ICS 点击此处添加 ICS 号
CCS 点击此处添加 CCS 号

T/UWA XX

世 界 超 高 清 视 频 产 业 联 盟 团 体 标 准

T/UWA XXXX—XXXX

标准动态范围视频技术：元数据及适配

Standard Dynamic Range Video Technology: Metadata and Processing

（工作组讨论稿）

（草案 V1.0）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

世界超高清视频产业联盟　　发 布

目　　次

前　　言 III
引　　言 IV
1 范围 1
2 规范性引用文件 1
3 术语和定义 1
3.1 元数据 metadata 1
3.2 动态元数据 dynamic metadata 1
3.3 适配 processing 1
3.4 扩展裕量 extended headroom 1
3.5 显示裕量 display headroom 1
4 缩略语 1
5 符号与运算 1
5.1 总体要求 2
5.2 算数运算符 2
5.3 逻辑运算符 2
5.4 关系运算符 2
5.5 位运算符 2
5.6 赋值运算符 3
5.7 数学函数 3
5.8 转换函数 4
5.9 结构关系符 4
5.10 位流语法的描述方法 4
5.11 函数 6
5.12 描述符 6
5.13 保留、禁止和标记位 7
6 端到端系统要求 7
6.1 总体要求 7
6.2 系统框图 7
7 元数据 8
7.1 动态元数据语法 8
7.2 动态元数据语义 9
8 元数据显示适配 10
8.1 显示适配过程 10
8.2 基础曲线参数获得过程 10
8.3 暗部区域映射曲线参数获得过程 11
8.4 亮部区间映射参数获得过程 12
8.5 色彩信号动态范围转换过程 13

8.6 色彩校正过程 14
附 录 A （资料性） 动态元数据提取方法 15
A.1 概述 15
A.2 SDR 信号预处理过程 15
A.3 动态元数据 maximum_maxrgb_e[w]的计算 15
A.4 动态元数据 average_maxrgb_o[w]的计算 15
A.5 动态元数据 shadow_maxrgb_e[w]和 highlight_maxrgb_e[w]的计算 16
A.6 动态元数据 extended_headroom [w]的计算 16
A.7 动态元数据的时域滤波 17
附 录 B （资料性） 元数据在 HEVC/VVC 码流中的封装 18
B.1 元数据在 HEVC/VVC 码流中的封装 18
B.2 语义 18
参 考 文 献 20

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这些专利的责任。

本文件由世界超高清视频产业联盟提出。

本文件由世界超高清视频产业联盟归口。

本文件起草单位：北京爱奇艺科技有限公司、荣耀终端股份有限公司、中央广播电视总台技术局、海信视像科技股份有限公司、聚好看科技股份有限公司、夏普电子研发(南京)有限公司、华为技术有限公司、海思技术有限公司、咪咕文化科技有限公司、天翼数字生活科技有限公司、北京数码视讯软件技术发展有限公司、北京小米电子产品有限公司、维沃移动通信有限公司、中国电子技术标准化研究院、北京数字电视国家工程实验室有限公司、北京市博汇科技股份有限公司、西安诺瓦星云科技股份有限公司、深圳市奥拓电子股份有限公司、浪潮智能终端有限公司、上海数字电视国家工程研究中心有限公司马栏山音视频实验室、北京集创北方科技股份有限公司。

本文件主要起草人：王志航、丁岳、耿晨晖、李勇鹏、贾霖、张嘉森、蔡幼鹏、潘波、张宏伟、佘祥顺、马博翼、余全和、袁乐、毕蕾、贾立鼎、周骋、孙良、孙国展、陈仁伟、毛珂、李思远、郭忠武、刘必龙、陈鹏飞、王勇、房兰涛、殷惠清、于婧、樊磊。

引　　言

本文件的发布机构提请注意，声明符合本文件时，可能使用以下涉及的相关专利：

本文件的发布机构对于该专利的真实性、有效性和范围无任何立场。

该专利持有人已向本文件的发布机构保证，其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：

联 系 人：
通讯地址：
邮政编码：
电子邮件：
电　　话：
传　　真：
网　　址：

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。

标准动态范围视频技术：元数据及适配

1 范围

本文件规定了标准动态范围（SDR）视频技术元数据定义、封装以及终端显示适配过程。

本文件适用于网络电视、网络视频、数字存储媒体等视频应用。主要用于在显示亮度超过100坎德拉每平方米（cd/m²）的设备上播放SDR视频。

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 46269.1-2025 高动态范围(HDR)视频技术 第1部分:元数据及适配

GB/T 41809-2022 超高清晰度电视系统节目制作和交换参数值

ITU-R BT.709 节目制作和国际节目交换中使用高清晰度电视标准的参数值(Parameter values for the HDTV standards for production and international programme exchange)

3 术语和定义

下列术语和定义适用于本文件。

3.1 元数据 metadata

描述视频或者图像处理过程中需要的关键信息和特征的数据。

3.2 动态元数据 dynamic metadata

与每帧图像相关联且随画面不同而改变的元数据。

3.3 适配 processing

适配是一种将一组颜色映射到另一组颜色，以在更高动态范围的介质中呈现标准动态范围图像的处理过程。

3.4 扩展裕量 extended headroom

标准动态范围内容可扩展的动态范围。

3.5 显示裕量 display headroom

显示设备可以提供用于标准动态范围扩展的显示设备动态范围。

4 缩略语

下列缩略语适用于本文件。

SDR 标准动态范围（Standard Dynamic Range）
EOTF 电光转换函数（Electro-Optical Transfer Function）
OETF 光电转换函数（Optical-Electro Transfer Function）
HEVC 高效视频编码（High Efficiency Video Coding）
VVC 多功能视频编码（Versatile Video Coding）
MSB 最高有效位（Most Significant Bit）

5 符号与运算

5.1 总体要求

本文件中使用的数学运算符和优先级参照C语言。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 算数运算符

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（二元运算符）或取反（一元前缀运算符）
×	乘法运算
a^b	幂运算，表示a的b次幂。也可表示上标
/	整除运算，沿向0的取值方向截断。例如，7/4和-7/-4截断至1，-7/4和7/-4截断至-1
÷	除法运算，不做截断或四舍五入
$\frac{a}{b}$	除法运算，不做截断或四舍五入
$\sum_{i=a}^{b} f(i)$	自变量i取由a到b（含b）的所有整数值时，函数$f(i)$的累加和
a % b	模运算，a除以b的余数，其中a与b都是正整数
$\lfloor . \rfloor$	下取整

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之间的与逻辑运算
a || b	a和b之间的或逻辑运算
!	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
a >> *b*	将*a*以2的补码整数表示的形式向右移*b*位。仅当*b*取正数时定义此运算
a << *b*	将*a*以2的补码整数表示的形式向左移*b*位。仅当*b*取正数时定义此运算

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	定义
=	赋值运算符
++	递增，*x*++相当于*x* = *x* + 1。当用于数组下标时，在自加运算前先求变量值
--	递减，*x*--相当于*x* = *x* - 1。当用于数组下标时，在自减运算前先求变量值
+=	自加指定值，例如*x* += 3相当于*x* = x + 3，*x* += (-3)相当于*x* = *x* + (-3)
-=	自减指定值，例如*x* -= 3相当于*x* = *x* - 3，*x* -= (-3)相当于*x* = *x* - (-3)

5.7 数学函数

数学函数定义见公式（1）至公式（10）。

$$\mathrm{Abs}(x) = \begin{cases} \mathrm{x},\ \mathrm{x} \geq 0 \\ -\mathrm{x},\ \mathrm{x} < 0 \end{cases} \tag{1}$$

式中：

x——自变量。

$$\mathrm{Floor}(x) = \lfloor x \rfloor \tag{2}$$

式中：

x——自变量。

$$\mathrm{Clip3}(\mathrm{i},\mathrm{j},\mathrm{x}) = \begin{cases} \mathrm{i},\ \mathrm{x} < \mathrm{i} \\ \mathrm{j},\ \mathrm{x} > \mathrm{j} \\ \mathrm{x}, \text{其他} \end{cases} \tag{3}$$

式中：

x——自变量；

i——下界；

j——上界。

$$\mathrm{Median}(\mathrm{x},\mathrm{y},\mathrm{z}) = \mathrm{x} + \mathrm{y} + \mathrm{z} - \mathrm{Min}\big(\mathrm{x}, \mathrm{Min}(\mathrm{y},\mathrm{z})\big) - \mathrm{Max}\big(\mathrm{x}, \mathrm{Max}(\mathrm{y},\mathrm{z})\big) \tag{4}$$

式中：

x——自变量；

y——自变量；

z——自变量。

$$\mathrm{Min}(x,y)=\begin{cases}x, & x\le y\\ y, & x>y\end{cases} \tag{5}$$

式中：
x——自变量；
y——自变量。

$$\mathrm{Max}(x,y)=\begin{cases}x, & x\ge y\\ y, & x<y\end{cases} \tag{6}$$

式中：
x——自变量；
y——自变量。

$$\mathrm{Sign}(x)=\begin{cases}1, & x\ge 0\\ -1, & x<0\end{cases} \tag{7}$$

式中：
x——自变量。

$$\mathrm{Log}(x)=\log_2 x \tag{8}$$

式中：

x——自变量。

$$\mathrm{Ln}(x)=\log_e x \tag{9}$$

式中：

x——自变量；

e——自然对数的底，其值为2.718281828…。

$$\mathrm{pow}(x,y)=x^y \tag{10}$$

式中：

x——自变量；

y——自变量。

5.8 转换函数

OETF函数定义见式（11），EOTF函数定义见式（12）。

$$\mathrm{OETF}(L)=\begin{cases}4.5\times L, & L\le 0.018\\ 1.099L^{0.45}-0.099, & L>0.018\end{cases} \tag{11}$$

$$\mathrm{EOTF}(V)=\begin{cases}V/4.5, & V\le 0.081\\ \left(\frac{V+0.099}{1.099}\right)^{1/0.45}, & V>0.081\end{cases} \tag{12}$$

式中：
L ——自变量L，光信号值。
V ——自变量V，电信号值。

5.9 结构关系符

结构关系符定义见表6。

表 6 结构关系符

结构关系符	定义
->	例如：*a*->*b*表示*a*是一个结构，*b*是*a*的一个成员变量

5.10 位流语法的描述方法

位流语法描述方法类似C语言。位流的语法元素使用粗体字表示，每个语法元素通过名字（用下划线分割的英文字母组，所有字母都是小写）、语法和语义来描述。语法表和正文中语法元素的值用常规字体表示。

某些情况下，可在语法表中应用从语法元素导出的其他变量值，这样的变量在语法表或正文中用带下划线的小写字母命名，或者用小写字母和大写字母混合命名。大写字母开头的变量用于解码当前以及相关的语法结构，也可用于解码后续的语法结构。小写字母开头的变量只在它们所在的小节内使用。

语法元素值的助记符和变量值的助记符与它们的值之间的关系在正文中说明。在某些情况下，二者等同使用。

位串的长度是4的整数倍时，可使用十六进制符号表示。十六进制的前缀是“0x”，例如“0x1*a*”表示位串“0001 1010”。

条件语句中0表示FALSE，非0表示TRUE。

语法表描述了所有符合本文件的位流语法的超集，附加的语法限制在相关条中说明。

表7给出了描述语法的伪代码例子。当语法元素出现时，表示从位流中读一个数据单元。

表 7 语法描述的伪代码

伪代码	描述符
/*语句是一个语法元素的描述符，或者说明语法元素的存在、类型和数值，下面给出两个例子。*/	
syntax_element	
conditioning statement	
/*花括号括起来的语句组是复合语句，在功能上视作单个语句。*/	
{	
statement	
…	
}	
/*“while”语句测试condition是否为TRUE，如果为TRUE，则重复执行循环体，直到condition不为TRUE。*/	
while (condition)	
statement	
/*“do … while”语句先执行循环体一次，然后测试condition是否为TRUE，如果为TRUE，则重复执行循环体，直到condition不为TRUE。*/	
do	
statement	
while (condition)	
/*“if … else”语句首先测试condition，如果为TRUE，则执行primary语句，否则执行alternative语句。如果alternative语句不需要执行，结构的“else”部分和相关的alternative语句可忽略。*/	
if (condition)	
primary statement	

else	
alternative statement	
/*“for”语句首先执行initial语句，然后测试condition，如果conditon为TRUE，则重复执行primary语句和subsequent语句直到condition不为TRUE。*/	
for (initial statement; condition; subsequent statement)	
primary statement	

解析过程和解码过程用文字和类似C语言的伪代码描述。

5.11 函数

5.11.1 byte_aligned( )

如果位流的当前位置是字节对齐的，返回TRUE，否则返回FALSE。

5.11.2 next_start_code( )

在位流中寻找下一个起始码，将位流指针指向起始码前缀的第一个二进制位。函数定义应符合表8的规定。

表 8 next_start_code 函数的定义

函数定义	描述符
next_start_code( ) {	
stuffing_bit	’1’
while (! byte_aligned( ))	
stuffing_bit	’0’
while (next_bits(24) != ’0000 0000 0000 0000 0000 0001’)	
stuffing_byte	’00000000’
}	

stuffing_byte应出现图像头之后和第一个片起始码之前。

5.11.3 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等于0，则返回0，位流指针不前移。

函数也用于解析过程和解码过程的描述。

5.12 描述符

描述符表示不同语法元素的解析过程，见表9。

表 9 描述符

描述符	说明
b(8)	一个任意取值的字节。解析过程由函数read_bits(8)的返回值规定
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

r(n)	连续n个‘0’。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

5.13 保留、禁止和标记位

本文件定义的位流语法中，某些语法元素的值被标注为“保留”（reserved）或“禁止”（forbidden）。

“保留”定义了一些特定语法元素值用于将来对本文件的扩展。这些值不应出现在符合本文件的位流中。

“禁止”定义了一些特定语法元素值，这些值不应出现在符合本文件的位流中。

“标记位”（marker_bit）指该位的值应为‘1’。

位流中的“保留位”（reserved_bits）表明保留了一些语法单元用于将来对本文件的扩展，解码处理应忽略这些位。

6 端到端系统要求

6.1 总体要求

标准动态范围内容制作和交换参数应符合GB/T 41809-2022 超高清晰度电视系统节目制作和交换参数值的规定。

元数据应符合本文规定。

6.2 系统框图

SDR的动态元数据端到端系统可分为两种类型，分别为编码前提取动态元数据和解码后提取动态元数据。系统应优先采用编码前提取动态元数据方案（见6.2.1）；当码流中未携带动态元数据时，可采用解码后提取动态元数据方案（见6.2.2）。

6.2.1 编码前提取动态元数据

图1描述了编码前提取动态元数据的系统框图，主要包括SDR前处理、编码、解码、扩展显示净空、SDR显示模块。

SDR前处理模块：提取SDR视频动态元数据，第7章规定了元数据的语法语义。附录A规定了元数据的提取方法。

编码模块：把SDR视频和SDR动态元数据进行视频编码，生成视频码流。第7章规定了元数据封装方式。

解码模块：将视频码流进行解码，输出SDR解码视频和SDR动态元数据。

SDR显示模块：使用SDR动态元数据和显示终端参数处理SDR解码视频，并在显示终端显示。第8章规定了SDR内容利用元数据和显示终端参数在显示终端的适配过程。

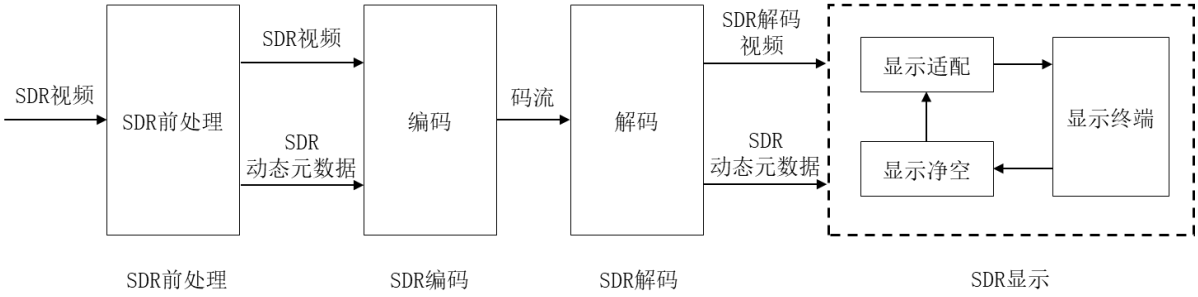

图 1 编码前提取 SDR 动态元数据端到端系统

6.2.2 解码后提取动态元数据

图2描述了解码后提取动态元数据的系统框图，主要包括编码、解码、SDR后处理、扩展显示净空、显示模块。

编码模块：把SDR视频进行视频编码，生成视频码流。

解码模块：将码流进行解码，输出SDR解码视频。

SDR后处理模块：提取SDR解码视频的动态元数据，第7章规定了元数据的语法定义。附录A规定了元数据的提取方法。

SDR显示模块：使用SDR动态元数据和显示终端参数处理SDR解码视频，并在显示终端显示。第8章规定了SDR内容利用SDR动态元数据和显示终端参数在显示终端的适配过程。

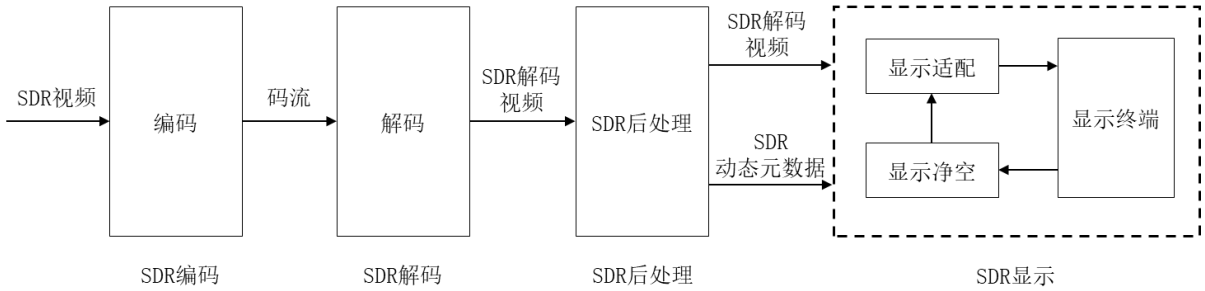

图 2 解码后提取 SDR 动态元数据端到端系统

7 元数据

7.1 动态元数据语法

动态元数据详细语法见表10。

表 10 动态元数据语法

动态元数据语法	描述符
dynamic_metadata ( ) {	
system_start_code	u(8)
if(system_start_code==0x01){	
num_blocks_h	u(8)
num_blocks_v	u(8)
for(v=0; v<num_blocks_v; v++){	
for(h=0; h<num_blocks_h; h++){	
shadow_maxrgb_e[v][h]	u(12)
highlight_maxrgb_e[v][h]	u(12)
max_maxrgb_e[v][h]	u(12)
average_maxrgb_o[v][h]	u(12)
extended_headroom[v][h]	u(16)
tone_mapping_factor_flag[v][h]	u(1)
if(tone_mapping_factor_flag[v][h]){	

动态元数据语法	描述符
shadow_factor[v][h]	u(8)
highlight_factor[v][h]	u(8)
tone_factor[v][h]	u(8)
}	
color_saturation_mapping_factor_flag[v][h]	u(1)
if(color_saturation_mapping_factor_flag[v][h]){	
color_saturation_factor[v][h]	u(8)
}	
}	
}	
}	
}	

7.2 动态元数据语义

7.2.1 系统起始码（system_start_code）

8位无符号整数，表示系统版本号,参见附录B.2。

7.2.2 水平分割块数量（num_blocks_h）

8位无符号整数，表示当前帧图像水平方向可分割块的数量。当前帧图像总窗口数量为num_blocks_h*num_blocks_v，每个窗口可保存该窗口区域内图像的动态元数据。规定帧图像原点位置为左上(TOP_LEFT)，窗口遍历顺序为先水平方向，后垂直方向。num_blocks_h默认值为1，num_blocks_h取值为范围为1~255。

7.2.3 垂直分割块数量（num_blocks_v）

8位无符号整数，表示当前帧图像垂直方向可分割块的数量。当前帧图像总窗口数量为num_blocks_h*num_blocks_v，每个窗口可保存该窗口区域内图像的动态元数据。规定帧图像原点位置为左上(TOP_LEFT)，窗口遍历顺序为先水平方向，后垂直方向。num_blocks_v默认值为1，num_blocks_v取值为范围为1~255。

7.2.4 RGB 分量最大值中的暗部特征值（shadow_maxrgb_e[v][h]）

12位无符号整数，表示显示内容在非线性空间的暗部特征值。shadow_maxrgb_e的取值为shadow_maxrgb_e[w]/4095；以1.0/4095为单位，范围从0.0到1.0。

7.2.5 RGB 分量最大值中的亮部特征值（highlight_maxrgb_e[v][h]）

12位无符号整数，表示显示内容在非线性空间的亮部特征值。highlight_maxrgb_e的取值为highlight_maxrgb_e[w]/4095；以1.0/4095为单位，范围从0.0到1.0。

7.2.6 RGB 分量最大值中的平均值（average_maxrgb_o[v][h]）

12位无符号整数，表示显示内容在线性空间的平均值。average_maxrgb_o的取值为average_maxrgb_o[w]/4095；以1.0/4095为单位，范围从0.0到1.0。

7.2.7 RGB 分量最大值中的最大值（max_maxrgb_e[v][h]）

12位无符号整数，表示显示内容在非线性空间的最大值。max_maxrgb_e的取值为max_maxrgb_e[w]/4095；以1.0/4095为单位，范围从0.0到1.0。

7.2.8 可扩展裕量（extended_headroom[v][h]）

16位无符号整数，表示显示内容在线性空间扩展裕量的最大值。extended_headroom的取值为extended_headroom[w]/1023 + 1.0；以1.0/1023为单位，范围从1.0到65.0。

7.2.9 色调映射标识（tone_mapping_factor_flag[v][h]）

1位无符号整数，表示传送色调映射的标识。。若tone_mapping_factor_flag[w]为0，则不传输曲线补偿参数；若tone_mapping_factor_flag[w]为1，则传输曲线补偿参数。

7.2.10 暗部补偿系数（shadow_factor[v][h]）

8位无符号整数，表示暗部补偿系数。shadow_factor的取值为 shadow_factor[w]/255 - 0.5；以1.0/255为单位，范围从-0.5到0.5。

7.2.11 亮部补偿系数（highlight_factor[v][h]）

8位无符号整数，表示亮部补偿系数。highlight_factor的取值为 highlight_factor[w]/255；以1.0/255为单位，范围从0.0到1.0。

7.2.12 色调补偿系数（tone_factor[v][h]）

8位无符号整数，表示色调补偿系数。tone_factor[w]的取值为tone_factor[w]/80；以1.0/80为单位，范围从0.0到3.0。

7.2.13 颜色校正标识（color_saturation_mapping_factor_flag[v][h]）

1位无符号整数，若color_saturation_mapping_flag[w]为1，表示应传输颜色校正参数，若color_saturation_mapping_flag[w]为‘0’，表示不应传输颜色校正参数。

7.2.14 颜色校正强度（color_saturation_factor[v][h]）

8位无符号整数，表示颜色校正强度参数。color_saturation_factor[w]取值为color_saturation_factor[w]/128，以1.0/128为单位，范围为0.0~2.0。

8 元数据显示适配

8.1 显示适配过程

输入：显示设备的显示亮度DisplayBrightness，显示设备的显示裕量DisplayHeadroom，待处理帧的RGB色域像素缓冲区$f[\mathrm{N_{frame}}][[3]$，metadata数据信息。

输出：待处理帧经过显示适配处理的RGB色域像素缓冲区$f_{process}[\mathrm{N_{frame}}][[3]$。

SDR显示适配过程如下：

a） 调用8.2节基础曲线参数获得过程；

b） 调用8.3节暗部区域映射曲线参数获得过程；

c） 调用8.4节亮部区域间映射曲线参数获得过程；

d） 调用8.5节色彩信号动态范围转换过程生成待处理帧经过动态范围转换处理的RGB色域像素缓冲区$f_{TM}[\mathrm{N_{frame}}][[3]$；

e） 调用8.6节色彩校正过程对$f_{TM}[\mathrm{N_{frame}}][[3]$信号进行色彩校正，得到经过显示适配处理的RGB色域像素缓冲区$f_{process}[\mathrm{N_{frame}}][[3]$。

8.2 基础曲线参数获得过程

8.2.1 概述

输入：metadata数据信息，显示设备的显示亮度DisplayBrightness，显示设备的显示裕量DisplayHeadroom。

输出：亮部区域阈值P2，基础曲线参数$P_{midtone}$：包含mp、ma、mb，得到基础曲线。

$$Midtone(L) = ma\times\left(\frac{mp\times \mathrm{L}}{(mp-1)\times \mathrm{L}+1}\right)^{2.4} + mb \quad \cdots\cdots(13)$$

8.2.2 基础曲线参数获得过程

基础曲线参数获得过程步骤如下：

a） 根据元数据extended_headroom与显示设备的显示裕量DisplayHeadroom得到当前待处理帧的headroom

$$\mathrm{headroom} = \mathrm{Min}(\mathrm{extended_headroom}, \mathrm{DisplayHeadroom})$$

b） 根据元数据中的highlight_maxrgb_e与显示设备的显示裕量DisplayHeadroom，得到亮部区域阈值P2

$$\mathrm{P2} = \mathrm{Max}(\mathrm{highlight_maxrgb_e}, 0.55 + 0.2 \times \mathrm{Max}(2 - \mathrm{display_headroom}, 0))$$

c） 根据元数据中的highlight_maxrgb_e与显示设备的显示亮度DisplayBrightness($DisL$)，计算mp0

$$\mathrm{mp0} = \begin{cases} 2.4 & \mathrm{hlightE} > 0.9 \\ (0.9 - \mathrm{hlightE}) \times slope + 2.4 & 0.5 \le \mathrm{hlightE} \le 0.9 \\ mpmax & \mathrm{hlightE} < 0.5 \end{cases} \quad \cdots\cdots(14)$$

式中，

$\mathrm{hlightE} = \mathrm{highlight_maxrgb_e}$，$mpmax = 4.2 - 1.2 \times \left(\frac{DisL}{2000}\right)^{\frac{1}{2}}$，$slope = \frac{mpmax-2.4}{0.4}$。

d） 根据元数据中的average_maxrgb_o计算mp

$$\mathrm{mp} = \begin{cases} \mathrm{mp0} & \mathrm{avgO} > 0.13 \\ \mathrm{mp0} + w0 \times 10 \times (0.13 - \mathrm{avgO}) & 0.03 \le \mathrm{avgO} \le 0.13 \\ \mathrm{mp0} + w0 & \mathrm{avgO} < 0.03 \end{cases} \quad \cdots\cdots(15)$$

式中，

$\mathrm{avgO} = \mathrm{average_maxrgb_o}$，$w0 = 1 - \frac{DisL}{2000}$。

e） 将ma，mb分别设为预设值1.0，0.0

$$ma = 1.0,\ mb = 0.0 \quad \cdots\cdots(16)$$

8.3 暗部区域映射曲线参数获得过程

8.3.1 概述

输入：metadata数据信息，显示设备的显示亮度DisplayBrightness，显示设备的显示裕量DisplayHeadroom。

输出：暗部区域阈值P1，暗部区域映射曲线参数P_{shadow}：$S1$，$S2$，得到暗部区域映射曲线：

$$Shadow(L) = S1 \times L + S2 \times \left(\left(\frac{L}{\mathrm{P1}}\right)^3 + \frac{1}{2} \times \left(\frac{L}{\mathrm{P1}}\right)^4\right) \quad \cdots\cdots(17)$$

8.3.2 暗部区域映射曲线参数获得过程

暗部区域映射曲线参数获得过程步骤如下：

a） 根据元数据中的shadow_maxrgb_e与显示设备的显示亮度DisplayBrightness($DisL$)，计算斜率下限值MinSlope：

$$\mathrm{MinSlope} = \frac{\mathrm{OETF}\left(\frac{\mathrm{DarkO}}{14.14*\sqrt{DisL}}\right)}{\mathrm{Max(shadowE,0.18)}} \quad \cdots\cdots(18)$$

其中，

$shadowE = \mathrm{shadow_maxrgb_e}$，$\mathrm{DarkO} = \mathrm{EOTF}(\mathrm{Max}(shadowE, 0.18)) * 200$。

b） 根据元数据中的shadow_factor，补偿斜率下限值MinSlope：

$$\mathrm{MinSlope} = \mathrm{Min}(\mathrm{MinSlope} + \mathrm{shadow_factor}, 1) \quad \cdots\cdots(19)$$

c） 若MinSlope大于0.99.则暗部特征斜率$darkslow = \mathrm{MinSlope}$。否则，根据元数据中的average_maxrgb_o与显示设备的显示亮度DisplayBrightness($DisL$)，计算暗部特征斜率darkslow：

$$\mathrm{dar}kslow = \begin{cases} \mathrm{MinSlope} & \mathrm{avgO} > 0.05 \\ 1 - slople \times avglog^{mi} & 0.005 \le \mathrm{avgO} < 0.05 \\ 1 & \mathrm{avgO} < 0.005 \end{cases} \quad \cdots\cdots(20)$$

式中，

$slople = Max(1 - \mathrm{MinSlope}, 0)$，$avglog = \log_{10}(\mathrm{avgO}/0.005)$，$mi = 1.1 + DisL/10000$。

d） 根据darkslow与基础曲线，求解下述方程得到暗部区域阈值P1：

$$darkslow = \frac{Midtone(\mathrm{P1})}{\mathrm{P1}} \quad \cdots\cdots(21)$$

使用遍历法：

```
darks=0.0;
darkEtemp = 1.0 / 4095.0;
bins = 1.0 / 1024.0;
while (darks - darkslow < -0.000001 && darkEtemp < 1.0) {
        darks = Midtone(darkEtemp, mp) / darkEtemp;
        darkEtemp += bins;
    }
P1 = darkEtemp;
```

e） 计算暗部区域映射曲线参数s1、s2

$$\mathrm{ad} = 1.0 - \mathrm{Max}(1.2 - \mathrm{DisplayHeadroom}) \times \mathrm{Min}(\frac{\mathrm{avgO}}{0.05}, 1.0) \quad \cdots\cdots(22)$$

$$\mathrm{s1} = \frac{Midtone(\mathrm{P1})}{P1} \times \mathrm{ad} \quad \cdots\cdots(23)$$

$$\mathrm{s2} = (\mathrm{TMGD3(P1)} - \mathrm{s1}) \times \mathrm{P1} \quad \cdots\cdots(24)$$

其中，TMGD3(L)表示基础曲线的导数形式如公式 25 所示。

$$\mathrm{TMGD3(L)} = 2.4 \times \mathrm{mp} \times \left(\frac{\mathrm{mp} \times \mathrm{L}}{(\mathrm{mp} - 1.0) \times \mathrm{L} + 1.0}\right)^{3.4} \times \left(\frac{1}{\mathrm{L} \times \mathrm{mp}}\right)^{2} \quad \cdots\cdots(25)$$

f） 根据headroom，更新基础曲线参数ma、mb

$$ma = 1.0 \quad \cdots\cdots(26)$$

$$mb = \mathrm{P1} \times 0.5 \times \left(\mathrm{TMGD3(P1)} - 2.0 \times \frac{\mathrm{Midtone(P1)}}{\mathrm{P1}}\right) + (\mathrm{Midtone(P1)} \times 0.5) \times ad \quad \cdots\cdots(27)$$

g） 根据元数据中的tone_factor，更新暗部区域映射曲线参数S1、S2以及基础曲线参数ma、mb

$$S1 = s1 \times \mathrm{tone_factor} \quad \cdots\cdots(28)$$

$$S2 = s2 \times \mathrm{tone_factor} \quad \cdots\cdots(29)$$

$$ma = ma \times \mathrm{tone_factor} \quad \cdots\cdots(30)$$

$$mb = mb \times \mathrm{tone_factor} \quad \cdots\cdots(31)$$

8.4 亮部区间映射参数获得过程

8.4.1 概述

输入：metadata数据信息，显示设备的显示裕量DisplayHeadroom。

输出：亮部区域映射曲线参数$P_{highlight}$：A，B，C，D，E，得到亮部区域映射曲线：

$$Highlight(L) = (A + B \times (L - \mathrm{P2}) + C \times (L - \mathrm{P2})^2 + D \times (L - \mathrm{P2})^3)^E \quad \cdots\cdots(32)$$

8.4.2 亮部区域映射曲线参数获得过程

亮部区域映射曲线参数获得过程步骤如下：

a） 根据headroom和max_maxrgb_e计算参数E

$$E = \mathrm{Max}(\mathrm{headroom}, \mathrm{Min}\left(\frac{1}{EOTF(\mathrm{max_maxrgb_e})}, 1.5\right)) \quad \cdots\cdots(33)$$

b） 计算亮部区域映射曲线在P2处的边界条件

$$Y_{P2} = Midtone(\mathrm{P2}) \quad \cdots\cdots(34)$$

$$K_{P2} = \frac{ma \times \mathrm{TMGD3(P2)}}{E \times Y_{P2}{}^{(E-1)/E}} \quad \cdots\cdots(35)$$

c） 计算亮部区域映射曲线在端点WP处的边界条件

$$Y_{WP} = (\mathrm{headroom} \times \mathrm{highlight_factor})^{1/E} \quad \cdots\cdots(36)$$

$$K_{WP} = K_{P2} \times \mathrm{headroom} \quad \cdots\cdots(37)$$

d） 根据边界条件计算亮部区域映射曲线参数A，B，C，D

$$A = Y_{P2} \quad \cdots\cdots(38)$$

$$B = K_{P2} \quad \cdots\cdots(39)$$

$$C = 3 \times \frac{Y_{WP} - Y_{P2}}{h^2} - \frac{Y_{WP} + 2Y_{P2}}{h} \quad \cdots\cdots(40)$$

$$D = 2 \times \frac{Y_{WP} - Y_{P2}}{h^3} + \frac{Y_{WP} + Y_{P2}}{h^2} \quad \cdots\cdots(41)$$

式中，$h = 1 - \mathrm{P2}$。

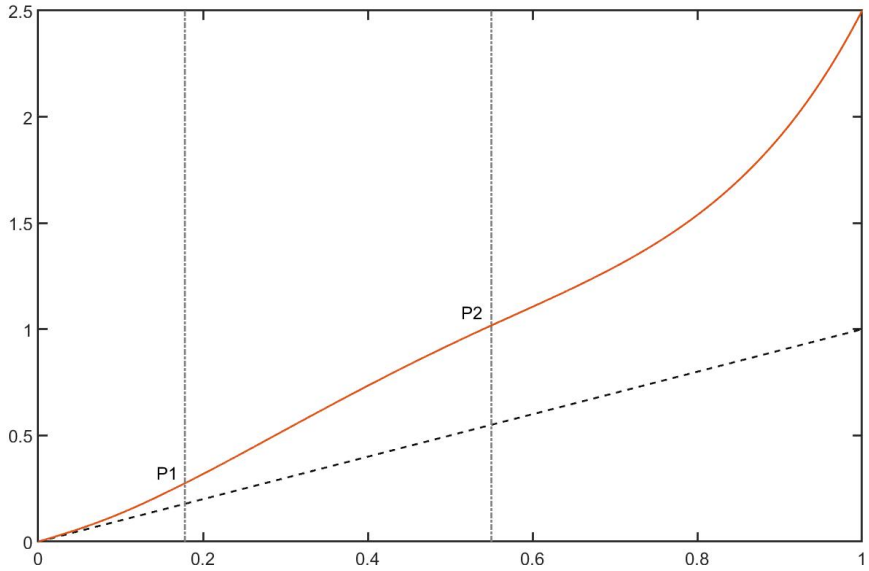

图 3　映射曲线示意图（headroom=2.5）

8.5　色彩信号动态范围转换过程

输入：待处理帧的RGB色域像素缓冲区$f[\mathrm{N_{frame}}][[3]$，暗部区域阈值P1，亮部区域阈值P2，色调映射曲线：基础曲线参数$P_{midtone}$，包含mp、ma、mb；暗部区域映射曲线参数P_{shadow}，包含$S1$、$S2$；亮部区域映射曲线参数$P_{hightlight}$，包含A，B，C，D，E。

输出：待处理帧经过动态范围转换处理的RGB色域像素缓冲区$f_{TM}[\mathrm{N_{frame}}][[3]$。

转换过程如下：

a） 计算特征值$f_{eigen}[i]$，i是当前待处理帧内像素索引，i的取值范围是[0，$\mathrm{N_{frame}}$)。

$$f_{eigen}[i] = \frac{MAX(\mathrm{f\,[i][0],f\,[i][1],f\,[i][2]})+\mathrm{y[i]}}{2} \quad \cdots\cdots(42)$$

其中，y[i]表示亮度值：　y[i] = 0.2126 * f[i][0] + 0.7152 * f[i][1] + 0.0722 * f[i][2]。

b） 对特征值$f_e[i]$进行色彩信号动态范围转换，计算$f_{e_TM}[i]$：

$$f_{e_TM}[i]=\begin{cases} Shadow(f_{eigen}[i]) & f_e[i]\le P1 \\ Midtone(f_{eigen}[i]) & P1<f_e[i]<P2 \\ Highlight(f_{eigen}[i]) & P2\le f_e[i] \end{cases} \quad \cdots\cdots(43)$$

c） 计算增益K：

$$K=f_{e_TM}[i]/f_{eigen}[i] \quad \cdots\cdots(44)$$

d） 将所有色域分量转换成线性信号：

$$\begin{cases} f_L[i][0]=EOTF(f[i][[0]) \\ f_L[i][1]=EOTF(f[i][[1]) \\ f_L[i][2]=EOTF(f[i][[2]) \end{cases} \quad \cdots\cdots(45)$$

e） 对于色域分量进行色彩信号动态范围转换：

$$\begin{cases} f_{TM}[i][0]=f_L[i][0]\times K \\ f_{TM}[i][1]=f_L[i][1]\times K \\ f_{TM}[i][2]=f_L[i][2]\times K \end{cases} \quad \cdots\cdots(46)$$

8.6 色彩校正过程

输入：待处理帧的RGB色域线性信号像素缓冲区$f_L[\mathrm{N_{frame}}][[3]$，待处理帧经过动态范围转换处理的RGB色域像素缓冲区$f_{TM}[\mathrm{N_{frame}}][[3]$，颜色校正强度color_saturation_factor。

输出：待处理帧经过显示适配处理的RGB色域像素缓冲区$f_{process}[\mathrm{N_{frame}}][[3]$。

色彩校正的具体过程如下：

a） 计算每一个像素点的色彩调整系数$\mathrm{S_{ca}}$：

$$S_{ca}=CLIP3\left(1.0,\ 1.1,\ \frac{sat_{pre}}{sat_{TM}}\right)\times \mathrm{sat_factor} \quad \cdots\cdots(47)$$

其中，

$$sat_{pre}=\mathrm{Max}(f_L[i][[0],f_L[i][[1],f_L[i][[2])-\mathrm{Min}\ (f_L[i][[0],f_L[i][[1],f_L[i][[2])$$

$$sat_{TM}=\mathrm{Max}(f_{TM}[i][[0],f_{TM}[i][[1],f_{TM}[i][[2])-\mathrm{Min}\ (f_{TM}[i][[0],f_{TM}[i][[1],f_{TM}[i][[2])$$

b） 对$f_{TM}[i][[3]$信号进行色彩调整，计算$f_{color}[i][[3]$：

$$\begin{cases} f_{color}[i][[0]=luma+(f_{TM}[i][0]-luma)\times S_{ca} \\ f_{color}[i][[1]=luma+(f_{TM}[i][1]-luma)\times S_{ca} \\ f_{color}[i][[2]=luma+(f_{TM}[i][2]-luma)\times S_{ca} \end{cases} \quad \cdots\cdots(48)$$

其中，

$$luma=0.2126\times f_{TM}[i][0]\ +0.7152\times f_{TM}[i][0]\ +\ 0.0722\times f_{TM}[i][0]$$

c） 将$f_{color}[i][[3]$信号转换成非线性信号，计算$f_{process}[i][[3]$：

$$\begin{cases} f_{process}[i][[0]=OETF(f_{color}[i][0]) \\ f_{process}[i][[1]=OETF(f_{color}[i][1]) \\ f_{process}[i][[2]=OETF(f_{color}[i][2]) \end{cases} \quad \cdots\cdots(49)$$

附 录 A
（资料性）
动态元数据提取方法

A.1 概述

本附录描述了在SDR前处理阶段如何提取动态元数据的推荐方法。

动态元数据提取过程如下：

a） 调用A.2 SDR信号预处理过程得到预处理后的色域像素缓冲区fpre[Nframe][3]；

b） 调用A.3计算元数据maximum_maxrgb_e；

c） 调用A.4计算元数据average_maxrgb_o；

d） 调用A.5计算元数据shadow_maxrgb_e和highlight_maxrgb_e；

e） 调用A.6计算元数据extended_headroom；

f） 调用A.7元数据时域滤波对动态元数据进行时域滤波；

A.2 SDR 信号预处理过程

输入参数包含：RGB像素缓冲区f[Nframe][3]或者YUV像素缓冲区YUV[Nframe][3]。

输出参数为：预处理后的RGB像素缓冲区fpre[Nframe][3]

预处理过程如下：

a） 若当前帧为RGB像素，则

$$fpre[Nframe][3] = f[Nframe][3]$$

其中f[Nframe][3]为当前帧或当前场景中的所有像素的R、G、B值，fpre[Nframe][3]为预处理后的当前帧或当前场景中的所有像素的R、G、B值。

b） 若当前帧为YUV格式，则

$$fpre[Nframe][1] = Clip3(0.0, 1.0, (f[Nframe][1] + 1.5748 \times f[Nframe][3]))$$

$$fpre[Nframe][2] = Clip3\big(0.0, 1.0, (f[Nframe][1] - 0.1873 \times f[Nframe][2] - 0.4681 \times f[Nframe][3])\big)$$

$$fpre[Nframe][3] = Clip3(0.0, 1.0, (f[Nframe][1] + 1.855 \times f[Nframe][2]))$$

其中 YUV[Nframe][3]为当前帧或当前场景中的所有预处理像素的Y、U、V值，fpre[Nframe]为预处理后的当前帧或当前场景中的所有像素的R、G、B值。

A.3 动态元数据 maximum_maxrgb_e[w]的计算

maximum_maxrgb_e[w]主要用于指示当前帧或当前场景主要内容的最大亮度值，具体计算过程如下。

a） 对于当前帧或当前场景中的所有预处理像素的R、G、B值（fpre[index][0]、fpre[index][1]和fpre[index][2]）计算其最大值（fMAX[index]）：

$$fMAX[Nframe][3] = Max(Max(f[index][0], f[index][1]), f[index][2])$$

式中，index为像素索引值，0≤index<Nframe。

b） 计算当前帧或当前场景中的所有像素fMAX[index]中的最大值fMAX_MAX：

```
fMAX_MAX=0.0;
for (i=0;i<Nframe;i++) {
    fMAX_MAX=Max(fMAX_MAX,fMAX[i])
}
```

c） 按照公式（A.1）计算当前帧或当前场景中的元数据maximum_maxrgb_e[w]。

$$maximum_maxrgb_e[w] = Floor(fMAX_MAX \times 4095) \quad \cdots\cdots (A.1)$$

A.4 动态元数据 average_maxrgb_o[w]的计算

average_maxrgb_o[w]主要用于指示当前帧或当前场景主要内容的亮度平均值，其计算过程如下：

a) 对于当前帧或当前场景中的所有预处理像素的R、G、B值（fpre[index][0]、fpre[index][1]和fpre[index][2]）计算其最大值（fMAX[index]）：

$$\mathrm{fMAX[Nframe][3] = Max(Max(f[index][0], f[index][1]), f[index][2])}$$

式中，index为像素索引值，0≤index<Nframe。

b) 计算当前帧或当前场景中的所有像素fMAX[index]在线性空间的对应值：

$$\mathrm{fOMAX[index] = EOTF(fMAX[index])} \quad \cdots\cdots（A.2）$$

c) 计算当前帧或当前场景中的所有像素fOMAX[index]的平均值fMAX_AVG。

$$f_{MAX_AVG} = \frac{\sum_{i=0}^{Nframe-1} fOMAX[i]}{\mathrm{Nframe}} \quad \cdots\cdots（A.3）$$

d) 将fMAX_AVG按照公式（A.4）计算当前帧或当前场景中的元数据average_maxrgb_o[w]：

$$\mathrm{average_maxrgb_o[w] = Floor(fMAX_AVG \times 4095)} \quad \cdots\cdots（A.4）$$

A.5 动态元数据 shadow_maxrgb_e[w]和 highlight_maxrgb_e[w]的计算

shadow_maxrgb_e[w]和highlight_maxrgb_e[w]主要用于指示当前帧或当前场景主要内容的暗部特征和亮部特征值，具体计算过程如下：

a) 对于当前帧或当前场景中的所有预处理像素的R、G、B值（fpre[index][0]、fpre[index][1]和fpre[index][2]）计算其最大值（fMAX[index]）：

$$\mathrm{fMAX[index]=Max(Max(f[index][0],f[index][1]),f[index][2])}$$

式中，index为像素索引值，0≤index<Nframe。

b) 计算fMAX[Nframe]的直方图His[i],0≤i≤1024

```
for(i=0; i<Nframe; i++){
    His[Floor(fMAX[i]×1024)]++;
}
```

c) 计算当前帧或当前场景中的中灰值Midgrey：

for(i=0; i<10; i++){

$$f_{shadow_AVG} = \frac{\sum_{i=0}^{Midgrey} His[[i] \times (i/1024)}{\sum_{i=0}^{midgrey} His[i]}$$

$$f_{highlight_AVG} = \mathrm{OETF}\left(\frac{\sum_{i=Midgrey}^{1024} His[[i] \times \mathrm{EOTF}(i/1024)}{\sum_{i=Midgrey}^{1024} His[i]}\right)$$

$$Midgrey = \mathrm{OETF}\left(\mathrm{Pow}\left(\mathrm{EOTF}\left(f_{shadow_{AVG}}\right) * \mathrm{EOTF}\left(f_{highlight_{AVG}}\right), 0.5\right)\right)$$

}

d) 按照公式（A.5）计算当前帧或当前场景中的highlight_maxrgb_e[w]:

$$\mathrm{highlight_maxrgb_e[w] = Floor}(f_{highlight_AVG} \times 4095) \quad \cdots\cdots（A.5）$$

e) 按照公式（A.6）计算当前帧或当前场景中的shadow_maxrgb_e[w]:

$$\mathrm{shadow_maxrgb_e[w] = Floor}((f_{shadow_{AVG}} + Midgrey)/2 \times 4095) \quad \cdots\cdots（A.6）$$

A.6 动态元数据 extended_headroom [w]的计算

extended_headroom [w]表示显示内容在线性空间扩展裕量的最大值。输入帧的RGB像素数据fpre[Nframe][3] 需通过亮度转换与直方图统计进行预处理，将RGB色域像素转化为亮度分量，亮度值量化为32个区间，统计每个区间的频数His[i]：

a) 对于当前帧，计算fpre[Nframe]的归一化直方图His_norm[i], 0≤i＜32：

计算fMAX[Nframe]的直方图His[i], 0≤i＜32;

```
  for(i=0; i< Nframe; i++){
      Gray = 0.299 * fpre[Nframe][0] + 0.587 * fpre[Nframe][1] + 0.114 * fpre[Nframe][2];
      His[Floor(Gray×256/8)]++;
  }
```

His[i]按总像素数归一化为His_norm[i]

```
  for(i=0; i< 32; i++){
```

```
        His_norm[i] = His[i] / (Width * Height);
  }
```

b) 基于直方图的场景亮度分布，确定动态范围扩展需求扩展系数HR_Hist

$$\text{hist_factor} = \frac{\sum_{i=0}^{7} w[i] \times His_norm[24+i]}{32} \quad \cdots\cdots(A.7)$$

其中$w = \{0.0, 0.2857, 0.5714, 0.8571, 1.1429, 1.4286, 1.7143, 2.0\}$；

$$\text{HR_HIST} = \begin{cases} \text{HR_Thresh_MAX} & \text{hist_factor} < 0.2 \\ 2.78 \times \text{hist_factor}^{2.0} + (-5.28 \times \text{hist_factor}) + 3.94 & 0.2 \le \text{hist_factor} \le 0.8 \\ \text{HR_Thresh_MIN} & \text{hist_factor} > 0.8 \end{cases} \quad \cdots\cdots(A.8)$$

c) 结合平均亮度（apl）场景调整扩展净空，防止高动态范围场景过曝

apl计算：

$$\text{apl} = \frac{1}{32}\sum_{i=1}^{32} His_norm[i] \quad \cdots\cdots(A.9)$$

slope斜率系数：

$$\text{slope} = \frac{\text{HR_Thresh_MAX} - \text{HR_Thresh_MIN}}{\text{aplThresh} - \text{highlimit}} \quad \cdots\cdots(A.10)$$

d) 计算apl场景扩展净空HR_APL

$$\text{HR_APL} = \begin{cases} \text{HR_Thresh_MAX} & \text{apl} < 0.05 \\ \text{slope} \times (\text{apl} - \text{aplThresh}) + \text{HR_Thresh_MAX} & 0.05 \le \text{apl} \le 0.5 \\ \text{HR_Thresh_MIN} & \text{apl} > 0.5 \end{cases} \quad \cdots\cdots(A.11)$$

e) 通过weight权重因子平衡直方图与APL的动态范围扩展系数：

$$\text{extended_headroom}[w] = \text{weight} \times \text{headroomAPL} + (1 - \text{weight}) \times \text{headroomHist} \quad \cdots\cdots(A.12)$$

A.7 动态元数据的时域滤波

对于当前帧提取的动态元数据进行时域滤波的过程如下：

a) 创建动态元数据队列dynamic_metadata_fifo，队列长度为M，M为32， dynamic_metadata_fifo[dynamic_metadata_fifo_Num]表示队列中第dynamic_metadata_fifo_Num个元数据，dynamic_metadata_fifo_Num为队列中有效数据的数量，初始化为0。

b) 调用A.2～A.7生成当前第N帧动态元数据dynamic_metadata_org，其中N为帧序号，N≥0。

c) 如果N等于0或当前帧为场景切换帧，则dynamic_metadata_fifo[0]= dynamic_metadata_org，dynamic_metadata_fifo_Num=1。

否则:

```
  如果dynamic_metadata_fifo_Num小于M则:
    dynamic_metadata_fifo[dynamic_metadata_fifo_Num]= dynamic_metadata_org
    dynamic_metadata_fifo_Num= dynamic_metadata_fifo_Num+1
  如果dynamic_metadata_fifo_Num等于M，则:
    for (n=0; n<M-1; n++) {
        dynamic_metadata_fifo[n+1]= dynamic_metadata_fifo[n]
    }
    dynamic_metadata_fifo[M-1]= dynamic_metadata_org
```

d) 输出时域滤波之后的元数据dynamic_metadata_fliter，见公式（A.13）。

$$\text{dynamic_metadata_fliter} = \frac{\sum_{i=0}^{\text{dynamic_metadata_fifo_Num}-1} \text{dynamic_metadata_fifo}[i]}{\text{dynamic_metadata_fifo_Num}} \quad \cdots\cdots(A.13)$$

附 录 B
（资料性）
元数据在 HEVC/VVC 码流中的封装

B.1 元数据在 HEVC/VVC 码流中的封装

元数据在HEVC/VVC码流中，按表B.1要求封装。

表 B.1 HEVC/VVC 动态元数据扩展定义

user_data_registered_itu_t_t35(payloadSize) {	Descriptor
itu_t_t35_country_code	b(8)
if(itu_t_t35_country_code != 0xFF){	
i = 1	
}	
else {	
itu_t_t35_country_code_extension_byte	b(8)
i = 2	
}	
do {	
itu_t_t35_payload_byte	b(8)
i++	
} while(i < payloadSize)	
}	

动态元数据封装在表B.1的user_data_registered_itu_t_t35()中的itu_t_t35_payload_byte，其封装结构如表B.2所示。

表 B.2 HEVC/VVC 动态元数据扩展定义

itu_t_t35_payload_bye(){	Descriptor
terminal_provide_code	f(16)
terminal_provide_oriented_code	f(16)
dynamic_metadata ()	
}	

B.2 语义

动态元数据dynamic_metadata ()
动态元数据dynamic_metadata ()的语法定义参见本文件7.1相关内容。
ITU-T T.35 **国家码**itu_t_t35_country_code
8位无符号整数，标识ITU-T T.35规定的国家识别码，0x26代表中国。
ITU-T T.35 **终端制造商码**itu_t_t35_terminal_provide_code
16位无符号整数，标识ITU-T T.35规定的终端制造商码，0x0004代表UWA。
ITU-T T.35 **终端制造商指向码**itu_t_t35_terminal_provide_oriented_code
16位无符号整数，标识ITU-T T.35规定的终端制造商指向码，0x0005代表版本1.0。
其中表B.3中的terminal_provide_code本方案中使用0x0030。terminal_provide_oriented_code版本号与标识的映射表参见本文件表B.3的内容。

表 B.3 版本号与标识的映射表

SDR动态元数据 版本号	terminal_provide_oriented_code码字
1.0	0x0030
2.0	0x0031

3.0	0x0032
4.0	0x0033

参 考 文 献

[1] GB/T 46269.1-2025 高动态范围(HDR)视频技术 第1部分:元数据及适配
[2] GB/T 41809-2022 超高清晰度电视系统节目制作和交换参数值
[3] ITU-R BT.709 节目制作和国际节目交换中使用高清晰度电视标准的参数值(Parameter values for the HDTV standards for production and international programme exchange)