本的三维观看空间的处理方式在样本格式中指示。

handling\_present取值为0时表示该三维空间的处理方式为缺省；取值为1时表示该三维空间的处理方式在样本入口及各样本中指示。

viewing\_space\_id指示三维观看空间的标识。

8.3.6.3样本

aligned(8) class ViewingSpaceSample() {  
 for(int j=0;j<=num\_viewing\_space;j++) {   
 unsigned int(16) viewing\_space\_id;  
 if(static\_viewing\_space\_flag[j] == 0)  
 ViewingSpaceStruct();

if(space\_handling\_flag[j]==1 && handling\_present[j] ==1 )){

unsigned int(8) num\_handling\_options;  
 for (j=0; j<= num\_handling\_options; j++) {  
 unsigned int(8) handling\_device\_class;  
 unsigned int(8) handling\_application\_class;  
 unsigned int(8) handling\_method;  
 }  
}

}

}

8.4非时序沉浸式媒体封装

8.4.1 全景图片

8.4.1.1概述

根据ISO/IEC 23008-12的规范，全景图片在文件中应以图片数据项的形式存储。全景图片数据项中应包含ProjectionFormatProperty属性。若全景图片数据项包含双目立体内容，则数据项中应包含FramePackingProperty属性。若全景图片数据项采用了区域封装，则数据项中应包含RegionWisePackingProperty属性。

8.4.1.2帧排布数据项属性

8.4.1.2.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'stvi' |
| 属性类型: | Descriptive item property |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个或1个 |

全景图片数据项包含FramePackingProperty时， 表明图片中包含两个空间上联合排布的图像帧。

'stvi'数据项属性中的essential 字段应取值为1。

8.4.1.2.2语法

FramePackingProperty与ISO/IEC 14496-12中定义的 StereoVideoBox 数据盒语法一致。

8.4.1.2.3语义

FramePackingProperty 中各语法元素的语义与ISO/IEC 14496-12中定义的 StereoVideoBox 数据盒的各元素语义一致。

8.4.1.3投影格式数据项属性

8.4.1.3.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'prfr' |
| 属性类型: | Descriptive item property |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个或1个 |

ProjectionFormatProperty指示全景图片的投影格式。

'prfr'数据项属性中的essential 字段应取值为1。

ProjectionFormatProperty的使用规范与8.2.1.1中定义的全景视频投影数据盒相同。

8.4.1.3.2语法

aligned(8) class ProjectionFormatProperty extends ItemFullProperty('prfr', 0, 0) {  
 ProjectionFormatStruct(); //在8.1.1.1 投影格式结构中定义  
}

8.4.1.4区域封装数据项属性

8.4.1.4.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'rwpk' |
| 属性类型: | Descriptive item property |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个或1个 |

RegionWisePackingProperty指示封装区域和对应的投影区域之间的映射关系，以及可能的保护带位置和尺寸信息。

'rwpk'数据项属性中的essential 字段应取值为1。

8.4.1.4.2语法

aligned(8) class RegionWisePackingProperty extends ItemFullProperty('rwpk', 0, 0) {  
 RegionWisePackingStruct(); //在8.1.1.5 区域封装结构中定义  
}

8.4.1.4.3语义

区域封装数据项属性的语义与8.1.1.5中的语义相同，并遵循以下约束。

packed\_picture\_width取值应为image\_width的整数倍；packed\_picture\_height 取值应为image\_height 的整数；其中image\_width 和 image\_height 为图像属性 ImageSpatialExtentsProperty 中的语法元素。

8.4.1.5旋转信息数据项属性

8.4.1.5.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'rotn' |
| 属性类型: | Descriptive item property |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个或1个 |

当全景图片是基于本地坐标系时，图片数据项中应存在RotationProperty属性。

RotationProperty指示本地坐标系到全局坐标系之间的旋转信息。若图片数据项为双目图片，则该属性对每个单目图片分别生效。若图片数据项中不包含该属性，则3D\_rotation\_type ，rotation\_yaw, rotation\_pitch, rotation\_roll缺省值均为0。

'rotn'数据项属性中的essential 字段应取值为1。

8.4.1.5.2语法

aligned(8) class RotationProperty extends ItemFullProperty('rotn', 0, 0) {  
 RotationStruct();//在8.1.1.2 旋转结构中定义  
}

8.4.1.6视点信息数据项属性

8.4.1.6.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'vpip' |
| 属性类型: | Descriptive item property |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个或1个 |

视点信息属性用于描述全景图片的视点信息。

'vpip'数据项属性中的essential 字段应取值为1。'vpip'数据项属性的视点信息结构中group\_alignment\_flag字段应取值为0。

8.4.1.6.2语法

aligned(8) class ViewpointInfoProperty extends ItemFullProperty('vpip', 0, 0) {

ViewpointInfoStruct(); //在8.1.1.6 视点信息结构中定义

}

8.4.1.7全景图片初始观看方向数据项属性

8.4.1.7.1定义

数据盒类型: 'ivop'

属性类型: 描述性属性

数据盒容器: ItemPropertyContainerBox

Mandatory (per item): 否

Quantity (per item): 0个或1个

全景图片初始观看方向数据项属性用于描述全景图片的初始观看区域，当缺少这一信息时，按照默认初始观看方向播放，其中centre\_Azimuth、centre\_Elevation和centre\_tilt，range\_included\_flag分别为0。

8.4.1.7.2语法

aligned(8) class InitialViewingOrientationProperty extends ItemFullProperty('ivop', 0, 0) {

SphereRegionStruct();

}

8.4.1.8全景视点关联数据盒

8.4.1.8.1定义

数据盒类型（Box Type）: ‘vpia’

数据盒容器（Container）: GroupsListBox

强制性（Mandatory）: 否

数量（Quantity）: 0个，1个或多个

当静态全景内容由多个不同视点的全景图片组成时，使用‘vpia’类型的EntityToGroupBox关联所有视点对应的全景图片。

8.4.1.8.2语法

aligned(8) class ViewpointItemAssociationBox extends EntityToGroupBox('vpia') {

for(i=0; i<num\_entities\_in\_group; i++){

unsigned int(16) viewpoint\_id;

unsigned int(1) initial\_viewpoint\_flag;

unsigned int(1) transition\_orientation\_type;

unsigned int(1) neighbor\_viewpoint\_flag;

bit(5) reserved = 0;

if(neighbor\_viewpoint\_flag == 1){

unsigned int(8) num\_neighbor\_vp;

for(j=0; j< num\_neighbor\_vp; j++){

unsigned int(16) neighbor\_vp\_id；

}

}

string viewpoint\_label;

}

}

8.4.1.8.3语义

viewpoint\_id指示每个视点的标识符。

initial\_viewpoint\_flag取值为1时表示相应视点为初始视点；取值为0时表示相应视点不是初始视点。

transition\_orientation\_type取值为0时表示该视点切换至下一视点时，下一视点的初始视窗区域与当前视点的用户观看视窗相同；取值为1时表示该视点切换至下一视点时，下一视点的初始视窗区域由下一视点的初始观看方向数据项属性决定。该字段默认取值为0。

neighbor\_viewpoint\_flag取值为1时表示当前视点仅能切换至相邻的视点；取值为0时表示当前视点可以切换至当前实体组中的任一视点。该字段默认取值为0。

num\_neighbor\_vp指示当前视点可切换的目标邻近视点的数目。

neighbor\_vp\_id指示目标邻近视点的视点标识符。

viewpoint\_label以空字符结尾的字符串，指示视点的标签信息。

8.4.2 非时序容积媒体

8.4.2.1概述

根据ISO/IEC 23008-12的规范，非时序容积媒体在文件中应以数据项的形式存储。非时序容积媒体的数据项类型包括容积媒体图集数据项、容积媒体图集分片数据项以及容积媒体组件数据项。包含上述容积媒体数据项的文件中，其MetaBox中的HandlerBox数据盒的处理类型字段应为'volv'。

8.4.2.2容积媒体图集数据项

容积媒体图集数据项用于表示一个可独立编解码的图集访问单元，图集数据项的数据项类型为'v3c1', 'v3cb'或 'v3a1'。图集数据项中存储一个或多个ACL或non-ACL NAL单元。数据项类型为'v3c1'或'v3cb'的图集数据项应关联一个V3CConfigurationProperty属性。所有图集数据项均应关联一个V3CUnitHeaderProperty属性。

若非时序容积媒体内容仅包含一个图集，则该图集数据项类型应为'v3c1'。若'v3c1'类型的数据项与多个图集分片数据项相关联，则'v3c1'数据项中存储一个或多个与图集分片数据项关联的non-ACL NAL单元。否则，该'v3c1'类型的数据项存储一个或多个ACL或non-ACL NAL单元。

若非时序容积媒体内容包含多个图集，每个图集码流需要以不同的图集数据项存储。其中一个图集数据项的类型应为'v3cb'，且其余数据项的类型应为'v3a1'。'v3cb'数据项与'v3a1'数据项之间使用'v3cs'类型的数据项索引关联。

'v3cb'数据项不应包含任何ACL NAL单元。若'v3a1'类型的数据项与多个图集分片数据项相关联，则'v3a1'数据项中存储一个或多个与V3CUnitHeaderProperty属性中vuh\_atlas\_id字段所指示的图集关联的non-ACL NAL单元。

若PrimaryItemBox数据盒存在，则该数据盒中的item\_ID字段取值应对应类型为'v3c1' 或 'v3cb'的图集数据项。

8.4.2.3容积媒体图集分片数据项

容积媒体图集分片数据项的数据项类型为'v3t1'，其包含一个或多个属于同一个图集的ACL NAL单元。

每个图集分片数据项均应关联一个V3CAtlasTileConfigurationProperty数据项属性。该数据项属性应指示图集分片数据项中的图集分片的标识符。图集数据项与图集分片数据项之间应使用'v3ct'类型的数据项索引从图集数据项关联至图集分片数据项。

8.4.2.4容积媒体组件数据项

容积媒体组件数据项表示容积媒体的组件数据。组件数据项中存储V3C\_OVD, V3C\_GVD和 V3C\_AVD类型的V3C单元的单元负载。一个组件数据项中应仅存储一个相应视频组件数据的访问单元。

组件数据项的类型应取决于相应视频组件的编码类型。一个组件数据项应关联对应的V3C单元头数据项属性以及编码类型相关的配置数据项属性。组件数据项应被标记为隐藏数据项。

为了表示图集数据项与组件数据项之间或图集分片数据项与组件数据项之间的关系，本标准定义了三种类型分别为'v3vo', 'v3vg'和'v3va'的数据项索引。该数据项索引从图集数据项或图集分片数据项索引至相应的组件数据项。三种数据项索引具体对应的组件数据类型如下：

'v3vo': 被索引的组件数据为占位图数据。

'v3vg': 被索引的组件数据为几何数据。

'v3va': 被索引的组件数据为属性数据。

8.4.2.5容积媒体相关数据项属性

8.4.2.5.1概述

容积媒体的数据项属性用于指示容积媒体数据项的参数集信息、V3C单元头信息以及其他元数据信息。

8.4.2.5.2容积媒体配置数据项属性

8.4.2.5.2.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'v3cp' |
| 属性类型: | 数据项描述属性 |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 是，对于'v3c1' 或 'v3cb'数据项 |
| 数量(每数据项): | 0个，1个或多个 |

在非时序容积媒体封装中，V3C参数集以数据项描述属性的形式在数据项中指示，且该数据项属性需关联至'v3c1'或'v3cb'类型的图集数据项中。在当前版本的标准中，容积媒体配置数据项属性中仅存储一个V3C参数集。

容积媒体数据项配置属性为必选属性，ItemProperyAssociationBox中对应的essential字段需设置为1。

8.4.2.5.2.2语法

aligned(8) class V3CConfigurationProperty

extends ItemProperty('v3cp', version=0, flags) {

unsigned int(16) v3c\_parameter\_set\_length;

// v3c\_parameter\_set()由ISO/IEC FDIS 23090-5标准定义

v3c\_parameter\_set v3c\_parameter\_set(v3c\_parameter\_set\_length);

}

8.4.2.5.2.3语义

v3c\_parameter\_set\_length指示v3c\_parameter\_set()的字节长度。

v3c\_parameter\_set包含ISO/IEC 23090-5中定义的参数集。

8.4.2.5.3 V3C单元头数据项属性

8.4.2.5.3.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'vutp' |
| 属性类型: | 数据项描述属性 |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 是，对于'v3c1' , 'v3a1','v3cb'类型的图集数据项或V3C组件数据项 |
| 数量(每数据项): | 1个 |

在非时序容积媒体封装中，V3C单元头以数据项描述属性的形式在数据项中指示，且该数据项属性需关联至图集数据项和V3C组件数据项。

V3C单元头数据项属性为必选属性，ItemProperyAssociationBox中对应的essential字段需设置为1。

8.4.2.5.3.2语法

aligned(8) class V3CUnitHeaderProperty() extends ItemFullProperty('vutp', version=0, 0) {

v3c\_unit\_header header(); // 4 字节

}

8.4.2.5.3.3语义

header 包含ISO/IEC 23090-5中定义的V3C单元头。

8.4.2.5.4图集分片配置数据项属性

8.4.2.5.4.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'v3tp' |
| 属性类型: | 数据项描述属性 |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 是，对于'v3t1'类型的图集分片数据项 |

在非时序容积媒体封装中，图集分片配置信息通过数据项描述属性的形式在数据项中指示，且该数据项属性需关联至图集分片数据项。

图集分片配置数据项属性为必选属性，ItemProperyAssociationBox中对应的essential字段需设置为1。

8.4.2.5.4.2语法

aligned(8) class V3CAtlasTileConfigurationProperty () extends ItemFullProperty(‘v3tp’, version=0, 0) {

unsigned int(16) num\_tiles;

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

unsigned int(16) tile\_id;

}

}

8.4.2.5.4.3语义

num\_tiles 指示对应图集分片数据项中包含的分片数量。

tile\_id 指示对应图集分片数据项中包含的图集分片的标识符。其取值应与ISO/IEC 23090-5中的相应的afti\_tile\_id字段相同。

8.4.2.5.5 容积媒体空间信息属性

8.4.2.5.5.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'v3sp' |
| 属性类型: | 数据项描述属性 |
| 数据盒容器: | ItemPropertyContainerBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个或1个 |

容积媒体空间信息属性用于描述非时序容积媒体的三维空间信息。该属性可关联至图集分片数据项。

8.4.2.5.5.2语法

aligned(8) class V3CSpatialInfoProperty extends ItemFullProperty('v3sp', 0, 0) {

3DSpatialRegionStruct (); //在8.1.2.3中定义

}

8.4.2.6非时序容积媒体可替换组

8.4.2.6.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'valy' |
| 数据盒容器: | GroupsListBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个，1个或多个 |

容积媒体组件数据项可以存在可替换关系，V3CAlternativeEntityToGroupBox用于指示可替换的组件数据项间的质量差异信息。相互可替换的组件数据项中，仅有一个能被相应的图集数据项或图集分片数据项索引。

8.4.2.6.2语法

aligned(8) class V3CAlternativeEntityToGroupBox extends EntityToGroupBox('valy') {

V3CAlternativeInfoStruct();//在8.1.2.6中定义

}

8.4.2.7非时序容积媒体播放组

8.4.2.7.1定义

|  |  |
| --- | --- |
| 数据盒类型: | 'eply' |
| 数据盒容器: | GroupsListBox |
| 强制性(每数据项): | 否 |
| 数量(每数据项): | 0个，1个或多个 |

当容积媒体组件数据项的多个可替换版本之间，仅有某些特定组合应被联合播放时，应使用容积媒体播放实体组'eply'来指示这些联合播放的组合。对于一个播放实体组的所有组件数据项，可以有选择地定义其联合质量等级，用以指示不同质量的媒体内容。

8.4.2.7.2语法

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

unsigned int(1) quality\_ranking\_flag;

bit(7) reserved = 0;

if(quality\_ranking\_flag == 1){

unsigned(8) int quality\_ranking;

}

}

8.4.2.7.3语义

quality\_ranking\_flag取值为1时表示容积视频播放轨道组的所有组件轨道具有联合质量等级；取值为0时表示容积视频播放轨道组的所有组件轨道不具有联合质量等级。

quality\_ranking 指示一个容积视频播放轨道组内所有组件轨道的联合质量等级，该字段取值越小，表明联合质量等级越高。

沉浸媒体DASH传输信令

9.1 沉浸媒体DASH传输架构

图 17示例了沉浸媒体系统中的DASH传输功能的内容流。



图 17 沉浸媒体DASH传输的内容流

应支持如下接口：

Fs/F's：初始化及媒体片段。

G：DASH媒体表述描述，包括全景视频特定的元数据，如投影信息、区域封装信息、推荐视窗信息等，容积视频的预选择及其特定的元数据，如容积视频的预选择信息、组件信息、空间区域信息。

H/H’：发送/接收媒体呈现描述文件或片段。

9.1.1基于全景视频的DASH传输

基于片段（Fs）和表述相同内容的其他媒体文件创建一个媒体呈现描述（MPD）。DASH MPD创建器包括全景媒体特定的描述子（descriptor）。描述子包括投影类型、区域封装类型、内容覆盖、推荐视窗、视点、球形区域质量分类、二维平面区域质量分类等信息。这些信息可根据片段（segments）中的相同信息产生。

DASH客户端获取媒体呈现描述（MPD）中相关描述子，包括：投影类型描述子、内容覆盖描述子、推荐视窗描述子、视点描述子等，解析其中携带的投影类型、内容覆盖、推荐视窗、视点描述等信息。投影类型用于确定播放全景视频的投影格式。内容覆盖信息用于确定全景视频覆盖的球面区域。推荐视窗信息用于确定播放全景视频的推荐视窗。其中，推荐视窗在全景视频中的球面区域不同于当前全景视频播放视窗的球面区域。视点描述用于确定全景视频的视点。DASH客户端根据终端能力、网络带宽、用户偏好等选择播放全景视频的投影格式、球面区域、推荐视窗、视点。DASH客户端向服务器请求获取对应于选择播放全景视频的投影格式、球面区域、推荐视窗、视点的一个或多个全景视频片段。

服务器可根据客户端发送的获取请求提供一个或多个片段（Fs）给客户端，也可提供MPD（作为接口H的一部分），或者MPD通过其他方式提供给客户端。片段和MPD通过网络传递给客户端，接收来自服务器的片段和MPD的接口被标记为H'。服务器（H）的输出被看作为等同于DASH MPD和片段接收块（H'）的输入。接收的片段通过DASH MPD和片段接收模块输出到文件/片段解封装模块。在一些媒体配置文件中，片段接收功能可能包括一个遵循片段序列的重构，这被认为是在F's接口。

9.1.2基于容积视频的DASH传输

9.1.2.1 概述

基于容积视频的二维表示的组件数据、图集数据、参数集数据的媒体文件创建MPD。DASH MPD创建器使用预选择描述容积视频参数集数据、图集数据和组件数据并生成特定的描述子，描述子包含容积视频的图集及其组件数据、空间区域等描述信息。DASH客户端获得由服务器或其他方式传输/提供的MPD，从MPD中获取并使用其包含的容积视频预选择元素（Preselection）和相关特定描述子信息（如容积视频描述子、空间区域描述子），选择相应的自适应集（Adaptation Set）和表示（Representation）访问服务器端的容积视频的一个或多个图集数据和/或一个或多个图集分片数据，以及相应的视频组件数据，获取相应的一个或多个片段，并根据片段中的图集数据和视频组件数据重构容积视频。

根据容积视频的内容在媒体文件中的封装方式，在DASH中支持单轨模式或多轨模式的流传输，其中，单轨模式是对以单轨方式封装的容积视频内容的媒体文件的流传输，多轨模式是对以多轨方式封装的容积视频内容的媒体文件的流传输。

9.1.2.2 单轨模式

单轨模式应由唯一的Adaptation Set和一个或多个Representation来描述。Adaptation Set中的所有 Representation应使用相同的编码器来编码相应的视频组件，但是不应要求相同的编码器编码每一个视频组件（比如占位图组件和几何组件可采用不同的编码器）。

如果一个Representation包含2个以上的媒体片段，则应该存在一个初始化的媒体片段。初始化的片段应该包含一个含有v3c\_parameter\_set语法结构的 V3CDecoderConfigurationRecord，以及一个组件编码器映射的SEI消息。

一个媒体片段的第一个样本应该包含一个媒体流接入点（SAP）。这意味着第一个样本的每一个子样本应该有一个SAP。

基于容积视频单轨模式DASH传输的部分属性的使用要求如下：

@mimeType参数应该是‘video/mp4’。

@codecs参数应该出现在adaptation set层级，并且标识出最大的能力以要求可解码adaptation set中的任意Representation。如果@codecs参数与在adaptation set层级中的不一样，应该在representation 层级标识。

@codecs参数呈现在representation 层级应该表示要求可解码representation中任意组件的能力。

@maxWidth 和@maxHeight参数不应该在任何Adaptation Set中出现。

@frameRate参数应该只在AdaptationSet元素中表示，即该参数值不应该与同一Adaptation Set中的其他Representation中的该参数值不同。@width 和@height参数不应该在任何Representation中指示。

9.1.2.3多轨模式

9.1.2.3.1概述

在多轨模式中，每个容积视频的视频组件在MPD中应该表示为单独的Adaptation Set，被称为视频组件自适应集（Component Adaptation Set）。图集信息由一个额外的Adaptation Set来表示，作为一个容积视频的主自适应集（Main Adaptation Set）。如果几何或属性组件有多个层级，每一个层级应该分别使用一个Component Adaptation Set元素来描述。

如果@codecs参数不在AdaptationSet元素中指示，并且容积视频的所有Component Adaptation Set或者这些Adaptation Set中的所有Representation的@codecs属性是基于视频组件各自使用的编码器进行设置，Main Adaptation Set的 @codecs属性应该设置为'v3c1'。@codecs的值应该设置为'resv.vvvc.XXXX'，XXXX对应编码器的四字符码（4CC）。

Main Adaptation Set应该包含一个Adaptation set层级的单独初始化片段。初始化片段应该包含用于初始化容积视频解码器的所有参数集，包含容积视频参数集和其他组件子流的参数集。

Main Adaptation Set的Representation中的媒体片段应该包含一个或多个容积视频图集轨道的轨道片段。Component Adaptation Set的Representation中的媒体片段应该包含一个或多个相应组件轨道的轨道片段。

9.1.2.3.2容积视频预选择

在MPD文件中包含一个或多个Preselection（定义在ISO/IEC 23009-1中）元素用于指示容积视频的预选择，其中，Preselection元素在Period元素中。容积视频的预选择提供容积视频的图集数据和组件数据的访问机制。该元素的@preSelectionComponents属性中的标识列表包含一个或多个Component Adaptation Set的标识符以及Main Adaptation Set的标识符。Preselection元素的@codecs属性应该设置为'v3c1'，指示Preselection描述的媒体内容是基于容积视频编码的媒体。

图 18展示了容积视频在DASH MPD文件中容积视频组件的分组示例，其中，容积视频只有一个图集且在一个容积视频图集轨道中。



图 18 使用Preselection分组容积视频组件的示例

当一个容积视频中存在多个图集时，每个图集所在的容积视频图集轨道应该分别通过单独的图集自适应集(Atlas Adaptation Set)来表示。Atlas Adaptation Set中的Representation的 @dependencyId属性应设置为容积视频Main Adaptation Set中Representation的标识符。一个Preselection包括一个Atlas Adaptation Set（可看作该Preselection的Main Adaptation Set）和该图集对应的视频组件的Component Adaptation Set。承载图集数据的Main Adaptation Set不应由任何Preselection元素选择。访问一个或多个Atlas Adaptation Set中承载的图集数据时，应要访问Atlas Adaptation Set关联的Main Adaptation Set中承载的容积视频入口图集的图集数据。

基于多图集的容积视频的部分访问，除了访问由一个或多个Preselection元素描述的容积视频预选择中指示的图集数据和对应的基于视频编码的容积组件，还应该同时检索承载通用图集数据的Adaptation Set。

图 19展示了多个图集情况下在DASH MPD文件中的容积视频组件的分组配置示例。

图 图 19 多个图集使用Preselection分组容积视频组件示例

9.1.2.3.3容积视频图集分片预选择

当容积视频图集分片数据分别承载在单独的轨道中时，应该通过单独的Adaptation Set来表示，其中，Adaptation Set的@codecs属性设置为'v3t1'。与一个图集分片轨道关联并承载基于视频编码的容积视频组件的轨道，也应承载在单独的Adaptation Set中，其中，Adaptation Set的@codecs属性设置为'resv.vvvc.XXXX'，XXXX对应视频编码器的四字符码（例如“ avc1”或“ hvc1”）。

关联并包含图集分片轨道的图集分片自适应集（Atlas Tile Adaptation Set）应成为MPD中一个图集分片Preselection的一部分，并且Atlas Tile Adaptation Set可看作是图集分片Preselection的Main Adaptation Set （即Atlas Tile Adaptation Set的标识符是Preselection元素中@preselectionComponents属性所表示的适应集标识符列表中的第一个标识符）。 图集分片Preselection的Atlas Tile Adaptation Set中的Representation所包含的@dependencyId属性应设置为对应Atlas Adaptation Set的一个Representation的标识符。

当容积视频存在一个或多个容积视频图集分片预选择时，承载一个或多个分片所属图集的图集数据的Adaptation Set不应由任何Preselection元素选择。基于图集分片的容积视频的部分访问，除了访问由一个或多个Preselection元素描述的容积视频预选择中指示的图集数据和对应的基于视频编码的容积视频组件，还应该同时检索承载一个或多个图集分片对应图集的图集数据的Adaptation Set。

图 20展示了多个图集的图集分片情况下在DASH MPD文件中容积视频组件的分组配置。



图 20 基于多图集的图集分片使用preselection元素的分组示例

9.1.2.4 支持多版本的容积视频

同一容积视频的多个版本应使用单独的Preselection来标识。表示相同容积视频的替代版本的Preselection应包含具有相同@vId值的V3C描述子。 在Preselection元素里最多应存在一个V3C描述子，因此，这些Preselection是彼此替代的，并且Preselection的Main Adaptation Set的标识（即@preSelectionComponents属性的Adaptation Set标识列表中的第一个标识）可能是不同的，其中，容积视频的每个版本都有单独的Main Adaptation Set在MPD文件中指示。

当需要从不同的组件Adaptation Set中选择特定的Representation集合作为特殊版本进行呈现时，使用Preselection对该版本进行标识。此时同一个容积视频多个版本的Preselection可具备可选属性@combineQualityRanking，用于表示该Preselection中所有容积组件对应的Representation集合的整体质量等级。此时该Preselection的@preSelectionComponents属性中，第一个标识为Main Adaptation Set的标识，其余标识为容积组件Representation的标识。

9.1.2.5 容积视频组件的编码器切换

如果使用不同的视频编解码器对容积视频组件的多个版本进行编码，则应根据所使用的视频编解码器在单独的Adaptation Set中用@codecs属性集的值来指示每个版本。 此外，每个Adaptation Set都应包含一个补充属性（SupplementalProperty）描述子，其@schemeIdURI设置为urn:mpeg:dash:adaptation-set-switching:2016，而@value是一个逗号分隔的Adaptation Set 标识列表，与其他可用版本相对应， 表示支持这些版本的Adaptation Set的Representation之间是无缝切换。 适用于ISO / IEC 23009-1:2019中章节5.3.3.5定义的支持在Adaptation Set之间进行切换的任何其他规则。

9.1.3基于自由视角视频的DASH传输

9.1.3.1 概述

基于自由视角视频的媒体文件进行切片封装并创建MPD。DASH MPD创建器生成特定自由视角视频的描述子，描述自由视角视频以及相机的元数据信息，上述这些信息可根据媒体文件中的相应元数据信息产生。

DASH客户端获得由服务器或其他方式传输/提供的MPD，从MPD中获取并使用自由视角视频相关特定描述子信息（如自由视角相机描述子），选择相应的自适应集（Adaptation Set）和表示（Representation）访问服务器端的自由视角视频的一个或多个相机对应的视频数据，获取待播放视角对应的一个或多个媒体文件的片段，根据片段中的合成任意视角视频，或直接播放片段中的自由视角视频。

当自由视角视频的纹理图信息和深度图信息分别位于不同的Representation时，深度图信息的Representation应使用@dependencyId关联至相应纹理图信息的Representation。若自由视角视频的Representation中不存在@dependencyId，则该Representation为包含纹理信息的Representation。

9.1.3.2 初始化片段

自由视角视频轨道在媒体文件中以受限的视频轨道进行表示，其轨道中包含一个 scheme\_type设置为'as3f'的RestrictedSchemeInfoBox。相应地，自由视角视频的初始化片段中应包含一个scheme\_type设置为'as3f'的RestrictedSchemeInfoBox。

9.2 沉浸媒体的DASH MPD描述子

9.2.1 XML命名空间和规则

本标准中，定义并使用了一些的XML元素和属性。新的XML元素定义在一个"urn:avs:ims:2018"的命名空间。命名空间指示符“ims：”指示了本文档中的名字空间（沉浸媒体系统）。在每一个章节定义的新的MPD描述子中，这些元素和属性被定义在标准规则文档中。命名空间指示符“xs:”xs:”对应命名空间<http://www.w3.org/2001/XMLSchema>，并作为XML规则（Schema）的第一部分，数据类型及其含义定义在XML Schema的第二部分。

9.2.2基于全景视频的DASH MPD描述子

9.2.2.1 投影类型描述子

一个@schemeIdUri属性为"urn:avs:ims:2018:pf"的必要属性（EssentialProperty）元素表示一个投影格式（Projection Format,PF）描述子。

至多一个投影格式描述子可能在MPD层被指定。至多一个投影格式描述子可能在Adaptation set层被指定。至多一个投影格式描述子可能在Representation层被指定。

一个投影格式描述子的ims:@projection\_type属性存在于一个低层级中，则覆盖高层级中的ims:@projection\_type属性。例如，当AdaptationSet元素中的AdaptationSet元素和Representation元素都包含投影格式描述子时，采用低层级Representation元素中的表述。

投影格式描述子的@value元素不应该被表述，投影格式描述子应该包含ims:@projection\_type属性，投影类型描述子包含的属性值如表19中所示。

表20 投影格式描述子属性及语义

| **属性** | **使用** | **描述** |
| --- | --- | --- |
| ims:@projection\_type | 必选 | 根据在表中定义的投影格式列表（参见表6，指定投影类型值，每个值在0到31的范围内，32到255的值保留。列表中的每一个值应该唯一。  对于ISOBMFF的片段，projection\_type应该与初始化片段的样本入口中的ProjectionFormatBox的projection\_type相同。 |

9.2.2.2 内容覆盖描述子

一个@schemeIdUri属性值为"urn:avs:ims:2018:cc"的SupplementalProperty元素表示一个内容覆盖（Content Coverage, CC）描述子。

至多一个内容覆盖描述子在Adaptation set层被指定，而在MPD 层或Representation层不应被指定。

内容覆盖描述子指示每个Representation覆盖的球面区域信息，表示该球面区域的shape\_type和SphereRegionStruct中的centre\_azimuth，centre\_elevation，centre\_tilt, azimuth\_range和elevation\_range都应包含在该描述子中。

当适用于一个Representation的内容覆盖描述子存在时，CoverageInformationBox存在于与Representation对应的轨道中，该描述子中应该带有与CoverageInformationBox相同的信息。

该描述子中@value属性不应被指定，内容覆盖描述子包含的元素和属性如表20所示：

表21 内容覆盖描述子属性及语义

| **元素及属性** | **使用** | **描述** |
| --- | --- | --- |
| cc | 0..1 | 包含指定球面区域覆盖信息的属性和元素 |
| cc@shape\_type | 可选 | 指出球面区域的形状类型，在章节8.1.4定义，当没有呈现，则该值为0 |
| cc@view\_idc\_presence\_flag | 可选 | 值为0指明cc.coverageInfo@view\_idc不被信令表示；  值为1指明cc.coverageInfo@view\_idc被信令表示，并指出含有双眼视图或单目内容的球面区域的关联；  当没有呈现，则该值为0. |
| cc@default\_view\_idc | CM | 值为0指出所有的球面区域为单目；  值为1指出所有的球面区域为立体内容的左眼视图；  值为2指出所有的球面区域为立体内容的右眼视图；  值为3指出所有的球面区域为双眼视图。  当cc@view\_idc\_presence\_flag为0时，cc@default\_view\_idc应该存在。cc@view\_idc\_presence\_flag为1时，cc@default\_view\_idc不应存在。 |
| cc.coverageInfo | 1..255 | 当cc.coverageInfo@view\_idc属性存在时，该元素提供关于球面区域覆盖的视图的信息，包括球面区域中心点位置、中心点的倾斜角度以及方位角和俯仰角的范围的属性。 |
| cc.coverageInfo @view\_idc | CM | 值为1指出球面区域在立体内容的左眼视图；  值为2指出球面区域在立体内容的右眼视图；  值为3指出球面区域在双眼视图。  值为0保留。  当cc@view\_idc\_presence\_flag为0时cc.coverageInfo@view\_idc应该不存在。当cc@view\_idc\_presence\_flag为1时，cc.coverageInfo@view\_idc应该存在。 |
| cc.coverageInfo @centre\_azimuth | 可选 | 指出基于全局坐标系统的球面区域中心点的方位角，如果没有呈现，则该值为0. |
| cc.coverageInfo @centre\_elevation | 可选 | 指出基于全局坐标系统的球面区域中心点的俯仰角，如果没有呈现，则该值为0. |
| cc.coverageInfo @centre\_tilt | 可选 | 指出基于全局坐标系统的球面区域的倾斜角度。如果没有呈现，则该值为0. |
| cc.coverageInfo @azimuth\_range | 可选 | 以2−16度为单位指定通过球面区域中心点的球面区域的方位角范围。如果没有呈现，则该值为 2π\* 216 |
| cc.coverageInfo @elevation\_range | 可选 | 以2−16度为单位指定通过球面区域中心点的球面区域的仰俯角范围。如果没有呈现，则该值为 2π\* 216 |

如果该描述子中没有内容覆盖元素，则指出当适用于Representation的投影格式描述子存在时，每个Representation覆盖整个球面。

当投影格式描述子在MPD层或Adaptation set层不存在，则内容覆盖描述子不应该存在于Adaptation set的元素中。

9.2.2.3 推荐视窗描述子

一个@schemeIdUri属性值为"urn:avs:ims:2018:vp"的SupplementalProperty元素表示一个推荐视窗（Viewport,VP）描述子，指示与该描述子相关联的Representation中包含在特定时间点的推荐视窗信息。

至多一个推荐视窗描述子在Adaptation set层被指定。

推荐视窗描述子的@value属性不应被指定。推荐视窗描述子包含的元素和属性如表22所示：

表 22 推荐视窗描述子属性及语法

| **元素与属性** | **用法** | **描述** | |
| --- | --- | --- | --- |
| vp | 0..1 | 包含推荐视窗信息的元素和属性 |
| vp@viewport\_type | 可选 | 指定推荐视窗的类型，在表3中定义，缺省值为0 |
| vp@description | 可选 | 推荐视窗的文本描述，以空字符结尾的UTF-8字符串 |
| vp@playback\_time | 必选 | 指定推荐视窗的播放时间 |
| vp.regioninfo | 1..255 | 该元素指定推荐视窗覆盖的球面区域位置信息 |
| vp.regioninfo  @shape\_type | 可选 | 指出球面区域形状类型 |
| vp.regioninfo @centre\_azimuth | 可选 | 指出基于全局坐标系统的球面区域中心点的方位角，如果没有呈现，则该值为0 |
| vp.regioninfo @centre\_elevation | 可选 | 指出基于全局坐标系统的球面区域中心点的俯仰角，如果没有呈现，则该值为0 |
| vp.regioninfo @centre\_tilt | 可选 | 指出基于全局坐标系统的球面区域的倾斜角度。如果没有呈现，则该值为0 |
| vp.regioninfo @azimuth\_range | 可选 | 以2−16度为单位指定通过球面区域中心点的球面区域的方位角范围。如果没有呈现，则该值为0 |
| vp.regioninfo @elevation\_range | 可选 | 以2−16度为单位指定通过球面区域中心点的球面区域的仰俯角范围。如果没有呈现，则该值为0 |

推荐视窗描述子可以关联一个或多个媒体轨道，指示在特定时间点建议播放的推荐视窗。客户端根据推荐视窗描述子信息可以提前获取相关联媒体轨道中的媒体内容，减少视窗切换时延。

9.2.2.4 视点信息描述子

一个@schemeIdUri属性为"urn:avs:ims:2018:vwpt"的SupplementalProperty元素表示一个视点（Viewpoint, VWPT）描述子。

至多一个视点描述子在Adaptation set层被指定，而在MPD层或Representation层不应被指定。如果未出现视点描述子，媒体呈现描述仅包含一个视点。

视点描述子指示全景视频的视点信息，包含的元素和属性如表23所示：

表 23 视点描述子属性及语法

| **元素或属性** | **使用** | **描述** |
| --- | --- | --- |
| @value | 必选 | 指示视点的视点标识 |
| vwpt | 1 | 容器元素，其子元素和属性提供@value所指定视点的视点信息。 |
| vwpt.Position | 1 | 此元素的属性提供@value所指定视点的三维空间位置信息。 |
| vwpt.Position@x | 0..1 | 在以（0,0,0）为公共参考坐标系中心的三维空间中指定视点的X轴坐标位置（以1/10毫米为单位）。如果该视点与动态视点定时元数据表示相关联，则该属性指定该视点的初始位置的X轴坐标，否则该属性指定静态视点的X轴坐标位置。 |
| vwpt.Position@y | 0..1 | 在以（0,0,0）为公共参考坐标系中心的三维空间中指定视点的Y轴坐标位置（以1/10毫米为单位）。如果该视点与动态视点定时元数据表示相关联，则该属性指定该视点的初始视点的Y轴坐标位置，否则该属性指定静态视点的Y轴坐标位置。 |
| vwpt.Position@z | 0..1 | 在以（0,0,0）为公共参考坐标系中心的三维空间中指定视点的Z轴坐标位置（以1/10毫米为单位）。如果该视点与动态视点定时元数据表示相关联，则该属性指定该视点的初始视点的Z轴坐标位置，否则该属性指定静态视点的Z轴坐标位置。 |
| vwpt.gps | 0..1 | 容器元素，其属性提供@value所指定视点的GPS位置。 |
| vwpt.gps@longitude | 1 | 以2-23度为单位指定视点地理位置的经度值。 经度应在-180 \* 223到180 \* 223 -1的范围内。 |
| vwpt.gps@latitude | 1 | 以2-23度为单位指定视点地理位置的纬度值。 纬度应在-90 \* 223到90 \* 223 -1的范围内。 |
| vwpt.gps@altitude | 1 | 以毫米为单位指定视点地理位置的高度值 |
| vwpt@initialViewpoint | 0..1 | 表示初始视点标志位。  如果等于true，则该属性指定此视点是当前时段（Period）所有视点的初始视点；  如果等于false，则该属性指定此视点不是当前Period中的初始视点。  在一个Period中，最多只能有一个视点的@ initialViewpoint属性等于true。如果Period中没有视点的@ initialViewpoint属性等于true或者如果不存在@ initialViewpoint属性，则初始视点由关联的初始视点定时元数据表示指定。 |
| vwpt. ViewPointGroup | 0..1 | 容器元素，其属性提供@value所指定视点的视点群组 |
| vwpt.ViewPointGroup@group\_id | 0..1 | 指示视点群组的标识符，该视点群组中的所有视点共用一个公共参考坐标系，用于确定播放媒体轨道的视点所属的视点群组。 |
| vwpt.ViewPointGroup@ group\_description | 0..1 | 提供视点群组的描述，以空值结尾的UTF-8字符串。 |
| vwpt@label | 0..1 | 指示视点的标签，为视点提供人类可读的以空字符结尾的UTF-8字符串。 |

如果视点描述子与一个样本入口为“dyvp”类型的定时元数据关联，则该视点是动态视点。否则，该视点为静态视点。基于关联关系，与视点描述子对应的媒体轨道相关联的动态视点定时元数据轨道需向服务器请求获取。

9.2.2.5 球面区域缩放描述子

方案标识属性（@schemeIdUri）等于"urn:avs:ims: 2018:srwz"的SupplementalProperty元素称为球面区域缩放（Sphere Region Zoom,SRWZ）描述子。

媒体呈现描述文件的Representation层级中最多存在一个球面区域缩放描述子。球面区域缩放描述子不应出现在MPD或者AdaptationSet层级中。

球面区域缩放描述子指示其对应Representation所携带的全景视频轨道中的全景视频的球面区域，以及球面区域在全景视频的投影图像上的一个或者多个缩放区域。

当存在适用于Representation的球面区域缩放描述子并且SphereRegionZoomingBox也存在于该Representation相对应的轨道中时，球面区域缩放描述子应携带与SphereRegionZoomingBox等效信息。

球面区域缩放描述子应包含下表中定义的元素和属性。

表 24 球面区域缩放描述子元素属性及语义

|  |  |  |  |
| --- | --- | --- | --- |
| **元素和属性** | **使用** | **数据类型** | **描述** |
| sphRegionZoom | 1 | omaf:sphRegionZoomType | 容器元素，其属性和元素指示球面区域及其对应的缩放区域 |
| sphRegionZoom @shape\_type | 可选 | xs:unsignedByte | 指示球面区域的形状类型。取值0球面区域通过四个大圆圈指示，取值1球面区域通过两个方位角和两个俯仰角圆指示 |
| sphRegionZoom@remaining\_area\_flag | 可选 | xs:boolean | 取值0表示所有球面区域由**SphRegionZoom**.**sphRegionInfo**元素定义，取值1表示除了最后球面区域的所有球面区域由**SphRegionZoom**.**sphRegionInfo**元素定义，并且最后剩余的球面区域是内容覆盖范围内未被**SphRegionZoom**.**sphRegionInfo**元素定义的球面区域集合覆盖的球面区域。 |
| sphRegionZoom @view\_idc\_presence\_flag | 可选 | xs:boolean | 取值0表示不存在**SphRegionZoom**.**sphRegionInfo**@view\_idc属性。取值1存在**SphRegionZoom**.**sphRegionInfo**@view\_idc属性，指示球面区域与特定视图（左、右或两者）或者单目图像关系。 |
| sphRegionZoom @default\_view\_idc | 条件必选 | omaf:ViewType | 取值0表示球面区域是单目图像，取值1表示球面区域是立体图像的左视图，取值2表示球面区域是立体图像的右视图，取值3表示球面区域包含立体图像的左视图和右视图。 |
| sphRegionZoom.  sphRegionInfo | 1..255 | omaf:zoomInfoType | 元素，其属性描述**sphRegionZoom**元素中的球面区域信息。其中，指定的球面区域数量至少为一个。 |
| sphRegionZoom.sphRegionInfo@view\_idc | 条件必选 | omaf:ViewType | 取值0表示球面区域是单目图像，取值1表示球面区域是立体图像的左视图，取值2表示球面区域是立体图像的右视图，取值3表示球面区域包含立体图像的左视图和右视图。 |
| sphRegionZoom.sphRegionInfo@centre\_azimuth | 条件必选 | omaf:Range1 | 以2−16度为单位指示球面区域的中心点的方位角。 |
| sphRegionZoom.sphRegionInfo@centre\_elevation | 条件必选 | omaf:Range2 | 以2−16度为单位指示球面区域的中心点的俯仰角。 |
| sphRegionZoom.sphRegionInfo@centre\_tilt | 条件必选 | omaf:Range1 | 以2−16度为单位指示球面区域的中心点的倾斜角。 |
| sphRegionZoom.sphRegionInfo@azimuth\_range | 条件必选 | omaf:HRange | 通过其中心点以2−16度为单位定义球面区域的方位角范围。 |
| sphRegionZoom.sphRegionInfo@elevation\_range | 条件必选 | omaf:HRange | 通过其中心点以2−16度为单位定义球面区域的俯仰角范围。 |
| sphRegionZoom.  zoomInfo | 1..255 | omaf:sphRegionInfoType | 元素，其属性描述sphRegionZoom.sphRegionInfo元素中所定义球面区域对应的缩放区域信息。其中，指定的缩放区域数量至少为一个。 |
| sphRegionZoom.zoomInfo@zoom\_region\_left | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域左上角的水平坐标。 |
| sphRegionZoom.zoomInfo@zoom\_region\_right | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域左上角的垂直坐标。 |
| sphRegionZoom.zoomInfo@zoom\_region\_width | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域的宽度。 |
| sphRegionZoom.zoomInfo@zoom\_region\_height | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域的高度。 |
| SphRegionZoom.zoomInfo@zoom\_ratio | 条件必选 | xs:unsignedByte | 指示投影图像内缩放区域的缩放比例。 |

表 24 （续）

|  |  |  |  |
| --- | --- | --- | --- |
| **元素和属性** | **使用** | **数据类型** | **描述** |
| SphRegionZoom.zoomInfo@zoom\_algorithm\_type | 可选 | omaf:listofUnsignedByte | 指示投影图像内缩放区域的缩放算法。 |
| SphRegionZoom.zoomInfo@zoom\_symbolization\_type | 可选 | omaf:listofUnsignedByte | 指示投影图像内缩放区域的缩放边界符号化类型。 |
| SphRegionZoom.zoomInfo@zoom\_description | 可选 | xs:string | 指示投影图像内缩放区域的描述信息。 |

根据球面区域缩放描述子，终端请求获取与全景视频的球面区域缩放操作对应的视频文件。

9.2.2.6 二维区域缩放描述子

方案标识属性（@schemeIdUri）等于"urn:mpeg:mpegI:omaf:2018:二维wz"的 SupplementalProperty元素称为二维区域缩放（Two Dimensional Region Zoom,2DWZ）描述子。

媒体呈现描述文件的Representation层级中最多存在一个二维区域缩放描述子。二维区域缩放描述子不应出现在MPD或者AdaptationSet层级中。

二维区域缩放描述子指示其对应Representation所携带的全景视频轨道中的全景视频的投影图像上的二维区域，以及二维区域在全景视频的投影图像上的一个或者多个缩放区域。

当存在适用于Representation的二维区域缩放描述子并且2DRegionZoomingBox也存在于该Representation相对应的轨道中时，二维区域缩放描述子应携带与2DRegionZoomingBox等效信息。二维区域缩放描述子应包含下表中定义的元素和属性。

表 25 二维区域缩放描述子的元素属性及语义

| **元素和属性** | **使用** | **数据类型** | **描述** |
| --- | --- | --- | --- |
| twoDRegionZoom | 1 | omaf:twoDRegionZoomType | 容器元素，其属性和元素指示二维区域及其对应的缩放区域 |
| twoDRegionZoom@remaining\_area\_flag | 可选 | xs:boolean | 取值0表示所有二维区域由twoDRegionZoom.twoDRegionInfo元素定义，取值1表示除了最后二维区域的所有二维区域由twoDRegionZoom.twoDRegionInfo元素定义，并且最后剩余的二维区域是内容覆盖范围内未被twoDRegionZoom.twoDRegionInfo元素定义的二维区域集合覆盖的二维区域。 |
| twoDRegionZoom@view\_idc\_presence\_flag | 可选 | xs:boolean | 取值0表示不存在twoDRegionZoom.twoDRegionInfo @view\_idc属性。取值1表示存在twoDRegionZoom.twoDRegionInfo@view\_idc属性，指示二维区域与特定视图（左、右或两者）或者单目图像关系。 |
| twoDRegionZoom@default\_view\_idc | 条件必选 | omaf:ViewType | 取值0表示二维区域是单目图像，取值1表示二维区域是立体图像的左视图，取值2表示二维区域是立体图像的右视图，取值3表示二维区域包含立体图像的左视图和右视图。 |
| twoDRegionZoom.twoDRegionInfo | 1..255 | omaf:twoDRegionInfoType | 元素，其属性描述twoDRegionZoom元素中的二维区域信息。其中，指定的二维区域数量至少为一个。 |
| twoDRegionZoom.twoDRegionInfo@view\_idc | 条件必选 | omaf:ViewType | 取值0表示二维区域是单目图像，取值1表示二维区域是立体图像的左视图，取值2表示二维区域是立体图像的右视图，取值3表示二维区域包含立体图像的左视图和右视图。 |
| twoDRegionZoom.twoDRegionInfo@left\_offset | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内二维区域左上角水平坐标。 |
| twoDRegionZoom.twoDRegionInfo@top\_offset | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内二维区域左上角垂直坐标。 |
| twoDRegionZoom.twoDRegionInfo@region\_width | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内二维区域的宽度。 |
| twoDRegionZoom.twoDRegionInfo@region\_height | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内二维区域的高度。 |
| twoDRegionZoom.zoomInfo | 1..255 | omaf:zoomInfoType | 元素，其属性描述twoDRegionZoom.twoDRegionInfo元素所定义二维区域对应的缩放区域信息。其中，指定的缩放区域数量至少为一个。 |
| twoDRegionZoom.zoomInfo@zoom\_region\_left | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域的左上角的水平坐标。 |
| twoDRegionZoom.zoomInfo@zoom\_region\_right | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域的左上角的垂直坐标。 |
| twoDRegionZoom.zoomInfo@zoom\_region\_width | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域的宽度。 |
| twoDRegionZoom.zoomInfo@zoom\_region\_height | 条件必选 | xs:unsignedShort | 以亮度样本为单位指定投影图像内缩放区域的高度。 |
| twoDRegionZoom.zoomInfo@zoom\_ratio | 条件必选 | xs:unsignedByte | 指示投影图像内缩放区域的缩放比例 |
| twoDRegionZoom.zoomInfo@zoom\_algorithm\_type | 可选 | omaf:listofUnsignedByte | 指示投影图像内缩放区域的缩放算法 |
| twoDRegionZoom.zoomInfo@zoom\_symbolization\_type | 可选 | omaf:listofUnsignedByte | 指示投影图像内缩放区域的缩放边界符号化类型 |
| twoDRegionZoom.zoomInfo@zoom\_description | 可选 | xs:string | 指示投影图像内缩放区域的描述信息 |

根据二维区域缩放描述子，终端请求获取与全景视频的投影图像上的二维区域缩放操作对应的视频文件。

9.2.2.7 独立编解码区域群组描述子

方案标识属性（@schemeIdUri）等于" urn:avs:ims: 2018:icrg"的 SupplementalProperty元素为独立编解码区域群组（Independently Coded Region Group,icrg）描述子。该描述子用于描述同一个独立编解码区域轨道组内独立编解码区域轨道对应的Representation，如表26所示。

独立编解码区域群组描述子可以出现在AdaptationSet层级，且不能出现在其他层级。

独立编解码区域群组描述子可携带一个可选属性compositionId。

独立编解码区域群组描述子可携带一个可选属性noSingleSelection。

表 26 IndependentlyCodedRegionGroup元素及属性语法

|  |  |  |  |
| --- | --- | --- | --- |
| **元素和属性** | **使用** | **数据类型** | **描述** |
| IndependentlyCodedRegionGroup | 0..N | xs: unsignedShort | 指示当前Adaptation Set 中携带的独立编解码区域轨道所属的独立编解码区域组ID。 |
| IndependentlyCodedRegionGroup@compositionId | O | xs: unsignedByte | 指示当前Adaptation Set 中携带的独立编解码区域轨道所属的独立编解码区域组内的重构坐标系ID。 |
| IndependentlyCodedRegionGroup@noSingleSelection | O | xs:boolean | 该字段为true时，说明当前Adaptation Set 中携带的独立编解码区域轨道必须与同一IndependentlyCodedRegionGroup下的其他轨道同时呈现。  该字段为false时，说明当前Adaptation Set 中携带的独立编解码区域轨道与同一IndependentlyCodedRegionGroup下的其他轨道可以同时呈现，也可以不同时呈现。  当该字段缺失时，默认该字段取值为false。 |

9.2.3基于容积视频的DASH MPD描述子

9.2.3.1 容积视频组件描述子

应使用容积视频组件（V3CComponent）描述子识别视频组件Adaptation Set中的容积视频组件的类型。一个容积视频组件描述子是一个EssentialsProperty描述子，其@schemeIdUri属性设置为“urn:avs:ims:2018:component”。

在Adaptation set层级上，存在于视频组件Adaptation Set的Representation中的每个容积视频的视频组件应由一个容积视频组件描述子来表述。

容积视频组件描述子的@value属性不应存在。 容积视频组件描述子应包括表27中指定的元素和属性。

表27 容积视频组件描述子的元素和属性

| **元素和属性** | **使用** | **数据类型** | **描述** |
| --- | --- | --- | --- |
| V3CComponent | 0..N | v3c:v3cComponentType | 一个元素，其属性指示Adaptation set的Representation中存在的一个容积视频组件的信息。 |
| V3CComponent@component\_type | 必选 | xs:string | 指示容积视频组件的类型。 'geom'表示几何图形组件，'occp'表示占用组件，'attr'表示属性组件。 |
| V3CComponent@is\_raw | 条件可选 | xs:boolean | 一个标志，用于指示由存在V3CComponent描述子的Adaptation set所表示的容积视频组件信息是否用于单独的原始点图块。 值为true表示该信息是针对原始点图块。  如果不存在，则默认值为false。 |
| V3CComponent@is\_single\_map | 条件可选 | xs:boolean | 一个标志，指示由存在V3CComponent描述子的Adaptation set所表示的容积视频几何或属性组件是否包含容积视频一个或全部层级。 值为false表示存在该组件的所有层级。 值为true表示仅存在一个层级，其索引由@map\_index给出。  仅当@component\_type的值为“ geom”或“ attr”时，该属性应存在。  如果不存在，并且@component\_type设置为“ geom”或“ attr”，则默认值为false。 |
| V3CComponent@map\_index | 条件可选 | xs:integer | 指示一个包含V3CComponent描述子的Adaptation set表示的组件层级的索引。 仅在如果存在@is\_single\_map并且其值设置为true时，该属性应存在。 |
| V3CComponent@attribute\_type | 条件可选 | xs:unsignedByte | 指示属性的类型（参考ISO / IEC FDIS 23090-5）。仅允许使用0到15之间的值（包括0和15）。  仅当组件是容积视频属性（即component\_type的值为“ attr”）时，该属性应存在。 |
| V3CComponent@attribute\_index | 条件可选 | xs:unsignedByte | 指示属性的索引。 值应介于0到127之间（含）。  仅当组件是容积视频属性（即component\_type的值为“ attr”）时，该属性应存在。 |
| V3CComponent@attribute\_dim\_partition\_index | 条件可选 | xs:unsignedByte | 指示包含V3CComponent描述子的Adaptation set中所载属性的维度分区索引。  仅当组件是容积视频属性（即component\_type的值为“attr”）时，该属性应存在。  如果不存在，则默认值为1。 |
| V3CComponent@atlas\_id | 缺省可选 | xs:integer | 指示其中包含V3CComponent描述符的Adaptation set所表示组件对应的图集标识符。  如果不存在@atlas\_id，则默认值为0。 |
| V3CComponent@tile\_ids | 可选 | xs:UIntVectorType | 通过提供空格分隔的分片标识符的列表来指定与Adaptation set中包含的数据相关的图集分片。 在没有@tile\_ids的情况下，适应集将包含图集的所有分片。 |

9.2.3.2 容积视频描述子

容积视频（V3C）描述子是一个SupplementalProperty元素，其@schemeIdUri属性等于“urn:avs:ims:2018:v3cd” 。 在容积视频Main Adaptation Set层级至多可存在一个容积视频描述子。

每一个PreSelection元素可以包含一个容积视频描述子，并进一步包含图集数据所在轨道标识和该图集对应图集分片的地址。容积视频描述子应包含在表28指定的属性。

表28 容积视频描述子属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **使用** | **数据类型** | **描述** |
| v3c:@vId | 条件必选 | xs:string | 一个容积视频的标识符。 如果在单独的Adaptation set中指示同一容积视频的多个版本，则应存在此属性。 |
| v3c:@atlas\_id | 缺省可选 | xs:integer | 指示包含容积视频描述子的图集适应集中所携带的图集的标识符，  如果不存在@atlas\_id，则默认值为0。 |
| v3c:@tile\_ids | 可选 | xs:UIntVectorType | 指示对应容积视频图集（由@atlas\_id指示）的分片地址列表，以空格分隔。  @tile\_ids属性的值是一个由空格分隔的图集分片的地址列表，指示对应容积视频图集（由@atlas\_id指示）的分片地址。 |
| v3c:@regionIds | 条件可选 | xs:stringVectorType | 由空格分隔的空间区域的标识符列表。如果该容积视频内容对应一个或多个容积视频的三维空间区域，该属性应存在。 |
| v3c:@viewIds | 条件可选 | xs:stringVectorType | 以空格分隔的视图标识符列表。 如果容积视频内容对应于一个或多个视图，则该属性应存在。 |

9.2.3.3 静态空间区域描述子

如果三维空间区域是静态的（即每个区域的位置和尺寸在呈现时间内没有变化），则应使用容积视频空间区域（V3CSpatialRegion， VSR）描述子指示容积视频的三维空间区域的特征以及这些区域与容积视频的图集分片之间的映射。

静态空间区域描述子是一个SupplementalProperty元素，其@schemeIdUri属性等于“ urn:avs:ims:2018:vsr”。静态空间区域描述子应存在于Main Adaptation set或其Representation中。

V3CSpatialRegions描述子的@value属性不应存在。V3CSpatialRegions描述子应包括表29中指定的元素和属性。

表29 容积视频空间区域描述子的元素和属性

| **元素和属性** | **使用** | **数据类型** | **描述** |
| --- | --- | --- | --- |
| V3CspatialRegion | 0..N | v3c:spatialRegionType | 三维空间区域元素，其属性定义三维空间区域，并提供定义的区域和多个容积视频图集分片之间的映射。 |
| V3CspatialRegion@id | 必选 | xs:unsignedByte | 三维空间区域的标识符。  该属性的值应与为ISOBMFF容器中的相应区域的3d\_region\_id字段的值匹配。 |
| V3CspatialRegion@ reference\_x | 必选 | xs:int | 定义空间区域的边界框参考点在笛卡尔坐标中的x坐标。 |
| V3CspatialRegion@ reference\_y | 必选 | xs:int | 定义空间区域的边界框参考点在笛卡尔坐标中的y坐标。 |
| V3CspatialRegion@ reference\_z | 必选 | xs:int | 定义空间区域的边界框参考点在笛卡尔坐标中的z坐标。 |
| V3CspatialRegion@delta\_x | 必选 | xs:int | 空间区域边界框在笛卡尔坐标中沿x轴的长度（即宽度）。 负值表示在x轴的负方向上延伸的长度。 |
| V3CspatialRegion@ delta\_y | 必选 | xs:int | 空间区域边界框在笛卡尔坐标中沿y轴的长度（即高度）。 负值表示在y轴的负方向上延伸的长度。 |
| V3CspatialRegion@ delta\_z | 必选 | xs:int | 空间区域边界框在笛卡尔坐标中沿z轴的长度（即深度）。 负值表示在z轴的负方向上延伸的长度。 |
| V3CspatialRegion@trackGroupIds | 条件可选 | StringVectorType | 使用空格分隔的标识符的列表，指示该描述子表示的三维空间区域对应的一个或多个容积是篇组件轨道组的track\_group\_id值。  在单轨封装容积视频内容的情况下，应缺少此属性。 |

9.2.3.4 动态空间区域描述子

当三维空间区域是动态的时候，用于在呈现时间轴上指示每个三维空间区域的位置和尺寸的定时元数据轨道应在包含单个Representation的独立的Adaptation Set中承载，使用@associationId属性（在ISO / IEC 23009-1中定义）与Main Adaptation Set中的Representation相关联，并且对应的@associationType值为'dyvm' 。

9.2.3.5容积视频视图描述子

使用容积视频视图（V3CView，VPVW）描述子识别Main AdaptationSet 中容积视频内容及其各自关联的容积视频图集轨道的视图。 容积视频视图描述子是一个 EssentialProperty 或 SupplementalProperty 描述子，其 @schemeIdUri 属性等于“urn:avs:ims:2018:vpvw”。

在Main AdaptationSet 中，至多存在一个容积视频视图描述子在Adaptation Set层级或Representation层级，或者存在于容积视频的Preselection层级。

容积视频视图描述子的@value 属性不应存在。 容积视频视图描述子应包括表 30中规定的元素和属性。

表 30 容积视频视图描述子的元素和属性

| **元素和属性** | **使用** | **数据类型** | **描述** |
| --- | --- | --- | --- |
| V3Cview | 0..N | v3c:viewGroupType | 一个元素，其元素和属性指定容积视频及其关联的容积视频媒体轨道的视图信息。 |
| V3Cview@viewId | 必选 | xs:unsignedByte | 一个属性，指定视图标识符 |
| V3Cview@viewGroupId | 必选 | xs:unsignedByte | 一个属性，指定视图组标识符 |
| V3Cview@is\_basic\_View | 可选 | xs:boolean | 指示视图是否被选为基础视图的标志。 如果不存在，则默认值为 false。 |
| V3Cview.cameralParameter | 可选 | v3c:cameraParameterType | 一个元素，指定视图对应的真实或虚拟相机位置和方向信息的元素。 |
| V3Cview.cameralParameter@cameraId | 必选 | xs:unsignedByte | 一个属性，指定相机标识符 |
| V3Cview.cameralParameter@camera\_pos\_x | 必选 | xs:int | 全局参考坐标系中相机位置的 x 坐标值（以米为单位），值应以 2-16 米为单位。 |
| V3Cview.cameralParameter@camera\_pos\_y | 必选 | xs:int | 全局参考坐标系中相机位置的 y 坐标值（以米为单位），值应以 2-16 米为单位 |
| V3Cview.cameralParameter@camera\_pos\_z | 必选 | xs:int | 全局参考坐标系中相机位置的 z 坐标值（以米为单位），值应以 2-16 米为单位。 |
| V3Cview.cameralParameter@camera\_quat\_x | 必选 | xs:float | 使用四元数表示的相机方向的 x 分量。 |
| V3Cview.cameralParameter@camera\_quat\_y | 必选 | xs:float | 使用四元数表示的相机方向的 y 分量。 |
| V3Cview.cameralParameter@camera\_quat\_z | 必选 | xs:float | 使用四元数表示的相机方向的 z 分量。 |
| V3Cview.cameralParameter@cameraType | 必选 | xs:int | 指示视窗相机的投影算法类型。取值为0表示ERP投影；取值为1表示透视投影；取值为2表示正交投影；其余取值保留。 |
| V3Cview.cameralParameter@erp\_horizontal\_fov | 条件必选 | v3c:Range | 指示ERP投影时，视窗区域水平方向的经度范围。 |
| V3Cview.cameralParameter@erp\_vertical\_fov | 条件必选 | v3c:Range | 指示ERP投影时，视窗区域垂直方向的维度范围。 |
| V3Cview.cameralParameter@perspective\_horizontal\_fov | 条件必选 | v3c:Range | 指示透视投影时，视窗区域的水平范围。 |
| V3Cview.cameralParameter@perspective\_aspect\_ratio | 条件必选 | xs:float | 指示透视投影时，视窗的纵横比例(水平/垂直)。 |
| V3Cview.cameralParameter@ ortho\_aspect\_ratio | 条件必选 | xs:float | 指示正交投影时，视窗的纵横比例(水平/垂直)。 |
| [V3Cview.cameralParameter@ortho\_horizontal\_size](mailto:V3Cview.cameralParameter@ortho_horizontal_size) | 条件必选 | xs:int | 指示正交投影时，视窗水平方向的尺寸，以米为单位。 |
| V3Cview.cameralParameter@clipping\_near\_plan | 条件必选 | xs:float | 指示视窗近点的深度 |
| V3Cview.cameralParameter@clipping\_far\_plane | 条件必选 | xs:float | 指示原点剪裁平面的深度 |
| V3Cview@v3cTrackIds | 可选 | xs:StringVectorType | 一个空格分隔的标识符列表，对应于容积视频图集轨道的 track\_id 值，这些轨道携带视图对应的图集数据。 |

9.2.3.6三维观看空间描述子

三维观看空间（Viewing Space，VS）描述子是一个SupplementalProperty元素，其@schemeIdUri属性为“urn:avs:ims:2018:vs”，通过三维观看空间描述子可参考用户观看位姿（位置、朝向等）与三维观看空间的关系确定并获取对应的媒体数据。

至多一个三维观看空间描述子在Adaptation Set层被指定。如果未出现三维观看空间描述子，则可能无法移动观看位置或者可在空间中任意移动观看位置。

三维观看空间描述子包含的元素和属性如下表31所示。

表31 三维观看空间描述子元素和属性

|  |  |  |
| --- | --- | --- |
| **元素和属性** | **使用** | **描述** |
| vs | 0…255 | 包含三维观看空间信息的元素和属性 |
| vs@ViewingSpaceid | O | 指示三维观看空间标识 |
| vs.struct | 1…255 | 其属性描述vs元素中的组合三维观看空间的信息。 |
| vs.struct@shape\_operation | CM | 指示对组合成三维观看空间的复杂几何体的操作类型；  取值0表示进行几何体相并的组合方式；取值1表示进行几何体相差的组合方式；取值2表示进行几何体相交的组合方式。 |
| vs.struct.primitive | 1…255 | 指示组成三维观看空间的基础几何体的结构 |
| vs.struct.primitive@view\_id | CM | 指示基础几何结构对应的视点标识，如果该基础几何结构不对应任何摄像机，则没有对应的视点。 |
| vs.struct.primitive@ shape\_type | M | 取值0表示基础几何结构为立方体；取值1表示基础几何结构为球形体；取值2表示基础几何结构为半球体。 |
| vs.struct.primitive@operation\_type | CM | 取值0表示使用CSG模式对基础几何体进行相加形成新的复杂几何体；取值1时，指示将沿着基础几何体中心形成的路径对基础几何体进行插值操作，以形成新的复杂几何体。 |
| vs.struct.primitive @center\_x | O | 指示基础几何结构在坐标系中x坐标值，如果vs.struct.primitive@ shape\_type为0或者1时，缺省值为0；如果vs.struct.primitive@ shape\_type为2时，则不需要该属性 |
| vs.struct.primitive @center\_y | O | 指示基础几何结构在坐标系中y坐标值，如果vs.struct.primitive@ shape\_type为0或者1时，缺省值为0；如果vs.struct.primitive@ shape\_type为2时，则不需要该属性 |
| vs.struct.primitive @center\_z | O | 指示基础几何结构在坐标系中y坐标值，如果vs.struct.primitive@ shape\_type为0或者1时，缺省值为0；如果vs.struct.primitive@ shape\_type为2时，则不需要该属性 |
| vs.struct.primitive @size\_x | CM | 指示立方体的x轴方向的边长度，当vs.struct.primitive@shape\_type为0时，该属性存在 |
| vs.struct.primitive @size\_y | CM | 指示立方体的y轴方向的边长度，当vs.struct.primitive@shape\_type为0时，该属性存在 |
| vs.struct.primitive @size\_z | CM | 指示立方体的z轴方向的边长度，当vs.struct.primitive@shape\_type为0时，该属性存在 |
| vs.struct.primitive @radius\_x | CM | 指示球形体的x轴方向的半径长度，当vs.struct.primitive@shape\_type为1时，该属性存在 |
| vs.struct.primitive @radius\_y | CM | 指示球形体的y轴方向的半径长度，当vs.struct.primitive@shape\_type为1时，该属性存在 |
| vs.struct.primitive @radius\_z | CM | 指示球形体的z轴方向的半径长度，当vs.struct.primitive@shape\_type为1时，该属性存在 |
| vs.struct.primitive @normal\_x | CM | 指示半球体四元数组的沿着X轴的法向量，当vs.struct.primitive@shape\_type为2时，该属性存在 |
| vs.struct.primitive @normal\_y | CM | 指示半球体四元数组的沿着Y轴的法向量，当vs.struct.primitive@shape\_type为2时，该属性存在 |
| vs.struct.primitive @normal\_z | CM | 指示半球体四元数组的沿着Y轴的法向量，当vs.struct.primitive@shape\_type为2时，该属性存在 |

表31（续）

|  |  |  |
| --- | --- | --- |
| **元素和属性** | **使用** | **描述** |
| vs.struct.primitive @size | CM | 指示从原点沿着法向量到半球体平面方向的距离，当vs.struct.primitive@shape\_type为2时，该属性存在 |
| vs.struct.primitive @guard\_band | O | 指示简单几何体的保护带的大小 |
| vs.struct.primitive.DirectionConstrain | O | 其属性描述组成复杂几何体的基础几何体中观看方向限制信息 |
| vs.struct.primitive. DirectionConstrain@ direction\_center\_x | O | 指示在基础几何体中建议观看方向的中心的四元数分量x |
| vs.struct.primitive. DirectionConstrain@ direction\_center\_y | O | 指示在基础几何体中建议观看方向的中心的四元数分量y |
| vs.struct.primitive. DirectionConstrain@ direction\_center\_z | O | 指示在基础几何体中建议观看方向的中心的四元数分量z |
| vs.struct.primitive. DirectionConstrain@ direction\_yaw\_range | O | 指示在基础几何体中观看方向的偏航范围二分之一； |
| vs.struct.primitive. DirectionConstrain@ direction\_pitch\_range | O | 指示在基础几何体中观看方向的倾斜范围二分之一； |
| vs.struct.primitive.rotation | O | 指示基础几何体的旋转信息 |
| vs.struct.primitive.rotation@rotation\_x | O | 指示用于基础几何体的旋转四元数的x分量 |
| vs.struct.primitive.rotation@rotation\_y | O | 指示用于基础几何体的旋转四元数的y分量 |
| vs.struct.primitive.rotation@rotation\_z | O | 指示用于基础几何体的旋转四元数的z分量 |
| vs.struct.primitive.rotation@guard\_band\_direction\_size | O | 指示基础几何体内观看方向的保护带的大小，以度数为单位来表示 |

9.2.3.7三维观看空间处理方式描述子

三维观看空间处理方式（Viewing Space Handing, VWPH）描述子是一个SupplementalProperty元素，其@schemeIdUri属性为“urnurn:avs:ims:2018:vwph”。

至多一个三维观看空间处理方式描述子在MPD 层被指定。三维观看空间处理方式描述子不能同时出现在两个层级中，如果表述在低层级中，则在高层级的中重用。

三维观看空间处理方式描述子包含的元素和属性如下表32所示。

表 32 三维观看空间处理方式描述子元素和属性

|  |  |  |
| --- | --- | --- |
| **元素和属性** | **使用** | **描述** |
| vwph | 0…255 | 其属性描述vwph元素中的三维观看空间处理方式的信息 | |
| vwph@spaceid | O | 指示三维观看空间标识 | |
| vwph@deviceclass | O | 指示三维观看空间处理的设备类型。 | |
| vwph@applicationclass | O | 指示三维观看空间处理的应用类型 | |
| vwph@handlingmethod | O | 指示三维观看空间处理的方式类型 | |

9.2.4基于自由视角视频的DASH MPD描述子

9.2.4.1 自由视角相机描述子

自由视角相机描述子（AvsFreeViewCamInfo）为SupplementalProperty元素，其@schemeIdUri属性为"urn:avs:ims:2018:av3l"。该描述子可存在于adaptation set层级或representation层级。该描述子存在于adaptation set层级时，描述该adaptation set内所有的representation；该描述子存在于representation层级时，描述对应的representation。自由视角相机描述子指示自由视角相机信息的相关属性，具体属性如下表所示。

表33 自由视角相机描述子元素和属性

|  |  |  |  |
| --- | --- | --- | --- |
| **元素和属性** | **使用** | **数据类型** | **描述** |
| AvsFreeViewCamInfo | 0..N | avs:ims:2018:av3l | 该元素的属性指示对应的representation或representation set对应的自由视角拍摄相机的属性。该元素可存在多个。 |
| AvsFreeViewCamInfo@cameraId | M | xs:unsignedInt | 指示每个相机对应的相机标识符，该标识符与文件轨道中AvsFreeViewInfoBox中的取值对应。 |
| AvsFreeViewCamInfo@camera\_pos\_x | M | xs:float | 指示该拍摄相机位置的x分量取值。 |
| AvsFreeViewCamInfo@camera\_pos\_y | M | xs:float | 指示该拍摄相机位置的y分量取值。 |
| AvsFreeViewCamInfo@camera\_pos\_z | M | xs:float | 指示该拍摄相机位置的z分量取值。 |
| AvsFreeViewCamInfo@virtual\_flag | CM | xs:bool | 当自由视角视频中同时包含原始相机和虚拟相机的视频图像时使用。取值为1时表示视角对应相机为虚拟相机。取值为0时表示视角对应的相机为原始相机。缺省时表示视角对应的相机为原始相机。 |
| AvsFreeViewCamInfo@related\_camera | O | xs:intVectorType | 当virtual\_flag为1时使用，指示虚拟相机关联的物理相机的标识，用逗号分隔，即指示虚拟相机对图像所对应的物理相机。 |
| AvsFreeViewCamInfo@combinationId | O | xs:unsignedInt | 指示相应自由视角视频所属的组合标识符。该字段取值对应AvsFreeViewGroupBox中的track\_group\_id字段。 |

沉浸媒体SMT传输信令

10.1 概述

SMT信令功能区中包含了一整套用于传输和消费SMT数据包的必要信令消息，为了使SMT流传输全景视频、容积视频和自由视角视频编码数据，本章定义了针对沉浸式媒体的SMT扩展信令表、描述符或者传输相关信息的消息格式，包括：

——虚拟现实资源描述符

——ROI引导信息表

——全景视频字幕在SMT上的传输扩展

——容积视频视角依赖媒体数据包消息

——容积视频资源索引描述符

——媒体资源组消息

——Asset关系信息描述符

——自由视角相机描述符

——沉浸媒体交互反馈信息

10.2 全景视频的SMT传输信令

10.2.1 虚拟现实信息资源描述符

10.2.1.1概述

资源描述符用于向SMT虚拟现实接收实体和虚拟现实应用程序告知关于当前携带虚拟现实内容的资源具体内容。信息资源描述符描述了使用的投影类型，默认的初始观看位置，这些内容所覆盖的球体区域以及该部分视频具体的3DoF+约束信息。

10.2.1.2语法

下表展示了虚拟现实信息资源描述符的语法。

表 34 虚拟现实信息资源描述符

|  |  |  |  |
| --- | --- | --- | --- |
| **语法** | **值** | **比特数** | **类型** |
| VR\_information\_descriptor () {  descriptor\_tag  descriptor\_length  3DoF+\_flag  reserved  ProjectionFormatStruct()  InitialViewingOrientationSample()  SphereRegionStruct ()  if(3DoF+\_flag) {  HXmax  HXmin  HYmax  HYmin  HZmax  HZmin  }  } |  | 16  8  1  7  16  16  16  16  16  16 | uimsbf  uimsbf  bslbf  bslbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf |

10.2.1.3语义

descriptor\_tag指示描述符的类型。

descriptor\_length指示从该字段的下一个字节到描述符最后一个字节的字节长度。

3DoF+\_flag取值为1时表示存在3DoF+视频内容；取值为0时表示不存在3DoF+视频内容。

ProjectionFormatStruct指示有关投影格式的信息，包括经纬图投影和正六面体投影等。

InitialViewingOrientationSample指示在播放相应媒体内容时默认的初始观看方向。当缺少这一信息时，播放相应媒体内容的默认初始视点。

SphereRegionStruct指示单位球面的一个球面区域。

HXmax、HXmin、HYmax、HYmin、HZmax、HZmin分别指示观看者在x轴上头部移动的最大值、x轴上头部移动的最小值、y轴上头部移动的最大值、y轴上头部移动的最小值、z轴上头部移动的最大值、z轴上头部移动的最小值，取值范围均为(0, 216-1)，单位为毫米。

10.2.2 ROI引导信息表

10.2.2.1概述

SMT发送端通过下发ROI引导信息表指示接收端从交付的整个视频内容中寻找需要呈现的指定区域的信息。如果SMT接收端接收了整个视频的媒体数据，则仅应呈现由ROI引导信息表指定的区域的视频内容。

10.2.2.2语法

下表定义了ROI引导信息表的语法。

表 35 ROI引导信息表的语法

|  |  |  |  |
| --- | --- | --- | --- |
| **语法** | **值** | **比特数** | **类型** |
| ROIGuide\_table (){  table\_id  version  length  reserved  region\_type  description\_length  for(i=0;i<N1;i++) {  description\_byte[i]  }  guide\_shape\_type  SphereRegionStruct(region\_type)  guide\_start  } | '11111111'  N1 | 8  8  16  8  8  16  8  8  32 | uimsbf  uimsbf  uimsbf  bslbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf |

10.2.2.3语义

table\_id指示ROI引导信息表的标识符。

version指示ROI引导信息表的版本。新的版本所携带的信息将覆盖任何之前的旧版本。

length指示以字节计算的ROI引导信息表的长度，即从下一字段起直到ROI引导信息表最后一个字节的长度。

region\_type指示区域类型字段，指示引导区域的类型，引导区域类型如下表所示。

表 36 region\_type取值

|  |  |
| --- | --- |
| **值** | **描述** |
| 0 | 球面上的视窗的中心点。 |
| 1 | 球面上的视窗。 |

description\_length指示推荐视窗的文本描述长度。

description\_byte指示推荐视窗的文本描述。

guide\_shape\_type指示引导球面区域形状类型字段，指示引导球面区域的形状，guide\_shape\_type 与8.3.1.2节定义的shape\_type具有相同的语义。

guide\_start指示区域引导开始的媒体数据的呈现时间。该字段是NTP格式的UTC时间，长度为32位。

10.2.3 全景视频的字幕在SMT上传输

全景视频的字幕在SMT上的传输参照GB/T 20214118-T-469 信息技术 可关闭字幕附录A定义的扩展。

10.3 容积视频的SMT传输信令

10.3.1 容积视频视角依赖媒体数据包消息

容积视频视角依赖媒体数据包消息用于服务器向客户端指示与用户视角相关的媒体数据包索引信息，以及对应的容积视频空间区域信息。服务器根据客户端发送的当前视角相关状态信息确定视角依赖点云空间区域信息，并选取对应的空间区域媒体数据。以数据包消息方式将确定的发送给客户端。客户端可以根据用户需求处理（如渲染）与用户视角相关的容积视频空间区域信息及对应的数据。

表 37 容积视频视角依赖媒体数据包消息

|  |  |  |  |
| --- | --- | --- | --- |
| **语法** | **值** | **比特数** | **类型** |
| Viewdependent\_packet\_message() {  message\_id  version  length  message\_payload {  packet\_id  num \_regions;  for (i=0; i<num\_regions; i++) {  3DSpatialRegionStruct(1);  }  }  } |  | 16  8  32  16  16 | uimsbf  uimsbf  uimsbf  uimsbf  uimsbf |

message\_id指示容积视频视角依赖媒体数据包消息的标识符。

version指示容积视频视角依赖媒体数据包消息的版本。新的版本所携带的信息将覆盖任何之前的旧版本。

length指示以字节计算的容积视频视角依赖媒体数据包消息的长度，即从下一字段起直到该消息最后一个字节的长度。“0”值在此字段无效。

packet\_id指示与容积视频asset关联的媒体数据包索引信息。此字段将被传递到渲染器，用以指示当前正在接收和播放的容积视频asset的索引，基于该asset的索引，渲染器能够确定与当前asset对应的空间覆盖信息。

num \_regions指示与用户视窗产生重叠的容积视频三维空间区域数量。

3DSpatialRegionsStruct()指示与视角相关的三维空间区域结构。

10.3.2 容积视频资源索引描述符

10.3.2.1 概述

容积视频资源索引描述符采用Volumetric\_media\_descriptor结构，对于容积视频，该资源索引描述符应当位于图集数据码流对应的包含媒体资源描述符的信令消息或信令表中。

当容积视频存在图集分片时，该资源索引描述符指示其和图集分片媒体资源之间的索引关系。

10.3.2.2 语法

表 38 容积视频资源索引描述符语法

| **语法** | **值** | **比特数** | **类型** |
| --- | --- | --- | --- |
| Volumetric\_media\_descriptor () {  descriptor\_tag  descriptor\_length  data\_type  all\_tiles\_slices\_present\_flag  reserved  if (!all\_tiles\_slices\_present\_flag) {  num\_tiles  for (i=0; i<N1; i++) {  tile\_slices \_id  }  }  } | ‘1111111’  N1 | 16  16  8  1  7  16  16 | uimsbf  uimsbf  uimsbf  bslbf  bslbf  uimsbf  bslbf  uimsbf |

10.3.2.3 语义

descriptor\_tag指示标识符，用于表示描述符类型。

descriptor\_length指示标识符的长度，单位为字节。

data\_type 指示容积媒体[[1]](#footnote-1)的媒体资源类型，类型取值详见下表，在基于SMT的容积视频传输中，data\_type取值范围为0x00~0x05。

表 39 data\_type取值

|  |  |
| --- | --- |
| **值** | **描述** |
| 0x00 | 所有容积媒体组件数据 |
| 0x01 | 容积视频图集组件数据 |
| 0x02 | 容积视频占用图组件数据 |
| 0x03 | 容积媒体几何组件数据 |
| 0x04 | 容积媒体属性组件数据 |
| 0x05 | 动态容积媒体时序元数据信息 |
| 0x06-0xFF | 保留 |

all\_tiles\_slices\_present\_flag指示当前媒体资源是否包含容积媒体的所有分片，取值为1时表示当前媒体资源包含了图集组件的所有分片。取值为0时表示当前媒体资源仅包含一部分图集分片。

num tiles\_slices 指示当前媒体资源包含的容积媒体分片数目，即图集分片数量。

tile slice\_id 指示容积媒体分片的标识符，即图集分片的标识符。

10.3.3 媒体资源组消息

10.3.3.1 概述

当通过SMT发送容积视频内容时，应使用媒体资源组消息向接收实体提供与容积视频内容相关联的资源信息。该消息还可用于指示这些资源中哪些资源正在传输至接收实体。

10.3.3.2 语法

表 40 媒体资源组消息语法

| **语法** | **值** | **比特数** | **类型** |
| --- | --- | --- | --- |
| Asset\_group\_message () {  message\_id  version  length  num\_asset\_groups  start\_time  for (i=0; i<N1; i++) {  asset\_group\_id  pending\_flag  3D\_spatial\_region\_info\_flag  reserved  if (3D\_spatial\_region\_info\_flag){  num\_regions  for (j=0; j<N2; j++) {  3DSpatialRegionStruct(1)  }  }  }  } | N1  ‘111111’  N2 | 16  8  16  8  16  8  1  1  6  16 | uimsbf  uimsbf  uimsbf  uimsbf  bslbf  bslbf  bslbf  uimsbf |

10.3.3.3 语义

message\_id 指示媒体资源组消息的标识符。

version指示媒体资源组消息的版本。

length指示媒体资源组消息的长度，以字节为单位，长度计算从下一个字段起直至消息结尾。该字段取值不能为0。

num\_asset\_groups指示媒体资源组的数目，每一个媒体资源组中的媒体资源都对应一个媒体组件。

start\_time指示容积视频内容的起始呈现时间，该起始呈现时间对本消息内所有的媒体资源均适用。

asset\_group\_id指示容积视频媒体资源组标识符，该标识符定义在Asset关系信息描述符中。

pending\_flag指示是否媒体资源组中所有的数据组件均可被渲染。该字段置为1时，所有数据均可被渲染，否则该字段置为0。

3D\_spatial\_region\_info\_flag指示是否携带三维空间区域信息。该字段置为1时，表示消息中携带三维空间信息，否则该字段置为0。

num\_regions指示三维空间区域的数量。

3DSpatialRegionStruct(1)指示该媒体资源组对应的空间区域的信息。

10.3.4 Asset关系信息描述符

10.3.4.1概述

SMT中定义了Asset关系信息描述符，用于指示同一个SMT数据包中资源间的关联关系。在容积视频多个组件轨道具备联合质量等级时，通过combine\_quality\_flag定义新的联合质量等级关系，对相应的联合质量等级进行指示。”

10.3.4.2语法

表 41展示了Asset\_relationship\_information\_descriptor()的语法。

表 41 Asset关系信息描述符语法

| **语法** | **值** | **比特位数** | **备注** |
| --- | --- | --- | --- |
| Asset\_relationship\_information\_descriptor() {  descriptor\_tag  descriptor\_length  library\_flag  reserved  combine\_qaulity\_flag  dependency\_flag  composition\_flag  equivalence\_flag  similarity\_flag  if(dependency\_flag) {  num\_dependencies  for(i = 0; i <N1; i++) {  asset\_id()  }  }  if(composition\_flag) {  asset\_group\_id\_flag  if(asset\_group\_id\_flag){  asset\_group\_id  }  num\_compositions  for(i = 0; i <N2; i++) {  asset\_id() | ‘11’  N1  N2 | 16  16  1  2  1  1  1  1  1  8  1  8  8 | uimsbf  uimsbf  blsbf  blsbf  blsbf  blsbf  blsbf  blsbf  blsbf  uimsbf  blsbf  uimsbf  uimsbf |
| }  }  if(equivalence\_flag) {  equivalence\_selection\_level  num\_equivalences  for(i = 0; i <N3; i++) {  asset\_id()  equivalence\_selection\_level  }  }  if(similarity\_flag) {  similarity\_selection\_level  num\_similarities  for(i = 0; i <N4; i++) {  asset\_id()  similarity\_selection\_level  }  }  if(combine\_quality\_flag) {  combine\_quality\_ranking  num\_combine\_assets  for(i=0; i<N5;i++){  asset\_id()  }  }  if(library\_flag){  num\_libraries  for(i = 0; i <N6; i++) {  asset\_id()  }  }  } | N3  N4  N5  N6 | 8  8  8  8  8  8  8  8  8 | uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  unimsbf |

10.3.4.3 语义

descriptor\_tag 该字段为16位。用于指示此类型描述符的标签值。

descriptor\_length 该字段为16位。指示此描述符的字节长度，从下一个字段计算至最后一个字段。

library\_flag：表示是否需要添加非对齐时间段的知识位流依赖关系。值为‘0’表示不需要添加，值为‘1’表示需要添加。

combine\_quality\_flag 该字段为1位。取值为1时表示具备组合关系的多个Asset整体具备一个联合质量等级；取值为0时表示具备组合关系的多个Asset不存在联合质量等级。

dependency\_flag 该字段为1位。指示在此描述符中是否需要添加依赖关系。值‘0’意味着不需要添加。

composition\_flag 该字段为1位。指示在此描述符中是否需要添加组合关系。值‘0’意味着不需要添加。

equivalence\_flag 该字段为1位。指示在此描述符中是否需要添加等同关系。值‘0’意味着不需要添加。

similarity\_flag 该字段为1位。指示在此描述符中是否需要添加相似关系。值‘0’意味着不需要添加。

num\_dependencies 该字段为8位。指示此描述符所描述的Asset所依赖的Asset的数目。

asset\_group\_id\_flag 指示组合关系中是否包含asset\_group\_id标识符。

asset\_group\_id 指示该组合关系的标识符。

num\_compositions 该字段为8位。指示与此描述符所描述的Asset有组合关系的Asset的数目。

equivalence\_selection\_level 该字段为8位。指示所对应的Asset在等同关系组中的呈现等级。‘0’值表示该Asset被默认呈现。当默认Asset无法被选择时，拥有呈现等级较小的Asset会作为替代被选择和呈现。

num\_equivalences 该字段为8位。指示与此描述符所描述的Asset有等同关系的Asset的数目。

similarity\_selection\_level 该字段为8位。指示所对应的Asset在相似关系组中的呈现等级。‘0’值表示该Asset被默认呈现。当默认Asset无法被选择时，拥有呈现等级较小的Asset会作为替代被选择和呈现。

num\_similarities 该字段为8位。指示与此描述符所描述的Asset有相似关系的Asset的数目。

combine\_quality\_ranking 指示多个Asset整体的联合质量等级。拥有联合质量等级较小的Asset整体具有更好的呈现质量。

num\_combine\_assets 指示与此描述符所描述的Asset有联合质量等级关系的Asset的数目。

num\_libraries：指示此描述符所描述的Asset所依赖的非对齐时间段的知识位流Asset的数目。

asset\_id

该字段指示Asset的标识符， Asset关系信息描述符中的asset\_id：

——当Asset关系信息描述符用于指示依赖关系时，asset\_id字段指示此描述符所描述的Asset所依赖的Asset的标识符，同时，此描述符中提供的Asset 标识符顺序与其内部编码依赖层次相对应；

——当Asset关系信息描述符用于指示组合关系时，asset\_id字段指示与此描述符所描述的Asset有组合关系的Asset的标识符；

——当Asset关系信息描述符用于指示等同关系时，asset\_id字段指示与此描述符所描述的Asset有等同关系的Asset的标识符；

——当Asset关系信息描述符用于指示相似关系时，asset\_id字段指示与此描述符所描述的Asset有相似关系的Asset的标识符；

——当Asset关系信息描述符用于指示联合质量关系时，asset\_id字段指示与此描述符所描述的Asset有联合质量关系关系的Asset的标识符；

——当Asset关系信息描述符用于指示非对齐时间段的知识位流依赖关系时，asset\_id字段指示与此描述符所描述的Asset有非对齐时间段的知识位流依赖关系的Asset的标识符；

该字段位数及具体用法参见 GB/T 33475-6。

10.4 自由视角视频的SMT传输信令

10.4.1 自由视角相机描述符

10.4.1.1 概述

自由视角相机描述符用于描述自由视角视频的相机相关信息，一个媒体资源可具备一个或多个自由视角相机描述符。

10.4.1.2 语法

表 42 自由视角相机描述符语法

|  |  |  |  |
| --- | --- | --- | --- |
| **语法** | **值** | **比特数** | **类型** |
| Avs\_freeview\_camera\_descriptor() {  descriptor\_tag  descriptor\_length  camera\_id  camera\_pos\_x  camera\_pos\_y  camera\_pos\_z  virtual\_flag  reserved  if(virtual\_flag){  num\_related\_camera  for(i = 0; i <N1; i++) {  related\_camera\_id  }  }  } | ‘1111111’  N1 | 16  16  16  32  32  32  1  7  8  16 | uimsbf  uimsbf  uimsbf  float  float  float  blsbf  blsbf  uimsbf  uimsbf |

10.4.1.2 语义

camera\_id视角对应的拍摄相机标识符。

camera\_pos\_x，camera\_pos\_y，camera\_pos\_z分别指示拍摄相机位置的x，y，z坐标分量值。

virtual\_flag取值为1时表示视角对应相机为虚拟相机。取值为0时表示视角对应的相机为原始相机。缺省时表示视角对应的相机为原始相机。

num\_related\_camera指示虚拟相机关联的物理相机的数量。

related\_camera\_id指示虚拟相机关联的物理相机的标识符。

10.4.2 自由视角媒体资源关联

当自由视角视频的纹理图信息和深度图信息分别位于不同的媒体资源中时，包含深度图信息的媒体资源应使用SMT信令资源Asset关系信息描述符（参考10.3.4章节）中的依赖关系关联至相应纹理图信息的媒体资源。若自由视角视频的媒体资源中不存在依赖关系，则该媒体资源为包含纹理信息的媒体资源。

当自由视角视频的不同相机拍摄的纹理和深度信息分别位于不同的媒体资源中时，应使用SMT信令Asset关系信息描述符中的组合关系进行关联。

10.5 沉浸媒体交互反馈信息

10.5.1 全景视频反馈消息结构体定义

10.5.1.1 用户实时位置

用户实时位置结构体表示用户在虚拟场景中相对起始位置的位移，在3DoF时该结构中所有值为0，在3DoF+时为非0值且取值范围应在约束范围内。

aligned(8) class ClientPosition () {

signed int(16) position\_x;

signed int(16) position\_y;

signed int(16) position\_z;

}

position\_x指示用户实时位置相对起始位置沿着x轴位移，取值范围为（-215，215 − 1），以毫米为单位。

position\_y指示用户实时位置相对起始位置沿着y轴位移，取值范围为（-215，215 − 1），以毫米为单位。

position\_z指示用户实时位置相对起始位置沿着z轴位移，取值范围为（-215，215 − 1），以毫米为单位。

10.5.1.2 视角方向

视角方向结构体表示用户实时的视角相对初始视角的变化。

aligned(8) class ClientRotation () {

unsigned int(8) 3D\_rotation\_type;

if(3D\_rotation\_type ==0){

signed int(32) rotation\_yaw;

signed int(32) rotation\_pitch;

signed int(32) rotation\_roll;

}

if(3D\_rotation\_type ==1){

float(32) rotation\_x;

float(32) rotation\_y;

float(32) rotation\_z;

float(32) rotation\_w;

}

}

3D\_rotation\_type 指示旋转信息的表示类型。该字段取值为0表示旋转信息以欧拉角的形式给出；该字段取值为1表示旋转信息以四元数的形式给出。其余取值保留。

rotation\_yaw指示用户实时的视角相对初始视角的沿着x轴的偏航角度，取值范围为（−180 \* 216，180 \*216 – 1）。

rotation\_pitch指示用户实时的视角相对初始视角的沿着y轴的俯仰角度，取值范围为（−90 \* 216， 90 \* 216-1）。

rotation\_roll指示用户实时的视角相对初始视角的沿着z轴的翻滚角度，取值范围为（−180 \* 216，180 \*216 – 1）。

rotation\_x, rotation\_y, rotation\_z and rotation\_w 分别指示四元数x，y，z和w分量的取值，表示用户实时的视角相对初始视角的旋转信息。

10.5.1.3 视窗信息

视窗信息结构体表示客户端视窗的尺寸和屏幕分辨率。

aligned(8) class ClientRegion (){

signed int(32) Region\_width\_angle;

signed int(32) Region\_height\_angle;

unsigned int(16) Region\_width\_resolution;

unsigned int(16) Region\_height­\_resolution;

}

region\_width\_angle指示客户端视窗在横向的张角，精度为2−16度，取值范围为（−90 \* 216， 90 \* 216-1）。

region\_height\_angle指示客户端视窗在纵向的张角，精度为2−16度，取值范围为（−90 \* 216， 90 \* 216-1）。

region\_width\_resolution指示客户端视窗横向的分辨率，取值范围为（0，216 ）。

region\_height\_resolution指示客户端视窗在纵向的分辨率，取值范围为（0，216 ）。

10.5.2 容积视频反馈消息结构体定义

10.5.2.1 用户实时观看方向

用户实时观看方向结构体表示用户在以当前位置建立的笛卡尔坐标系中的观看方向。

aligned(8) class ClientOrientation () {

signed int(32) dirx;

signed int(32) diry;

signed int(32) diry;

}

Dirx指示用户观看方向在x轴上的坐标，以用户所在位置为原点建立笛卡尔坐标系。

Diry指示用户观看方向在y轴上的坐标，以用户所在位置为原点建立笛卡尔坐标系。

Dirz指示用户观看方向在z轴上的坐标，以用户所在位置为原点建立笛卡尔坐标系。

10.5.3 交互反馈消息

10.5.3.1 定义

交互反馈消息（Interaction Feedback Message）提供沉浸式媒体消费时，服务器与客户端之间的交互反馈。当沉浸式媒体消费中的服务器与客户端之间需要发送交互反馈信息时，使用此消息进行会话。一个交互反馈消息信令中可包含一个或多个交互反馈信令表。

10.5.3.2 语法

表 43 交互反馈消息语法

| **语法** | **值** | **比特数** | **备注** |
| --- | --- | --- | --- |
| interaction\_feedback\_message() {  message\_id  version  length  extension{  number\_of\_tables  for(i-0;i<N1;i++){  table\_id;  table\_version;  table\_length  }  message\_payload {  for (i=0; i<N1; i++) {  table() }  }  }  } | N1 | 16  8  32  8  8  8  16 | uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  uimsbf |

10.5.3.3 语义

message\_id指示交互反馈消息的标识符。

version指示交互反馈消息的版本。新的版本所携带的信息将覆盖任何之前的旧版本。

length指示了以字节计算的交互反馈消息的长度，即从下一字段起直到交互反馈消息最后一个字节的长度。“0”值在此字段无效。

number\_of\_tables指示交互反馈消息中包含的信令表的数量。

table\_id指示交互反馈消息中包含的表识别符。这是表中包含在交互反馈消息的有效负载中的table\_id 字段的一个副本。

table\_version指示交互反馈消息中所包含的表的版本。这是包含在交互反馈消息的有效负载中的表的版本字段的一个副本。

table\_length包含在交互反馈消息的有效负载中的表的长度字段的一个副本。

table()指示一个交互反馈信令表实体。在有效负载中的该表与扩展域中table\_id出现的顺序相同。一个交互反馈信令表可以作为一个table()的实例。

10.5.4 交互反馈信令表

10.5.4.1 定义

交互反馈信令表中包含了服务器和客户端之间交互反馈的信息，不同类型的交互反馈信令表用于指示不同类型的交互反馈信息。

10.5.4.2 语法

表 44 交互反馈信令表语法

| **语法** | **值** | **比特数** | **类型** |
| --- | --- | --- | --- |
| interaction\_feedback\_table() {  table\_id  version  length  table\_payload {  table\_type  timestamp  message\_source  asset\_group\_flag  reserved  if(asset\_group\_flag){  asset\_group\_id  }  else{  asset\_id()  }  if (table\_type == 0) {  3DoF+\_flag  interaction\_target  interaction\_type  if(interaction\_target==1){  ClientRegion()  }  if(interaction\_target==2){  ClientRotation()  }  if(3DoF+\_flag==1 && interaction target ==3){  ClientPosition()  behavior\_coefficient  }  }  if(table\_type == 1){  ClientPosition()  ClientOrientation()  last\_processed\_media\_timestamp  }  if(table\_type == 2){  ClientCam\_pos\_x;  ClientCam\_pos\_y;  ClientCam\_pos\_z;  }  }  } |  | 8  8  16  8  32  1  1  6  8  1  8  8  16  32  32  32 | uimsbf  uimsbf  uimsbf  uimsbf  uimsbf  bool  bool  uimsbf  uimsbf  bool  uimsbf  uimsbf  uimsbf  float  float  float |

10.5.4.3 语义

table\_type指示交互反馈信令表携带的信息类型。其取值含义如下表所示：

表 45 table\_type取值表

|  |  |
| --- | --- |
| **取值** | **描述** |
| 0 | 全景视频用户位置变动信息 |
| 1 | 容积媒体用户位置变动信息 |
| 2 | 自由视角视频用户位置变动信息 |
| 3..255 | 未定义 |

timestamp指示当前交互产生的时间，使用UTC时间。

message\_source指示消息源，0表示交互反馈消息是客户端发往服务器，1表示交互反馈消息是服务器发往客户端。该值此处置0。

asset\_group\_flag取值为1表示客户端当前消费内容属于一个资源组；取值为0表示客户端当前消费内容不属于资源组。

asset\_group\_id指示客户端当前消费内容的资源组标识符，与Asset关系信息描述符中组合关系下的asset\_group\_id对应。

asset\_id指示客户端当前消费内容的asset\_id。

3DoF+\_flag指示3DoF+视频内容。

reserved保留字节位。

interaction\_num指示当前信令中包含的交互的数目。

interaction\_target指示客户端当前交互的目标，包括头盔设备当前状态、用户关注目标、用户当前状态等；交互目标的取值如表43所示。

表 46 interaction\_target取值表

|  |  |  |
| --- | --- | --- |
| **类型** | **取值** | **描述** |
| Null | 0 | 交互目标为空，即没有特定交互目标 |
| HMD\_status | 1 | 交互目标为头盔设备当前状态 |
| Object of interests | 2 | 交互目标为用户关注区域当前状态 |
| User\_status | 3 | 交互目标为用户当前状态 |

interaction\_type指示交互类型，该值此处应置0。

interaction\_content\_length指示客户端对于当前的交互行为，其具体交互内容的长度。该字段取值以字节为单位，为从该字段下一字节起到当前交互内容结束的长度。

ClientRegion指示客户端视窗的尺寸和屏幕分辨率。

ClientRotation指示视角方向，即用户实时的视角相对初始视角的变化。

behavior\_coefficient指示放大行为系数。

ClientPosition()指示用户当前在全局坐标系中的位置，详见10.5.1.1。

ClientOrientation()指示用户在以当前位置建立的笛卡尔坐标系中的观看方向，详见10.5.2.1。

last\_processed\_media\_timestamp指示已加入解码器缓冲区的最后一个媒体单元的时间戳。SMT发送实体使用此字段从容积视频播放器的新asset中确定下一个传输的媒体单元。下一个媒体单元是紧随该时间戳后的带有时间戳或序号的媒体单元。SMT发送实体从随后的媒体时间戳开始，从传输先前的asset（根据先前的视窗确定）切换到传输新的asset（根据新的视窗确定），以减少接收对应于新视窗媒体内容的延迟。

ClientCam\_pos\_x、ClientCam\_pos\_y、ClientCam\_pos\_z分别指示目标视角对应的虚拟相机的x轴、y轴，z轴分量。

（资料性）  
沉浸媒体介绍

虚拟现实对现实世界中视觉和听觉的感官刺激的渲染并呈现给用户。用户通过头戴设备（但不限于），实现模拟的视觉组件来替代用户的视线区域，并佩戴耳机为用户提供相应的音频。用户从一个三维空间的显示区域开始向周围观看，同时根据视窗得到关联的音频。

1） 三自由度

VR用户在一个三维空间的中心点固定，用户头部沿着X,Y,Z轴旋转观看周围的景象。

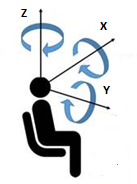
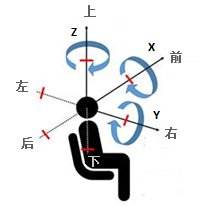


图 A.1三自由度示意图

2） 增强的三自由度

当虚拟的场景具有一定的深度信息，VR用户头部可以基于三自由度在一个有限的空间内移动，称为3DOF+。



图A.2三自由度+示意图

3） 六自由度

当用户可以在模拟的场景中较自由的移动时，称为6DOF。6DOF分为窗口6DOF、全方向6DOF和6DOF。

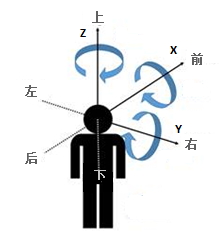


图 A.3六自由度示意图

* 窗口6DOF，用户在X轴、Y轴的旋转移动，和Z轴的平移受限。例如，用户不能够看到窗户框架外的景象，以及穿过窗户。
* 全方向6DOF，用户在X轴、Y轴和Z轴的旋转移动受限（特别是多步行走距离）。例如，用户在受限的移动区域中不能自由的穿过三维的360度VR内容。
* 6DOF，用户基于3DOF，可以沿着X,Y,Z轴的自由平移。例如，用户再三维的360度VR内容中自由的走动。

（资料性）  
AVS-I客户端参考模型

B.1 基于沉浸媒体的客户端参考模型

本章节提出了基于沉浸媒体应用的客户端参考模型，见图B.1，主要规定了客户端各功能组件。



图B.1 AVS-I客户端参考模型

用户终端通过远端服务器推荐或用户自己需求的方式对媒体文件选择，从远端服务器下载或接收远端服务器推送的媒体文件，经过并由解析器、解码器、转化/合成器、渲染器等一系列组件进行处理后，实现虚拟现实媒体内容的显示。同时，用户终端可以依据用户需求进行远程渲染。

解析器：解析器提供对媒体文件或分片的处理，提取基本流以及解析元数据，解析出的元数据用于渲染。解析器可依据用户动作进行动态的信息处理（如用户头动、位置的跟踪信息），如动态选择下载的媒体分片。

解码器：解码器用于解码解析器提供的媒体流，并将解码流输出到转换器。

转换器：转换器根据解析器提供的元数据，将解码后的媒体转换为球形三维视频。例如3DoF时将平面图像映射为球形，在基于映射、投影的6DoF处理时将二维信息流重建成三维数据。如果有必要，可使用解析器解析的转换元数据。

渲染器：渲染器使用解码的信令、渲染元数据、以及视窗的信息（或者考虑其他的可能的信息）对视频/音频进行渲染。3DoF和3DoF+主要基于当前视点、视差、深度信息等对球形媒体内容进行渲染，6DoF对当前视点对视窗内的三维媒体内容进行渲染。

传感装置：传感装置依据用户的移动获取当前视窗的方向以及用户的位置信息，并反馈给用户终端解析器。用户终端可依据视窗、视窗的方向以及用户的位置信息选择下载适当的媒体，或者解析器依据视窗、用户位置信息选择适当的媒体文件。

远程渲染平台：远程渲染平台部署在远端服务器，依据用户终端反馈的视窗、视窗的方向以及用户的位置信息或者媒体文件中的渲染元数据进行渲染，用户终端依据远程渲染平台的渲染媒体直接显示。

1. 容积媒体包含容积视频以及几何编码点云 [↑](#footnote-ref-1)