GY

中华人民共和国广播电视和网络视听行业标准

GY/T 398. 1—2024

视频浅压缩编码
第 1 部分：超高清视频分层编码

Video mezzanine coding—Part 1：Scalable ultra high definition video coding

2024－01－02 发布　　2024－01－02 实施

国家广播电视总局　发布

目　次

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GY/T 398《视频浅压缩编码》的第1部分。GY/T 398已经发布了以下部分：

——第 1 部分：超高清视频分层编码。

本文件由全国广播电影电视标准化技术委员会（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、国家广播电视总局广播电视规划院、中广电广播电影电视设计研究院有限公司、成都索贝数码科技股份有限公司、四川新视创伟超高清科技有限公司、电子科技大学、北京广播电视台、江苏省广播电视总台、广东广播电视台、吉林广播电视台、陕西广电融媒体集团有限公司、深圳广播电影电视集团、青岛市广播电视台、重庆广播电视集团（总台）、宁夏广播电视台、联通在线信息科技有限公司、北京格非科技股份有限公司、北京中科大洋科技发展股份有限公司、北京新奥特集团有限公司、深圳市康维讯视频科技有限公司、视溪科技（上海）有限公司、成都东方盛行电子有限责任公司、浪潮智能终端有限公司、卓曜（北京）科技有限公司、广州博冠光电科技股份有限公司、成都卓元科技有限公司。

本文件主要起草人：郭晓强、宁金辉、王嘉、王炜、姚平、谢超平、周芸、张金沙、胡潇、龚坤、黎政、朱策、毕江、施冬、曾志群、李志明、任辉、赵为纲、张建栋、王晓、李海平、李韩、鲍放、褚震宇、戴霖、郑珺、林建泉、侯定光、孙业志、张承、王旭耀、张乾、牛睿、刘汉源、姚琼、罗天、张天宇、宋小民、王威、查晓辉、姚仕元、魏娜、付光涛、李小雨、吴成志、罗雷、郭红伟、杨桂明、张宁、邢卫东、刘涛、成六祥、姜殿宇、宋博、郗望、韩轲、白帆、张培仕、封连伟、郝延华、邹西山、林金怡、邓宇翔、李硕、陈域、韩涛、段晓峰、安静。

引　言

GY/T 398《视频浅压缩编码》规定了视频浅压缩编解码方法，适用于节目制播域的视频编解码。考虑到规范内容、应用范围的不同，GY/T 398《视频浅压缩编码》拟由以下两个部分组成。

——第 1 部分：超高清视频分层编码。规定了超高清视频浅压缩分层编码的编码框架、增强层编码位流语法语义及解码过程、图像重建过程、HD-SDI 接口传输方法和文件封装方法，适用于超高清视频节目制播域的视频编码，支持高清制播系统传输 4K 超高清视频、4K 超高清制播系统传输 8K 超高清视频。

——第 2 部分是常规的视频浅压缩编码，不作分层处理，可用于第 1 部分的基本层编码，也可以单独使用，适用于高清或超高清节目制播域的视频编解码。

本文件的发布机构提请注意，声明符合本文件时，可能适用涉及本文件第6章～第10章、附录A和附录D有关内容的相关专利的使用如下：

序号	标准章条编号	专利名称	专利权利人
1	6、10	一种支持高效编辑的超高清视频分层编解码方法	成都索贝数码科技股份有限公司
2	7～9	一种针对视频图像小波变换高频系数的低复杂度编码方法	成都索贝数码科技股份有限公司
3	附录A	视频转换方法及装置	成都索贝数码科技股份有限公司
4	附录D	一种实现HD和4K HDR视频信号同传输的方法以及SDI设备	成都索贝数码科技股份有限公司

本专利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：

专利权利人	联系地址	联系人	邮政编码	电话	电子邮件
成都索贝数码科技股份有限公司	四川省成都市高新区新园南二路2号	龚坤	610041	028-85121111	gongkun@sobey.com

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。

视频浅压缩编码
第1部分：超高清视频分层编码

1 范围

本文件规定了超高清视频浅压缩分层编码的编码框架、增强层编码位流语法语义及解码过程、图像重建过程及HD-SDI接口传输方法和文件封装方法。

本文件适用于超高清视频节目制播域的视频编码。

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 41808—2022 高动态范围电视节目制作和交换图像参数值

GB/T 41809—2022 超高清晰度电视系统节目制作和交换参数值

GY/T 155—2000 高清晰度电视节目制作及交换用视频参数值

GY/T 157—2000 演播室高清晰度电视数字视频信号接口

GY/T 160—2000 数字分量演播室接口中的附属数据信号格式

GY/T 161—2000 数字电视附属数据空间内数字音频和辅助数据的传输规范

ISO/IEC 9899 信息技术 编程语言C（Information technology—Programming languages—C）

3 术语和定义

下列术语和定义适用于本文件。

3.1

小波子带 wavelet subband

经离散小波变换得到的处于特定频段的子信号，由一组图像小波域系数组成。

3.2

奇数帧 odd-numbered frame

在视频序列中，以1开始计数的第1帧、第3帧、第5帧等奇数编号的帧。

3.3

偶数帧 even-numbered frame

在视频序列中，第2帧、第4帧、第6帧等偶数编号的帧。

3.4

奇数场 top field

在隔行扫描中，以1开始计数的第1行、第3行、第5行等奇数行构成的场。

注：奇数场又称为顶场。

3.5

偶数场 bottom field

在隔行扫描中，第2行、第4行、第6行等偶数行构成的场。

注：偶数场又称为底场。

3.6

基本层图像 base layer image

由使用离散小波变换算法从原始图像中提取的低频分量组成的图像。

3.7

增强层图像 enhancement layer image

原始图像与基本层图像通过某种数学方式计算出的残差图像。

3.8

增强层编码位流 enhancement layer coding bitstream

增强层编码数据形成的二进制位流。

注：经解码后可与基本层解码数据一起重建出增强层图像。

3.9

条带 slice

由若干行增强层编码位流组成的结构。

3.10

块 block

由若干增强层图像系数组成的方块状结构。

3.11

块组 block group

由同一个条带中的若干个块组成的结构。

3.12

系数解码 coefficient decoding

从无损压缩的比特位串中恢复出压缩前数据的过程。

3.13

零区块 zero-block

由一定尺寸的全0系数组成的区域。

3.14

Z编码组 Z coding group

对块组的块模式以及零区块进行描述的编码位流。

3.15

P编码组 P coding group

块组中非零区块区域系数的变长前缀编码位流。

3.16

S编码组 S coding group

块组中非零区块区域系数的变长后缀编码位流。

3.17

反量化 inverse quantization

将量化系数缩放后映射为重构系数的过程。

3.18

量化参数 quantization parameter

对量化系数进行反量化的参数。

3.19

量化步长 quantization step

将信号的连续取值近似为若干个离散取值区间的步长。

3.20

量化权重 quantization weight

根据不同小波子带系数的重要性不同而设计的参数，其影响量化参数。

3.21

死区反量化器 dead-zone inverse quantizer

一种反量化器，0所在区间的量化步长不同于其他区间的量化步长。

3.22

反哈达玛变换 inverse Hadamard transform

使用特定正交矩阵对系数进行反变换的过程。

3.23

块内系数反重排 inverse rearrangement of in-block coefficients

对块内系数的顺序进行重新排列的过程。

3.24

条带反映射 inverse reflection of slice

将条带内的解码后系数反映射到YUV平面进行排列的过程。

注：Y代表图像像素的亮度分量，U代表图像像素的蓝色色度分量，V代表图像像素的红色色度分量，U和V共同描述像素的颜色。

4 缩略语

下列缩略语适用于本文件。

ADF 附属数据标志（Ancillary Data Flag）

CRCC 循环冗余校验码（Cyclic Redundancy Check Code）

CS 检验和（CheckSum）

DBN 数据块序号数（Data Block Number）

DC 数据计数（Data Count）

DID 数据标志（Data IDentifier）

DWT 离散小波变换（Discrete Wavelet Transform）

EAV 有效视频结束（End of Active Video）

HD 高清（High Definition）

HDR 高动态范围（High Dynamic Range）

HD-SDI 高清串行数字接口（High Definition Serial Digital Interface）

HLG 混合对数伽马（Hybrid Log Gamma）

IDWT 离散小波反变换（Inverse Discrete Wavelet Transform）

LN 行序号数（Line Number）

LSB 最低有效位（Least Significant Bit）

SUVC 超高清视频分层编码（Scalable Ultra high definition Video Coding）

MSB 最高有效位（Most Significant Bit）
PQ 感知量化（Perceptual Quantization）
SAV 有效视频开始（Start of Active Video）
SDR 标准动态范围（Standard Dynamic Range）
UDW 用户数据字（User Data Word）

5 约定

5.1 通则

本文件使用的数学运算符和优先级与C语言类似，应符合ISO/IEC 9899的规定，但对整型除法和算术移位操作进行了特定定义。除特别说明外，约定编号和计数从0开始。

5.2 算术运算符

算术运算符定义见表1。

表 1 算术运算符定义

算术运算符	定义
+	加法运算
−	减法运算
*	乘法运算
/	整除运算，沿向 0 的取值方向截断
$\frac{a}{b}$	整除运算，沿向 0 的取值方向截断
<<	算术左移运算
>>	算法右移运算

5.3 逻辑运算符

逻辑运算符定义见表2。

表 2 逻辑运算符定义

逻辑运算符	定义
a && b	a 和 b 之间的逻辑与运算
a || b	a 和 b 之间的逻辑或运算
!	逻辑非运算

5.4 关系运算符

关系运算符定义见表3。

表 3 关系运算符定义

关系运算符	定义
>	大于
>=	大于或等于
<	小于
<=	小于或等于
==	等于
!=	不等于

5.5 位运算符

位运算符定义见表4。

表 4 位运算符定义

位运算符	定义
&	与运算
|	或运算
～	取反运算

5.6 赋值运算符

赋值运算符定义见表 5。

表 5 赋值运算符定义

赋值运算符	定义
=	赋值运算
++	递增运算
——	递减运算
+=	自加指定值运算
-=	自减指定值运算

5.7 位流语法、解析过程和解码过程的描述方法

5.7.1 通则

位流语法描述方法类似 C 语言。解析过程和解码过程采用文字和类似 C 语言的伪代码描述。

5.7.2 函数定义

read_bits(n)

返回位流的随后 n 个二进制位，MSB 在前，同时位流指针前移 n 个二进制位。如果 n 为 0，则返回 0，位流指针不前移。

5.7.3 描述符

描述符表示不同语法元素的解析过程，见表 6。

表 6 描述符定义

描述符	定义
f(n)	取特定值的连续 n 个二进制位。解析过程由函数 read_bits(n)的返回值规定
u(n)	n 位无符号整数。在语法表中，如果 n 是“v”，其位数由其他语法元素值确定。解析过程由函数 read_bits(n)的返回值规定，该返回值用高位在前的二进制表示
me(v)	块编码模式的变长编码语法元素。解析过程在 9.2.3 中定义
pe(v)	系数变长编码的前缀码语法元素。解析过程在 9.2.5 中定义

6 编码框架

6.1 通则

超高清视频浅压缩分层编码支持图像格式应为 YUV 4:2:2、10bit、GB/T 41808—2022 规定的 HDR（HLG 或 PQ）、GB/T 41809—2022 规定的 BT.2020 色域的 4K 超高清视频和 8K 超高清视频的浅压缩编码。

6.2 4K 超高清视频浅压缩分层编码框架

4K 超高清视频经过 DWT 变换后，分层进行 HD SDR 基本层编码、HD HDR 增强层编码、4K HDR 增强层编码，可分别得到 HD SDR 基本层码流、HD HDR 增强层码流、4K HDR 增强层码流。HD HDR 增强层码流和 4K HDR 增强层码流应符合第 7 章～第 11 章的规定。DWT 变换应符合附录 A 的规定，4K 超高清视频浅压缩分层编码过程见附录 B。

4K 超高清视频浅压缩分层编码框架见图 1。

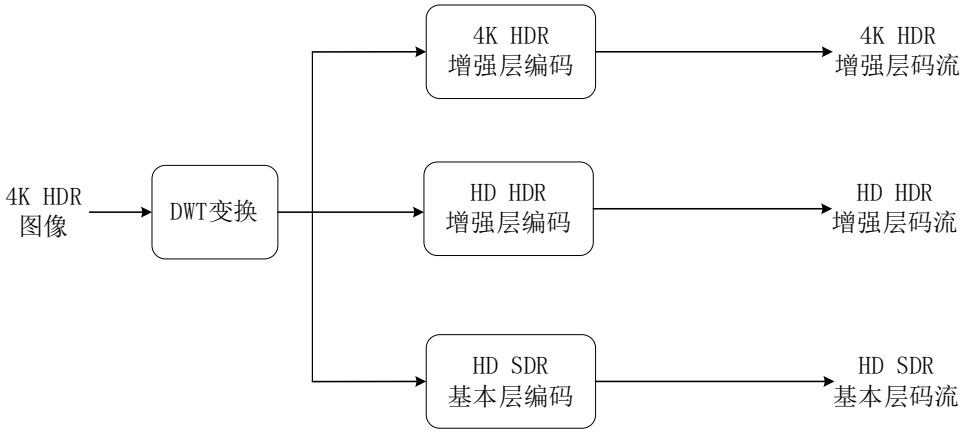

图1 4K 超高清视频浅压缩分层编码框架

6.3 8K 超高清视频浅压缩分层编码框架

8K 超高清视频经过 DWT 变换后，分层进行 4K 基本层编码、8K 增强层编码，可分别得到 4K 基本层码流、8K 增强层码流，4K 基本层码流和 8K 增强层码流应符合第 7 章～第 11 章的规定。DWT 变换应符合附录 A 的规定，8K 超高清视频浅压缩分层编码过程见附录 B。

8K 超高清视频浅压缩分层编码框架见图 2。

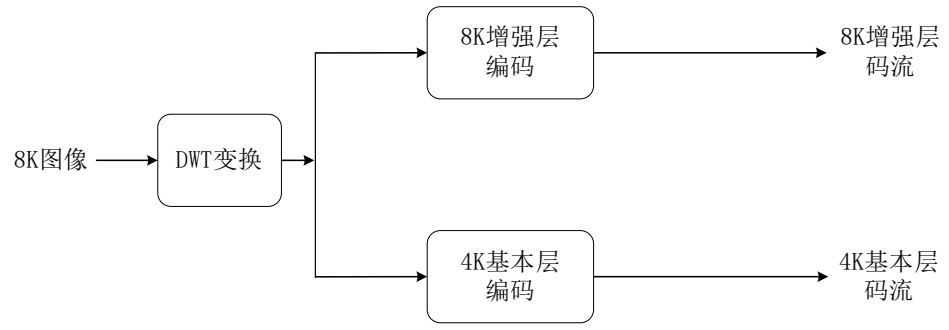

图2 8K 超高清视频浅压缩分层编码框架

7 增强层图像和编码位流结构

7.1 增强层图像

每帧增强层图像是一帧小波域图像。小波域图像和原始图像具有相同的高度和宽度。对于做了一次水平小波变换和一次垂直小波变换后的小波域图像，具有垂直低频水平低频 LL、垂直低频水平高频 LH、垂直高频水平低频 HL、垂直高频水平高频 HH 四个子带，Y、U、V 分量各自具有 LL、LH、HL、HH 四个子带，其中 LL 被替换为基本层对 LL 编码后的残差，见图 3。

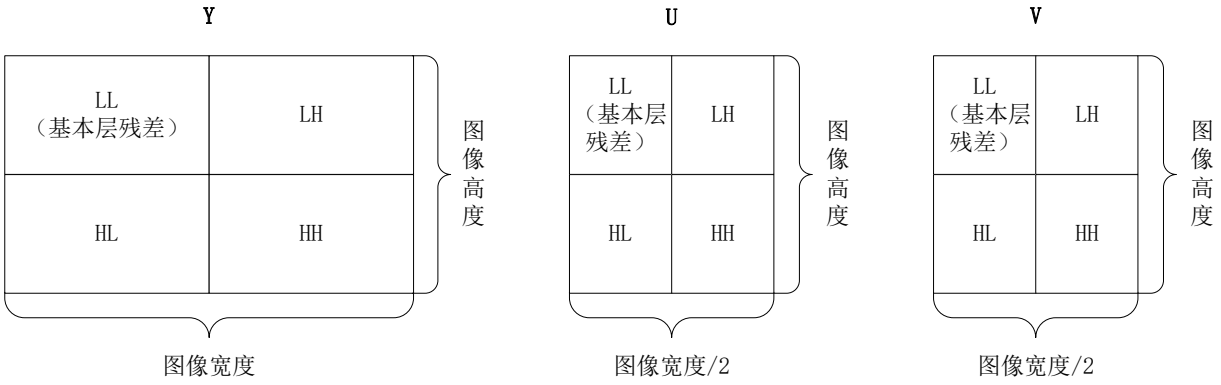

图3 对 YUV 4:2:2 图像做了一次水平小波变换和一次垂直小波变换后的增强层图像

7.2 增强层编码位流结构

增强层编码位流序列由若干帧独立的增强层图像编码位流所构成。每帧增强层图像的编码位流由图像头和若干个条带构成，增强层编码位流结构应符合图 4 的规定。

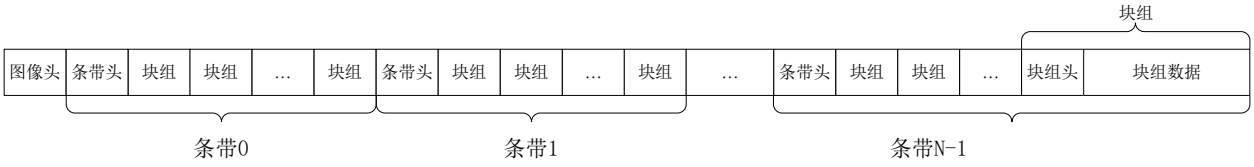

图4 增强层编码位流结构

图像头记录一帧增强层图像的基本信息，以及对这一帧增强层编码位流进行解码所需的必要信息。

一个条带对应增强层图像的若干行，包含 Y、U、V 三个分量各个小波子带的完整编码数据。条带的基本信息记录在条带头中。

一个条带由若干个块组组成。块组是增强层语法结构的基本单元，其基本信息记录在块组头中。块组头后面记录的即是块组数据的编码位流。

8 位流的语法和语义

8.1 语法描述

8.1.1 增强层图像位流结构

增强层图像位流结构见表7。

表 7　增强层图像位流结构

增强层图像定义
picture(){
picture_header()
for(sliceIdx = 0; sliceIdx < SliceCount; sliceIdx++) {
slice_header()
for(bgIdx = 0; bgIdx< SliceBlockGroupCount; bgIdx++) {
block_group_header()
block_group_data()
}
}
}

8.1.2　图像头定义

图像头定义见表 8。

表 8　图像头定义

图像头定义	描述符
picture_header() {	
pich_syncwords	f(64)
frame_bytes_count	u(32)
pich_size	u(8)
version	u(8)
bit_depth	u(8)
chroma	u(8)
width	u(16)
height	u(16)
slice_height	u(16)
block_width	u(8)
block_height	u(8)
block_group_size	u(8)
dwt_horizontal_count	u(8)
dwt_vertical_count	u(8)
inverse_hadamard_size	u(8)
vlc_mode_option	u(16)
reserved1	u(272)
quantizer_type	u(8)
weight_table_size	u(8)
weight_table	u(240)
reserved2	u(256)
}	

8.1.3　条带头定义

条带头定义见表 9。

表 9　条带头定义

条带头定义	描述符
slice_header() {	
slice_syncwords	f(32)
slice_index	u(16)
slice_bytes_count	u(24)
slice_qp	u(8)
}	

8.1.4　块组头定义

块组头定义见表 10。

表 10　块组头定义

块组头定义	描述符
block_group_header() {	
block_group_bytes_count	u(16)
}	

8.1.5　块组数据定义

块组数据定义见表11。

表 11　块组数据定义

块组数据定义	描述符
block_group_data() {	
Z_data()	
byte_align_padding	u(v)
P_data()	
byte_align_padding	u(v)
S_data()	
byte_align_padding	u(v)
}	

8.1.6　Z 编码组定义

Z编码组数据定义见表12。

表 12　Z 编码组数据定义

Z 编码组数据定义	描述符
Z_data() {	
for(blockIdx = 0; blockIdx < block_group_size; blockIdx++) {	
block_mode_code[blockIdx];	me(v)
if(BlockMode[blockIdx] == 1) {	
if(BlockCoeffCount == 256) {	
for(i = 0; i < 4; i++)	
z64_flag[blockIdx, i];	u(1)
for(i = 0; i < 4;i++)	
if(z64_flag[blockIdx, i] == 1)	
for(j = 0; j < 4; j++)	

表 12（续）

Z 编码组数据定义	描述符
z16_flag[blockIdx, i * 4 + j];	u(1)
for(i = 0; i < 16; i++)	
if(z16_flag[blockIdx, i] == 1)	
for(j = 0; j < 4; j++)	
z4_flag[blockIdx, i * 4 + j];	u(1)
}	
else if(BlockCoeffCount == 64) {	
for(i = 0; i < 4; i++)	
z16_flag[blockIdx, i];	u(1)
for(i = 0; i < 4;i++)	
if(z16_flag[blockIdx, i] == 1)	
for(j = 0; j < 4; j++)	
z4_flag[blockIdx, i * 4 + j];	u(1)
}	
}	
else if(BlockMode[blockIdx] == 2) {	
for(i = 0; i < BlockCoeffCount/4; i++)	
z4_flag[blockIdx, i];	u(1)
}	
if(BlockMode[blockIdx] == 1 || BlockMode[blockIdx] == 2) {	
for(i = 0; i < BlockCoeffCount / 4; i++){	
if(z4_flag[blockIdx, i] == 1)	
vlc_flag[blockIdx, i];	u(1)
}	
for(i = 0; i < BlockCoeffCount / 4; i++) {	
if(vlc_flag[blockIdx, i] == 0)	
pattern0001_code[blockIdx, i];	u(3)
}	
}	
}	
}	

8.1.7 P 编码组定义

P编码组数据定义见表13。

表 13 P 编码组数据定义

P 编码组数据定义	描述符
P_data() {	
for(blockIdx = 0; blockIdx < block_group_size; blockIdx++) {	
if(BlockMode[blockIdx] == 1 || BlockMode[blockIdx]==2) {	
for(i = 0; i< BlockCoeffCount/4; i++){	
if(z4_flag[blockIdx, i] == 1 && vlc_flag[blockIdx, i] == 1) {	
for(j = 0; j < 4; j++)	
vlc_prefix_code[blockIdx, i * 4 + j];	pe(v)
}	
}	
}	
else if(BlockMode[blockIdx] == 3 || BlockMode[blockIdx] == 4) {	
for(i = 0; i < BlockCoeffCount; i++)	

表 13（续）

P 编码组数据定义	描述符
vlc_prefix_code[blockIdx, i];	pe(v)
}	
}	
}	

8.1.8 S 编码组定义

S编码组数据定义见表14。

表 14 S 编码组数据定义

S 编码组数据定义	描述符
S_data() {	
for(blockIdx = 0; blockIdx < block_group_size; blockIdx++) {	
if(BlockMode[blockIdx] == 1 || BlockMode[blockIdx] == 2 || BlockMode[blockIdx] == 3) {	
for(i = 0; i < BlockCoeffCount; i++)	
if(VLCPrefixLen[blockIdx, i] >= 5)	
suffix[blockIdx, i];	u(v)
}	
else if(BlockMode[blockIdx] == 4) {	
for(i = 0; i < BlockCoeffCount; i++)	
if(VLCPrefixLen[blockIdx, i] >= 1)	
suffix[blockIdx, i];	u(v)
}	
}	
}	

8.2 语义描述

8.2.1 图像头语义

图像头同步字 pich_syncwords

8 字节固定字符串，SUVCPICH。表示图像头开始。

帧字节数 frame_bytes_count

32 位无符号整数。表示该帧的编码字节数（包含图像头字节数）。

版本号 version

8 位无符号整数。表示编码码流的版本，当前版本为 1。

图像头大小 pich_size

64 位无符号整数。表示图像头的字节数，固定大小为 0x0080。

位深 bit_depth

8 位无符号整数。表示编码系数的比特位深度，位深的取值应为 12。

色度采样 chroma

8 位无符号整数。表示编码系数的 YUV 色度采样方式，见表 15。

表 15　色度采样方式

chroma	含义
1	4:2:2
其他	预留

图像宽度　width

16 位无符号整数。表示该编码帧的图像宽度。

图像高度　height

16 位无符号整数。表示该编码帧的图像高度。

条带高度　slice_height

16 位无符号整数。表示该编码帧的条带高度。条带高度应与块高度相等。

块宽度　block_width

8 位无符号整数。表示块的宽度。块宽度取值为 16、32。当块高度为 16 时，块宽度应为 16；当块高度为 8 时，块宽度应为 32；当块高度为 4 时，块宽度应为 16。

块高度　block_height

8 位无符号整数。表示块的高度。块高度取值为 16、8、4。

块组大小　block_group_size

8 位无符号整数。表示一个块组中包含的块的个数。块组大小的最小取值为 1，最大取值为 60。

小波水平变换次数　dwt_horizontal_count

8 位无符号整数。表示该层增强层数据的小波水平变换次数，取值为 1。

小波垂直变换次数　dwt_vertical_count

8 位无符号整数。表示该层增强层数据的垂直水平变换次数，取值为 1。

反哈达玛变换尺寸　inverse_hadamard_size

8 位无符号整数。表示反哈达玛变换所使用矩阵的尺寸，见表 16。

表 16　反哈达玛变换尺寸

inverse_hadamard_size	含义
0	不使用反哈达玛变换
2	使用矩阵大小2×2的反哈达玛变换
其他	预留

变长编码模式选择　vlc_mode_option

16 位无符号整数。表示变长编码所使用的具体编码方式，默认取值为 3。

预留字段1　reserved1

第 1 个预留字段区域。272 个预留比特位。

量化器类型　quantizer_type

8 位无符号整数。表示量化器类型，见表 17。

表 17　量化器类型

quantizer_type	含义
0	量化使用死区量化器，反量化时使用死区反量化器
其他	预留

量化权重表大小　weight_table_size

8位无符号整数。表示量化权重表的大小。表大小应和YUV图像做小波变换后的子带个数相同，取值应为12。

量化权重表　weight_table

24个8位无符号整数。前12个数依次表示YUV图像做小波变换后的各个子带的量化参数相对于条带量化参数slice_qp的增量，子带排列顺序为LL-Y、LL-U、LL-V、LH-Y、LH-U、LH-V、HL-Y、HL-U、HL-V、HH-Y、HH-U、HH-V。后12个数取值应为0。

预留字段2　reserved2

第2个预留字段区域。256个预留比特位。

8.2.2　条带头语义

条带头同步字　slice_syncwords

4字节固定字符串，SLIC。表示条带头开始。

条带序号　slice_index

16位无符号整数。表示该条带在该编码帧中的条带序号。

条带字节数　slice_bytes_count

24位无符号整数。表示该条带的编码字节数（包含条带头）。

条带量化参数　slice_qp

8位无符号整数。表示该条带的量化参数。

8.2.3　块组头语义

块组字节数　block_group_bytes_count

16位无符号整数。表示该块组的编码字节数（包含块组头）。

8.2.4　Z编码组语义

块编码模式码　block_mode_code[blockIdx]

块组中第blockIdx个块的编码模式码。根据块编码模式码可解码出块编码模式BlockMode[blockIdx]。块编码模式共有5种，具体含义和解析过程见9.2.3。

64系数组非零标志　z64_flag[blockIdx,i]

1位无符号整数。表示块组中第blockIdx个块的第i个64系数组的非零标志。如果该标志值为0，表示该64系数组的所有系数为0；如果该标志值为1，表示该64系数组至少存在一个非零系数。

16系数组非零标志　z16_flag[blockIdx,i]

1位无符号整数。表示块组中第blockIdx个块的第i个16系数组的非零标志。如果该标志值为0，表示该16系数组的所有系数为0；如果该标志值为1，表示该16系数组至少存在一个非零系数。

4系数组非零标志　z4_flag[blockIdx,i]

1位无符号整数。表示块组中第blockIdx个块的第i个4系数组的非零标志。如果该标志值为0，表示该4系数组的所有系数为0；如果该标志值为1，表示该4系数组至少存在一个非零系数。

4系数组变长编码标志　vlc_flag[blockIdx,i]

1位无符号整数。表示块组中第blockIdx个块的第i个4系数组是否使用变长编码。如果该标志为0，表示该4系数组属于0001样式，由0001样式码定长编码表示；如果该标志为1，表示该4系数组使用变长编码表示。变长编码数据存放于P编码组和S编码组中。

4系数组0001样式码　pattern0001_code[blockIdx,i]

3位无符号整数。表示块组中第blockIdx个块的第i个4系数组的0001样式码。根据0001样式码可解码出该4系数组每个系数的值。0001样式共有8种取值可能，具体含义和解析过程见9.2.4。

8.2.5　P 编码组语义

变长编码前缀码　vlc_prefix_code[blockIdx,i]

块组中第blockIdx块的第i个系数的变长编码前缀码。根据变长编码前缀码可解码出变长编码前缀位数VLCPrefixLen[blockIdx,i]。具体解析过程见9.2.5。

8.2.6　S 编码组语义

变长编码后缀　suffix[blockIdx,i]

块组中第blockIdx块的第i个系数的变长编码后缀。变长编码后缀位数VLCSuffixLen[blockIdx,i]由块编码模式和变长编码前缀位数决定。具体解析过程见9.2.6。

8.2.7　字节对齐填充语义

字节对齐填充　byte_align_padding

如果位流当前的比特位数不是8的整数倍，则存在数量为1至7个的填充比特，使位流的比特位数补齐到8的整数倍。如果位流当前的比特位数是8的整数倍，则没有填充比特。

9　解码过程

9.1　增强层位流解码过程

增强层位流数据的解码过程见图5。编码数据经码流解析后进行系数解码，系数解码具体过程为：

a)　解码 Z 编码组的位流数据得到数组 z4table[]；
b)　以数组 z4table[]作为输入，解码 P 编码组的位流数据得到数组 prefix[]；
c)　以数组 z4table[]、数组 prefix[]作为输入，解码 S 编码组的位流数据得到块组系数解码数组 v[]；
d)　依据量化表对块组系数解码数组 v[]进行反量化得到块组反量化数组 dq[]；
e)　对块组反量化数组 dq[]进行反哈达玛变换得到块组反哈达玛变换数组 dh[]；
f)　对块组反哈达玛变换数组 dh[]进行反重排和反映射得到最终的增强层解码数据。

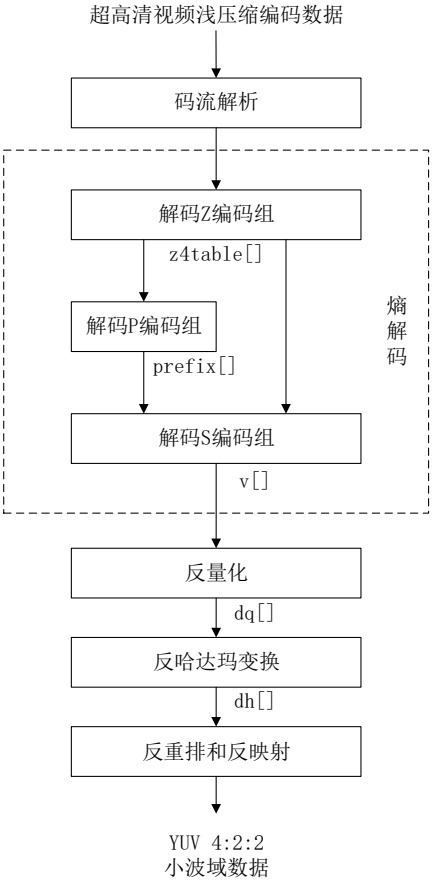

图5　增强层位流解码过程

9.2　位流解析

9.2.1　位流头解析

位流头解析按照图像头、条带头、块组头的顺序依次解析。

当位流中出现图像头同步字“SUVCPICH”时，可定位到图像头。解析图像头，可以获得图像的各个基本参数。从帧字节数字段可以获得该帧位流数据的总字节数大小，从图像头大小字段可以获得图像头的字节数，同时可以推算获得第0个条带的起始位置。

图像的第0个条带紧接着图像头之后出现。解析条带头，可以获得条带的各个参数。从条带字节数字段可以获得该条带的字节数，同时可以推算获得下一个条带的起始位置。

每个条带的第0个块组头紧接着条带头之后出现。解析块组头，可以获得块组的各个参数。从块组字节数字段可以获得该块组的字节数，同时可以推算获得下一个块组的起始位置。

9.2.2　解析解码所需变量

条带数量　SliceCount

表示图像包含的条带数量。条带数量由公式（1）推导获得。

$$\mathrm{SliceCount}=\frac{\mathrm{height}/2+\mathrm{slice\_height}-1}{\mathrm{slice\_height}} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$

块系数数量　BlockCoeffCount

表示图像中每个块包含的系数数量。块系数数量由公式（2）推导获得。

$$BlockCoeffCount = block\_width \times block\_height \quad (2)$$

块4系数组数量　BlockZ4GroupCount

表示每个块包含的4系数组数量。块4系数组数量由公式（3）推导获得。

$$BlockZ4GroupCount = \frac{BlockCoeffCount}{4} \quad (3)$$

块16系数组数量　BlockZ16GroupCount

表示每个块包含的16系数组数量。块16系数组数量由公式（4）推导获得。

$$BlockZ16GroupCount = \frac{BlockCoeffCount}{16} \quad (4)$$

块64系数组数量　BlockZ64GroupCount

表示每个块包含的64系数组数量。块64系数组数量由公式（5）推导获得。

$$BlockZ64GroupCount = \frac{BlockCoeffCount}{64} \quad (5)$$

块组系数数量　BlockGroupCoeffCount

表示每个块组包含的系数数量。块组系数数量由公式（6）推导获得。

$$BlockGroupCoeffCount = BlockCoeffCount \times block\_group\_size \quad (6)$$

条带块组数量　SliceBlockGroupCount

表示每个条带包含的块组的数量。条带块组数量由公式（7）推导获得。

$$SliceBlockGroupCount = \frac{width \times 4 \times slice\_height}{BlockGroupCoeffCount} \quad (7)$$

块组序号　block_group_index

表示在一个条带中块组的序号。一个条带包含slice_block_group_count个块组，块组序号block_group_index依次为0，1，2，…，slice_block_group_count-1。

块组子带序号　block_group_band_index

表示一个块组所对应的小波子带序号。对于chroma、dwt_horizontal_count、dwt_vertical_count均等于1的情形，一共有12个小波子带，块组子带序号依次为0，1，2，…，11。块组子带序号与子带内块组序号范围的对应关系见表18。

表 18　块组子带序号与子带内块组序号范围的对应关系

块组子带序号	小波子带	子带内块组序号范围
0	LL-Y	[0, SliceBlockGroupCount×2/16-1]
1	LL-U	[SliceBlockGroupCount×2/16, SliceBlockGroupCount×3/16-1]
2	LL-V	[SliceBlockGroupCount×3/16, SliceBlockGroupCount×4/16-1]
3	LH-Y	[SliceBlockGroupCount×4/16, SliceBlockGroupCount×6/16-1]
4	LH-U	[SliceBlockGroupCount×6/16, SliceBlockGroupCount×7/16-1]
5	LH-V	[SliceBlockGroupCount×7/16, SliceBlockGroupCount×8/16-1]
6	HL-Y	[SliceBlockGroupCount×8/16, SliceBlockGroupCount×10/16-1]
7	HL-U	[SliceBlockGroupCount×10/16, SliceBlockGroupCount×11/16-1]
8	HL-V	[SliceBlockGroupCount×11/16, SliceBlockGroupCount×12/16-1]
9	HH-Y	[SliceBlockGroupCount×12/16, SliceBlockGroupCount×14/16-1]
10	HH-U	[SliceBlockGroupCount×14/16, SliceBlockGroupCount×15/16-1]
11	HH-V	[SliceBlockGroupCount×15/16, SliceBlockGroupCount×16/16-1]

9.2.3　块编码模式码 me(v)的解析过程

块编码模式码是变长编码。块编码模式码block_mode_code与块编码模式BlockMode的对应关系及含义见表19。

表 19　块编码模式对应关系及含义

块编码模式码 block_mode_code	位数	块编码模式 BlockMode	含义
0	1	0	本块的所有系数全为0
10	2	1	本块存在非0系数，使用多个层级的系数组非零标志数组描述，对系数的编码方式同时使用0001样式定长编码和第一套变长编码 [a]
110	3	2	本块存在非0系数，只使用4系数组非零标志数组描述，对系数的编码方式同时使用0001样式定长编码和第一套变长编码 [a]
1110	4	3	本块存在非0系数，不使用系数组非零标志数组描述，对系数的编码方式只使用第一套变长编码 [a]
1111	4	4	本块存在非0系数，不使用系数组非零标志数组描述，对系数的编码方式只使用第二套变长编码 [b]
[a] 第一套变长编码解码伪代码见 9.3.4。 [b] 第二套变长编码解码伪代码见 9.3.4。			

9.2.4　4 系数组 0001 样式码的解析过程

4系数组0001样式码是3位定长编码。4系数组0001样式码与该组4个系数值的对应关系见表20。

表 20　4 系数组 0001 样式码与该组 4 个系数值的对应关系

4系数组0001样式码	4系数组解码数值
000	1, 0, 0, 0
001	-1, 0, 0, 0
010	0, 1, 0, 0
011	0, -1, 0, 0
100	0, 0, 1, 0
101	0, 0, -1, 0
110	0, 0, 0, 1
111	0, 0, 0, -1

9.2.5　变长编码前缀码 pe(v)的解析过程

对变长编码前缀码的解析，从位流的当前位置开始寻找第一个非零位，找到的零位个数即为变长编码前缀位数VLCPrefixLen。pe(v)解析伪代码如下。

```
VLCPrefixLen[blockIdx, i] = 0;
while(read_bits(1) == 0)
      VLCPrefixLen[blockIdx, i]++;
```

9.2.6　变长编码后缀的解析过程

变长编码后缀位数VLCSuffixLen，由块编码模式和变长编码前缀位数推导得出。变长编码后缀解析伪代码如下。

```
VLCSuffixLen[blockIdx, i] = 0;
if(BlockMode[blockIdx] == 1 || BlockMode[blockIdx] == 2 || BlockMode[blockIdx] == 3){
  if(VLCPrefixLen[blockIdx, i] >= 5)
    VLCSuffixLen[blockIdx, i] = VLCSuffixLen[blockIdx, i] - 4;
}
else if(BlockMode[blockIdx] == 4){
  if(VLCPrefixLen[blockIdx, i] >= 1)
    VLCSuffixLen[blockIdx, i] = VLCSuffixLen[blockIdx, i];
}
suffix[blockIdx, i] = read_bits(VLCSuffixLen[blockIdx, i]);
```

9.2.7 语义变量的初始化过程

在对一帧增强层位流数据进行解析解码之前，应对以下语义变量进行初始化：

——z64_flag 数组中的每个值应初始化为 0；

——z16_flag 数组中的每个值应初始化为 0；

——z4_flag 数组中的每个值应初始化为 0；

——vlc_flag 数组中的每个值应初始化为 1。

9.3 系数解码

9.3.1 系数解码综述

系数解码以块组为基本单元，每个块组都可以独立解码。一个块组的编码位流，由块组头、Z编码组、P编码组和S编码组构成，块组位流结构见图6。

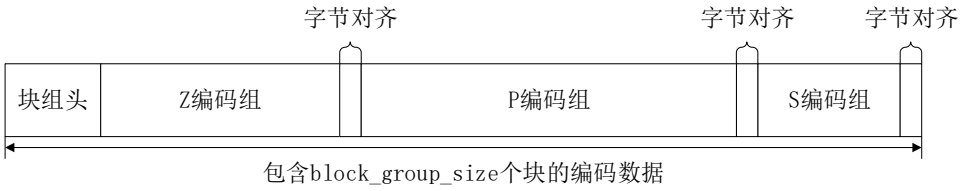

图6 块组位流结构示意图

若块组头记录的块组字节数等于2，则表示该块组的所有系数全部为0，该块组无需系数解码。若块组头记录的块组字节数大于2，表示该块组存在非0数据，应按照9.3.2～9.3.4所述方法进行系数解码。

9.3.2 Z 编码组的解码过程

输入：Z编码组的位流数据。

输出：4系数组的标志数组z4table[]。

一个块组包含block_group_size数量的块，Z编码组依次包含这些块的Z编码位流，Z编码组结构见图7。

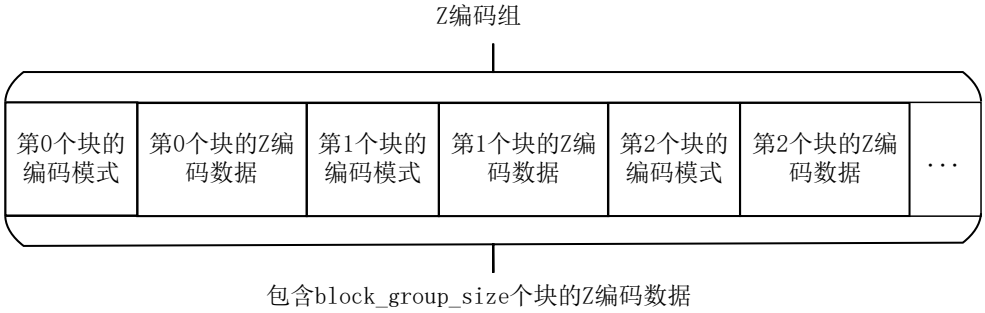

图7　Z 编码组的结构示意图

每个块的编码模式共有5种，含义见9.2.3。

对于块编码模式0，该块所有系数全为0，不存在Z编码数据。

对于块编码模式3，该块不存在Z编码数据，所有系数使用第1套变长编码，变长编码的前缀码字和后缀码字分别记录在P编码组和S编码组。

对于块编码模式4，该块不存在Z编码数据，所有系数使用第2套变长编码，变长编码的前缀码字和后缀码字分别记录在P编码组和S编码组。

对于块编码模式1，如果块系数数量为256，则该块的Z编码数据由64系数组的非零标志、16系数组的非零标志、4系数组的非零标志、非零4系数组的编码模式、0001样式的4系数组编码数据依次组成。块编码模式1的Z编码数据结构见图8。

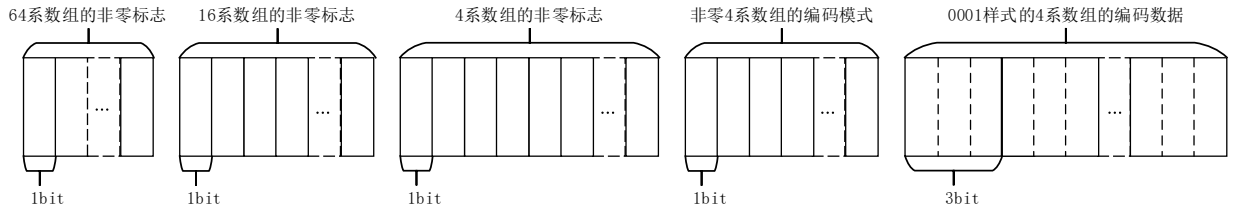

图8　块编码模式为 1 的 Z 编码数据结构示意图

64系数组的非零标志表示连续64个系数是否全为0，每个非零标志用1个bit表示。如果该bit为0，表示该组的64个系数全为0；如果该bit为1，表示该组的64个系数不全为0。

16系数组的非零标志表示连续16个系数是否全为0，每个非零标志用1个bit表示。如果该bit为0，表示该组的16个系数全为0；如果该bit为1，表示该组的16个系数不全为0。在64系数组非零标志里面已经表示过为全0系数的64系数组，已隐含表示了其包含的所有16系数组为全0，故不用标记特殊标识。

4系数组的非零标志表示连续4个系数是否全为0，每个非零标志用1个bit表示。如果该bit为0，表示该组的4个系数全为0；如果该bit为1，表示该组的4个系数不全为0。在16系数组非零标志里面已经表示过为全0系数的16系数组，已隐含表示了其包含的所有4系数组为全0，故不用标记特殊标识。

如果块系数数量为64，则不存在64系数组的非零标志，该块的Z编码数据由16系数组的非零标志、4系数组的非零标志、非零4系数组的编码模式、0001样式的4系数组编码数据依次组成。

非零4系数组的编码模式表示该组的4个系数的具体编码模式，每个模式用1个bit表示。如果该bit为0，表示该组的4个系数属于0001样式（3个系数为0，1个系数为±1），具体样式记录在0001样式4系数组编码数据；如果该bit为1，表示该组的4个系数使用第1套变长编码，变长编码的前缀码字和后缀码字分别记录在P编码组和S编码组里。

对于块编码模式2，该块的Z编码数据由4系数组的非零标志、非零4系数组的编码模式、0001样式的4系数组编码数据依次组成。块编码模式2的Z编码数据结构见图9。

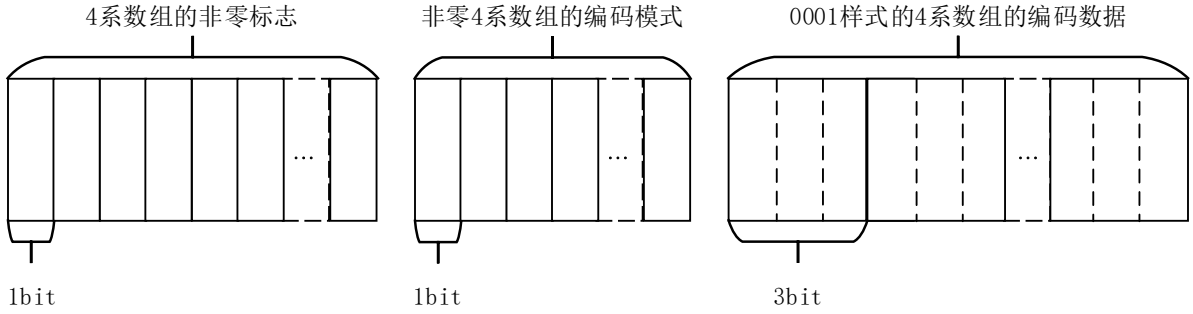

图9　块编码模式为 2 的 Z 编码数据结构示意图

解码Z编码组，得到4系数组标志数组z4table[]。z4table[]共有11种取值，含义见表21。

表 21　4 系数组标志数组取值含义

取值	含义
0	该4个系数的值全为0
1	该4个系数的值依次为1，0，0，0
2	该4个系数的值依次为-1，0，0，0
3	该4个系数的值依次为0，1，0，0
4	该4个系数的值依次为0，-1，0，0
5	该4个系数的值依次为0，0，1，0
6	该4个系数的值依次为0，0，-1，0
7	该4个系数的值依次为0，0，0，1
8	该4个系数的值依次为0，0，0，-1
9	该4个系数使用第一套变长编码，需要解码P编码组和S编码组
10	该4个系数使用第二套变长编码，需要解码P编码组和S编码组

解码Z编码组得到z4table[]的伪代码如下。

```
for(blockIdx = 0; blockIdx < block_group_size; blockIdx++){
  if(BlockMode[blockIdx] == 0){
    for(i = 0; i < BlockZ4GroupCount; i++)
      z4table[blockIdx * BlockZ4GroupCount + i] = 0;
  }
  else if(BlockMode[blockIdx] == 1 || BlockMode[blockIdx] == 2){
    for(i = 0; i < BlockZ4GroupCount; i++){
      if(z4_flag[blockIdx, i] == 0)
        z4table[blockIdx * BlockZ4GroupCount + i] = 0;
      else if(z4_flag[blockIdx, i] == 1){
        if(vlc_flag[blockIdx, i] == 0)
          z4table[blockIdx * BlockZ4GroupCount + i] = pattern0001_code[blockIdx, i] + 1;
        else if(vlc_flag[blockIdx, i] == 1)
          z4table[blockIdx * BlockZ4GroupCount + i] = 9;
      }
    }
  }
  else if(BlockMode[blockIdx] == 3){
    for(i = 0; i < BlockZ4GroupCount; i++)
      z4table[blockIdx * BlockZ4GroupCount + i] = 9;
  }
  else if(BlockMode[blockIdx] == 4){
    for(i = 0; i < BlockZ4GroupCount; i++)
      z4table[blockIdx * BlockZ4GroupCount + i] = 10;
  }
}
```

9.3.3 P 编码组的解码过程

输入：P编码组的位流数据，4系数组的标志数组z4table[]。

输出：P编码组的解码数据prefix[]。

计算块组内P编码组的解码数据prefix[]的伪代码如下。

```
for(z4group = 0; z4group < block_group_size * BlockZ4GroupCount; z4group++){
  blockIdx = z4group / BlockZ4GroupCount;
  groupIdx = z4group % BlockZ4GroupCount;
  if(z4table[z4group] == 9 || z4table[z4group] == 10){
    for(i = 0; i < 4; i++)
      prefix[z4group * 4 + i] = VLCPrefixLen[blockIdx, groupIdx * 4 + i];
  }
  else{
    for(i = 0; i < 4; i++)
      prefix[z4group * 4 + i] = 0;
  }
}
```

9.3.4 S 编码组的解码过程

输入：S编码组的位流数据，P编码组的解码数据prefix[]，4系数组的标志数组z4table[]。

输出：块组的解码数据v[]。

计算块组的解码数据v[]的伪代码如下。

```
for(z4group = 0; z4group < block_group_size * BlockZ4GroupCount; z4group++){
  blockIdx = z4group / BlockZ4GroupCount;
  groupIdx = z4group % BlockZ4GroupCount;
  if(z4table[z4group] == 9){
    for(i = 0; i < 4; i++)
      v[z4group * 4 + i] = GetVLC_mode1(prefix[z4group * 4 + i], suffix[blockIdx, groupIdx * 4 + i]);
  }
  else if(z4table[z4group] == 10){
    for(i = 0; i < 4; i++)
      v[z4group * 4 + i] = GetVLC_mode2(prefix[z4group * 4 + i], suffix[blockIdx, groupIdx * 4 + i]);
  }
  else if (z4table[z4group] >= 1 && z4table[z4group] <= 8){
    v[z4group * 4 + 0] = GetPattern0001(z4table[z4group], 0);
    v[z4group * 4 + 1] = GetPattern0001(z4table[z4group], 1);
    v[z4group * 4 + 2] = GetPattern0001(z4table[z4group], 2);
    v[z4group * 4 + 3] = GetPattern0001(z4table[z4group], 3);
  }
  else{
    for(i = 0; i < 4; i++)
      v[z4group * 4 + i] = 0;
  }
}
```

输入：0001样式ptn。

输出：系数在4系数组中的序号idx。

GetPattern0001的伪代码如下。

```
GetPattern0001(ptn, idx){
  if((ptn==1 && idx==0) || (ptn==3 && idx==1) || (ptn==5 && idx==2) || (ptn==7 && idx==3))
    v = 1;
  else if((ptn==2 && idx==0) || (ptn==4 && idx==1) || (ptn==6 && idx==2) || (ptn==8 && idx==3))
    v = -1;
  else
    v = 0;
}
```

第一套变长编码的解码伪代码GetVLC_mode1()如下。

```
GetVLC_mode1(prefix, suffix){
  if(prefix == 0)
    return 0;
  else if(prefix == 1)
    return -1;
  else if(prefix == 2)
    return 1;
  else if(prefix == 3)
    return -2;
  else if(prefix == 4)
    return 2;
  else{
    v = (suffix >> 1) + (1 << (prefix - 5)) + 2;
    if(suffix & 1 == 1)
      v = 0 - v;
    return v;
  }
}
```

当使用第一套变长编码时，P编码组的解码数据prefix、S编码组的码字suffix及系数解码数值v的对应关系见表22。

表 22　第一套变长编码的解码对应表

prefix	suffix	系数解码数值v
0	无	0
1	无	-1
2	无	1
3	无	-2
4	无	2
5	1	-3
5	0	3
6	01	-4
6	00	4
6	11	-5
6	10	5
7	001	-6
7	000	6
7	011	-7
7	010	7
7	101	-8
7	100	8
7	111	-9
7	110	9
…	…	…
16	111111111011	-4095
16	111111111010	4095

第二套变长编码的解码伪代码GetVLC_mode2()如下。

```
GetVLC_mode2(prefix, suffix){
  if(prefix == 0)
    return 0;
  else{
    v =  (suffix >> 1) + (1 << (prefix  -  1));
    if(suffix & 1 == 1)
      v = 0 - v;
    return v;
  }
}
```

当使用第二套变长编码时，P编码组的解码数据prefix、S编码组的码字suffix及系数解码数值v的对应关系见表23。

表 23　第二套变长编码的解码对应表

prefix	suffix	系数解码数值v
0	无	0
1	1	-1
1	0	1
2	01	-2
2	00	2

表 23（续）

prefix	suffix	系数解码数值v
2	11	-3
2	10	3
3	001	-4
3	000	4
3	011	-5
3	010	5
3	101	-6
3	100	6
3	111	-7
3	110	7
4	0001	-8
4	0000	8
4	0011	-9
4	0010	9
…	…	…
12	111111111111	-4095
12	111111111110	4095

9.4 反量化

输入：块组的解码数据v[]，块组序号block_group_index，条带量化参数slice_qp，量化权重表weight_table[]。

输出：块组的反量化数据dq[]。

量化表共分为88档，量化参数qp取值为0～87。量化参数qp和量化步长qstep的关系应符合表24的规定。

表 24　量化参数 qp 和量化步长 qstep 的关系

qp	qstep	qp	qstep	qp	qstep	qp	qstep
0	1	22	7	44	48	66	320
1	1.125	23	7.5	45	52	67	352
2	1.25	24	8	46	56	68	384
3	1.375	25	9	47	60	69	416
4	1.5	26	10	48	64	70	448
5	1.625	27	11	49	72	71	480
6	1.75	28	12	50	80	72	512
7	1.875	29	13	51	88	73	576
8	2	30	14	52	96	74	640
9	2.25	31	15	53	104	75	704
10	2.5	32	16	54	112	76	768
11	2.75	33	18	55	120	77	832
12	3	34	20	56	128	78	896
13	3.25	35	22	57	144	79	960
14	3.5	36	24	58	160	80	1024
15	3.75	37	26	59	176	81	1152
16	4	38	28	60	192	82	1280
17	4.5	39	30	61	208	83	1408
18	5	40	32	62	224	84	1536
19	5.5	41	36	63	240	85	1664
20	6	42	40	64	256	86	1792
21	6.5	43	44	65	288	87	1920

对于系数解码后的块组，其块组序号为block_group_index，块组所属于的小波子带序号block_group_band_index由9.2.2中所述的方法获得。计算块组的反量化数据dq[]的伪代码如下。

```
qp = slice_qp + weight_table[block_group_band_index];
if(qp < 0)
  qp = 0;
if(qp > 87)
  qp = 87;
for(i = 0; i < block_group_coeff_count; i++){
  dq[i] = v[i] * qstep;
}
```

为获得好的图像重建效果，编码时的量化过程应符合下述过程：

——输入：哈达玛变换后的块组系数 coeffs[]，块组序号 block_group_index，条带量化参数 slice_qp，量化权重表 weight_table[]；

——输出：块组的量化系数 v[]。

量化参数qp和量化步长qstep的关系应符合表24的规定。计算量化的伪代码如下。

```
qp = slice_qp + weight_table[block_group_band_index];
if(qp < 0)
  qp = 0;
if(qp > 87)
  qp = 87;
for(i = 0; i < block_group_coeff_count; i++){
  if(coeffs[i] > 0)
    v[i] = (coeffs[i] + qstep / 3) / qstep;
  else if(coeffs[i] < 0)
    v[i] = (coeffs[i] - qstep / 3) / qstep;
  else
    v[i] = 0;
}
```

9.5 反哈达玛变换

输入：块组的反量化数据dq[]。

输出：块组的反哈达玛变换数据dh[]。

如果图像头中记录的inverse_hadamard_size为2，需要对反量化数据进行2×2的反哈达玛变换。2×2反哈达玛变换所使用的矩阵如下。

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

计算块组的反哈达玛变换数据dh[]和编码时使用的哈达玛变换伪代码如下；编码时输入为块组的重排数据，输出为块组的哈达玛变换数据。

```
if(inverse_hadamard_size == 2){
  for(k = 0; k < BlockGroupCoeffCount; k+=4){
    for(i = 0; i < 4; i++){
      dh[k + i] = 0;
      for(j = 0; j < 4; j++)
        dh[k + i] += dq[k + j] * HDM2x2[i * 4 + j];
      dh[k + i] = (dh[k + j] + 1) >> 1;
    }
  }
}
else {
  for(k = 0; k < BlockGroupCoeffCount; k++)
    dh[k] = dq[k];
}
```

9.6 反重排和反映射

9.6.1 块内系数反重排

对于高度为4，宽度为16的块，将其反重排为16×4的形式，应符合图10的规定。

0	1	4	5	16	17	20	21	32	33	36	37	48	49	52	53
2	3	6	7	18	19	22	23	34	35	38	39	50	51	54	55
8	9	12	13	24	25	28	29	40	41	44	45	56	57	60	61
10	11	14	15	26	27	30	31	42	43	46	47	58	59	62	63

图10 16×4 块内系数反重排

对于高度为8，宽度为32的块，将其反重排为32×8的形式，应符合图11的规定。

0	1	4	5	16	17	20	21	64	65	68	69	80	81	84	85	128	129	132	133	144	145	148	149	192	193	196	197	208	209	212	213
2	3	6	7	18	19	22	23	66	67	70	71	82	83	86	87	130	131	134	135	146	147	150	151	194	195	198	199	210	211	214	215
8	9	12	13	24	25	28	29	72	73	76	77	88	89	92	93	136	137	140	141	152	153	156	157	200	201	204	205	216	217	220	221
10	11	14	15	26	27	30	31	74	75	78	79	90	91	94	95	138	139	142	143	154	155	158	159	202	203	206	207	218	219	222	223
32	33	36	37	48	49	52	53	96	97	100	101	112	113	116	117	160	161	164	165	176	177	180	181	224	225	228	229	240	241	244	245
34	35	38	39	50	51	54	55	98	99	102	103	114	115	118	119	162	163	166	167	178	179	182	183	226	227	230	231	242	243	246	247
40	41	44	45	56	57	60	61	104	105	108	109	120	121	124	125	168	169	172	173	184	185	188	189	232	233	236	237	248	249	252	253
42	43	46	47	58	59	62	63	106	107	110	111	122	123	126	127	170	171	174	175	186	187	190	191	234	235	238	239	250	251	254	255

图11 32×8 块内系数反重排

对于高度为16，宽度为16的块，将其反重排为16×16的形式，应符合图12的规定。

0	1	4	5	16	17	20	21	64	65	68	69	80	81	84	85
2	3	6	7	18	19	22	23	66	67	70	71	82	83	86	87
8	9	12	13	24	25	28	29	72	73	76	77	88	89	92	93
10	11	14	15	26	27	30	31	74	75	78	79	90	91	94	95
32	33	36	37	48	49	52	53	96	97	100	101	112	113	116	117
34	35	38	39	50	51	54	55	98	99	102	103	114	115	118	119
40	41	44	45	56	57	60	61	104	105	108	109	120	121	124	125
42	43	46	47	58	59	62	63	106	107	110	111	122	123	126	127
128	129	132	133	144	145	148	149	192	193	196	197	208	209	212	213
130	131	134	135	146	147	150	151	194	195	198	199	210	211	214	215
136	137	140	141	152	153	156	157	200	201	204	205	216	217	220	221
138	139	142	143	154	155	158	159	202	203	206	207	218	219	222	223
160	161	164	165	176	177	180	181	224	225	228	229	240	241	244	245
162	163	166	167	178	179	182	183	226	227	230	231	242	243	246	247
168	169	172	173	184	185	188	189	232	233	236	237	248	249	252	253
170	171	174	175	186	187	190	191	234	235	238	239	250	251	254	255

图12　16×16 块内系数反重排

9.6.2　条带反映射

对于chroma、dwt_horizontal_count、dwt_vertical_count均等于1的情形，条带的小波子带排列应与图13相符合，块组排列应与图14相符合。

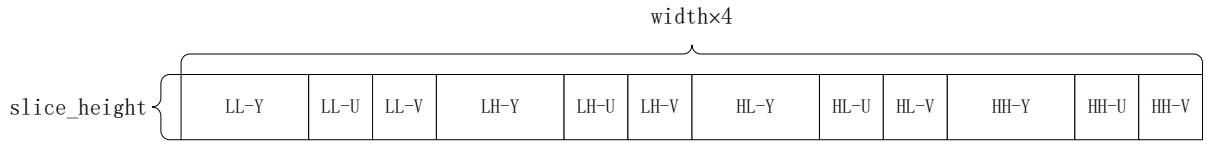

图13　条带的小波子带排列

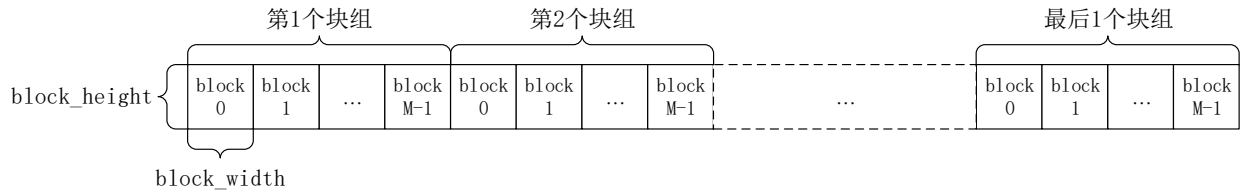

图14　条带的块组排列

对于chroma、dwt_horizontal_count、dwt_vertical_count均等于1的情形，从条带到YUV 4:2:2的小波域数据的条带反映射方法应与图15相符合。

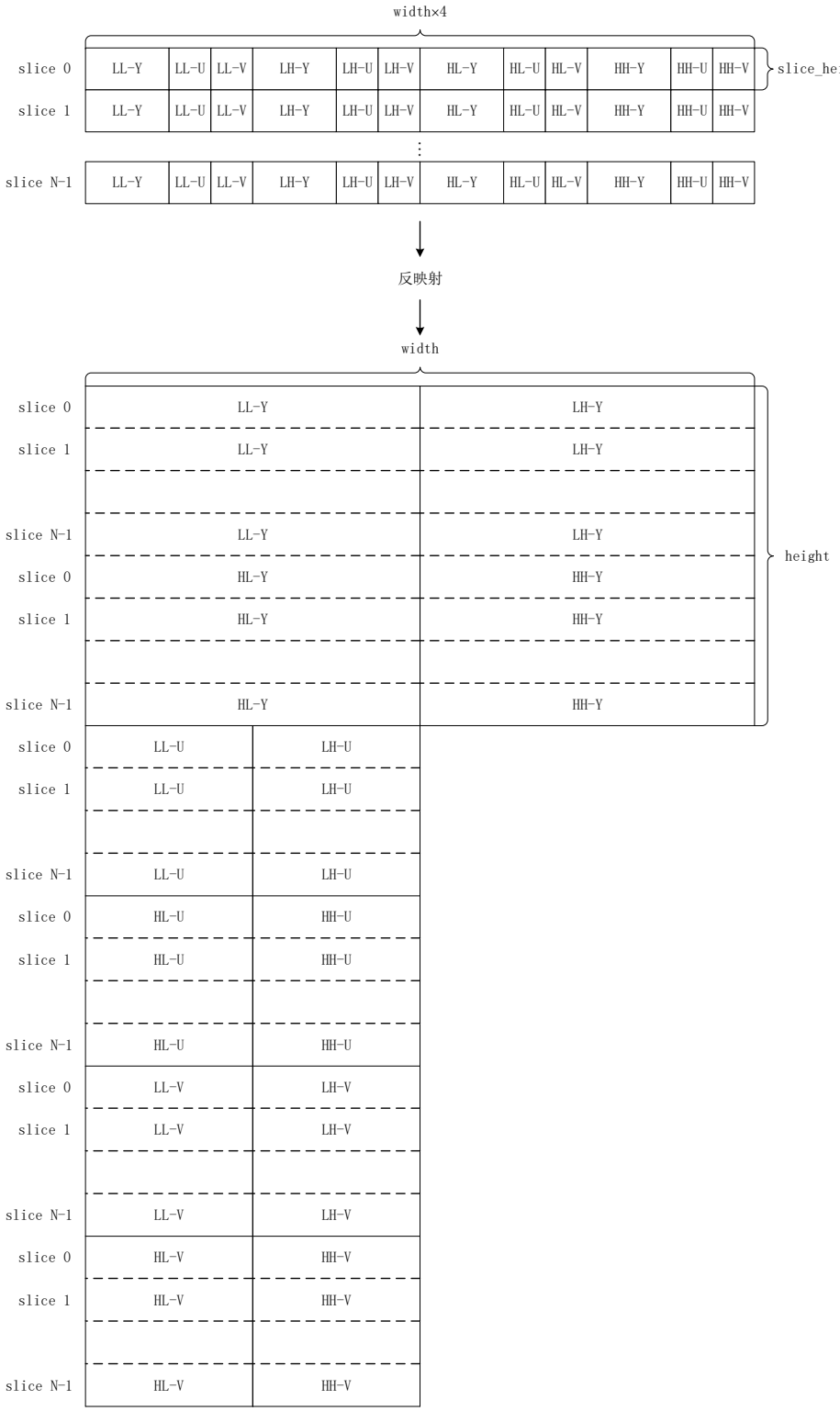
width×4
slice 0
slice 1
slice N-1
LL-Y
LL-U
LL-V
LH-Y
LH-U
LH-V
HL-Y
HL-U
HL-V
HH-Y
HH-U
HH-V
slice_height
反映射
width
height

图15　条带反映射

10 图像重建过程

10.1 4K 超高清图像重建过程

4K超高清图像重建过程步骤如下：

a) 解码HD SDR基本层码流，得到HD SDR隔行扫描重建图像，基本层码流的解码方法不属于本文件的范围；

b) 对HD SDR隔行扫描重建图像进行SDR到HDR的动态范围上变换和GY/T 155—2000规定的BT.709色域到GB/T 41809—2022规定的BT.2020色域的上变换，得到HD HDR隔行扫描重建图像；

c) 解码HD HDR增强层码流，得到LL’、LH’、HL’、HH’四个重建小波子带；

d) 对c)得到的四个重建小波子带采用IDWT算法进行一次水平反变换，得到LL-L重建残差和LL-H重建小波子带；

e) 将HD HDR隔行扫描重建图像与LL-L重建残差逐像素相加，得到LL-L重建小波子带，即HD HDR隔行扫描重建图像；

f) 若HD增强图像为奇数场图像，对e)得到的LL-L重建残差和d)得到的LL-H重建小波子带采用IDWT算法进行一次垂直反变换（见表A.11），得到LL重建小波子带，即HD HDR逐行扫描图像；

g) 若HD增强图像为偶数场图像，对e)得到的LL-L重建残差和d)得到的LL-H重建小波子带采用IDWT算法进行一次偶数场垂直反变换（见表A.12），得到LL重建小波子带，即HD HDR逐行扫描图像；

h) 解码4K HDR增强层码流，得到Zeros、LH、HL、HH四个重建小波子带；

i) 将LL、LH、HL、HH采用IDWT算法进行一次水平反变换和一次垂直反变换，得到4K HDR重建图像。

4K超高清图像重建过程见图16。

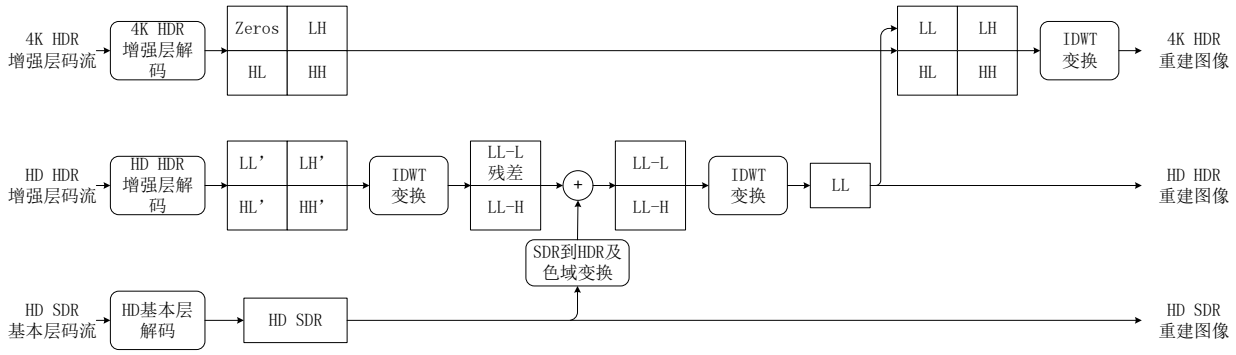

图16 4K 超高清图像重建过程

10.2 8K 超高清图像重建过程

8K超高清图像重建过程步骤如下：

a) 解码4K基本层码流，得到LL’重建小波子带，即4K超高清重建图像，基本层码流的解码方法不属于本文件的范围；

b) 解码8K增强层码流，得到LL残差、LH、HL、HH四个重建小波子带；

c) 将LL’重建小波子带与LL残差逐像素相加，得到LL重建小波子带；

d) 对LL、LH、HL、HH重建小波子带采用IDWT算法进行一次水平反变换和一次垂直反变换，得到8K超高清重建图像。

8K超高清图像重建过程见图17。

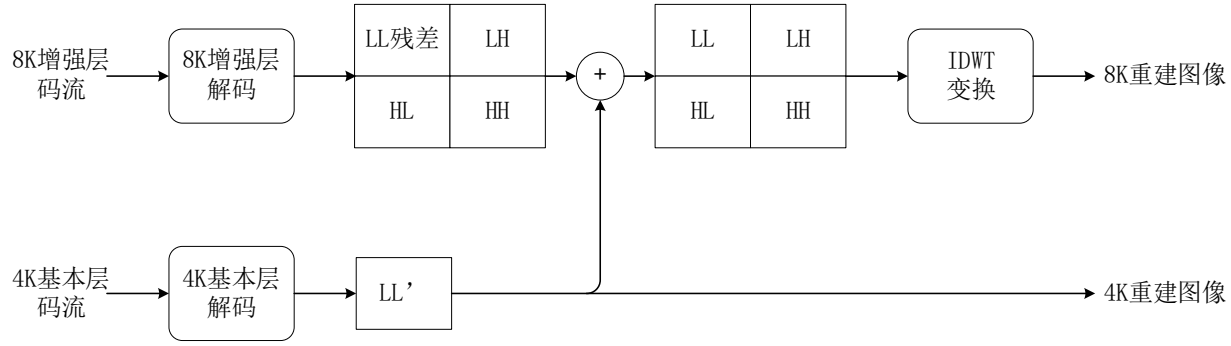

图17　8K 超高清图像重建过程

11　文件封装和 HD-SDI 传输

超高清视频浅压缩分层编码码流的文件封装方法见附录 C。

超高清视频浅压缩分层编码码流的 HD-SDI 传输方法应符合附录 D 的规定。

支持超高清视频浅压缩分层编码的文件封装辅助信息结构见附录 E。

附 录 A
（规范性）
DWT 变换

A.1 概述

本附录规定了编码过程与解码过程中超高清浅压缩分层编码使用的基于Le Gall 5/3小波的DWT算法与IDWT算法。

A.2 位宽规定和位宽缩放

A.2.1 位宽规定

为了保证DWT和IDWT计算精度以及基本层和增强层的编码质量，对各层的位宽做以下规定。

——基本层的位宽空间为 10bit。基本层编码的输入，以及基本层解码的输出，均为 10bit。

——增强层的位宽空间为 12bit。增强层编码的输入，以及增强层解码的输出，均为 12bit。

——DWT 和 IDWT 的计算以及基本层残差的计算，应在位宽为 20bit 的位宽空间中进行。

——在编码过程与解码过程中，做 DWT 和 IDWT 计算时，从低位宽空间到高位宽空间，应按 A.2.2 的方式做位宽放大处理。从高位宽空间到低位宽空间，应按 A.2.3 的方式做位宽缩小处理，同时应按 A.2.4 的方式做钳位处理。

A.2.2 位宽放大

位宽放大的伪代码见表A.1。

输入：小波系数数组X[x]，数组的规模Z，位宽放大的位数bits。

输出：已被位宽放大的小波系数数组X[x]。

表 A.1 位宽放大伪代码

伪代码	注释
upscale_bits(Z, bits) {	
for(i = 0; i < Z; i = i + 1) {	
X[i] = X[i] << bits;	位宽放大bits位
}	
}	

A.2.3 位宽缩小

位宽缩小的伪代码见表A.2。

输入：小波系数数组X[x]，数组的规模Z，位宽缩小的位数bits。

输出：已被位宽缩小的小波系数数组X[x]。

表 A.2 位宽缩小伪代码

伪代码	注释
downscale_bits(Z, bits) {	
for(i = 0; i < Z; i = i + 1) {	
X[i] = (X[i] + (1 << (bits - 1))) >> bits;	位宽缩小bits位，并做四舍五入
}	
}	

A.2.4 钳位

钳位的伪代码见表A.3。

输入：小波系数数组X[x]，数组的规模Z，位宽空间bits。

输出：已被钳位的小波系数数组X[x]。

表 A.3 钳位伪代码

伪代码	注释
clamp(Z, bits) {	
for(i = 0; i < Z; i = i + 1) {	
if(X[i] < 0)	如果系数小于0
X[i] = 0;	系数最小值钳位为0
if(X[i] > (1 << bits) - 1)	如果系数大于位宽空间的最大值
X[i] = (1 << bits) - 1;	系数最大值钳位为位宽空间的最大值
}	
}	

A.3 对称扩展

DWT算法与IDWT算法在计算过程中都需要将临时数组X中的样本扩展到条带边界上，计算伪代码见表A.4。

输入：小波系数数组X[x],数组X的规模Z。假设该数组位置0到Z-1已被填充。

输出：已被对称扩展的小波系数数组X。

表 A.4 对称扩展计算伪代码

伪代码	注释
extend_sample(Z) {	
for(i = 1; i ≤ 2; i = i + 1) {	在临时数组边界之外的两个样本上循环
X[-i] = X[i];	在左边界映射样本
X[Z + i - 1] = X[Z - i - 1];	在右边界映射样本
}	
}	

A.4 DWT算法

A.4.1 概述

编码过程中，对输入4K、8K图像采用DWT算法的具体处理见第6章，涉及的水平小波变换、垂直小波变换、偶数帧垂直小波变换见A.4.2与A.4.3。

A.4.2 水平小波变换

输入：分量索引k，输出小波滤波器类型β_0和两种输入滤波器类型，低通β_L和高通β_H；临时输出条带T中的小波系数[β_0，x，y]。

输出：临时输出条带[β_L，x，y]和[β_H，x，y]中的过滤小波系数。

水平小波变换的计算伪代码见表A.5。

表 A.5　水平小波变换计算伪代码

伪代码	注释
HFDWT(k, β_0, β_L, β_H) {	将低通系数β_L和高通系数β_H进行水平变换到分量k中的输出条带β_0
for(y = 0; y < H_b[β_0,k]; y = y + 1) {	迭代条带b的所有行
for(x = 0; x< W_b[β_0,k]; x = x + 1) {	迭代条带b的所有列
X[x] = T[β_0, x, y];	将输入系数复制到临时行
}	
extend_samples(W_b[β_0,k])	对称地将样本X穿过边界
filter_1D(W_b[β_0,k])	对临时数组X执行小波滤波
for(x = 0; x < W_b[b]; x = x + 1) {	迭代条带b的所有列
i =⌊x / 2⌋;	计算源条带中的输入样本位置
if(x umod 2 == 0) {	如果是偶数样本位置
T[β_L, i, y] = Y[x];	将偶数样本分配给低通输出
}	
else {	如果是奇数样本位置
T[β_H, i, y] = Y[x];	将奇数样本分配给高通输出
}	
}	
}	
}	

其中filter_1D见表A.8。

A.4.3　垂直小波变换

输入：分量索引k，输出小波滤波器类型β_0和两种输入滤波器类型，低通β_L和高通β_H；临时输出条带T中的小波系数[β_0, x, y]。

输出：临时条带[β_L, x, y]和[β_H, x, y]中的过滤小波系数。

垂直小波变换的计算伪代码见表A.6，偶数帧垂直小波变换的计算伪代码见表A.7。

表 A.6　垂直小波变换计算伪代码

伪代码	注释
VFDWT(k, β_0, β_L, β_H) {	将输入条带β_0的系数垂直变换到低通系数β_L和高通系数β_H
for(x = 0; x< W_b[β_0,k]; x = y + 1) {	迭代所有列
for(y = 0; y < H_b[β_0,k]; y = y + 1) {	迭代所有行
X[y]=T[β_0,x,y];	将输入系数复制到临时行
}	
extend_samples(H_b[β_0,k])	对称地将样本X穿过边界
filter_1D(H_b[β_0,k])	对临时数组X执行一维小波滤波
for(y = 0; y < H_b[β_0,k]; y = y + 1){	迭代所有行
i =⌊y / 2⌋;	计算源条带中的输入样本位置
if(y umod 2 == 0){	如果是偶数样本位置
T[β_L, x, i] = Y[y];	将偶数样本分配给低通输出
}	
else {	如果是奇数样本位置
T[β_H, y, i] = Y[y];	将奇数样本分配给高通输出
}	
}	
}	
}	

表 A.7　偶数帧垂直小波变换计算伪代码

伪代码	注释
VFDWT_even (k, β_0, β_L, β_H) {	将输入条带β_0的系数垂直变换到低通系数β_L和高通系数β_H
for(x = 0; x< $W_b[\beta_0,k]$; x = y + 1) {	迭代所有列
for(y = 0; y < $H_b[\beta_0,k]$; y = y + 1) {	迭代所有行
X[y]=T[β_0,x,y];	将输入系数复制到临时行
}	
extend_samples($H_b[\beta_0,k]$)	对称地将样本X穿过边界
filter_1D_even($H_b[\beta_0,k]$)	对临时数组X执行偶数帧一维小波滤波
for(y = 0; y < $H_b[\beta_0,k]$; y = y + 1) {	迭代所有行
i = ⌊y / 2⌋;	计算源条带中的输入样本位置
if(y umod 2 == 0) {	如果是偶数样本位置
T[β_L, x, i] = Y[y];	将偶数样本分配给低通输出
}	
else {	如果是奇数样本位置
T[β_H, y, i] = Y[y];	将奇数样本分配给高通输出
}	
}	
}	
}	

其中filter_1D_even见表A.9。

A.4.4　小波滤波

基于Le Gall 5/3小波的一维小波滤波计算伪代码见表A.8，偶数帧一维小波滤波计算伪代码见表A.9。

输入：小波系数数组X[x]，数组X的规模Z。

输出：小波变换系数的数组Y[x]，至少对索引0到Z-1有效。

表 A.8　一维小波滤波计算伪代码

伪代码	注释
filter_1D (Z) {	
for(i = -1; i < Z + 1; i = i + 2) {	迭代奇数样本
Y[i] = X[i] − ((X[i−1] + X[i + 1]) >> 1)	在奇数样本中生成高通
}	
for(i = 0; i < Z; i = i + 2) {	迭代偶数样本
Y[i] = X[i] + ((Y[i −1] + Y[i + 1]) + 2 >> 2)	更新偶数样本以生成低通
}	
}	

表 A.9　偶数帧一维小波滤波计算伪代码

伪代码	注释
filter_1D_even (Z) {	
for(i = 0; i < Z; i = i + 2) {	迭代偶数样本
Y[i] = X[i] − ((X[i −1] + X[i + 1]) >> 1)	在偶数样本中生成高通
}	
for(i = 1; i < Z + 1; i = i + 2) {	迭代奇数样本
Y[i] = X[i] + ((Y[i −1] + Y[i + 1]) + 2 >> 2)	更新奇数样本以生成低通
}	
}	

A.5 IDWT算法

A.5.1 水平小波反变换

输入：分量索引k，输出小波滤波器类型β_0和两种输入滤波器类型，低通β_L和高通β_H；临时条带T[β_L，x，y]和T[β_H，x，y]中的小波系数

输出：临时输出条带T[β_0，x，y]的小波系数。

水平小波反变换的计算伪代码见表A.10。

表A.10 水平小波反变换计算伪代码

伪代码	注释
HIDWT(k, β_0, β_L, β_H) {	将低通系数β_L和高通系数β_H水平逆变换到输出条带β_0
for(y = 0; y < H_b[β_0,k]; y = y + 1) {	迭代条带b的所有行
for(x = 0; x< W_b[β_0,k]; x = x + 1) {	迭代条带b的所有列
i = ⌊x / 2⌋;	计算源条带中的输入样本位置
if(x umod 2 == 0) {	如果是偶数样本位置
X[x] = T[β_L,i,y];	为临时数组X的偶数样本分配低通输入
}	
else {	如果是奇数样本位置
X[x] = T[β_H,i,y];	为临时数组X的奇数样本分配高通输入
}	
}	
extend_samples(W_b[β_0,k])	对称地将样本X穿过边界
inverse_filter_1D(W_b[β_0,k])	对临时数组X进行一维反滤波；
for(x = 0; x< W_b[β_0,k]; x = x + 1) {	迭代条带b的所有列
T[β_0, x, y] = Y[x];	将反变换的小波系数分配到输出条带
}	
}	
}	

A.5.2 垂直小波反变换

输入：分量索引k，输出小波滤波器类型β_0和两种输入滤波器类型，低通β_L和高通β_H；临时条带T[β_L，x，y]和T[β_H，x，y]中的小波系数

输出：临时输出条带T[β_0，x，y]的小波系数。

垂直小波反变换的计算伪代码见表A.11，偶数场垂直小波反变换的计算伪代码见表A.12。

表A.11 垂直小波反变换计算伪代码

伪代码	注释
VIDWT(k, β_0, β_L, β_H) {	将低通系数β_L和高通系数β_H垂直逆变换到分量k的输出条带β_0
for(x = 0; x < W_b[β_0,k]; x = x + 1) {	迭代条带b的所有列
for(y = 0; y < H_b[β_0,k]; y = y + 1) {	迭代条带b的所有行
i = ⌊y / 2⌋;	计算源条带中的输入样本位置
if(y umod 2 == 0) {	如果是偶数样本位置
X[y] = T[β_L,x,i];	为临时数组X的偶数样本分配低通输入
}	
else {	如果是奇数样本位置
X[y] = T[β_H,y,i];	为临时数组X的奇数样本分配高通输入
}	
}	

表 A.11（续）

伪代码	注释
extend_samples(H_b[β_0,k])	对称地将样本X穿过边界
inverse_filter_1D(H_b[β_0,k])	对临时数组X进行一维反滤波
for(y = 0; y < H_b[β_0,k]; y = y + 1) {	迭代条带b的所有行
T[β_0, x, y] = Y[y];	将反变换的小波系数分配到输出条带
}	
}	
}	

表 A.12　偶数场垂直小波反变换计算伪代码

伪代码	注释
VIDWT_even(k, β_0, β_L, β_H) {	将低通系数β_L和高通系数β_H垂直逆变换到分量k的输出条带β_0
for(x = 0; x < W_b[β_0,k]; x = x + 1) {	迭代条带b的所有列
for(y = 0; y < H_b[β_0,k]; y = y + 1) {	迭代条带b的所有行
i = ⌊y / 2⌋;	计算源条带中的输入样本位置
if(y umod 2 == 0) {	如果是偶数样本位置
X[y] = T[β_L,x,i];	为临时数组X的偶数样本分配低通输入
}	
else {	如果是奇数样本位置
X[y] = T[β_H,y,i];	为临时数组X的奇数样本分配高通输入
}	
}	
extend_samples(H_b[β_0,k])	对称地将样本X穿过边界
inverse_filter_1D_even(H_b[β_0,k])	对临时数组X进行偶数场一维反滤波；
for(y = 0; y < H_b[β_0,k]; y = y + 1) {	迭代条带b的所有行
T[β_0, x, y] = Y[y];	将反变换的小波系数分配到输出条带
}	
}	
}	

A.5.3　反滤波

基于Le Gall 5/3小波的一维反滤波计算伪代码见表A.13，偶数场一维反滤波计算伪代码见表A.14。

输入：小波系数数组X[x]，数组X的规模Z。

输出：反小波变换系数的数组Y[x]，至少对索引0到Z-1有效。

表 A.13　一维反滤波计算伪代码

伪代码	注释
inverse_filter_1D (Z) {	
for(i = 0; i < Z + 1; i = i + 2) {	迭代偶数样本
Y[i] = X[i] − ((X[i − 1] + X[i + 1]) >> 2)	从低通中重建偶数样本
}	
for(i = 1; i < Z; i = i + 2) {	迭代奇数样本
Y[i] = X[i] + ((Y[i − 1] + Y[i + 1]) >> 1)	从高通中重建奇数样本
}	
}	

表 A.14　偶数场一维反滤波计算伪代码

伪代码	注释
inverse_filter_1D_even (Z) {	
for(i = 1; i < Z + 1; i = i + 2) {	迭代奇数样本
Y[i] = X[i] − ((X[i −1] + X[i + 1]) >> 2)	从低通中重建奇数样本
}	
for(i = 0; i < Z; i = i + 2) {	迭代偶数样本
Y[i] = X[i] + ((Y[i −1] + Y[i + 1]) >> 1)	从高通中重建偶数样本
}	
}	

附 录 B
（资料性）
参考编码过程

B.1 概述

本附录介绍了视频格式为YUV 4:2:2、10bit、GB/T 41808—2022规定的HDR（HLG或PQ）、GB/T 41809—2022规定的BT.2020色域的4K超高清视频和8K超高清视频的浅压缩分层编码过程。

B.2 4K超高清视频浅压缩分层编码过程

对于 4K 超高清视频，编码过程如下：

a) 对每帧图像采用DWT变换算法进行一次水平变换、一次垂直变换，得到LL、LH、HL、HH四个小波子带，DWT变换算法见附录A；
b) 对奇数帧图像的LL小波子带进行一次垂直小波变换（见表A.6），得到LL-L、LL-H两个小波子带；
c) 对偶数帧图像的LL小波子带进行一次偶数帧垂直小波变换（见表A.7），得到LL-L、LL-H两个小波子带；
d) 对b)得到的LL-L和c)得到的LL-L分别进行HDR到SDR的动态范围下变换、GB/T 41809—2022规定的BT.2020色域到GY/T 155—2000规定的BT.709色域下变换，分别得到HD SDR基本层图像的奇数场和偶数场；
e) 可对d)得到的HD SDR基本层图像进行浅压缩编码得到HD SDR基本层码流，也可不编码直接传输，基本层图像的浅压缩编码方法不属于本文件的范围；
f) 若对d)得到的HD SDR基本层图像进行浅压缩编码，则解码HD SDR基本层码流得到HD SDR解码图像；
g) 对f)得到的HD SDR解码图像，进行SDR到HDR的动态范围上变换、GY/T 155—2000规定的BT.709色域到GB/T 41809—2022规定的BT.2020色域上变换，得到HD HDR图像的奇数场和偶数场；
h) 将b)得到的LL-L和g)得到的HD HDR图像的奇数场逐像素相减，得到LL-L残差；
i) 将c)得到的LL-L与g)得到的HD HDR图像的偶数场逐像素相减，得到LL-L残差；
j) 对h)或i)得到的LL-L残差和LL-H采用DWT变换算法进行一次水平变换，得到LL’、LH’、HL’、HH’四个小波子带；
k) 对LL’、LH’、HL’、HH’四个小波子带进行浅压缩编码，得到HD HDR增强层码流，HD HDR增强层码流见第7章～第11章；
l) 对a)得到的LL小波子带进行0元素填充，记为Zeros子带；
m) 对Zeros子带和a)得到的LH、HL、HH小波子带进行浅压缩编码，得到4K HDR增强层码流，4K HDR增强层码流见第7章～第11章。

4K超高清视频浅压缩分层编码过程见图B.1。

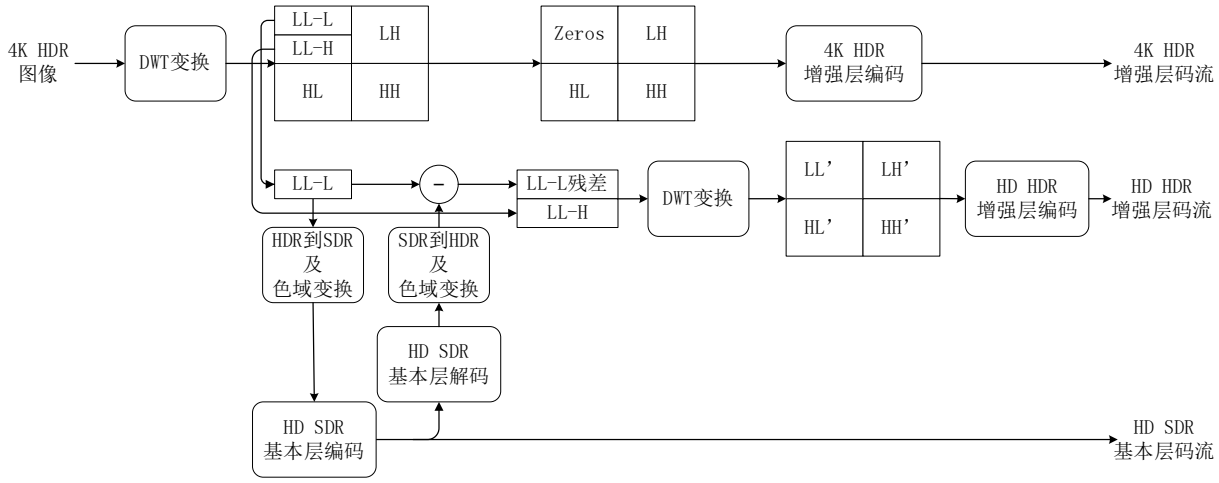

图 B.1　4K 超高清视频浅压缩分层编码过程

B.3　8K超高清视频浅压缩分层编码过程

对于 8K 超高清视频，编码过程如下：

a)　对每帧图像采用DWT变换算法进行一次水平变换、一次垂直变换，得到LL、LH、HL、HH四个小波子带，DWT变换算法见附录A；

b)　可对LL小波子带进行浅压缩编码得到4K基本层码流，也可不编码直接传输，基本层图像的浅压缩编码方法不属于本文件的范围；

c)　若对4K基本层码流进行浅压缩编码，则解码4K基本层码流得到LL’；

d)　将a)得到的LL与c)得到的LL’逐像素相减，得到基本层残差；

e)　对基本层残差、LH、HL、HH进行浅压缩编码，得到8K增强层码流，8K增强层码流见第7章～第11章。

8K超高清视频浅压缩分层编码过程见图B.2。

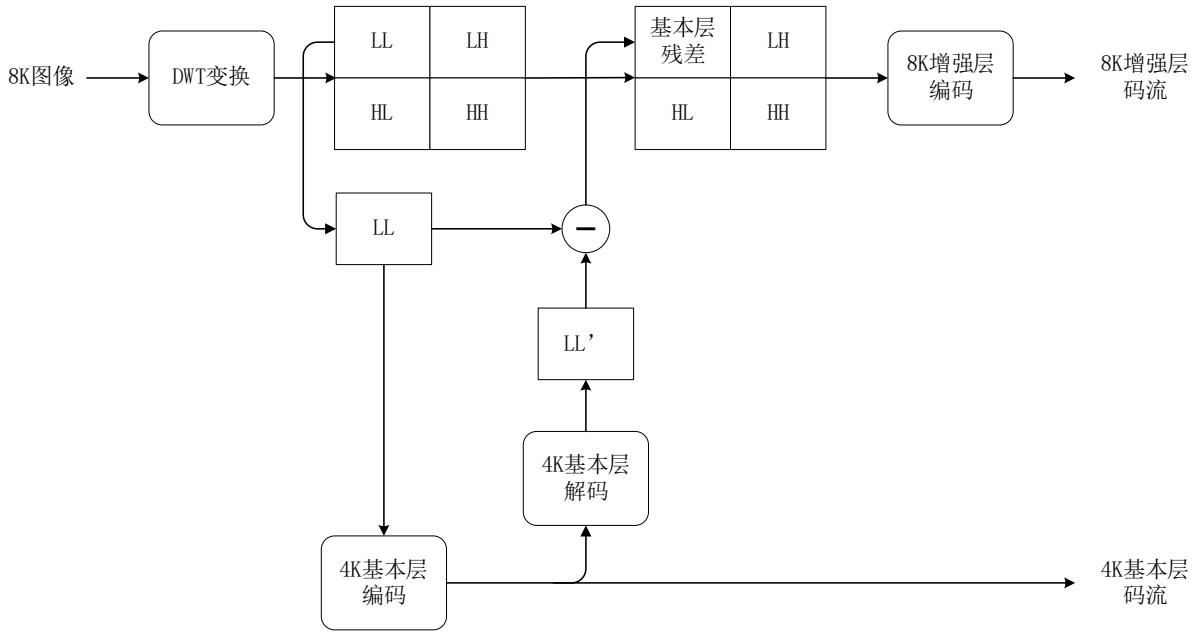

图 B.2　8K 超高清视频浅压缩分层编码过程

附 录 C
（资料性）
文件封装方法

C.1 概述

本附录介绍了超高清视频浅压缩分层编码的文件封装采用ISO/IEC 14496-12规定的文件封装格式以及基本结构。

C.2 文件封装

C.2.1 方法概述

本文件定义的文件封装方法通过video轨存储基本层编码数据，自定义meta轨存储增强层编码数据。该方式可实现基本层码流被专业媒体播放器识别，并且解析增强层码流得到全幅面解码图像。

C.2.2 文件结构

支持超高清视频浅压缩分层编码的文件结构见图C.1。

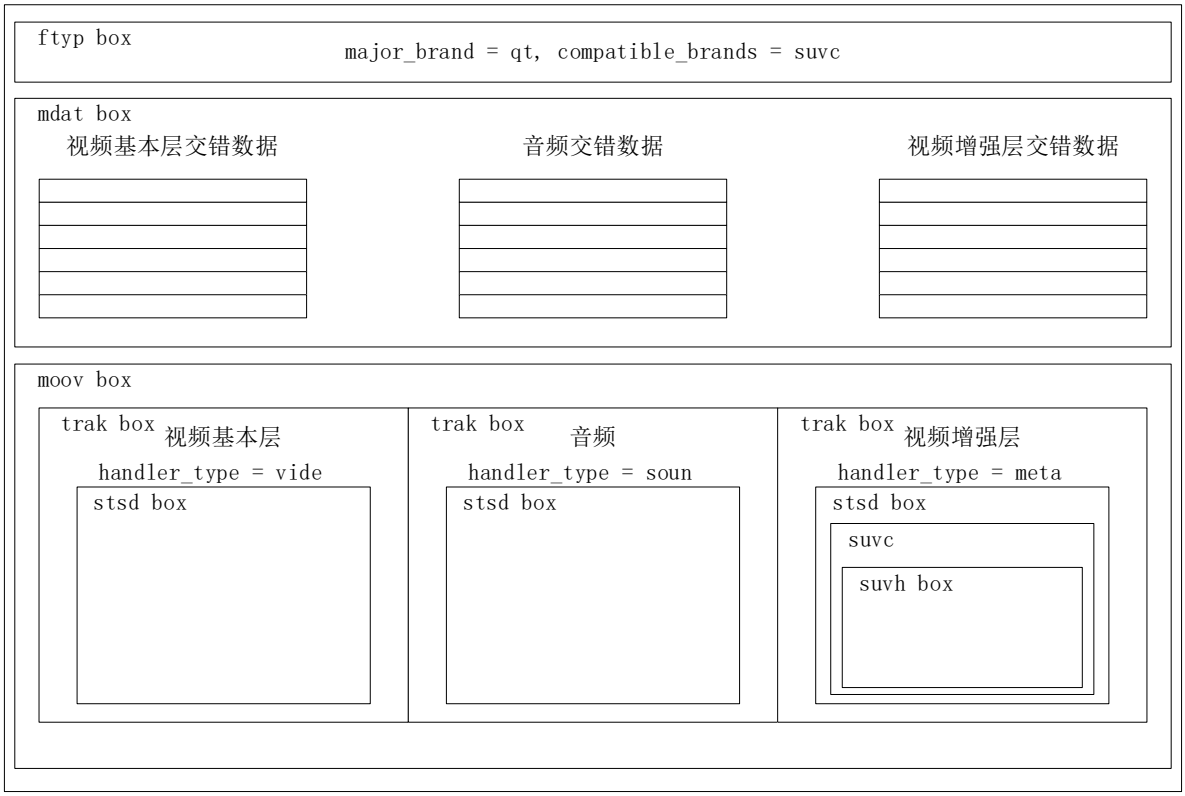

图 C.1 文件结构

图C.1中：

——ftyp box：用于存储文件类型；

——major_brand：定义主商标；
——qt：表示封装格式；
——compatible_brands：定义兼容商标；
——mdat box：用于存储音视频数据；
——moov box：用于存储元数据；
——trak box：用于存储一条音、视频轨/流/track的相关信息；
——handler_type：定义处理器类型；
——vide：表示视频轨；
——soun：表示音频轨；
——meta：表示meta轨；
——stsd box：用于存储解码所需参数；
——suvh box：用于存储视频增强层的文件封装辅助信息，suvh box的结构见附录E。

C.2.3 文件封装方法

视频基本层数据、音频数据、视频增强层数据封装在mdat box，三类数据按照逐帧交错或者若干帧交错的方式存放。

视频基本层参数信息封装在视频trak box中的stsd box。

音频参数信息封装在音频trak box中的stsd box。

视频增强层参数信息封装在meta trak box中的suvh box。

附 录 D
（规范性）
基于 HD-SDI 的 4K 视频传输方法

D.1 概述

本附录规定了采用一个HD-SDI传输接口实现支持超高清视频浅压缩分层编码的4K HDR视频和HD SDR视频同传的技术过程。

D.2 HD-SDI传输格式

D.2.1 HD-SDI信道帧格式

根据GY/T 155—2000和GY/T 157—2000中的定义，高清视频系统采用1920×1080/50i的数字帧，见图D.1，场周期定时见表D.1。

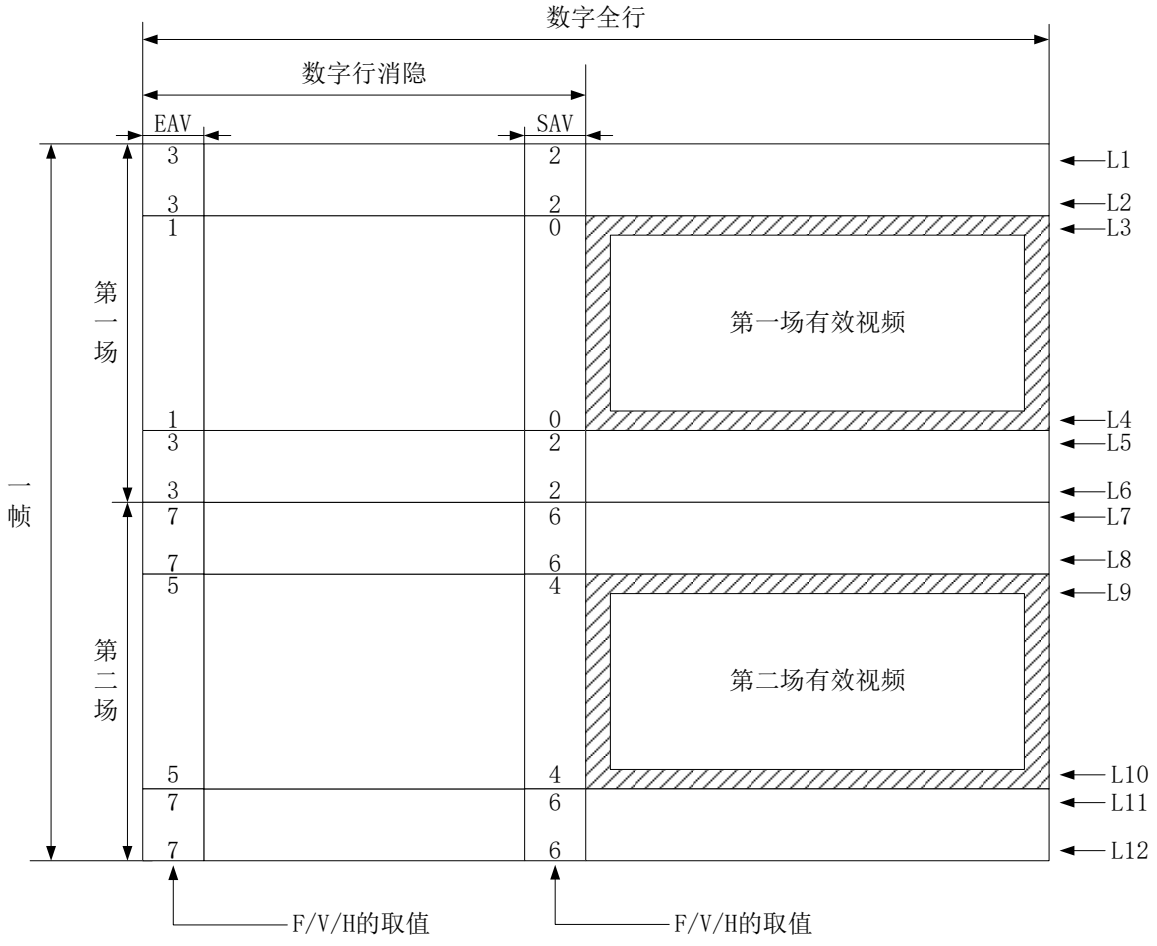

图 D.1 隔行扫描系统帧/场期间定时关系

表 D.1　隔行扫描系统场周期定时

序号	定义	数字行序号
L1	第一场的第一行	1
L2	第一场数字场消隐的最后一行	20
L3	第一场有效视频的第一行	21
L4	第一场有效视频的最后一行	560
L5	第二场数字场消隐的第一行	561
L6	第一场的最后一行	563
L7	第二场的第一行	564
L8	第二场数字场消隐的最后一行	583
L9	第二场有效视频的第一行	584
L10	第二场有效视频的最后一行	1123
L11	第一场数字场消隐的第一行	1124
L12	第二场的最后一行	1125
注：第一场数字场消隐表示在第一场有效视频之前的场消隐，第二场数字场消隐表示在第二场有效视频之前的场消隐。		

HD-SDI数字帧总行数为1125，总列数为2640。第0列至第1919列为数据有效行，第1920列至第2639列为数字行消隐。

每一帧包括第一场有效视频和第二场有效视频，有效行1080行，有效列1920列。第0列至第1919列、第21行至第560行是第一场有效视频，共540行；第0列至第1919列、第584行至第1123行是第二场有效视频，共540行。

每一帧垂直消隐间隔共45行。其中，第1124行、第1125行和第1行至第20行位于第一垂直消隐间隔，中间包含有一个用于切换容错的转换点，共22行。第561行至第583行是第二垂直消隐间隔，中间包含有一个用于切换容错的转换点，共23行。

视频定时基准码SAV和EAV，有效视频起始SAV自第2636列至第2639列、有效视频结束EAV自第1920列至第1924列，应符合GY/T 157—2000中4.3的规定。

LN、CRCC应符合GY/T 157—2000中6.1.3和6.1.4的规定。

音频数据应符合GY/T 161—2000中的规定。

辅助数据应符合GY/T 160—2000中的规定。

D.2.2　支持超高清视频浅压缩分层编码的4K超高清视频在HD-SDI的传输过程

支持超高清视频浅压缩分层编码的4K超高清视频包括HD SDR基本层和增强层，一起通过HD-SDI数字接口进行传输。

在GY/T 155—2000的要求下，本文件采用HD SDR基本层视频制式为YUV 4:2:2 8bit HD 50i SDR，即1920×1080分辨率，4:2:2采样，8bit色深，50i帧速率，色域和动态范围为BT.709 SDR。HD SDR基本层在HD-SDI数字帧的第一场有效视频和第二场有效视频进行传输，应符合GY/T 157—2000中的要求。

增强层包括HD HDR增强层和4K HDR增强层，在HD-SDI数字帧的消隐区部分和有效视频部分进行传输，消隐区包括水平消隐区和垂直消隐区。HD-SDI数字帧的有效视频中，每个视频数据有10位，HD SDR基本层只使用高8位，低位被用作增强层传输。

增强层数据传输从数字帧水平消隐区的第21行开始，在CRCC之后进行传输。增强层数据在水平消隐区或垂直消隐区中，插入多个数据包进行传输，除非消隐区已分配给其他应用。

增强层数据的编码位流结构见7.2。从位流头部开始，在第21行至第560行，依次放入水平消隐区、第一场有效视频的低2位；在第561行至第583行，依次放入水平消隐区、垂直消隐区，其中第569行、第

570行保留；在第584行至第1123行，依次放入水平消隐区、第二场有效视频的低2位；在第1124行至第1125行，依次放入水平消隐区、垂直消隐区。

增强层数据的传输顺序随着行数增加依次传输，在同一行内，增强层数据的传输顺序与数据存储顺序相同。

HD SDR基本层数据、增强层数据在HD-SDI数字帧的传输见图D.2。

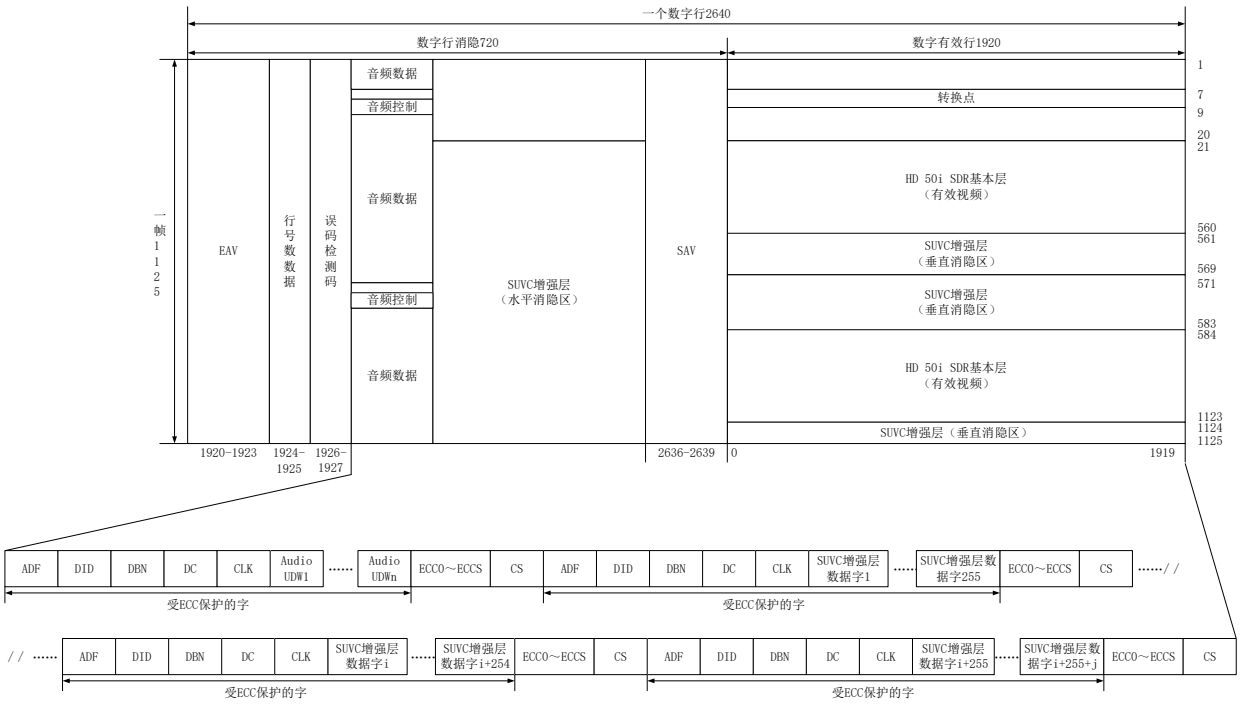

图 D.2　HD 50i SDR 基本层和增强层数据在 SDI 数字帧的传输

D.3　增强层数据在消隐区的传输

D.3.1　通则

增强层数据作为附属数据在水平消隐区和垂直消隐区传输，应符合GY/T 160—2000的规定。

D.3.2　增强层数据包结构

增强层数据采用附属数据包类型1，增强层数据包格式见图D.3。

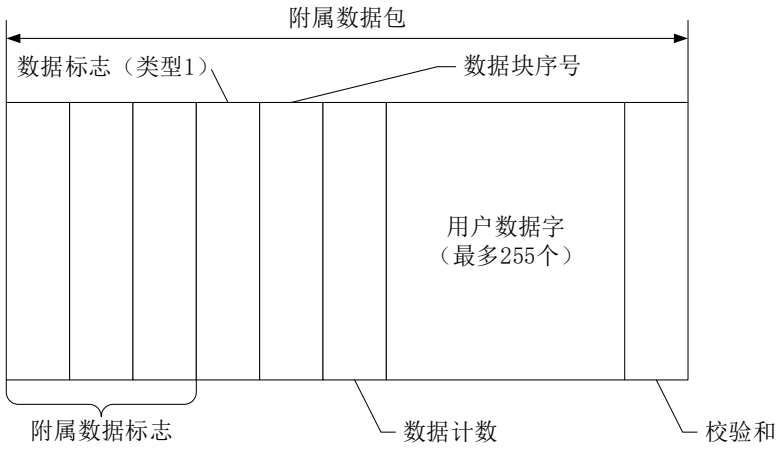

图 D.3　增强层数据包格式

D.3.3　ADF

ADF由三个字的序列组成，每个字10位，其值为000h 3FFh 3FFh，ADF表示附属数据包的开始，用以检测附属数据包。

D.3.4　DID字

DID定义附属数据包用户数据字中所运载数据性质，10位数值，其中低8位运载标识值，其余的2位运载校验码及其反码，分配如下：

a)　比特b7（MSB）～b0（LSB）组成标识值（00h～FFh）；

b)　比特b8为b7～b0的偶校验比特；

c)　比特b9为b8的反码。

增强层的DID值为2DEh，其中DE是标识值，2是比特b7～b0的偶校验码及其反码。

数据标识值DEh属于“用户应用”预留DID字，可以由用户或特定的设备厂家予以分配，应符合GY/T 160—2000中5.3.1的规定。

D.3.5　DBN

DBN用于区分带有共用数据标识的相继附属数据包，10位数值，其中8位运载DBN值，2位运载校验码及其反码，分配如下：

a)　比特b7（MSB）～b0（LSB）运载数据块（包）序号值；

b)　比特b8为b7～b0的偶校验比特；

c)　比特b9为b8的反码。

增强层数据包的DBN的比特b7～b0置为0，b8为0，b9为1，因此DBN的值为200h。

D.3.6　DC

DC字代表后随的范围为0～255个字的UDW数目，10位数值，其中7位运载数据计数值，3位运载校验码及反码，用于定义附属数据包中用户数据字的数量，分配如下：

a)　比特b7（MSB）～b0（LSB）运载数据计数值；

b)　比特b8为b7～b0的偶校验比特；

c)　比特b9为b8的反码。

D.3.7　UDW

UDW用于传输由DID标识的信息，10位数值，一个辅助数据包中UDW最大数目为255。

增强层数据封装在每个用户数据字的10位中，使用其中的低8位，即b7～b0。考虑到HD-SDI信号同步字竞争问题，高2位设置为01或10，以避免出现增强层数据和SAV、EAV等HD-SDI信号同步字的竞争，分配如下：

a) 比特b7（MSB）～b0（LSB）运载增强层数据；
b) 比特b8为b7～b0的偶校验比特；
c) 比特b9为b8的反码。

D.3.8 CS字

CS字用来确定DID至UDW的附属数据包的有效性，10位数值，其中9位运载校验和值，其余1位是反码，能进行误码检测，分配如下：

a) 比特b8（MSB）～b0（LSB）为CS值；
b) 比特b9为b8的反码。

D.4 增强层位流在HD-SDI传输时的参数限定和排列顺序

为了实现视频行级别的低延迟传输，增强层位流在HD-SDI传输时，编码参数应做以下限定：

——HD HDR 增强层的块高度应为 4，块宽度应为 16，条带高度应为 4；
——4K HDR 增强层的块高度应为 8，块宽度应为 32，条带高度应为 8；
——HD HDR 增强层和 4K HDR 增强层，各自包含的条带数量应为 135。

HD HDR增强层和4K HDR增强层应按照条带交错的方式在HD-SDI进行传输，排列顺序见图D.4。

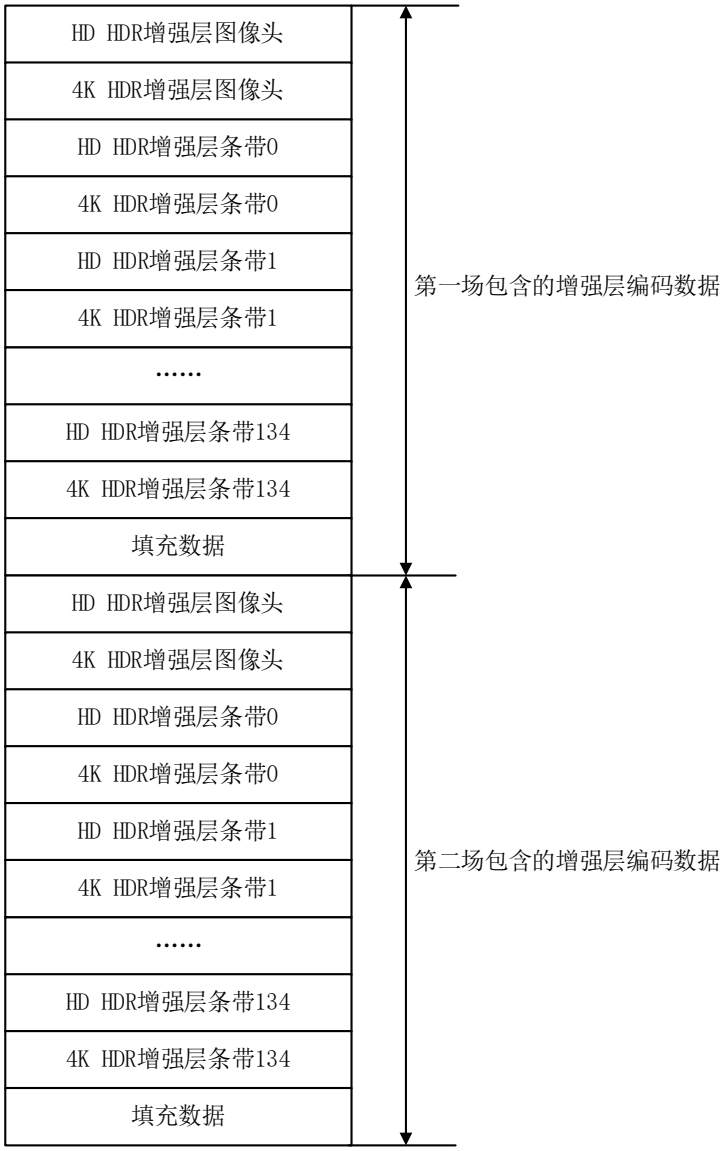

图 D.4　增强层位流在 HD-SDI 传输的排列顺序

附 录 E
（资料性）
支持超高清视频浅压缩分层编码的文件封装辅助信息结构

E.1 概述

本附录介绍了支持超高清视频浅压缩分层编码的文件封装辅助信息结构。

E.2 字段类型定义

字段类型定义见表E.1。

表 E.1 字段类型定义

字段类型	定义
FOURCC	32 位固定长度代码
b(n)	n 个连续的比特位
u(n)	n 位无符号整型
s(n)	n 位有符号整型

E.3 结构定义

E.3.1 概述

结构定义包含基本信息和扩展信息，基本信息长度为40个字节，存放于handler_type为meta的track box中，只在文件中出现一次。扩展信息长度不超过65535个字节，扩展信息包含的数据需要根据ExtID字段值来确定。扩展信息随增强层数据逐帧存放，出现在每帧增强层数据图像头的前面。

E.3.2 基本信息

基本信息字段见表E.2。

表 E.2 基本信息字段

字段	字段类型	定义
HeaderSize	u(32)	结构体长度
HeaderTag	FOURCC	码流 Codec 标记
FrameWidth	u(16)	图像宽度
FrameHeight	u(16)	图像高度
ChromaFormat	u(8)	色度分量格式
FrameRateDen	u(8)	视频帧率分母
FrameRateNum	u(16)	视频帧率分子
ColorPrimaries	u(8)	色域描述信息
TransferCharacteristic	u(8)	转换函数描述信息
MatrixCoefficient	u(8)	转换矩阵系数描述信息
FullRange	u(8)	信号范围描述信息
BaseVideoType	FOURCC	基本层视频格式
BaseWidthScale	u(8)	基本层视频宽度缩放系数
BaseHeightScale	u(8)	基本层视频高度缩放系数
BaseDepth	u(8)	基本层色度采样位宽
EnhancementDepth	u(8)	增强层色度采样位宽

表 E.2（续）

字段	字段类型	定义
EnhancementEncodeType	u(8)	增强层编码模式
TransformType	u(8)	变换模式
TransformPrecision	u(8)	变换精度
BaseFrameLayout	u(8)	基本层存放方式
HDRSDRConversionMethod	u(8)	图像上下变换算法类型
Reserved	b(56)	预留字段
注：所有的字段按顺序存放在二进制流中。		

HeaderSize

固定值40（0x00000028）个字节。

HeaderTag

固定为“suvh”（0x73757668）。

ChromaFormat

0x01表示YUV 4:2:2。

FrameRateDen

0x00表示1.000，0x01表示1.001。

BaseVideoType

基本层视频类型。

BaseWidthScale

基本层图像宽度等于FrameWidth/BaseWidthScale。

BaseHeightScale

基本层图像高度等于FrameHeight/BaseHeightScale。

EnhancementEncodeType

0x01表示超高清视频浅压缩分层编码。

BaseFrameLayout

0x00表示基本层按场编码存放，0x01表示基本层按帧编码存放。

E.3.3 扩展信息

当ExtID为0时，帧头扩展信息字段见表E.3。

表 E.3 帧头扩展信息字段

字段	字段类型	定义
ExtSize	u(16)	头部扩展部分字节长度
ExtID	u(8)	头部扩展部分数据类型
Interlacenformation	u(8)	当前帧码流的隔行扫描模式
TimeCode	u(32)	时码
HDRSDRInfo	u(8)	HDR-SDR 残差的编码方式
HDRSDRLayerSize	u(24)	HD HDR 增强层字节数
EnhancementLayerSize	u(32)	增强层字节数
BaseLayerSize	u(32)	基本层字节数
注：所有的字段按顺序存放在二进制流中。		

ExtSize

固定值为20(0X0014)。

Interlacenformation

描述当前码流的隔行扫描信息。低4位描述增强层，高3位描述基本层，具体如下：

a) 0x00：基本层按帧方式存放，增强层该帧对应基本层的帧；

b) 0x11：基本层按单场方式存放，增强层该帧对应基本层的顶场；

c) 0x22：基本层按单场方式存放，增强层该帧对应基本层的底场；

d) 0x31：基本层按两场方式存放，增强层该帧对应基本层的顶场；

e) 0x32：基本层按两场方式存放，增强层该帧对应基本层的底场。

TimeCode

时码，格式为HHMMSSFF，HH指定小时(0～23)、MM指定分钟(0～59)、SS指定秒(0～59)、FF指定帧数(1-n)。HH、MM、SS和FF字段是连续的单个字节。

HDRSDRInfo

固定值为2。

HDRSDRLayerSize

对于4K超高清图像，该字段表示HD HDR增强层字节数。对于8K超高清图像，该字段的值为0。

EnhancementLayerSize

对于8K超高清图像，该字段表示增强层字节数。对于4K超高清图像，该字段表示4K HDR增强层字节数。

BaseLayerSize

基本层字节数。

参 考 文 献

[1] ISO/IEC 14496-12 Information technology—Coding of audio-visual objects—Part 12: ISO base media file format

[2] ISO/IEC 21122-1:2022 Information technology—JPEG XS low-latency lightweight image coding system—Part 1:Core coding system