

GY

中华人民共和国广播电视和网络视听行业标准

GY/T 420—2025

超高清音视频编码码流网络传输封装技术 要求

Technical requirements of ultra-high-definition video and audio encoding streams
encapsulation for network transmission

2025 - 04 - 03 发布

2025 - 04 - 03 实施

国家广播电视总局 发布

目 次

前言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 约定	3
5.1 助记符	3
5.2 保留	3
6 通则	3
7 超高清音视频编码码流在 TS 中的封装要求	3
7.1 概述	4
7.2 AVS2 视频编码码流在 TS 中的封装要求	4
7.3 AVS3 视频编码码流在 TS 中的封装要求	5
7.4 Audio Vivid 音频编码码流在 TS 中的封装要求	12
8 超高清音视频编码码流在 SMT 中的封装要求	18
8.1 概述	18
8.2 AVS2 视频编码码流在 SMT 中的封装要求	18
8.3 AVS3 视频编码码流在 SMT 中的封装要求	19
8.4 Audio Vivid 音频编码码流在 SMT 中的封装要求	25
附录 A (规范性) 超高清音视频编码码流在 ISOBMFF 文件中的封装要求	28
A.1 概述	28
A.2 AVS2 视频编码码流在 ISOBMFF 文件中的封装要求	28
A.3 AVS3 视频编码码流在 ISOBMFF 文件中的封装要求	32
A.4 Audio Vivid 音频编码码流在 ISOBMFF 文件中的封装要求	40
附录 B (资料性) 超高清音视频编码码流在 HLS 中的封装	45
附录 C (资料性) 超高清音视频编码码流在 RTMP 中的封装	46
C.1 概述	46
C.2 AVS2 视频编码码流在 FLV 中的封装	46
C.3 AVS3 视频编码码流在 FLV 中的封装	46
C.4 Audio Vivid 音频编码码流在 FLV 中的封装	47
附录 D (资料性) 超高清音视频编码码流在 DASH 中的封装	49
D.1 概述	49
D.2 AVS3 视频编码码流在 DASH 中的封装	49

D.3 Audio Vivid 音频编码码流在 DASH 中的封装.....	52
附录 E (资料性) MIME 类型的 ‘codecs’ 参数	54
E.1 概述	54
E.2 AVS2 视频编码参数.....	54
E.3 AVS3 视频编码参数.....	54
E.4 Audio Vivid 音频编码参数.....	54
参考文献	55

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。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国广播电视和网络视听标准化技术委员会（SAC/TC 239）归口。

本文件起草单位：国家广播电视总局广播电视科学研究院、杭州当虹科技股份有限公司、鹏城实验室、中兴通讯股份有限公司、华为技术有限公司、上海交通大学、中央广播电视总台、北京广播电视台、广东广播电视台、湖南广播电视台、广东南方新媒体股份有限公司、国家广播电视总局广播电视规划院、咪咕文化科技有限公司、浙江大学、上海海思技术有限公司、北京数码视讯软件技术发展有限公司、广东博华超高清创新中心有限公司、央视频融媒体发展有限公司、湖南快乐阳光互动娱乐传媒有限公司、北京爱奇艺科技有限公司、优酷网络技术（北京）有限公司、杭州微帧信息科技有限公司、四川传媒学院、四川新视创伟超高清科技有限公司、北京中视广信科技有限公司、北京流金岁月传媒科技股份有限公司、寰宇信任（北京）技术有限公司、北京全景声信息科技有限公司。

本文件主要起草人员：周芸、郭晓强、王雪辉、郑建铎、黄成、刘帅、徐异凌、潘晓菲、胡潇、王东飞、付光涛、毕江、曾志群、方林、李小雨、黎政、覃毅力、刘文翰、王兵、陈志业、焦健波、庞超、王琦、于化龙、邹志铭、周骋、张伟民、李日、胡颖、高原、王立冬、朱易、管云峰、王振中、李康敬、袁乐、严玄、孙剑、周骋、罗准、雷威、梅大为、朱政、左旭舟、冉峡、宋小民、卢剑平、宁黎、赵鹏、许舒敏、陈丽丽、李凯东、温竹、周士琪、汪喜。

超高清音视频编码码流网络传输封装技术要求

1 范围

本文件规定了以GY/T 299.1—2016（简称“AVS2”）、GY/T 368—2023（简称“AVS3”）、GY/T 363—2023（简称“Audio Vivid”）等方式编码的超高清音视频编码流在传输流（TS）、智能媒体传输（SMT）等网络传输协议中的封装技术要求。

本文件适用于广播电视和网络视听超高清音视频编码传输系统开发、生产、应用、测试和维护。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；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 17975.1—2010 信息技术 运动图像及其伴音信息的通用编码 第1部分：系统

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

GY/T 299.1—2016 高效音视频编码 第1部分：视频

GY/T 363—2023 三维声编解码及渲染

GY/T 368—2023 先进高效视频编码

ISO/IEC 14496-12:2022 信息技术 音视频对象编码 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)

3 术语和定义

下列术语和定义适用于本文件。

3.1

节目流 program stream

由具有共同时间基准的一个或多个PES分组包合并而成的单一流。

3.2

位流 bitstream

编码图像所形成的二进制数据流。

[来源：GY/T 368—2023, 3.74]

3.3

主位流 sequence stream

由该位流以外的知识位流提供的知识图像进行解码的位流。

[来源：GY/T 368—2023, 3.102]

3.4

独立主位流 independent sequence stream

在不参考该位流以外的知识位流提供的知识图像的情况下可以完全解码的位流。

3.5

非独立主位流 dependent sequence stream

使用了该位流以外的知识位流提供的知识图像才能解码的位流。

3.6

片段 segment

媒体呈现描述中的HTTP统一资源定位符引用的媒体数据单元。

3.7

知识图像 library picture

知识位流中的图像，可以被其他位流中的图像参考。

[来源：GY/T 368—2023, 3.100, 有修改]

3.8

知识位流 library stream

包含知识图像的位流。

[来源：GY/T 368—2023, 3.101]

4 缩略语

下列缩略语适用于本文件。

AATF AVS音频传输格式 (AVS Audio Transport Format)

AU 访问单元 (Access Unit)

BBV 位流缓冲区校验 (Bitstream Buffer Verifier)

bslbf 位串, 左位在先 (bit string, left bit first)

CEU 通用封装单元 (Common Encapsulation Unit)

CPB 编码图像缓冲区 (Coded Picture Buffer)

DASH 基于HTTP的动态自适应流 (Dynamic Adaptive Streaming over HTTP)

DPB 解码图像缓冲区 (Decoded Picture Buffer)

DTS 解码时间戳 (Decoding Time Stamp)

FIFO 先进先出 (First Input First Output)

HLS 基于HTTP的实时流媒体 (HTTP Live Streaming)

HOA 高阶立体声场 (High Order Ambisonics)

LBM 知识位流数据缓存模型 (Library Buffer Model)

LBMF 知识位流数据缓存模型反馈 (Library Buffer Model Feedback)

LFU 最不常使用 (Least Frequently Used)

LRAP 知识随机访问点 (Library Random Access Point)

LRU 最近最少使用 (Least Recently Used)

MFU 媒体最小分割单元 (Media Fragment Unit)

MIME 多用途互联网邮件扩展 (Multipurpose Internet Mail Extensions)

MP SMT包 (SMT Packet)

MPD 媒体呈现描述 (Media Presentation Description)

PES	分组的基本流 (Packetized Elementary Streams)
PMT	节目映射表 (Program Map Table)
PTS	显示时间戳 (Presentation Time Stamp)
RL	参考层 (Reference Layer)
RTMP	实时消息协议 (Real-Time Messaging Protocol)
SAP	流访问点 (Stream Access Point)
SMT	智能媒体传输 (Smart Media Transport)
STD	系统目标解码器 (System Target Decoder)
TS	传输流 (Transport Stream)
URI	统一资源标识符 (Uniform Resource Identifier)
URL	统一资源定位符 (Uniform Resource Locator)
UUID	通用唯一标识符 (Universally Unique Identifier)
uimsbf	无符号整数, 最高有效位在前 (unsigned integer, most significant bit first)
XML	可扩展标记语言 (eXtensible Markup Language)

5 约定

5.1 助记符

助记符定义见表1。

表1 助记符定义

助记符	定义
bslbf	位串, 左位在前。位串是带单引号的1和0串。如‘1000 0001’。位串内的空格是便于阅读的, 无特殊意义
uimsbf	无符号整数, 最高有效位优先
UI8	无符号8位整数
SI24	有符号24位整数
UB	无符号位

5.2 保留

“保留”(reserved)定义了一些特定语法元素值用于将来对本文件的扩展。除非本文件另有规定, 否则所有保留位应设置为‘1’。码流解析处理应忽略这些位。

6 通则

超高清AVS2/AVS3视频和Audio Vivid音频编码码流在TS中的封装应符合第7章的规定; 在SMT中的封装应符合第8章的规定, 对应ISOBMFF文件格式应符合附录A的规定; 在HLS、RTMP、DASH中的封装见附录B~附录E。

7 超高清音视频编码码流在 TS 中的封装要求

7.1 概述

基于GB/T 17975.1—2010，对超高清音视频编码码流在TS中的封装进行扩展。

7.2 AVS2 视频编码码流在 TS 中的封装要求

7.2.1 流标识

AVS2视频编码码流对应的stream_id和stream_type赋值应符合表2的规定。

表2 stream_id 和 stream_type 赋值

字段名称	赋值
stream_id	1110 xxxx ^a
stream_type	0xD2 ^b
^a 符号 x 表示 ‘0’ 或 ‘1’ 均被允许且可产生相同的流类型，流号码由 x 的取值决定。 ^b 应配有 descriptor_tag 为 64 (0x40) 的 AVS2_video_descriptor。	

7.2.2 注册描述符

PMT 中应包含 registration_descriptor()。registration_descriptor() 的定义见 GB/T 17975.1—2010，其中，registration_descriptor() 的 Tag 值应为 5(0x05)，AVS2 视频的 format_identifier 应为 0x41-56-53-56(‘AVSV’)。

7.2.3 AVS2 视频流描述符

AVS2视频流描述符定义应符合表3的规定。

表3 AVS2 视频流描述符定义

AVS2视频流描述符定义	位数	助记符
AVS2_video_descriptor () {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimbsf
profile_id	8	uimbsf
level_id	8	uimbsf
extension_layer_number	8	uimbsf
multiple_frame_rate_flag	1	bslbf
frame_rate_code	4	uimbsf
AVS_still_present	1	bslbf
chroma_format	2	uimbsf
sample_precision	3	uimbsf
reserved	5	bslbf
}		

描述符标签 descriptor_tag

8位无符号整数。值为64 (0x40)，表示该描述符为AVS2视频流描述符。

描述符长度 descriptor_length

8位无符号整数。表示紧跟在该字段之后的描述符的字节数。

类标识 profile_id

8位无符号整数。表示AVS2视频编码码流的类，该字段与GY/T 299.1—2016中profile_id字段相同。

级标识 level_id

8位无符号整数。表示AVS2视频编码码流的级，该字段与GY/T 299.1—2016中level_id字段相同。

扩展层数 extension_layer_number

8位无符号整数。表示比特流的扩展层数，该字段为0。

复合帧率标志 multiple_frame_rate_flag

二值变量。值为‘1’表示AVS2视频编码码流中可能有多个帧率，值为‘0’表示只有单一帧率。

帧率代码 frame_rate_code

4位无符号整数。该字段与GY/T 299.1—2016中frame_rate_code字段类似，不同点在于multiple_frame_rate_flag值为‘1’时，一个特定的帧率意味着视频流中允许有某些其他的帧率，具体应符合表4的规定。

表4 帧率代码

frame_rate_code	帧率	multiple_frame_rate_flag字段置‘1’时，允许的其他帧率
0000	禁止	—
0001	23.976	—
0010	24	23.976
0011	25	—
0100	29.97	23.976
0101	30	23.976、24、29.97
0110	50	25
0111	59.94	23.976、29.97
1000	60	23.976、24、29.97、30、59.94
1001	100	50
1010	120	59.94、60、119.88、120
1011~1111	保留	—

AVS静态图像标识 AVS_still_present

二值变量。值为‘1’表示该视频编码码流只含静态图像数据；值为‘0’表示可包含运动的或静态的图像数据。

色度格式 chroma_format

2位无符号整数。表示色度分量的格式，该字段与GY/T 299.1—2016中chroma_format字段相同。

采样精度 sample_precision

3位无符号整数。表示亮度和色度样本的精度，该字段与GY/T 299.1—2016中sample_precision字段相同。

7.3 AVS3 视频编码码流在 TS 中的封装要求

7.3.1 基本要求

AVS3视频编码码流约束如下。

- a) AVS3 视频编码码流应是 GB/T 17975.1—2010 中节目的一个节目元素，基本流的 stream_type 字段值应等于 ‘0xD4’。
- b) AVS3 视频编码码流的序列头应存在于承载该视频流的基本流中。
- c) 在 stream_type 等于 ‘0xD4’ 且具有 GB/T 17975.1—2010 中层次描述符的基本流中，层次描述符中的 hierarchy_type 字段值应等于 ‘3’（时域可扩展性）。
- d) AVS3 视频编码码流的最高“级”以及所符合的“类”等编码参数应使用 AVS3 视频流描述符标识。如果存在与 AVS3 视频编码码流相关联的 AVS3 视频流描述符，则该描述符应包含在节目映射表中对应于基本流条目的描述符循环中。

如果 AVS3 视频编码码流包含主位流及其依赖的知识位流，约束如下：

- a) 当 AVS3 视频编码码流被封装为传输流时，主位流及其依赖的知识位流应被封装在同一个传输流中；
- b) 当 AVS3 视频编码码流被封装为节目流时，主位流及其依赖的知识位流应被封装在同一个节目流中。

7.3.2 PES 分组

7.3.2.1 流标识

AVS3 视频编码码流应作为 PES_packet_data_bytes 携带在 PES 分组数据包中，并通过节目映射表中分配的 stream_type 字段值 (0xD4) 标识。

以下约束适用于 AVS3 视频的 PES 数据包使用的 stream_id：

- a) PES 分组中的 stream_id 取值 ‘1111 1101’ (extended_stream_id)，表示 PES 分组数据包采用扩展语法标识 AVS3 视频流；
- b) PES 分组包头中 stream_id_extension_flag 取值 ‘0’，stream_id_extension 的取值范围应从 ‘0x41’ 到 ‘0x4F’，stream_id_extension 字段的取值设置为 ‘0x41’ 用于表示 AVS3 视频主位流，该字段取值设置为 ‘0x42’ 用于表示 AVS3 视频知识位流。

具体应符合表 5 的规定。

表5 AVS3 视频编码码流在 PES 分组中的流标识

字段名称	值
stream_id	1111 1101
stream_id_extension_flag	0
stream_id_extension	0x41 ^a
	0x42 ^b
stream_type	0xD4 ^c
^a 表示 AVS3 视频编码主位流。 ^b 表示 AVS3 视频编码知识位流。 ^c 应配有 descriptor_tag 为 209 (0xD1) 的 AVS3_video_descriptor。	

7.3.2.2 数据对齐指示

AVS3 视频编码码流的 PES 分组使用 data_alignment_indicator 和 7.3.3.3 中定义的数据流对齐描述符 data_stream_alignment_descriptor 来标识 PES 分组包头后数据的对齐方式：

- a) 如果 `data_alignment_indicator` 值为 ‘1’，并且存在 7.3.3.3 中定义的数据流对齐描述符 `data_stream_alignment_descriptor`，表示 PES 分组包头之后紧跟着该数据流对齐描述符所指出的视频起始码；
- b) 如果 `data_alignment_indicator` 值为 ‘1’ 且不存在 7.3.3.3 中定义的数据流对齐描述符 `data_stream_alignment_descriptor`，则使用本文件表 8 中 `alignment_type` 取值 ‘01’ 时所指示的对齐方式；
- c) 当 `data_alignment_indicator` 值为 ‘0’ 时，表示没有定义是否有任何对齐方式。

7.3.2.3 解码时间约束

对于知识位流，PES分组包的包头中的PTS_DTS_flags字段设置为 ‘11’，此时知识位流PES分组包的包头中携带的PTS的值应被忽略，解码后的知识图像应不被用于显示。

知识位流的PES分组包的包头中携带的DTS，应小于或等于AVS3视频主位流中依赖该PES分组包的所有PES分组包中携带的最小的DTS。

7.3.3 节目和节目元素描述符

7.3.3.1 注册描述符

PMT中应包含`registration_descriptor()`。`registration_descriptor()`的定义见GB/T 17975.1—2010，其中，`registration_descriptor()`的Tag值应为5(0x05)，AVS3视频的`format_identifier`应为0x41-56-53-56(‘AVSV’)。

7.3.3.2 AVS3 视频流描述符

AVS3视频流描述符语法应符合表6的规定。该描述符给出了基本信息（例如“类”和“级”）以标识AVS3视频流的编码参数。对于AVS3视频时域分层子码流，AVS3视频流描述符提供在其所适用的基本流中的AVS3视频最高时域层级的信息。

表6 AVS3 视频流描述符定义

AVS3视频流描述符定义	位数	助记符
<code>AVS3_video_descriptor () {</code>		
<code> descriptor_tag</code>	8	<code>uimsbf</code>
<code> descriptor_length</code>	8	<code>uimsbf</code>
<code> profile_id</code>	8	<code>uimsbf</code>
<code> level_id</code>	8	<code>uimsbf</code>
<code> multiple_frame_rate_flag</code>	1	<code>bslbf</code>
<code> frame_rate_code</code>	4	<code>uimsbf</code>
<code> sample_precision</code>	3	<code>uimsbf</code>
<code> chroma_format</code>	2	<code>uimsbf</code>
<code> temporal_id_flag</code>	1	<code>bslbf</code>
<code> td_mode_flag</code>	1	<code>bslbf</code>
<code> library_stream_flag</code>	1	<code>uimsbf</code>
<code> library_picture_enable_flag</code>	1	<code>uimsbf</code>
<code> reserved</code>	2	<code>bslbf</code>

表 6 (续)

AVS3视频流描述符定义	位数	助记符
<code>colour primaries</code>	8	<code>uimsbf</code>
<code>transfer characteristics</code>	8	<code>uimsbf</code>
<code>matrix coefficients</code>	8	<code>uimsbf</code>
<code>reserved</code>	8	<code>bslbf</code>
}		

描述符标签 descriptor_tag

8位无符号整数，值为209 (0xD1)。表示该描述符为AVS3视频描述符。

描述符长度 descriptor_length

8位无符号整数。表示紧跟在该字段之后的AVS3视频描述符的字节数。

类标识 profile_id

8位无符号整数。表示AVS3视频编码码流的类，该字段与GY/T 368—2023中`profile_id`字段相同。

级标识 level_id

8位无符号整数。表示AVS3视频编码码流的级，该字段与GY/T 368—2023中`level_id`字段相同。

复合帧速率标志 multiple_frame_rate_flag

二值变量。值为‘1’表示AVS3视频编码码流中可能有多种帧速率，值为‘0’表示只有单一帧速率。

帧率代码 frame_rate_code

4位无符号整数。该字段与GY/T 368—2023中`frame_rate_code`字段类似，不同点在于`multiple_frame_rate_flag`值为‘1’时，一个特定的帧率意味着视频流中允许有某些其他的帧率，具体应符合表7的规定。

表7 帧率代码

<code>frame_rate_code</code>	帧率	<code>multiple_frame_rate_flag</code> 字段置‘1’时，允许的其他帧率
0000	禁止	—
0001	23.976	—
0010	24	23.976
0011	25	—
0100	29.97	23.976
0101	30	23.976、24、29.97
0110	50	25
0111	59.94	23.976、29.97
1000	60	23.976、24、29.97、30、59.94
1001	100	50
1010	120	59.94、60、119.88、120
1011~1111	保留	—

样本精度字段 sample_precision

3位无符号整数。表示亮度和色度样本的精度，该字段与GY/T 368—2023中`sample_precision`字段相同。

色度格式字段 chroma_format

2 位无符号整数。表示色度分量的格式，该字段与 GY/T 368—2023 中 chroma_format 字段相同。

时间层标识允许标志 temporal_id_flag

二值变量。表示视频流是否允许使用时间层标识，该字段与 GY/T 368—2023 中 temporal_id_enable_flag 字段相同。

立体视频模式标志 td_mode_flag

2 位位串。表示视频流是单目视频流或多视点视频流，该字段与 GY/T 368—2023 中 td_mode_flag 字段相同。

知识位流标志 library_stream_flag

二值变量。表示在包含此描述符的节目特定信息的生效范围内的基本流是否为知识位流，该字段与 GY/T 368—2023 中 library_stream_flag 相同。

知识图像允许标志 library_picture_enable_flag

二值变量。表示描述符对应的主位流中是否存在帧间预测图像使用知识图像作为参考图像，该字段与 GY/T 368—2023 中 library_picture_enable_flag 相同。

彩色三基色 colour_primaries

8 位无符号整数。表示 AVS3 视频编码码流中源图像三基色的色度坐标。该字段与 GY/T 368—2023 中 colour_primaries 字段相同。

光电转移特性 transfer_characteristics

8 位无符号整数。表示 AVS3 视频编码码流中源图像的光电转移特性。该字段与 GY/T 368—2023 中 transfer_characteristics 字段相同。

彩色信号转换矩阵 matrix_coefficients

8 位无符号整数。表示从红绿蓝三基色转换为亮度和色度信号时采用的转换矩阵。该字段与 GY/T 368—2023 中 matrix_coefficients 字段相同。

7.3.3.3 数据流对齐描述符

如果 PES 分组包头中的 data_alignment_indicator 的值为‘1’并且存在数据流对齐描述符，则需要由描述符 data_stream_alignment_descriptor 指示对齐类型，对应语义定义如下。

对齐类型字段 alignment_type

该字段是数据流对齐描述符中的 8 位字段，它描述了 AVS3 视频基本流中存在的对齐类型。当 PES 分组包头中的 data_alignment_indicator 为‘1’时，具体对齐类型应符合表 8 的规定。如果 PES 分组包是一个视频流的起始，对齐应发生在第一个序列头的起始码处。

表8 AVS3 视频流对齐类型

对齐类型	描述
00	保留
01	AVS3 视频存取单元
02	AVS3 视频片
03	AVS3 视频存取单元或 AVS3 视频片
04	AVS3 序列
05~255	保留

一个 AVS3 视频存取单元包含一个图像的所有编码数据，以及紧随其后的任意填充数据，直到下一个存取单元的起始位置为止且不包含下一个存取单元。如果一个图像前面没有视频序列起始码（video_sequence_start_code），那么这个图像的存取单元以图像起始码开始。如果一个图像前面有视频序列起始码，那么这个图像的存取单元以这些起始码中的第一个起始码的第一个字节开始。如果一个图像是码流中在视频序列结束码（video_sequence_end_code）之前的最后一个图像，那该编码图像的最后一个字节和视频序列结束码之间的所有字节（包括视频序列结束码）属于该编码图像的存取单元。

7.3.4 DPB 缓存管理

使用本文件携带 AVS3 视频编码码流不影响 DPB 的大小。当在 STD 中解码 AVS3 视频编码码流时，DPB 的大小由 GY/T 368—2023 定义。DPB 应按照 GY/T 368—2023 中 9.2.4 的规定进行管理。在一个 AVS3 视频存取单元被解码之后，即在该 AVS3 视频存取单元从 CPB 中被移除的时刻，该被解码的 AVS3 视频存取单元立即进入 DPB。

一个被解码的 AVS3 主位流的存取单元在 DPB 输出时间所指示的时刻被显示，一个被解码的 AVS3 知识位流的存取单元在其被解码之后立即从 DPB 中被输出。DPB 输出时间是执行 GY/T 368—2023 中 9.2.6 中的图像输出操作的时间所指示的时刻。

如果 AVS3 视频码流不能提供足够信息用于决定 AVS3 主位流的存取单元的 CPB 移除时间和 DPB 输出时间，那么这些时间戳应在 STD 模型中根据 PTS 和 DTS 时间戳被决定，方法如下：

- a) AVS3 主位流的存取单元 n 的 CPB 移除时间是由 DTS(n) 指示的时刻，其中 DTS(n) 是 AVS3 主位流的存取单元 n 的 DTS 值；
- b) AVS3 主位流的存取单元 n 的 DPB 输出时间是由 PTS(n) 指示的时刻，其中 PTS(n) 是 AVS3 主位流的存取单元 n 的 PTS 值。

如果 AVS3 视频码流不能提供足够信息用于决定 AVS3 知识位流的存取单元的 CPB 移除时间，那么该时间戳应在 STD 模型中根据 DTS 时间戳被决定，方法如下：AVS3 知识位流的存取单元 n 的 CPB 移除时间是由 DTS(n) 指示的时刻，其中 DTS(n) 是 AVS3 知识位流存取单元 n 的 DTS 值。

7.3.5 面向 AVS3 的 T-STD 扩展

7.3.5.1 AVS3 传输模型

对于一个符合本文件的节目，当该节目中的码流是 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流时，T-STD 模型见图 1。

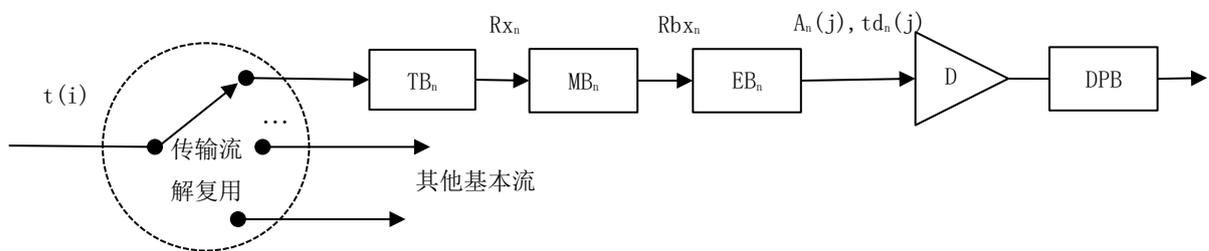


图1 面向 AVS3 的 T-STD 模型扩展

图 1 中下列符号用于描述 T-STD 模型扩展：

- t(i)：表示传输流的第 i 个字节进入系统目标解码器的时间，以秒为单位；
- TB_n：基本流 n 的传输缓存；

- TBS: 传输缓存 TB_n 的大小, 以字节为单位;
- MB_n: 基本流 n 的复用缓存;
- EB_n: AVS3 视频编码码流的基本流缓存;
- j: AVS3 视频编码码流中 AVS3 视频存取单元的索引;
- A_n(j): AVS3 视频编码码流的第 j 个存取单元;
- t_d(j): A_n(j) 在系统目标解码器中的解码时间, 以秒为单位;
- R_{xn}: 从传输缓存 TB_n 到复用缓存 MB_n 的传输速率;
- R_{bxn}: 从复用缓存 MB_n 到基本流缓存 EB_n 的传输速率;
- D: 虚拟 AVS3 视频解码器;
- DPB: 解码图像缓冲区。

7.3.5.2 缓存管理

缓存管理符合下列约束。

- a) 仅有一个传输缓存 TB_n 用于接收 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流, 其 TBS 大小固定为 512 字节。
- b) 仅有一个复用缓存 MB_n 用于接收 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流, 该复用缓存的大小 MBS_n 约束见公式 (1)。

$$MBS_n = BS_{\max} + BS_{oh} + \text{MaxCPB}[\text{level_id}] - \text{BBS} \cdots \cdots (1)$$

式中:

- MBS_n —— 复用缓存, 单位为二进制位 (bit);
- BS_{oh} —— 包头缓存, 单位为二进制位 (bit), 定义见公式 (2);
- MaxCPB[level_id] —— 最大 BBV 缓冲区大小, 单位为二进制位 (bit), 根据 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流的级别 level_id 由 GY/T 368—2023 中的 B.3 的参数限制中得到;
- BBS —— BBV 缓冲区大小, 单位为二进制位 (bit), 根据 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流中由 GY/T 368—2023 中 7.1.2.2 和 7.2.2.2 定义的序列头参数 bbv_buffer_size 得到;
- BS_{max} —— 额外复用缓存, 单位为位 (bit), 定义见公式 (3)。

$$BS_{oh} = (1/750) \times \max\{\text{MaxBR}[\text{level_id}], 2\ 000\ 000\} \cdots \cdots (2)$$

$$BS_{\max} = 0.004 \times \max\{\text{MaxBR}[\text{level_id}], 2\ 000\ 000\} \cdots \cdots (3)$$

式中:

- MaxBR[level_id] —— 最大位速率, 单位为位每秒 (bit/s), 根据 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流的级别 level_id 由 GY/T 368—2023 中的 B.3 的参数限制中得到。

- c) 仅有一个基本流缓存 EB_n 用于接收分层描述子对应的基本流集合中的所有基本流, 该基本流缓存的总大小 EBS_n 定义见公式 (4)。

$$EBS_n = \text{BBS} \cdots \cdots (4)$$

式中:

- EBS_n —— 基本流缓存的总大小, 单位为二进制位 (bit);
- BBS —— BBV 缓冲区大小, 单位为二进制位 (bit), 根据 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流中由 GY/T 368—2023 中 7.1.2.2 和 7.2.2.2 定义的序列头参数 bbv_buffer_size 得到。

- d) 从 TB_n 到 MB_n 的传输应按照如下方式执行:

- 当 TB_n 中没有数据时, R_{x_n} 设为 0;
- 否则, 计算方式见公式 (5)。

$$R_{x_n} = \text{BitRate} \times 400 \dots \dots \dots (5)$$

式中:

R_{x_n} ——从传输缓存 TB_n 到复用缓存 MB_n 的传输速率, 单位为位每秒 (bit/s);

BitRate ——AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流中的比特率参数, 单位为位每秒 (bit/s)。当 AVS3 视频编码码流不支持时间分层时, 比特率参数根据 GY/T 368—2023 中 7.1.2.2 和 7.2.2.2 定义的比特率低和比特率高位计算得到; 当 AVS3 视频编码码流支持时间分层时, 比特率参数根据 GY/T 368—2023 中 7.1.2.7 和 7.2.2.7 定义的时间层比特率低和时间层比特率高位计算得到。

- e) 从 MB_n 到 EB_n 的传输应按照公式 (6) 执行。

$$R_{b_{x_n}} = \text{MaxBR}[\text{level_id}] \dots \dots \dots (6)$$

式中:

$R_{b_{x_n}}$ ——从复用缓存 MB_n 到基本流缓存 EB_n 的传输速率, 单位为位每秒 (bit/s);

$\text{MaxBR}[\text{level_id}]$ ——最大位速率, 单位为位每秒 (bit/s), 根据 AVS3 视频编码码流或包含最低时域层级的 AVS3 视频编码码流的级别 level_id , 由 GY/T 368—2023 中的 B.3 的参数限制中得到。

如果 MB_n 中存在 PES 分组包负载数据且 EB_n 没有充满, PES 分组包负载以速率 $R_{b_{x_n}}$ 从 MB_n 被传输到 EB_n 。如果 EB_n 已经被充满, MB_n 中的数据不应被移除。当一个字节的数据从 MB_n 被传输到 EB_n 时, MB_n 中位于该字节之前的 PES 分组包头的所有字节被立即移除和丢弃。当 MB_n 中不存在 PES 分组包负载数据时, 没有数据被从 MB_n 中移除。所有进入 MB_n 的数据都应从其中离开。所有离开 MB_n 的 PES 分组包负载数据都要立即进入 EB_n 。

7.3.5.3 STD 延时

STD 延时是数据通过系统目标解码器中的 TB_n 、 MB_n 和 EB_n 缓存的时间, 符合 GY/T 368—2023 的任何数据的 STD 延时应满足: 对于所有的 j 对应的存取单元 $A_n(j)$ 中的所有字节 i , $t_{d_n}(j) - t(i) \leq 10s$ 。

7.3.5.4 缓存管理条件

传输流的构建缓存管理条件如下:

- 每个 TB_n 不应上溢, 且应在每一秒中至少有一次处于被清空的状态;
- 每个 MB_n 、 EB_n 和 DPB 不应上溢;
- EB_n 不应下溢。当 AVS3 视频存取单元 $A_n(j)$ 中一个或多个字节在解码时间 $t_{d_n}(j)$ 不存在于 EB_n 中时, EB_n 对于 $A_n(j)$ 发生下溢。

7.4 Audio Vivid 音频编码码流在 TS 中的封装要求

7.4.1 基本要求

Audio Vivid 音频编码码流应满足以下约束。

- Audio Vivid 音频编码码流应是 GB/T 17975.1—2010 中节目的一个节目元素, 基本流的 stream_type 字段值应等于 ‘0xD5’。
- Audio Vivid 音频使用 AATF 的封装格式, 即封装成 $\text{aatf_frame}()$ 。

- c) Audio Vivid 音频编码码流的常见编码参数，如 `audio_codec_id`，应使用 Audio Vivid 音频流描述符标识。如果存在与 Audio Vivid 音频流相关联的 Audio Vivid 音频流描述符，则该描述符应包含在节目映射表中相应基本流条目的描述符循环中。

7.4.2 PES 分组流标识

Audio Vivid 音频应作为 `PES_packet_data_bytes` 携带在 PES 分组数据包中，并通过节目映射表中分配的 `stream_type` 字段值 (0xD5) 标识。

Audio Vivid 音频的 PES 数据包应满足以下约束：

- PES 分组的 `stream_id` 取值 ‘1111 1101’ (`extended_stream_id`)；
- PES 分组包头中 `stream_id_extension_flag` 取值 ‘0’，`stream_id_extension` 字段的取值 ‘100 1111’ 用于表示 AVS3 音频；
- `elementary stream` 在 PES payload 里字节对齐，即 AVS3 音频的首字节位于 PES payload 的首字节中；
- 一个 PES 包包含一帧或多帧 `aatf_frame()`。

Audio Vivid 音频编码码流在 PES 分组中的流标识应符合表 9 的规定。

表 9 Audio Vivid 音频编码码流在 PES 分组中的流标识

字段名称	值
<code>stream_id</code>	1111 1101
<code>stream_id_extension_flag</code>	0
<code>stream_id_extension</code>	100 1111
<code>stream_type</code>	0xD5 ^a
^a 应配有 <code>descriptor_tag</code> 为 210 (0xD2) 的 <code>AVS3_audio_descriptor</code> 。	

7.4.3 节目和节目元素描述符

7.4.3.1 注册描述符

PMT 中应包含 `registration_descriptor()`。`registration_descriptor()` 的定义见 GB/T 17975.1—2010，其中，`registration_descriptor()` 的 Tag 值应为 5 (0x05)，Audio Vivid 音频的 `format_identifier` 应为 0x41-56-53-41 (‘AVSA’)。

7.4.3.2 Audio Vivid 音频流描述符

Audio Vivid 音频流描述符 `AVS3_audio_descriptor()` 位于 PMT 中的 `ES_info_length` 字段后面，语法应符合表 10 的规定。

表 10 Audio Vivid 音频流描述符定义

Audio Vivid 音频流描述符定义	位数	助记符
<code>AVS3_audio_descriptor() {</code>		
<code> descriptor_tag</code>	8	<code>uimsbf</code>
<code> descriptor_length</code>	8	<code>uimsbf</code>
<code> audio_codec_id</code>	4	<code>uimsbf</code>
<code> sampling_frequency_index</code>	4	<code>uimsbf</code>

表 10 (续)

Audio Vivid音频流描述符定义	位数	助记符
if(audio_codec_id==1) {		
if (sampling_frequency_index==0xf) {		
sampling_frequency	24	uimsbf
}		
anc_data_index	1	bslbf
coding_profile	3	bslbf
reserved	4	uimsbf
channel_number	8	uimsbf
}		
if(audio_codec_id==2) {		
nn_type	3	uimsbf
reserved	1	bslbf
content_type	4	uimsbf
if(content_type==0) {		
channel_number_index	7	uimsbf
reserved	1	bslbf
}else if(content_type==1) {		
object_channel_number	7	uimsbf
reserved	1	bslbf
}else if(content_type==2) {		
channel_number_index	7	uimsbf
reserved	1	bslbf
object_channel_number	7	uimsbf
reserved	1	bslbf
}else if(content_type==3) {		
hoa_order	4	uimsbf
reserved	4	bslbf
}		
total_bitrate	16	uimsbf
}		
resolution	2	uimsbf
reserved	6	bslbf
for (i=0; i<N; i++) {		
addition_info[i]	8	bslbf
}		
}		

descriptor_tag

8位无符号整数。值为210 (0xD2)，表示该描述符为Audio Vivid音频流描述符。

descriptor_length

8位无符号整数。表示紧跟在该字段之后的描述符的字节数。

audio_codec_id

4位无符号整数。表示音频编码类型，该字段与GY/T 363—2023中audio_codec_id字段相同。

sampling_frequency_index

4位无符号整数。表示音频采样率索引，该字段与GY/T 363—2023中sampling_frequency_index字段相同。

sampling_frequency

24位无符号整数。表示扩展用音频采样率，该字段与GY/T 363—2023中sampling_frequency字段相同。

anc_data_index

二值变量。表示编码码流中是否有辅助数据，该字段与GY/T 363—2023中anc_data_index字段相同。

coding_profile

3位无符号整数。表示编码类，该字段与GY/T 363—2023中coding_profile字段相同。

channel_number

8位无符号整数。该字段与GY/T 363—2023中channel_number字段相同。

nn_type

3位无符号整数。表示神经网络类型，该字段与GY/T 363—2023中nn_type字段相同。

content_type

4位无符号整数。表示音频内容类型，具体定义应符合表11的规定。

表11 content_type 定义

content_type值	音频内容类型	映射关系
0	声道信号	coding_profile 值为0
1	对象信号	coding_profile 值为1且soundBedType 值为0
2	声道信号和对象信号混合	coding_profile 值为1且soundBedType 值为1
3	HOA 信号	coding_profile 值为2
4~15	保留	

channel_number_index

7位无符号整数。表示声道数索引，该字段与GY/T 363—2023中channel_number_index字段相同。

object_channel_number

7位无符号整数。表示音频对象数量，该字段与GY/T 363—2023中object_channel_number字段相同。

hoa_order

4位无符号整数。表示HOA阶数，该字段为GY/T 363—2023中order字段值+1。

total_bitrate

16位无符号整数，表示音频编码总码率，单位为千比特每秒(kbps)，具体定义应符合表12的规定。

表12 total_bitrate 定义

content_type值	total_bitrate定义
0	GY/T 363—2023 A.2中声道信号的bitrate_index对应的比特率值
1	GY/T 363—2023 A.2中对象信号的bitrate_index_per_channel对应的比特率值 × (object_channel_number + 1)
2	GY/T 363—2023 A.2中声道信号的bitrate_index对应的比特率值 + 对象信号的bitrate_index_per_channel对应的比特率值 × (object_channel_number + 1)
3	GY/T 363—2023 A.2中HOA信号的bitrate_index对应的比特率值
4~15	保留

resolution

2 位无符号整数，表示输入信号的量化比特数索引，该字段与 GY/T 363—2023 中 resolution 相同。

addition_info

可选字段，表示附加信息。

7.4.4 Audio Vivid 音频 T-STD 模型拓展

7.4.4.1 Audio Vivid 传输模型

Audio Vivid 音频 T-STD 模型拓展中访问单元表示使用 AATF 封装格式的一个音频帧。对于包含 Audio Vivid 音频的传输流应符合 T-STD 模型，见图 2。

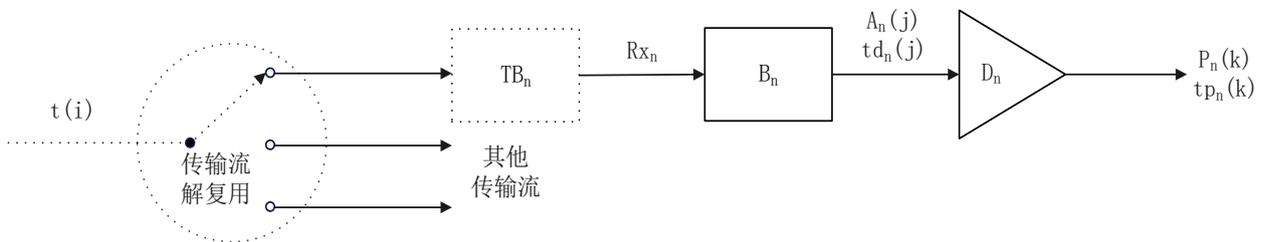


图2 面向 Audio Vivid 音频的 T-STD 模型拓展

图 2 中符号定义如下：

- $t(i)$ ：传输流的第 i 个字节进入系统目标解码器的时间，单位为秒(s)；
- TB_n ：基本流 n 的传输缓存；
- B_n ：基本流 n 的主缓存；
- R_{x_n} ：从 TB_n 到 B_n 的传输速率；
- $A_n(j)$ ：Audio Vivid 音频基本流的第 j 个存取单元；
- $t_{d_n}(j)$ ： $A_n(j)$ 在系统目标解码器中解码的时间，单位为秒(s)；
- D_n ：基本流 n 对应的解码器；
- $P_n(k)$ ：基本流 n 中第 k 个呈现单元；
- $t_{p_n}(k)$ ：基本流 n 中第 k 个呈现单元对应的时间，单位为秒(s)。

7.4.4.2 缓存管理

Audio Vivid 音频 T-STD 模型扩展中从 TB_n 到 B_n 的传输速率 R_{X_n} 计算见公式 (7)。

$$R_{X_n} = 1.2 \times R_{\max} \times N \dots \dots \dots (7)$$

式中:

R_{\max} ——Audio Vivid 音频单通道最大速率;

N ——基本流中包含的音频通道总数。

Audio Vivid 音频 T-STD 模型拓展中对应 R_{X_n} 取值, 见表 13。

表13 R_{X_n} 取值范围

声道数量	R_{X_n} bit/s
1~8	2000000
9~16	3686400
17~48	11059200
49~128	29491200

Audio Vivid 音频 T-STD 模型扩展中主缓存 BS_n 计算见公式 (8)。

$$BS_n = BS_{\max} + BS_{\text{dec}} + BS_{\text{oh}} \dots \dots \dots (8)$$

式中:

BS_{\max} ——额外复用缓存, 计算见公式 (9);

BS_{dec} ——基本流存储单元缓存, 取值为6144bits;

BS_{oh} ——PES包头缓存, 取值为528bytes。

$$BS_{\max} = 0.004 \times R_{\max} \times N \dots \dots \dots (9)$$

Audio Vivid 音频 T-STD 模型拓展中对应 BS_n 取值, 见表 14。

表14 BS_n 取值范围

声道数	BS_n 字节
1~8	7440
9~16	14352
17~48	42000
49~128	111120

7.4.4.3 缓存延时

Audio Vivid 音频传输流 STD 延时应满足: 对于所有 j 对应的存取单元 $A_n(j)$ 中的所有字节 i , 对应 $td_n(j) - t(i) \leq 10s$ 。

7.4.4.4 缓存管理条件

缓存管理条件如下:

- TB_n 不应上溢, 且应在每一秒中至少有一次处于清空状态;
- B_n 既不能上溢, 也不能下溢。

8 超高清音视频编码码流在 SMT 中的封装要求

8.1 概述

超高清音视频编码码流在SMT中的封装基于GB/T 33475.6—2024进行扩展，将超高清音视频编码码流对应的ISOBMFF文件封装成用于SMT协议的媒体资源(asset)，采用asset_type表示媒体资源的类型。

8.2 AVS2 视频编码码流在 SMT 中的封装要求

8.2.1 流标识

将AVS2视频编码码流封装成用于SMT协议的asset时，asset_type字段应为‘avst’。

8.2.2 AVS2 彩色信息描述符

8.2.2.1 概述

对MP表中的asset_descriptors()进行扩展，用于描述彩色信息。

8.2.2.2 语法

AVS2彩色信息描述符定义应符合表15的规定。

表15 AVS2 彩色信息描述符定义

AVS2彩色信息描述符定义	值	位数	助记符
Colour_info_descriptor () {			
descriptor_tag		16	uimsbf
descriptor_length		16	uimsbf
num_colour_asset	N	8	uimsbf
for(i=0; i<N; i++){			
asset_id()			
colour primaries		8	uimsbf
transfer characteristics		8	uimsbf
matrix coefficients		8	uimsbf
}			
}			

8.2.2.3 语义

descriptor_tag

16位无符号整数。值为0x0009，表示该描述符为彩色信息描述符。

descriptor_length

16位无符号整数。表示紧跟在该字段之后的描述符的字节数。

num_colour_asset

8位无符号整数。表示具有彩色信息描述的媒体资源数量。

asset_id()

表示媒体资源标识。

```

asset_id() {
    asset_id_scheme
    asset_id_length
    for(i = 0; i < asset_id_length;i++) {
        asset_id_byte
    }
}

```

其中：

- asset_id_scheme：32位无符号整数，表示asset_id的取值策略，值为‘UUID’或‘URI’；
- asset_id_length：asset_id字节长度；
- asset_id_byte：8位字符，不应用MFU所属的资源标识符编号。

colour primaries

8位无符号整数。表示AVS2视频编码码流中源图像三基色的色度坐标。该字段与GY/T 299.1—2016中colour_primaries字段相同。

transfer characteristics

8位无符号整数。表示AVS2视频编码码流中源图像的光电转移特性。该字段与GY/T 299.1—2016中transfer_characteristics字段相同。

matrix coefficients

8位无符号整数。表示从红绿蓝三基色转换为亮度和色度信号时采用的转换矩阵。该字段与GY/T 299.1—2016中matrix_coefficients字段相同。

8.3 AVS3 视频编码码流在 SMT 中的封装要求

8.3.1 流标识

将AVS3视频编码码流封装成用于SMT协议的asset时，asset_type字段应为‘avs3’。

8.3.2 缓存内容更新信令

8.3.2.1 概述

在SMT信令信息中新增LBM消息、LBMF消息，用于描述AVS3知识位流的相关信息。

8.3.2.2 AVS3 LBM消息

8.3.2.2.1 语法

AVS3 LBM消息的定义应符合表16的规定。

表16 AVS3 LBM 消息定义

LBM消息定义	位数	助记符
LBM_message () {		
message_id	16	uimsbf
version	8	uimsbf
length	16	uimsbf

表 16 (续)

LBM消息定义	位数	助记符
payload{		
required_buffer_size	32	uimsbf
required_buffer_Manage	8	uimsbf
}	16	
}		

8.3.2.2.2 语义

message_id

16位无符号整数。值为0xE006，表示LBM消息的消息标识符。

version

8位无符号整数。表示LBM消息的版本。

length

16位无符号整数。表示紧跟在该字段之后的LBM消息的字节数。

required_buffer_size

32位无符号整数。表示客户端接收该数据需准备的知识位流数据缓存的字节大小。

required_buffer_Manage

8位无符号整数。表示客户端管理知识位流数据缓存的方法。值为‘0’表示使用FIFO方法；值为‘1’表示使用LFU方法；值为‘2’表示使用LRU方法；其他值保留。

8.3.2.3 AVS3 LBMF消息

8.3.2.3.1 语法

AVS3 LBMF消息定义应符合表17的规定。

表17 AVS3 LBMF 消息定义

LBMF消息定义	值	位数	助记符
LBM_feedback_message() {			
message_id		16	uimsbf
version		8	uimsbf
length		16	uimsbf
payload{			
unavailable_mfu_number	N	32	uimsbf
for(i=0;i<N;i++){			
asset_id()			
sample_id		32	uimsbf
mfu_id		32	uimsbf
}			
}			
}			

8.3.2.3.2 语义

message_id

16位无符号整数。值为0xE007，表示LBMF消息的消息标识符。

version

8位无符号整数。表示LBMF消息的版本。

length

16位无符号整数。表示紧跟在该字段之后的LBMF消息的字节数。

unavailable_mfu_number

32位无符号整数。表示知识位流数据缓存中不可用的数据所属的MFU的数目。

asset_id()

表示媒体资源标识。

```
asset_id() {
    asset_id_scheme
    asset_id_length
    for(i = 0; i < asset_id_length;i++) {
        asset_id_byte
    }
}
```

其中：

——asset_id_scheme：32位无符号整数，表示asset_id的取值策略，值为‘UUID’或‘URI’；

——asset_id_length：asset_id字节长度；

——asset_id_byte：8位字符，不应用MFU所属的资源标识符编号。

sample_id

32位无符号整数。表示第i个不可用MFU所属的样本编号。

mfu_id

32位无符号整数。表示第i个不可用MFU的编号。

8.3.3 AVS3 包含非对齐 MFU 依赖关系的样本

8.3.3.1 概述

对CEU中的SMT提示轨道样本SMTHSample进行扩展，增加AVS3包含非对齐MFU依赖关系的样本SMTHRefSample，用于支持对序列图像和知识图像依赖关系的描述及对知识图像的管理。

8.3.3.2 语法

```
aligned(8) class SMTHRefSample extend SMTHSample {
    referenceMFUInfo();
}

aligned(8) class referenceMFUInfo extends Box( 'refm' ) {
    bit(1) reference_Sample_flag;
    bit(1) is_firstMFUinSample_flag;
```

```

    bit(6) reserved;
    if (reference_Sample_flag&&is_firstMFUinSample_flag) {
        unsigned int(32) depended_Sample_id;
    }
}

```

8.3.3.3 语义

reference_Sample_flag

1位无符号整数。表示当前MFU所属的sample是否依赖MFU，值为‘0’表示不依赖，值为‘1’表示依赖。

is_firstMFUinSample_flag

1位无符号整数。表示当前MFU是否为所属sample的第一个MFU，值为‘0’表示否，值为‘1’表示是。

depended_Sample_id

32位无符号整数。表示参考的sample编号。

8.3.4 Asset 关系信息描述符

8.3.4.1 语法

对Asset关系信息描述符进行扩展，增加library_flag、num_libraries和asset_id，用于标识当前AVS3主位流Asset与非对齐时间段的AVS3知识位流Asset的依赖关系。Asset关系信息描述符定义应符合表18的规定。

表18 Asset 关系信息描述符定义

Asset关系信息描述符定义	值	位数	助记符
Asset_relationship_information_descriptor() {			
descriptor_tag		16	uimsbf
descriptor_length		16	uimsbf
library_flag		1	blsbf
reserved	‘11’	2	blsbf
combine_quality_flag		1	blsbf
dependency_flag		1	blsbf
composition_flag		1	blsbf
equivalence_flag		1	blsbf
similarity_flag		1	blsbf
if(dependency_flag) {			
num_dependencies	N1	8	uimsbf
for(i = 0; i <N1; i++) {			
asset_id()			
}			
}			
if(composition_flag) {			

表 18 (续)

Asset关系信息描述符定义	值	位数	助记符
num_compositions	N2	8	uimsbf
for(i = 0; i <N2; i++) {			
asset_id()			
}			
}			
if(equivalence_flag) {			
equivalence_selection_level		8	uimsbf
num_equivalences	N3	8	uimsbf
for(i = 0; i <N3; i++) {			
asset_id()			
equivalence_selection_level		8	uimsbf
}			
}			
if(similarity_flag) {			
similarity_selection_level		8	uimsbf
num_similarities	N4	8	uimsbf
for(i = 0; i <N4; i++) {			
asset_id()			
similarity_selection_level		8	uimsbf
}			
}			
if(combine_quality_flag) {			
combine_quality_ranking		8	uimsbf
num_combine_assets	N5	8	uimsbf
for(i = 0; i < N5; i++) {			
asset_id()			
}			
}			
if(library_flag){			
num_libraries	N6	8	uimsbf
for(i = 0; i < N6; i++) {			
asset_id()			
}			
}			
}			

8.3.4.2 语义

library_flag

1位位串。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。

num_libraries

8位无符号整数。表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。其他语法对应的语义与GB/T 33475.6—2024中9.4.4.2相同。

asset_id()

表示媒体资源标识。

```
asset_id() {
    asset_id_scheme
    asset_id_length
    for(i = 0; i < asset_id_length;i++) {
        asset_id_byte
    }
}
```

其中：

- asset_id_scheme：32位无符号整数，表示asset_id的取值策略，值为‘UUID’或‘URI’；
- asset_id_length：asset_id字节长度；
- asset_id_byte：8位字符，不应用MFU所属的资源标识符编号。

8.3.5 AVS3 彩色信息描述符

8.3.5.1 概述

对MP表中的asset_descriptors()进行扩展，用于描述彩色信息。

8.3.5.2 语法

AVS3彩色信息描述符定义应符合表19的规定。

表19 AVS3 彩色信息描述符定义

AVS3彩色信息描述符定义	值	位数	助记符
Colour_info_descriptor () {			
descriptor_tag		16	uimsbf
descriptor_length		16	uimsbf
num_colour_asset	N	8	uimsbf
for(i=0; i<N; i++){			
asset_id()			
colour primaries		8	uimsbf
transfer characteristics		8	uimsbf
matrix coefficients		8	uimsbf
}			
}			

8.3.5.3 语义

descriptor_tag

16位无符号整数。值为0x0009，表示该描述符为彩色信息描述符。

descriptor_length

16位无符号整数。表示紧跟在该字段之后的描述符的字节数。

num_colour_asset

8位无符号整数。表示具有彩色信息描述的媒体资源数量。

asset_id()

表示媒体资源标识。

```
asset_id() {
    asset_id_scheme
    asset_id_length
    for(i = 0; i < asset_id_length;i++) {
        asset_id_byte
    }
}
```

其中：

——asset_id_scheme：32位无符号整数，表示asset_id的取值策略，值为‘UUID’或‘URI’；

——asset_id_length：asset_id字节长度；

——asset_id_byte：8位字符，不应用MFU所属的资源标识符编号。

colour_primaries

8位无符号整数。表示 AVS3 视频编码码流中源图像三基色的色度坐标。该字段与 GY/T 368—2023 中 colour_primaries 字段相同。

transfer_characteristics

8位无符号整数。表示 AVS3 视频编码码流中源图像的光电转移特性。该字段与 GY/T 368—2023 中 transfer_characteristics 字段相同。

matrix_coefficients

8位无符号整数。表示从红绿蓝三基色转换为亮度和色度信号时采用的转换矩阵。该字段与GY/T 368—2023中matrix_coefficients字段相同。

8.4 Audio Vivid 音频编码码流在 SMT 中的封装要求

8.4.1 流标识

将Audio Vivid音频编码码流封装成用于SMT协议的asset时，asset_type字段应为‘av3a’。

8.4.2 Audio Vivid 媒体资源描述符

8.4.2.1 概述

对MP表中的asset_descriptors()进行扩展，用于描述Audio Vivid编码码流的编码类和级、存储模式等信息。

8.4.2.2 语法

Audio Vivid 媒体资源描述符定义应符合表 20 的规定。

表20 Audio Vivid 媒体资源描述符定义

Audio Vivid媒体资源描述符定义	值	位数	助记符
Audio_info_descriptor() {			
descriptor_tag		16	uimsbf
descriptor_length		16	uimsbf
audio_format_type		4	uimsbf
audio_codec_id		4	uimsbf
coding_profile		3	uimsbf
average_bitrate_flag		1	bslbf
hoa_order_flag		1	bslbf
channel_number_flag		1	bslbf
object_info_flag		1	bslbf
reserved	'1'	1	uimsbf
if(average_bitrate_flag==1) {			
average_bitrate		16	uimsbf
}			
else {			
max_bitrate		16	uimsbf
min_bitrate		16	uimsbf
}			
if(hoa_order_flag) {			
max_hoa_order		8	uimsbf
}			
if(channel_number_flag) {			
max_channel_number		8	uimsbf
}			
if(object_info_flag) {			
max_object_channel_number		8	uimsbf
}			
bit_depth_resolution		8	uimsbf
sample_rate		24	uimsbf
}			

8.4.2.3 语义

descriptor_tag

16位无符号整数。值为0xEC05，表示该描述符为Audio Vivid媒体资源描述符。

descriptor_length

16位无符号整数。表示紧跟在该字段之后的描述符的字节数。

audio_format_type

4位无符号整数。表示 Audio Vivid 编码码流的类别。值应设置为‘1’，表示码流为 Audio Vivid AATF 传输格式的码流，其他值保留。

audio_codec_id

4位无符号整数。表示音频编码类型，该字段与GY/T 363—2023中audio_codec_id字段相同。

coding_profile

3位无符号整数。表示编码类，该字段与GY/T 363—2023中coding_profile字段相同。

average_bitrate_flag

1位位串。值为‘0’时表示音频媒体资源不具备平均码率；值为‘1’时表示音频媒体资源具备平均码率。

max_bitrate、average_bitrate、min_bitrate

16位无符号整数。分别表示音频媒体资源的最大码率、平均码率、最小码率，单位为千比特每秒(kbps)。

hoa_order_flag

1位位串。值为‘1’表示当前描述符中指示HOA阶数；取值为‘0’表示当前描述符中不指示HOA阶数。

channel_number_flag

1位位串。值为‘1’表示当前描述符中指示声道数；值为‘0’时表示当前描述符中不指示声道数。

object_info_flag

1位位串。值为‘1’表示当前描述符中指示声音对象信息；值为‘0’表示当前描述符中不指示声音对象信息。

max_hoa_order

8位无符号整数。表示当前媒体资源支持的最大HOA阶数。

max_channel_number

8位无符号整数。表示当前媒体资源支持的最大声道数。

max_object_channel_number

8位无符号整数。表示当前媒体资源包含的全部音频对象支持的最大声道数量。

bit_depth_resolution

8位无符号整数。表示音频输入信号的量化比特数。

sample_rate

24位无符号整数。表示音频输入信号的采样频率。

附录 A

(规范性)

超高清音视频编码码流在 ISOBMFF 文件中的封装要求

A.1 概述

超高清音视频编码码流在ISOBMFF文件中的封装基于ISO/IEC 14496-12:2022进行扩展。

A.2 AVS2 视频编码码流在 ISOBMFF 文件中的封装要求

A.2.1 AVS2视频编码码流和轨道定义

A.2.1.1 AVS2视频编码特性

AVS2视频编码码流的文件格式使用并扩展ISO/IEC 14496-12:2022规定的ISO基本媒体文件格式的现有功能，支持下列AVS2视频编码时域分层特性：AVS2视频编码支持将不同的编码图像按照解码时的依赖关系划分为不同时域层级，划分为低层级的编码图像，在解码时无需参考更高层级的编码图像。

AVS2视频内容的结构化分组机制包括时域层级样本群组，用于指示与AVS2视频编码图像关联的时域可伸缩性的不同层级。

A.2.1.2 码流结构

AVS2视频编码码流为符合GY/T 299.1—2016的由编码图像所形成的二进制数据流。

存储AVS2视频内容的位流中应包含序列头，并符合如下约束：

- a) 序列头应是位流中包含的AVS2样本（同步样本）的一部分；
- b) 序列头也可以存储在AVS2视频轨道的样本入口中。

A.2.1.3 AVS2轨道

AVS2轨道通过在样本入口中包含一个序列头，并在其样本中封装AVS2视频编码图像来表示AVS2视频编码码流。

A.2.1.4 时域分层存储

AVS2视频码流的时域层级支持多轨道存储。多轨道存储可用于选择性地访问和播放AVS2视频码流的时域层级。

如果AVS2视频编码码流支持时域分层，即序列头的“时间层标识允许标志字段”取值‘1’，AVS2视频编码码流中可以包含一个或多个不同时域层级的编码图像。AVS2视频编码码流中不同时域层级的编码图像可通过以下方式存储。

- a) 所有时域层级的编码图像存储在同一个AVS2轨道：
 - 1) AVS2轨道应使用A.2.3.1中定义的样本入口类型‘avst’；
 - 2) AVS2轨道应使用A.2.3.4中定义的时域层级样本群组来为轨道中的样本提供时域层级信息。
- b) 所有时域层级的编码图像存储在多个不同的AVS2轨道。其中，每个AVS2轨道中包含一个或多个时域层级对应的编码图像。
 - 1) 包含最低时域层级的AVS2轨道应使用A.2.3.1中定义的样本入口类型‘avst’。
 - 2) 不包含最低时域层级的AVS2轨道应使用A.2.3.2中定义的样本入口类型‘lav2’。

- 3) 其中, 样本入口类型为‘lav2’的AVS2轨道应使用‘ltrf’类型的轨道引用, 引用样本入口类型为‘avs2’的AVS2轨道, 并且可选地引用一个或多个样本入口类型为‘lav2’的包含较低时域层级的AVS2轨道。通过解析‘ltrf’类型的轨道引用, 包含不同时域层级的一个或多个AVS2轨道中的所有样本应包含解码至所有时域层级中的最高时域层级对应的AVS2位流中编码图像子集所需的所有数据。

示例:

当AVS2视频编码码流中不同时域层级的编码图像存储在不同的AVS2轨道时, 解码AVS2视频编码码流中时域分层的编码图像之前, 客户端需要获取包含对应时域层级的所有AVS2轨道, 重建AVS2视频编码码流中对应时域层级的子位流。

客户端首先确定AVS2视频编码码流中所要解码的最高时域层级, 并获取包含该时域层级的AVS2轨道。客户端通过解析‘ltrf’类型的轨道引用, 获取包含解码至所述最高时域层级所需AVS2视频编码码流子集的一个或多个AVS2轨道。

客户端按照解码时间对所获取的一个或多个AVS2轨道中的所有编码图像进行重新排序, 重建AVS2视频编码码流, 并按照A. 2. 2. 2中规定的解码器配置信息中的约束进行解码。

A. 2. 2 样本与解码器配置定义

A. 2. 2. 1 样本格式

AVS2样本: AVS2轨道中的样本应包含GY/T 299. 1—2016中规定的一个编码图像, 还可包含GY/T 299. 1—2016中规定的序列头和序列结束码。

A. 2. 2. 2 AVS2解码器配置记录

A. 2. 2. 2. 1 定义

AVS2解码器配置记录适用于不支持时域分层或者包含最低时域层级的AVS2视频位流。

A. 2. 2. 2. 2 语法

```
aligned(8) class Avs2DecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(16) sequence_header_length;
    bit(8*sequence_header_length) sequence_header;
}
```

A. 2. 2. 2. 3 语义

sequence_header_length: 以字节为单位指示配置记录所适用的位流中序列头的长度。

sequence_header: 包含GY/T 299. 1—2016中定义的sequence_header()。

A. 2. 2. 3 LAVS2解码器配置记录

A. 2. 2. 3. 1 定义

LAVS2解码器配置记录适用于支持时域分层的AVS2视频位流。

A. 2. 2. 3. 2 语法

```
aligned(8) class LAVs2DecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
```

```

    unsigned int(8) num_temporal_layers;
    for(i=0; i< num_temporal_layers; i++) {
        unsigned int(3) temporal_layer_id[i];
        unsigned int(4) frame_rate_code[i];
        bit(1) reserved = '1'b;
        unsigned int(18) temporal_bit_rate_lower[i];
        unsigned int(12) temporal_bit_rate_upper[i];
        bit(2) reserved = '11'b;
    }
}

```

A. 2. 2. 3. 3 语义

`num_temporal_layers`: 指示解码器配置记录所适用的位流中包含的时域层级数目。

`temporal_layer_id[i]`: 指示解码器配置记录所适用的位流中包含的时域层级标识。

`frame_rate_code[i]`: 指示当时域层级累计到 `temporal_layer_id[i]`时, AVS2 视频编码码流对应的帧率代码。

`temporal_bit_rate_lower[i]`: 指示当时域层级累计到 `temporal_layer_id[i]`时, AVS2 视频编码码流对应码率的低 18 位。

`temporal_bit_rate_upper[i]`: 指示当时域层级累计到 `temporal_layer_id[i]`时, AVS2 视频编码码流对应码率的高 12 位。

A. 2. 2. 4 AVS2 解码器配置数据盒

A. 2. 2. 4. 1 定义

`Avs2ConfigurationBox` 包含 A. 2. 2. 2 规定的 AVS2 解码器配置记录。

A. 2. 2. 4. 2 语法

```

class Avs2ConfigurationBox extends Box( 'av2c', version = 0) {
    Avs2DecoderConfigurationRecord () Avs2Config;
}

```

A. 2. 2. 4. 3 语义

`Avs2DecoderConfigurationRecord` 应符合 A. 2. 2. 2 的规定。

A. 2. 2. 5 LAVS2 解码器配置数据盒

A. 2. 2. 5. 1 定义

`LAvs2ConfigurationBox` 包含 A. 2. 2. 3 规定的 LAVS2 解码器配置记录。

A. 2. 2. 5. 2 语法

```
class LAvs2ConfigurationBox extends Box( '1a2c', version = 0) {
    LAvs2DecoderConfigurationRecord () LAvs2Config;
}
```

A. 2. 2. 5. 3 语义

LAvs2DecoderConfigurationRecord在A. 2. 2. 3中定义。

A. 2. 3 ISO基本媒体文件格式扩展

A. 2. 3. 1 AVS2样本入口

A. 2. 3. 1. 1 定义

样本入口类型：‘avsv’。

容器：Sample Table Box (‘stbl’)。

强制性：封装AVS2视频编码码流的轨道中的至少一个轨道应有一个‘avst’样本入口。

数量：一个或多个。

AVS2轨道的每个样本入口应是一个AVS2样本入口，其样本入口类型等于‘avst’或者A. 2. 3. 2定义的‘lav2’。

AVS2轨道应包含符合GY/T 299. 1—2016规定的AVS2视频位流，并符合如下约束：

- 不包含时域层级的AVS2轨道的每个样本入口应包含一个Avs2ConfigurationBox；
- 包含最低时域层级的AVS2轨道的每个样本入口应包含一个Avs2ConfigurationBox，以及一个LAvs2ConfigurationBox；
- 包含时域层级且不包含最低时域层级的AVS2轨道的每个样本入口应包含一个LAvs2ConfigurationBox。

A. 2. 3. 1. 2 语法

```
class Avs2SampleEntry() extends VisualSampleEntry ( 'avst' ) {
    Avs2ConfigurationBox    config;
    LAvs2ConfigurationBox (); // 可选 MPEG4ExtensionDescriptorsBox ();// 可选
}
```

A. 2. 3. 1. 3 语义

Avs2ConfigurationBox：提供AVS2视频编码码流的解码配置信息。

LAvs2ConfigurationBox：提供支持时域分层的AVS2视频编码码流的解码配置信息。

MPEG4ExtensionDescriptorsBox：定义见ISO/IEC 14496-15:2022中5. 4. 2。

A. 2. 3. 2 时域分层样本入口

A. 2. 3. 2. 1 定义

样本入口类型：‘lav2’。

容器：Sample Table Box (‘stbl’)。

强制性：‘lav2’样本入口不是强制的。

数量：一个或多个。

支持时域分层且不包含最低时域层级的AVS2轨道应使用样本入口类型等于‘lav2’的LAvs2SampleEntry。

A. 2. 3. 2. 2 语法

```
class LAvs2SampleEntry() extends VisualSampleEntry ( 'lav2' ) {
    LAvs2ConfigurationBox ();
    MPEG4ExtensionDescriptorsBox (); // 可选
}
```

A. 2. 3. 3 同步样本

AVS2 视频轨道（样本入口类型为‘avst’）中包含随机访问点编码图像的样本被定义为同步样本，同步样本的使用符合 ISO/IEC 14496-12:2022 中 8.6.2 的定义。

当样本入口类型为‘avst’时，适用以下约束。

- a) 如果样本是同步样本，解码该样本所需的序列头应包含在该样本中。此外，解码该样本所需的序列头还可以包含在样本入口中。
- b) 否则（样本不是同步样本），解码样本所需的序列头应包含在该样本的前一个同步样本中。此外，解码样本所需的序列头还可以包含在样本入口中。

A. 2. 3. 4 时域层级样本群组

时域层级样本群组（‘telg’）可用于根据时域层级对 AVS2 轨道中的样本进行分组，并为组中的样本提供时域层级信息。其语法和语义与 A. 3. 4. 3 中定义的时域层级样本群组相同。

A. 3 AVS3 视频编码码流在 ISOBMFF 文件中的封装要求

A. 3. 1 AVS3 视频编码码流和轨道定义

A. 3. 1. 1 AVS3 视频编码特性

AVS3 视频编码码流的文件格式使用并扩展 ISO/IEC 14496-12:2022 规定的 ISO 基本媒体文件格式的现有功能，支持下列 AVS3 视频编码特性。

a) 知识图像

知识图像是 AVS3 视频编码中用于帧间预测的特殊参考帧，RL 图像是 AVS3 视频编码中只使用知识图像作为参考图像进行帧间预测解码的图像。采用知识图像进行帧间预测，可在特定应用场景中大幅提升编码效率。

b) 时域分层

AVS3 视频编码支持将不同的编码图像按照解码时的依赖关系划分为不同时域层级，划分为低层级的编码图像，在解码时无需参考更高层级的编码图像。

用于支持 AVS3 视频内容的结构化分组机制包括：

- 1) 知识随机访问点样本群组：用于指示 AVS3 视频码流中作为 LRAP 样本的 RL 图像；
- 2) 知识图像样本群组：用于指示 AVS3 视频码流中知识图像；
- 3) 时域层级样本群组：用于指示与 AVS3 视频编码图像关联的时域可伸缩性的不同层级。

A. 3. 1. 2 码流结构

AVS3视频编码码流包括两种基本流的类型：

- a) 主基本流：可参考由该基本流以外的信息提供的知识图像进行解码的基本流；
- b) 知识基本流：只包含知识图像的基本流。其中，知识图像可以被其他基本流中的图像参考，而不用于显示输出。

使用上述基本流类型，存储AVS3视频内容的基本流中应包含序列头，并遵循如下约束：

- a) 序列头应是基本流中包含的AVS3样本（同步样本或者知识随机访问点样本）的一部分；
- b) 序列头也可以存储在AVS3视频轨道的样本入口中。

A.3.1.3 AVS3视频轨道类型

用于封装AVS3视频编码码流的AVS3视频轨道类型包括。

- a) AVS3轨道
 - 1) AVS3轨道通过在样本入口中包含一个或多个序列头，并在其样本中封装AVS3视频编码图像来表示AVS3视频编码码流中的主位流以及可能与其关联的知识位流。
 - 2) AVS3轨道通过在样本入口中包含一个或多个序列头，并在其样本中封装AVS3视频编码图像中的非知识图像来表示AVS3视频编码码流中的主位流。此时，AVS3轨道可以引用AVS3知识位流轨道。
 - 3) 当AVS3轨道通过‘a31r’轨道引用AVS3知识位流轨道时，也称为AVS3主位流轨道。
- b) AVS3知识位流轨道：AVS3知识位流轨道通过在样本入口中包含一个序列头，并在其样本中封装AVS3视频编码图像中的知识图像来表示AVS3视频编码码流中的知识位流。AVS3知识位流轨道由AVS3主位流轨道通过‘a31r’轨道引用来引用。

A.3.1.4 时域分层存储

AVS3视频编码码流的时域层级支持多轨道存储。多轨道存储可用于选择性地访问和播放AVS3视频编码码流的时域层级。

如果AVS3视频编码码流支持时域分层，即序列头的“时间层标识允许标志字段”取值‘1’，AVS3视频编码码流中可以包含一个或多个不同时域层级的编码图像。AVS3视频主位流中不同时域层级的编码图像可通过以下方式存储。

- a) 所有时域层级的编码图像存储在同一个AVS3轨道：
 - 1) AVS3轨道应使用A.3.3.1中定义的样本入口类型‘avs3’；
 - 2) AVS3轨道应使用A.3.4.3中定义的时域层级样本群组来为轨道中的样本提供时域层级信息。
- b) 所有时域层级的编码图像存储在多个不同的AVS3轨道。其中，每个AVS3轨道中包含一个或多个时域层级对应的编码图像：
 - 1) 包含最低时域层级的AVS3轨道应使用A.3.3.1中定义的样本入口类型‘avs3’；
 - 2) 不包含最低时域层级的AVS3轨道应使用A.3.3.2中定义的样本入口类型‘lav3’；
 - 3) 其中，样本入口类型为‘lav3’的AVS3轨道应使用‘ltrf’类型的轨道引用，引用样本入口类型为‘avs3’的AVS3轨道，并且可选地引用一个或多个样本入口类型为‘lav3’的包含较低时域层级的AVS3轨道。通过解析‘ltrf’类型的轨道引用，包含不同时域层级的一个或多个AVS3轨道中的所有样本应包含解码至所有时域层级中的最高时域层级对应的主位流中编码图像子集所需的所有数据。

示例：

当AVS3视频编码码流中不同时域层级的编码图像存储在不同的AVS3轨道时，解码AVS3视频编码码流中时域分层的编码图像之前，客户端需要获取包含对应时域层级的所有AVS3轨道，重建AVS3视频编码码流中对应时域层级的子码流。

客户端首先确定AVS3视频编码码流中所要解码的最高时域层级，并获取包含该时域层级的AVS3轨道。客户端通过解析‘ltrf’类型的轨道引用，获取包含解码至所述最高时域层级所需AVS3视频编码码流子集的一个或多个AVS3轨道。

客户端按照解码时间对所获取的一个或多个AVS3轨道中的所有编码图像进行重新排序，重建AVS3视频编码码流，并按照A.2.2.2中规定的解码器配置信息中的约束进行解码。

A.3.2 样本与解码器配置定义

A.3.2.1 AVS3样本格式

AVS3样本：AVS3轨道中的样本应包含GY/T 368—2023中规定的一个编码图像，还可包含GY/T 368—2023中规定的序列头和序列结束码。

AVS3知识图像样本：AVS3知识位流轨道中的样本包含GY/T 368—2023中规定的一个知识图像，还可包含GY/T 368—2023中规定的序列头和序列结束码。

A.3.2.2 AVS3解码器配置记录

A.3.2.2.1 定义

AVS3解码器配置记录适用于不支持时域分层或者包含最低时域层级的AVS3视频编码码流。

A.3.2.2.2 语法

```
aligned(8) class Avs3DecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(16) sequence_header_length;
    bit(8*sequence_header_length) sequence_header;
    bit(6) reserved = '111111'b;
    unsigned int(2) library_dependency_idc;
}
```

A.3.2.2.3 语义

sequence_header_length：以字节为单位指示配置记录所适用的基本流中序列头的长度。

sequence_header：包含GY/T 368—2023中定义的sequence_header()，并遵循以下约束。

- 如果配置记录适用于主基本流，则序列头中的library_stream_flag字段取值‘0’。而如果配置记录适用于知识基本流，则序列头中的library_stream_flag字段取值‘1’。
- 如果配置记录适用于主基本流，并且该主基本流不参考知识基本流中知识图像进行解码，则序列头中library_picture_enable_flag字段取值‘0’。如果配置记录适用于主基本流，并且该主基本流参考知识基本流中的知识图像进行解码，则序列头中library_picture_enable_flag字段取值‘1’。
- library_dependency_idc**：指示配置记录所适用的基本流的类型。如果配置记录适用于主基本流，library_dependency_idc还指示主基本流是否参考知识基本流中的知识图像进行解码。library_dependency_idc的值与描述见表A.1。

表 A.1 Library_dependency_idc 的值与描述

值	total_bitrate定义
00	当前解码器配置记录适用于主基本流，且该主基本流不参考知识基本流中的知识图像进行解码
01	当前解码器配置记录适用于主基本流，且该主基本流参考知识基本流中的知识图像进行解码
10	当前解码器配置记录适用于知识基本流
11	当前解码器配置记录所对应的轨道中同时封装主基本流和知识基本流，且主基本流参考知识基本流中的知识图像进行解码

A.3.2.3 LAVS3解码器配置记录

A.3.2.3.1 定义

LAVS3解码器配置记录适用于支持时域分层的AVS3视频码流。

A.3.2.3.2 语法

```
aligned(8) class LAVs3DecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) num_temporal_layers;
    for(i=0; i< num_temporal_layers; i++) {
        unsigned int(3) temporal_layer_id[i];
        unsigned int(4) frame_rate_code[i];
        bit(1) reserved = '1'b;
        unsigned int(18) temporal_bit_rate_lower[i];
        unsigned int(12) temporal_bit_rate_upper[i];
        bit(2) reserved = '11'b;
    }
}
```

A.3.2.3.3 语义

num_temporal_layers: 指示解码器配置记录所适用的基本流中包含的时域层级数目。

temporal_layer_id[i]: 指示解码器配置记录所适用的基本流中包含的时域层级标识。

frame_rate_code[i]: 指示当时域层级累计到 temporal_layer_id[i]时, AVS3 视频编码码流对应的帧率代码。

temporal_bit_rate_lower[i]: 指示当时域层级累计到 temporal_layer_id[i]时, AVS3 视频编码码流对应码率的低 18 位。

temporal_bit_rate_upper[i]: 指示当时域层级累计到 temporal_layer_id[i]时, AVS3 视频编码码流对应码率的高 12 位。

A.3.2.4 AVS3解码器配置数据盒

A.3.2.4.1 定义

Avs3ConfigurationBox包含A.3.2.2规定的AVS3解码器配置记录。

A.3.2.4.2 语法

```
class Avs3ConfigurationBox extends Box( 'avs3', version = 0) {
    Avs3DecoderConfigurationRecord () Avs3Config;
}
```

A.3.2.4.3 语义

Avs3DecoderConfigurationRecord在A.3.2.2中定义。

A.3.2.5 LAVS3解码器配置数据盒

A.3.2.5.1 定义

LAvs3ConfigurationBox包含A.3.2.3规定的LAVS3解码器配置记录。

A.3.2.5.2 语法

```
class LAvs3ConfigurationBox extends Box( 'lav3', version = 0) {
    LAvs3DecoderConfigurationRecord () LAvs3Config;
}
```

A.3.2.5.3 语义

LAvs3DecoderConfigurationRecord在A.3.2.3中定义。

A.3.3 ISO基本媒体文件格式扩展

A.3.3.1 AVS3样本入口

A.3.3.1.1 定义

样本入口类型： 'avs3' 。

容器： Sample Table Box ('stbl')。

强制性： 封装AVS3视频编码码流的轨道中的至少一个轨道应有一个 'avs3' 样本入口。

数量： 一个或多个。

AVS3轨道的每个样本入口应是一个AVS3样本入口，其样本入口类型等于 'avs3' 或者 'lav3' 。

AVS3轨道应包含符合GY/T 368—2023规定的AVS3视频码流，并符合如下约束：

- a) 不包含时域层级的 AVS3 轨道的每个样本入口应包含一个 Avs3ConfigurationBox;
- b) 包含最低时域层级的 AVS3 轨道的每个样本入口应包含一个 Avs3ConfigurationBox，以及一个 LAvs3ConfigurationBox;
- c) 不包含最低时域层级的 AVS3 轨道的每个样本入口应包含一个 LAvs3ConfigurationBox。

A.3.3.1.2 语法

```

class Avs3SampleEntry() extends VisualSampleEntry ( 'avs3' ) {
    Avs3ConfigurationBox  config;
    LAvs3ConfigurationBox (); // 可选
    MPEG4ExtensionDescriptorsBox (); // 可选
}

```

A.3.3.1.3 语义

Avs3ConfigurationBox: 提供 AVS3 视频编码码流的解码配置信息。

LAvs3ConfigurationBox: 提供支持时域分层的 AVS3 视频编码码流的解码配置信息。

MPEG4ExtensionDescriptorsBox: 定义见 ISO/IEC 14496-15:2022 中 5.4.2。

A.3.3.2 时域分层样本入口

A.3.3.2.1 定义

样本入口类型: ‘lav3’。

容器: Sample Table Box (‘stbl’)。

强制性: ‘lav3’ 样本入口是强制的。

数量: 一个或多个。

包含时域层级且不包含最低时域层级的 AVS3 轨道时，应使用样本入口类型等于 ‘lav3’ 的 LAvs3SampleEntry。

A.3.3.2.2 语法

```

class LAvs3SampleEntry() extends VisualSampleEntry ( 'lav3' ) {
    LAvs3ConfigurationBox ();
    MPEG4ExtensionDescriptorsBox (); // 可选
}

```

A.3.3.3 受限样本入口

AVS3 知识位流轨道仅包含用于参考而不用于显示输出的 AVS3 知识图像样本。

AVS3 知识位流轨道在文件中表示为受限视频，应使用符合下列附加约束的受限样本入口 ‘resv’。

- a) 原始样本入口类型应包含在 RestrictedSchemeInfoBox 中的 OriginalFormatBox。其中，OriginalFormatBox 的 data_format 字段取值 ‘avs3’。
- b) SchemeTypeBox 存在于 RestrictedSchemeInfoBox 中，并且 scheme_type 设置为 ‘av31’。

A.3.3.4 同步样本

AVS3 视频轨道（样本入口类型为 ‘avs3’）中包含随机访问点编码图像的样本被定义为同步样本，同步样本的使用应符合 ISO/IEC 14496-12:2022 中 8.6.2 的定义。

当样本入口类型为 ‘avs3’ 时，适用以下约束。

- a) 如果样本是同步样本或者知识随机访问点样本，解码该样本所需的序列头应包含在该样本中。此外，解码该样本所需的序列头还可以包含在样本入口中。

- b) 否则（样本不是同步样本或者知识随机访问点样本），解码样本所需的序列头应包含在该样本的前一个同步样本或知识随机访问点样本中。此外，解码样本所需的序列头还可以包含在样本入口中。

A.3.3.5 知识随机访问点样本

AVS3主位流轨道中的RL图像样本只使用AVS3知识位流轨道中的知识图像样本作为参考图像进行帧间预测解码。

AVS3主位流轨道中满足以下特性的RL图像样本被定义为LRAP样本，即，如果作为RL图像的参考图像的所有知识图像均可以获得，则按照解码顺序和显示顺序在LRAP图像之后的所有图像都可以正确解码。

LRAP样本具有A.2.4中定义的知识随机访问点类型（LRAP_type）。

A.3.3.6 片段索引数据盒

A.3.3.6.1 定义

数据盒类型：‘lidx’。

容器：File。

强制性：否。

数量：零个或多个。

LRAPSegmentIndexBox指示其引用的子片段中存在A.3.3.5中定义的知识随机访问点，并满足：

- a) 如果子片段封装的轨道分段包含LRAP，则子片段包含LRAP，并以LRAP开始；
- b) 子片段中的LRAP样本（LRAP_type类型取值‘1’‘2’或‘3’）应被标记为群组类型为‘lrap’的样本群组的成员。

A.3.3.6.2 语法

```
aligned(8) class LRAPSegmentIndexBox extends SegmentIndexBox( 'lidx', version, 0) {
    unsigned int(16) reserved = 0;
    unsigned int(16) reference_count;
    for(i=1; i <= reference_count; i++) {
        unsigned int(1) starts_with_LRAP;
        unsigned int(3) LRAP_type;
        unsigned int(28) reserved = 0;
    }
}
```

A.3.3.6.3 语义

reference_coun: 提供被引用项目的数量。

starts_with_LRAP: 取值为‘1’表示引用的子片段以LRAP开始；取值为‘0’表示引用的子片段不以LRAP开始。

LRAP_type: 指示子片段中LRAP的知识随机访问点类型。

A.3.4 样本群组定义

A.3.4.1 知识随机访问点样本群组

A.3.4.1.1 定义

知识随机访问点样本群组指示视频码流中的LRAP样本，以及作为参考图像样本的知识图像样本的信息。知识随机访问点样本群组的成员应当满足以下条件：

- a) LRAP 样本只能以知识图像样本作为参考图像样本；
- b) 在解码其所参考的知识图像样本之后，从 LRAP 样本处开始解码，可以正确解码 LRAP 样本以及所有显示顺序在该 LRAP 样本之后的样本。

注：LRAP样本只能与其所参考的知识图像样本联合使用。为创建可解码的样本序列，需要将知识图像样本与LRAP样本以及按显示顺序在LRAP样本之后的样本串联起来。

A.3.4.1.2 语法

```
class VisualLRAPEntry extend VisualSampleGroupEntry ( 'lrp' ) {
    unsigned int(3) LRAP_type;
    unsigned int(3) entry_count;
    unsigned int(2) reserved = 0;
    int i;
    for (i=0; i < entry_count; i++) {
        unsigned int(9) library_sample_number;
        unsigned int(7) reserved = 0;
    }
}
```

A.3.4.1.3 语义

LRAP_type 是一个非负整数，表示假设 LRAP 样本不依赖于知识图像样本时，LRAP 样本所对应的 ISO/IEC 14496-12:2022 附录 I 中定义的流访问点类型 (SAP_type)。LRAP_type 在 1 到 3 的范围内取值，其他类型值保留。

entry_count 是一个非负整数，表示群组中 LRAP 样本参考的知识图像样本的数量。entry_count 值为 '001' 表示群组中 LRAP 样本参考的知识图像样本数为 1；值 '000' '010' 和 '111' 保留。

library_sample_number 是一个非负整数，提供群组中 LRAP 样本所参考的知识图像样本的样本编号。

A.3.4.2 知识图像样本群组

A.3.4.2.1 定义

当AVS3轨道中包含来自自主基本流和知识基本流的样本时，知识图像样本群组（'a3lg'）用于标识AVS3轨道中的知识图像样本。

A.3.4.2.2 语法

```
class LibrarySampleGroupEntry extends VisualSampleGroupEntry ( 'a3lg' ) {
}
```

A.3.4.3 时域层级样本群组

A.3.4.3.1 定义

时域层级样本群组（‘telg’）可用于根据时域层级对AVS3轨道中的样本进行分组，并为组中的样本提供时域层级信息。

A.3.4.3.2 语法

```
class TemporalLayerEntry extend VisualSampleGroupEntry ( 'telg' ) {
    unsigned int(8) temporal_layer_id;
}
```

A.3.4.3.3 语义

temporal_layer_id 指示样本群组中的样本包含的编码图像的时域层级标识。

A.4 Audio Vivid音频编码码流在ISOBMFF文件中的封装要求

A.4.1 Audio Vivid 音频基本流定义

A.4.1.1 通则

Audio Vivid 音频基本流即 Audio Vivid 音频编码码流，应符合 GY/T 363—2023的规定。

A.4.1.2 Audio Vivid 音频编码特性

Audio Vivid 音频编码系统支持声道信号编码、对象信号编码、HOA 信号编码、元数据编码。

Audio Vivid 音频编码器由多种编码工具构成，包括：通用码率音频编码工具、无损音频编码工具和元数据编码工具。

A.4.1.3 基本流结构

根据Audio Vivid音频的编码特性，Audio Vivid音频的基本流结构包括：通用码率音频编码码流和无损音频编码码流。

A.4.1.4 基本流格式

Audio Vivid 音频基本流的传输采用 AATF 格式。

A.4.2 Audio Vivid 音频配置信息

A.4.2.1 Audio Vivid 音频通用码率音频编码特有配置

A.4.2.1.1 定义

适用于Audio Vivid音频通用码率音频编码内容的特有配置。

A.4.2.1.2 语法

```
class Avs3AudioGASpecificConfig {
    unsigned int(4) sampling_frequency_index;
    unsigned int(3) nn_type;
    unsigned int(1) reserved;
```

```

unsigned int(4) content_type;
if (content_type==0) {
    unsigned int(7) channel_number_index;
    unsigned int(1) reserved;
} else if(content_type==1) {
    unsigned int(7) number_objects;
    unsigned int(1) reserved;
} else if(content_type==2) {
    unsigned int(7) channel_number_index;
    unsigned int(1) reserved;
    unsigned int(7) number_objects;
    unsigned int(1) reserved;
} else if(content_type==3) {
    unsigned int(4) hoa_order;
}
}
unsigned int(16) total_bitrate;
unsigned int(2) resolution;
if (content_type==3) {
    unsigned int(2) reserved;
} else {
    unsigned int(6) reserved;
}
}

```

A.4.2.1.3 语义

sampling_frequency_index: 应符合 GY/T 363—2023附录 A 的规定。

nn_type: 应符合 GY/T 363—2023附录 A 的规定。

content_type: 表示音频内容类型, 应符合表 A.2 的规定。

表 A.2 content_type 配置表

content_type 值	音频内容类型	映射关系
0	声道信号	coding_profile 值为 0 时
1	对象信号	coding_profile 值为 1 且 soundBedType 值为 0 时
2	声道信号和对象信号混合	coding_profile 值为 1 且 soundBedType 值为 1 时
3	HOA 信号	coding_profile 值为 2 时
4~15	保留	
注: coding_profile 和 soundBedType 见 GY/T 363—2023 中的附录 A。		

channel_number_index: 应符合 GY/T 363—2023附录 A 的规定。

number_objects: 表示音频对象数量, 等于 GY/T 363—2023附录 A 中 object_channel_number+1。

hoa_order: 表示 HOA 信号阶数, 等于 GY/T 363—2023附录 A 中 order+1。

total_bitrate: 表示音频总比特率, 单位为千比特每秒 (kbps), 根据 **content_type** 的值计算方式不同, 见表 A.3。

表 A.3 **total_bitrate** 配置表

content_type 值	total_bitrate 计算方式
0	声道信号的 bitrate_index 对应的比特率
1	对象信号的 bitrate_index_per_channel 对应的比特率 × number_objects
2	声道信号的 bitrate_index 对应的比特率 + 对象信号的 bitrate_index_per_channel 对应的比特率 × number_objects
3	HOA 信号的 bitrate_index 对应的比特率
4~15	保留
注: bitrate_index 和 bitrate_index_per_channel 见GY/T 363—2023中的附录A。	

resolution: 应符合GY/T 363—2023附录A的规定。

A.4.2.2 Audio Vivid 音频无损音频编码特有配置

A.4.2.2.1 定义

适用于 Audio Vivid 音频无损音频编码内容的特有配置。

A.4.2.2.2 语法

```
class Avs3AudioLLSpecificConfig {
    unsigned int(4) sampling_frequency_index;
    if (sampling_frequency_index==0xF) {
        unsigned int(24) sampling_frequency;
    }
    unsigned int(1) anc_data_index;
    unsigned int(3) coding_profile;
    unsigned int(8) channel_number;
    unsigned int(2) resolution;
    unsigned int(16) addition_info_length;
    if (addition_info_length > 0) {
        bit(8*addition_info_length) addition_info;
    }
    unsigned int(2) reserved;
}
```

A.4.2.2.3 语义

sampling_frequency_index: 应符合 GY/T 363—2023附录 A 的规定。

sampling_frequency: 应符合 GY/T 363—2023附录 A 的规定。

anc_data_index: 应符合 GY/T 363—2023附录 A 的规定。

coding_profile: 应符合 GY/T 363—2023附录 A 的规定。

channel_number: 应符合 GY/T 363—2023附录 A 的规定。

resolution: 应符合 GY/T 363—2023附录 A 的规定。

addition_info_length: 指示 addition_info 的长度, 以字节为单位。

addition_info: 指示 Avs3AudioLLSpecificConfig 配置中的额外信息。

A. 4. 3 ISO 基本媒体文件格式扩展

A. 4. 3. 1 Audio Vivid 音频解码器配置数据盒

A. 4. 3. 1. 1 定义

数据盒类型: ‘dca3’。

容器: ‘av3a’ 类型的样本入口。

强制性: 强制包含于 ‘av3a’ 类型的样本入口。

数量: 一个。

Audio Vivid 音频解码器配置数据盒 CA3SpecificBox 包含 A. 4. 2中定义的音频编码配置。

A. 4. 3. 1. 2 语法

```
class CA3SpecificBox extends Box( 'dca3' ) {
    unsigned int(4) audio_codec_id;
    if(audio_codec_id == 2){
        Avs3AudioGASpecificConfig() Avs3AudioGAConfig;
    }
    else if (audio_codec_id == 1){
        Avs3AudioLLSpecificConfig() Avs3AudioLLConfig;
    }
}
```

A. 4. 3. 1. 3 语义

Avs3AudioSpecificGAConfig 在 A. 4. 2. 1中定义, Avs3AudioSpecificLLConfig 在 A. 4. 2. 2中定义。

audio_codec_id: 应符合 GY/T 363—2023中附录 A 的规定。

A. 4. 3. 2 Audio Vivid 音频样本入口

A. 4. 3. 2. 1 定义

样本入口类型: ‘av3a’。

容器: Sample Description Box (‘stds’)。

强制性: 封装Audio Vivid音频编码码流的轨道应包含一个 ‘av3a’ 样本入口。

数量: 一个。

AATF类型的Audio Vivid音频编码码流在 ‘av3a’ 类型的音频轨道中进行传输, 其轨道样本入口中应包含一个CA3SpecificBox数据盒。

A. 4. 3. 2. 2 语法

```
class AVS3ATSampleEntry() extends AudioSampleEntry ( 'av3a' ){  
    CA3SpecificBox config  
}
```

A. 4. 3. 2. 3 语义

CA3SpecificBox 提供 Audio Vivid 音频编码码流的解码配置信息。

由于 CA3SpecificBox 已经提供并扩展了对 ChannelCount、SampleSize、SampleRate 的描述，所以本文件解码器需要忽略 AudioSampleEntry 中的 ChannelCount、SampleSize、SampleRate。

A. 4. 3. 3 Audio Vivid 音频样本格式

音频轨道的样本入口类型为 'av3a'，其轨道中的每个样本对应一个 aatf_frame()，其中 aatf_frame() 的定义应符合 GY/T 363—2023 中附录 A 的规定。

附 录 B

(资料性)

超高清音视频编码码流在 HLS 中的封装

超高清音视频编码码流在HLS协议中的传输基于IETF RFC 8216对切片文件格式以及HLS M3U8进行扩展。

HLS传输采用TS或ISOBMFF文件格式，对应AVS2、AVS3、Audio Vivid文件格式见第7章和附录A。

HLS M3U8扩展在EXT-X-STREAM-INF tag中，CODEC属性见附录E ‘codecs’ 参数的定义。

附录 C

(资料性)

超高清音视频编码码流在 RTMP 中的封装

C.1 概述

AVS2视频编码码流、AVS3视频编码码流和Audio Vivid音频编码码流在RTMP中的传输采用FLV文件格式，并对FLV相关字段进行扩展。

C.2 AVS2 视频编码码流在FLV中的封装

AVS2 视频编码码流对应的VideoTagHeader定义见表C.1，VideoTagBody定义见表C.2，AVS2VIDEOPACKET定义见表C.3。

表 C.1 VideoTagHeader 字段定义

字段名称	助记符	定义
FrameType	UB[4]	值为‘1’表示 AVS2 I 帧，值为‘2’表示 AVS2 非 I 帧
CodecID	UB[4]	值为‘13’，表示视频类型为 AVS2
VideoPacketType	UI8	值为‘0’表示 AVS2 视频序列头信息，值为‘1’表示 AVS2 视频帧数据
CompositionTime	SI24	VideoPacketType 为 1 时，该值与 ISO/IEC 14496-12:2022 中 CompositionTime 相同，否则该值为 0

表 C.2 VideoTagBody 定义

字段	描述	定义
VideoTagBody	当 CodecID 等于 13 时，该字段为 AVS2VIDEOPACKET	见表 C.3

表 C.3 AVS2VIDEOPACKET 定义

字段	描述	定义
DATA	如果 VideoPacketType 为 0，该字段为 Avs2DecoderConfigurationRecord； 如果 VideoPacketType 为 1，该字段为一帧 AVS2 视频编码数据	Avs2DecoderConfigurationRecord 定义见 A.2.2

C.3 AVS3 视频编码码流在FLV中的封装

AVS3 视频编码码流对应的 VideoTagHeader 定义见表 C.4，VideoTagBody 定义见表 C.5，AVS3VIDEOPACKET 定义见表 C.6。

表 C.4 VideoTagHeader 字段定义

字段名称	助记符	定义
FrameType	UB[4]	值为‘1’表示 AVS3 I 帧，值为‘2’表示 AVS3 非 I 帧
CodecID	UB[4]	值为‘14’，表示视频类型为 AVS3
VideoPacketType	UI8	值为‘0’表示 AVS3 视频序列头信息，值为‘1’表示 AVS3 视频帧数据
CompositionTime	SI24	VideoPacketType 为 1 时，该值与 ISO/IEC 14496-12:2022 中 CompositionTime 相同，否则该值为 0

表 C.5 VideoTagBody 定义

字段	描述	定义
VideoTagBody	当 CodecID 等于 14 时，该字段为 AVS3VIDEOPACKET	见表 C.6

表 C.6 AVS3VIDEOPACKET 定义

字段	描述	定义
DATA	如果 VideoPacketType 为 0，该字段为 Avs3DecoderConfigurationRecord； 如果 VideoPacketType 为 1，该字段为一帧 AVS3 视频编码数据	Avs3DecoderConfigurationRecord 定义见 A.3.2

C.4 Audio Vivid 音频编码码流在 FLV 中的封装

Audio Vivid 视频编码码流对应的 AUDIODATA 定义见表 C.7，AudioTagBody 定义见表 C.8，AVS3AUDIODATA 定义见表 C.9。

表 C.7 AUDIODATA 字段定义

字段	助记符	定义
SoundFormat	UB[4]	值为‘9’，且 SourdFormatEx 为 17，表示 Audio Vivid 音频
SoundRate	UB[2]	3
SoundSize	UB[1]	1
SoundType	UB[1]	1
SourdFormatEx	UB[16]	值为‘17’，表示该音频类型为 Audio Vivid
AudioPacketType	UI8	值为‘0’表示 Audio Vivid 序列头信息，值为‘1’表示 Audio Vivid 帧数据

表 C.8 AudioTagBody 字段定义

字段	描述	定义
SoundData	当 SoundFormat 为 9 且 SoundFormatEx 为 17 时, 该字段为 AVS3AUDIODATA	见表 C.9

表 C.9 AVS3AUDIODATA 字段定义

字段	描述	定义
Data	当 AudioPacketType 为 0 时, 该字段为 CA3SpecificBox (); 当 AudioPacketType 为 1 时, 该字段为 aatf_frame ()	CA3SpecificBox () 定义见 A.4.3.1。 aatf_frame () 定义见 GY/T 363—2023 附录 A

附录 D (资料性) 超高清音视频编码码流在 DASH 中的封装

D.1 概述

超高清音视频编码码流在 DASH 中的封装基于 ISO/IEC 23009-1 进行扩展。

D.2 AVS3 视频编码码流在 DASH 中的封装

D.2.1 片段格式

D.2.1.1 概述

用于封装 AVS3 视频数据的 DASH 片段格式见 A.3，其样本入口类型支持 ‘avs3’。

D.2.1.2 初始化片段

每个 DASH 初始化片段包含一个 Avs3ConfigurationBox，其中包括 Avs3DecoderConfigurationRecord 解码器配置记录。

D.2.1.3 媒体片段

每个 DASH 媒体片段包含一个或多个 GY/T 368—2023 中规定的编码图像。

AVS3 视频编码图像解码所需的序列头信息在媒体片段中获取。每个 DASH 媒体片段中的第一个媒体样本符合以下任意一个约束。

- a) DASH 媒体片段中的第一个媒体样本是 ISO/IEC 14496-12:2022 附录 I 定义的 SAP_type 为 1 或 2 的 SAP 样本。知识图像不作为参考图像出现在编码视频序列中，即，编码视频序列中的首个序列头语法结构实例符合以下约束：
 - 1) library_stream_flag=0;
 - 2) library_picture_enable_flag=0。
- b) DASH 媒体片段中的第一个媒体样本是 LRAP_type 为 ‘1’ 或 ‘2’ 的 LRAP 样本。知识图像可以作为参考图像出现在编码视频序列中，即，编码视频序列中首个序列头语法结构实例符合以下约束：
 - 1) library_stream_flag=0;
 - 2) library_picture_enable_flag=1。

注：LRAP 样本解码时间之前，作为参考图像样本的所有知识图像样本均可以获得。

D.2.1.4 索引片段

索引片段提供动态自适应流媒体中的一个表示的所有媒体片段的索引。索引片段以一个 ‘styp’ 类型数据盒开始，表示中每个媒体片段包含若干个连续子片段，并由一个或多个符合以下约束条件之一的片段索引数据盒索引。

- a) 每个子片段由一个 ISO/IEC 14496-12:2022 中 8.16.3 定义的 ‘sidx’ 类型的片段索引数据盒索引。‘sidx’ 数据盒指示引用的子片段是否以 SAP 开头，以及子片段中流访问 SAP 的 SAP_type 类型。

- b) 每个子片段由一个 A.3.3.6 中定义的 ‘lidx’ 类型的片段索引数据盒索引。‘lidx’ 数据盒指示引用的子片段是否以 LRAP 开头，以及子片段中 LRAP 的 LRAP_type 类型。

D.2.2 DASH MPD 编码器参数

AVS3 视频编码码流在 MPD 中使用 @codecs 属性表示 “类” 和 “级”，属性值见 IETF RFC 6381 和 E.2 中定义的 MIME 类型的 ‘codecs’ 参数。

D.2.3 DASH MPD 描述子

D.2.3.1 XML 命名空间和规则

本文件中定义并使用了一些新的 XML 元素和属性，这些元素和属性在命名空间 “urn:avs:avs3:p6:2022” 中定义，命名空间指示符 “avs3” 用于指代此命名空间（智能媒体格式）。命名空间指示符 “xs:” 对应 XML Schema 第一部分定义的命名空间 (<http://www.w3.org/2001/XMLSchema>)，数据类型及其含义定义在 XML Schema 第二部分。

D.2.3.2 时域层信息描述子

一个 @schemeIdUri 属性为 “urn:avs:avs3:p6:2022:highest_temporal_id” 的补充属性元素为时域层级描述子，用于指示最高时域层级信息。时域层级描述子的 @value 属性值为最高时域层级标识值。

表示中可以包含多个不同时域层级的编码图像。当 AVS3 视频编码码流支持时域分层时，可以根据 AVS3 视频的主位流中不同时域层级的编码图像生成 DASH 的一个或多个表示，每个表示中包含一个时域层级描述子，并且每个表示中包含从最低时域层级到时域层级描述子指示的最高时域层级的全部时域层级。表示中的 AVS3 视频序列头可以应用于当前表示的全部时域层级。

D.2.3.3 知识位流依赖描述子

一个 @schemeIdUri 属性为 “urn:avs:avs3:p6:2022:LibraryDependency” 的基本属性元素为知识位流依赖描述子。知识位流依赖描述子包含于表示层级，用于指示描述子所适用的表示中封装的基本流类型以及知识位流依赖信息，其中基本流类型包括独立主位流、非独立主位流和知识位流。

该描述子的 @value 元素不出现，知识位流依赖描述子包含的元素及属性见表 D.1。

表 D.1 知识位流依赖描述子元素及属性

描述子元素及属性	类型	使用	描述
LibraryDependency	avs3:LibraryDependencyType	0..1	该元素指示表示层级的 AVS3 位流信息
LibraryDependency@library_dependency_idc	xs:unsignedByte	必选	取值为 0 时指示当前表示仅包含独立主位流； 取值为 1 时指示当前表示仅包含非独立主位流； 取值为 2 时指示当前表示仅包含知识位流； 取值为 3 时指示当前表示包含非独立主位流和知识位流

表 D.1 (续)

描述子元素及属性	类型	使用	描述
LibraryDependency.LibraryInfo	avs3:LibraryInfoType	0..N	@library_dependency_idc 属性取值为 1 或 3 时使用该元素指示表示中的媒体片段与包含知识图像的媒体片段间存在依赖关系, 其属性用于描述被依赖的媒体片段的定位信息和知识图像索引
LibraryDependency.LibraryInfo@library_pid	xs:unsignedInt	必选	指示被依赖媒体片段中包含的知识图像在编解码的索引
LibraryDependency.LibraryInfo@segmentURL	xs:anyURL	必选	指示被依赖媒体片段的定位信息 (URL 或 DASH MPD 中的媒体片段标识符)

LibraryDependency@library_dependency_idc 取值为 ‘1’ 的表示通过@associationId 关联至相应知识位流轨道对应的表示 (该表示的 LibraryDependency@library_dependency_idc 取值为 ‘2’), 且 @associationType 取值等于 ‘a3lr’。

位流依赖描述子的 XML Schema 如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:avs:avs3:p6:2022"
  xmlns:avs3="urn:avs:avs3:p6:2022"
  elementFormDefault="qualified">
  <xs:element name="LibraryDependency" type="avs3:LibraryDependencyType"/>
  <xs:complexType name="LibraryDependencyType">
    <xs:sequence>
      <xs:element name="LibraryInfo">
        <xs:complexType>
          <xs:attribute name="library_pid" type="xs:integer" use="required" />
          <xs:attribute name="segmentURL" type="xs:anyURI" use="required" />
        </xs:complexType>
      </xs:sequence>
      <xs:attribute name="library_dependency_idc" type="xs:unsignedByte" use="required" />
    </xs:complexType>
  </xs:schema>
```

D.2.3.4 彩色信息描述子

AVS3 视频编码码流使用的彩色三基色、彩色信号转换矩阵、光电转移特性信息通过基本属性元素描述子在 MPD 中进行描述, 见表 D.2, AVS3 彩色信息描述子包含于自适应集层级。

表 D.2 AVS3 彩色信息描述子

@schemeIdUri	基本属性元素描述子
urn:avs:avs3:p6:2022:ColourPrimaries	表示 AVS3 视频编码码流使用的彩色三基色, 其@value 取值与媒体片段包含的 AVS3 视频编码码流中 colour_primaries 的值相同
urn:avs:avs3:p6:2022:MatrixCoefficients	表示 AVS3 视频编码码流使用的彩色信号转换矩阵, 其@value 取值与媒体片段包含的 AVS3 视频编码码流中 matrix_coefficients 的值相同
urn:avs:avs3:p6:2022:TransferCharacteristics	表示 AVS3 视频编码码流使用的光电转移特性, 其@value 取值与媒体片段包含的 AVS3 视频编码码流中 transfer_characteristics 的值相同

AVS3 彩色信息描述子符合以下约束。

- a) 当自适应集中的 AVS3 视频编码码流支持 GY/T 155—2000 和 SDR 时, 包含以下基本属性元素描述子:
 - 1) @schemeIdUri=urn:avs:avs3:p6:2022:ColourPrimaries 且@value= ‘1’ ;
 - 2) @schemeIdUri=urn:avs:avs3:p6:2022:MatrixCoefficients 且@value= ‘1’ ;
 - 3) @schemeIdUri=urn:avs:avs3:p6:2022:TransferCharacteristics 且@value= ‘1’ 或 ‘6’ 。
- b) 当自适应集中的 AVS3 视频编码码流支持 GB/T 41809—2022 和 HLG 时, 包含以下基本属性元素描述子:
 - 1) @schemeIdUri=urn:avs:avs3:p6:2022:ColourPrimaries 且@value= ‘9’ ;
 - 2) @schemeIdUri=urn:avs:avs3:p6:2022:MatrixCoefficients 且@value= ‘8’ 或 ‘9’ ;
 - 3) @schemeIdUri=urn:avs:avs3:p6:2022:TransferCharacteristics 且@value= ‘14’ 。
- c) 当自适应集中的 AVS3 视频编码码流支持 GB/T 41809—2022 和 PQ 时, 包含以下基本属性元素描述子:
 - 1) @schemeIdUri=urn:avs:avs3:p6:2022:ColourPrimaries 且@value= ‘9’ ;
 - 2) @schemeIdUri=urn:avs:avs3:p6:2022:MatrixCoefficients 且@value= ‘8’ 或 ‘9’ ;
 - 3) @schemeIdUri=urn:avs:avs3:p6:2022:TransferCharacteristics 且@value= ‘12’ 。

D.3 Audio Vivid 音频编码码流在 DASH 中的封装

D.3.1 片段格式

D.3.1.1 概述

用于封装 Audio Vivid 音频数据的 DASH 片段格式见 A.4 规定的 Audio Vivid 音频编码码流的文件格式, 其样本入口类型支持 ‘av3a’ 。

D.3.1.2 初始化片段

每个 DASH 初始化片段包含一个 CA3SpecificBox 解码器配置记录。

D.3.1.3 媒体片段

每个 DASH 媒体片段包含一个或多个 GY/T 363—2023 中规定的音频编码数据。

每个 DASH 媒体片段中的第一个媒体样本符合以下任意一个约束:

- a) 每个 Audio Vivid 音频样本只能包含一帧且只有一帧 `aatf_frame()`;
- b) 每个 Audio Vivid 音频样本的 `audio_codec_id` 是不变的;
- c) 每个 Audio Vivid 音频样本的 SAP 的类型, 在 ISO/IEC 14496-12:2022 附录 I 定义, 值都为 1。

D.3.1.4 索引片段

Audio Vivid 音频索引片段受以下限制:

- a) 每个子片段由一个 ISO/IEC 14496-12:2022 中 8.16.3 定义的 ‘sidx’ 类型的片段索引数据盒索引;
- b) ‘sidx’ 数据盒指示引用的子片段的 `starts_with_SAP` 为 1, `SAP_type` 为 1。

D.3.2 DASH MPD 编码器参数

Audio Vivid 音频编码码流在 MPD 中的 `@codecs` 属性使用 E.4 中定义的 MIME 类型的 ‘codecs’ 参数。

D.3.3 DASH MPD 描述子

D.3.3.1 @mimeType 属性

`@mimeType` 属性设置为 “audio/mp4”。

D.3.3.2 @audioSampleRate 属性

音频采样率的属性源于 CA3SpecificBox 的 `sampling_frequency_index`。

D.3.3.3 @startWithSAP 属性

`@startWithSAP` 属性设置为 1。

D.3.3.4 AudioChannelConfiguration 描述子

`@schemeIdUri` 属性设置为 “urn:avs:avs3:p7:2024:audio_channel_configuration”, 用于描述音频声道配置的描述。

`@value` 属性值为 3 个字节, 计算方式如下。

- a) 如果 AVS3 音频编码位流在 MPD 中的 `@codecs` 属性为 ‘av3a.01’, 则 `@value` 的属性值:
 - 第 1 个字节为 0xF0;
 - 第 2 个字节等于 A.4.2.2.2 中 `channel_number` 的值;
 - 第 3 个字节为 0。
- b) 如果 AVS3 音频编码位流在 MPD 中的 `@codecs` 属性为 ‘av3a.02’, 则 `@value` 的属性值。
 - 第 1 个字节的高 4 位为 0xF, 低 4 位等于 A.4.2.1.2 中 `content_type` 的值, 其中, `content_type` 的取值范围为 0~3。
 - 第 2 个字节: 如果第 1 个字节等于 0xF0、0xF2 或者 0xF3, 则第 2 个字节最高 1 位等于 0, 低 7 位等于 A.4.2.1.2 中 `channel_number_index` 的值; 否则, 如果第 1 个字节等于 0xF1, 第 2 个字节等于 A.4.2.1.2 中 `number_objects` 的值。
 - 第 3 个字节: 如果第 1 个字节的值等于 0xF2, 则第 3 个字节的值等于 A.4.2.1.2 中 `number_objects` 的值; 否则, 第 3 个字节的值等于 0。

附 录 E

（资料性）

MIME 类型的 ‘codecs’ 参数

E.1 概述

当使用IETF RFC 6381定义的MIME类型的 ‘codecs’ 参数时，如果MIME类型对应本文件中定义的文件格式且 ‘codecs’ 参数值起始于本文件定义的样本入口类型，则 ‘codecs’ 参数的子参数见E.2~E.4。

E.2 AVS2 视频编码参数

如果 ‘codecs’ 参数第一个元素的取值代码对应于GY/T 299.1—2016定义的编码（‘avst’），且对应的轨道表示AVS2视频编码码流，则之后元素分别取值为AVS2解码器配置记录中定义的“类”和“级”字段值。 ‘codecs’ 参数的不同元素之间以句号隔开（“.”），在所有数字编码中，可以省略前导零。“类”和“级”字段如下：

- a) 类（profile_id）： ‘codecs’ 参数的第二个子参数，编码为十六进制数；
- b) 级（level_id）： ‘codecs’ 参数的第三个子参数，编码为十六进制数。

示例：

codecs= avst.22.54

——表示以基准 10 位类（Main 10bit profile），级 8.0.60 编码的 AVS2 视频编码码流。

E.3 AVS3 视频编码参数

如果 ‘codecs’ 参数第一个元素的取值代码对应于GY/T 368—2023定义的编码（‘avs3’），且对应的轨道表示AVS3视频编码码流，则之后元素分别取值为AVS3解码器配置记录中定义的“类”和“级”字段值。 ‘codecs’ 参数的不同元素之间以句号隔开（“.”），在所有数字编码中，可以省略前导零。

“类”和“级”字段如下：

- a) 类（profile_id）： ‘codecs’ 参数的第二个子参数，编码为十六进制数；
- b) 级（level_id）： ‘codecs’ 参数的第三个子参数，编码为十六进制数。

示例：

codecs= avs3.22.54

——表示以基准 10 位类（Main 10bit profile），级 8.0.60 编码的 AVS3 视频编码码流。

codecs= avs3.32.68

——表示以加强 10 位类（High 10bit profile），级 10.0.120 编码的 AVS3 视频编码码流。

E.4 Audio Vivid音频编码参数

Audio Vivid音频codecs参数定义如下：

codecs = ‘av3a.audio_codec_id’。

示例：

codecs= ‘av3a.02’

表示audio_codec_id为2。

参 考 文 献

- [1] GB/T 41809—2022 超高清清晰度电视系统节目制作和交换图像参数值
 - [2] GY/T 155—2000 高清晰度电视节目制作及交换用视频参数值
 - [3] IETF RFC 6381 The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types
 - [4] IETF RFC 8216 HTTP Live Streaming
 - [5] ISO/IEC 14496-15:2022 Information technology — Coding of audio-visual objects - Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format
 - [6] ISO/IEC 23009-1 Information technology — Dynamic adaptive streaming over HTTP (DASH) Part 1: Media presentation description and segment formats
-