GY

Radio, Television, and Network Audiovisual Industry Standard of the People's Republic of China

GY/T 358-2022

Technical Requirements for Display Adaptation Metadata of High Dynamic Range Television Systems

Issued on January 30, 2022 Implemented on January 30, 2022

Issued by the National Radio Television Administration

Contents

Foreword

This document is drafted in accordance with GB/T 1.1-2020 *Directives for standardization: Part 1: Rules for the structure and drafting of standardizing documents*.

This document is under the jurisdiction of the National Radio, Film and Television Standardization Technical Committee (SAC/TC 239).

Drafting organizations of this document: China Media Group; the Academy of Broadcasting Science, NRTA; the Academy of Broadcasting Planning, NRTA; Peking University; the National Testing and Inspection Center for Radio & TV Products of China; China Electronics Standardization Institute; Huawei Technologies Co., Ltd.; HiSilicon (Shanghai) Technologies Co., Ltd.; Sony (China) Ltd.; Guangdong Bohua UHD Innovation Center Co., Ltd; Zhongguancun Audio-Visual Industry Technology Innovation Alliance; Chengdu Sobey Digital Technology Co., Ltd.; Hangzhou Arcvideo Technology Co., Ltd.; Guangzhou Kuvision Digital Technology Co., Ltd.; and Shenzhen Skyworth Digital Technology Co., Ltd.

Drafters of this document: Pan Xiaofei, Zhou Yun, Ning Jinhui, Yu Quanhe, Yuan Le, Wang Shanshe, Wu Weihua, Niu Zhaohui/Chaohui, Cui Jianwei, Wang Pei, Guo Xiaoqiang, Hu Xiao, Wang Huiming, Zhang Qian, Ruan Weihong, Xu Weiwei, Wang Yichuan, Li Yan, Liu Xin, Wang Yaming, Xie Chaoping, Chen Yong, Zeng Zhihua, Xue Tao, wang Zhenzhong, Zhu Yi, Peng Fei, Yang Chen, Deng Xiangdong, Zhang Weimin, Ma Siwei, Liu Yi, Chen Hu, Wang Zheng, Zhu Jun, Liu Bo, Wang Dongfei, Wei Na, Li Xiaoyu, and Jia Tao.

Introduction

The issuing body of this document calls for attention to the fact that the declaration of compliance with this document may use the following related authorized and pending patents that relate to related contents of this document.

No.	Chapter No.	Patent name
1	10.4	VIDEO SIGNAL PROCESSING METHOD AND APPARATUS
2	10.4	VIDEO SIGNAL PROCESSING METHOD AND APPARATUS
3	10.1	VIDEO SIGNAL PROCESSING METHOD AND APPARATUS
4	7.2, 10.1, and 11.1	METHOD AND APPARATUS FOR PROCESSING IMAGE SIGNAL CONVERSION, AND TERMINAL DEVICE
5	10.4 and 10.5	IMAGE PROCESSING METHOD AND APPARATUS
6	10.1 and 11.1	IMAGE PROCESSING SYSTEM AND METHOD FOR GENERATING HIGH DYNAMIC RANGE IMAGE
7	7 and 8	IMAGE ENCODING AND DECODING METHOD AND DEVICE
8	10.4 and 10.5	IMAGE PROCESSING METHOD AND APPARATUS, AND TERMINAL DEVICE
9	10.1	PHOTOGRAPHING METHOD, RELATED DEVICE AND COMPUTER STORAGE MEDIUM
10	10.1 and 11.1	HIGH DYNAMIC RANGE IMAGE SYNTHESIS METHOD AND APPARATUS

The issuing body of this document has no position as to the authenticity, validity, and scope of the patents.

Patent holders have assured the issuing body of this document that they are willing to negotiate patent licenses under reasonable and non-discriminatory terms and conditions with any applicant. Patent holder's statements are registered with the issuing body of this document. The relevant information can be obtained using the following contact information:

Name of the patents holder: Huawei Technologies Co., Ltd.

Address: A, Huawei Base, Bantian, Longgang District, Shenzhen, Guangdong.

Attention is drawn to the possibility that some of the elements of this document may relate to patents other than those identified. The issuing body of this document shall not be held responsible for identifying any or all such patents.

Technical Requirements for Display Adaptation Metadata of High Dynamic Range Television Systems

1 Scope

This document specifies the technical requirements for high dynamic range (HDR) video display adaptation in the production, transmission, reception, display, and other links of HDR of ultra high definition televisions.

This document is applicable to HDR video display adaptation on various terminals such as cable televisions, direct broadcast satellites, terrestrial televisions, IPTVs/OTTs, and outdoor screens.

2 Normative References

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced documents (including any amendments) applies.

GY/T 315-2018 Image Parameter Values for High Dynamic Range Televisions for Use in Program Production and Exchange (ITU-R BT.2100-1, MOD)

ISO 11664-1:2007/CIE S 014-1:2006 Colorimetry - Part 1: Standard Colorimetric Observers

ITU-T T.35 Procedure for the Allocation of ITU-T Defined Codes for Non-standard Facilities

ITU-T H.265 High Efficiency Video Coding

3 Terms and Definitions

The following terms and definitions are applicable to this document.

3.1

Dynamic metadata

Metadata associated with each frame of picture. The metadata varies with pictures.

3.2

Static metadata

Metadata associated with a picture sequence. The metadata remains unchanged in the picture sequence.

3.3

Source picture

A high dynamic range picture obtained through program production.

4 Acronyms and Abbreviations

The following acronyms and abbreviations apply to this document.

AVS2 High Efficiency Coding of Audio and Video - Part 1: Video (High efficiency coding of audio and video—Part 1:video)

EOTF electro-optical transfer function (Electro-Optical Transfer Function)

HDR high dynamic range (High Dynamic Range)

HLG hybrid log-gamma (Hybrid Log-Gamma)

IPTV Internet Protocol television (Internet Protocol Television)

MSB most significant bit (Most Significant Bit)

OTT over the top (Over The Top)

PQ perception quantization (Perception Quantization)

SDR standard dynamic range (Standard Dynamic Range)

SEI supplemental enhancement information (Supplemental Enhancement Information)

5 Symbols and Operations

5.1 General Requirements

For the mathematical operators and precedences used in this document, refer to those in the C language. However, integer division and arithmetic shift operations are specifically defined. Unless otherwise specified, numbering and counting conventions generally begin from 0.

5.2 Arithmetic Operators

Definitions of arithmetic operators should comply with provisions in Table 1.

Table 1 Definition of arithmetic operator

Arithmetic operator	**Definition**
+	Addition
–	Subtraction (as a binary operator) or negation (as a unary prefix operator)
×	Multiplication
a^b	Exponentiation, indicating the *bth* power of *a*. It may also indicate a superscript.
/	Integer division with truncation of the result toward zero. For example, 7/4 and –7/–4 are truncated to 1, and –7/4 and 7/–4 are truncated to –1.
÷	Division without truncation or rounding

Arithmetic operator	Definition
$\frac{a}{b}$	Division without truncation or rounding
$\sum_{i=a}^{b} f(i)$	Cumulative sum of the function *f*(*i*) when the independent variable *i* takes all integer values from *a* to *b* (including *b*).
a % *b*	Modulo operation, which indicates a remainder of *a* divided by *b*, where both *a* and *b* are positive integers.
$\lfloor . \rfloor$	Rounding down

5.3 Logical Operators

Definitions of logical operators should comply with provisions in Table 2.

Table 2 Definition of logical operator

Logical operator	Definition
a && *b*	Logical AND operation between *a* and *b*
a || *b*	Logical OR operation between *a* and *b*

5.4 Relational Operators

Definitions of relational operators should comply with provisions in Table 3.

Table 3 Definition of relational operator

Relational operator	Definition
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	Not equal to

5.5 Bitwise Operators

Definitions of bitwise operators should comply with provisions in Table 4.

Table 4 Definition of bitwise operator

Bitwise operator	**Definition**
&	AND operation
|	OR operation
~	Negation operation
a>>*b*	Right shift of a two's complement integer representation of *a* by *b* digits. This operation is defined only when *b* is a positive number.
a<<*b*	Left shift of a two's complement integer representation of *a* by *b* digits. This operation is defined only when *b* is a positive number.

5.6 Assignment

Definitions of assignment operations should comply with provisions in Table 5.

Table 5 Definition of assignment operation

Assignment operation	**Definition**
=	Assignment operator
++	Increment. *x*++ is equivalent to *x* = *x* + 1. When the operator is used in an array subscript, a value of a variable is calculated prior to the auto-increment operation.
—	Decrement. x— is equivalent to *x* = *x* – 1. When the operator is used in an array subscript, a value of a variable is calculated prior to the auto-decrement operation.
+=	Auto-increment by amount specified. For example, *x* += 3 is equivalent to *x* = x + 3, and *x* += (–3) is equivalent to *x* = *x* + (–3).
–=	Auto-decrement by amount specified. For example, x –= 3 is equivalent to *x* = *x* – 3, and *x* –= (–3) is equivalent to *x* = *x* – (–3).

5.7 Mathematical Functions

For definitions of mathematical functions, refer to formula (1) to formula (10).

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

x is an independent variable.

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

x is an independent variable.

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

i is a lower bound.

j is an upper bound.

x is an independent variable.

$$\mathrm{Min}(x,y)=\begin{cases} x, & x\le y \\ y, & x>y \end{cases} \tag{4}$$

x is an independent variable.

y is an independent variable.

$$\mathrm{Max}(x,y)=\begin{cases} x, & x\ge y \\ y, & x<y \end{cases} \tag{5}$$

x is an independent variable.

y is an independent variable.

$$\mathrm{Median}(x,y,z)=x+y+z-\mathrm{Min}\big(x,\mathrm{Min}(y,z)\big)-\mathrm{Max}(x,\mathrm{Max}(y,z)) \tag{6}$$

x is an independent variable.

y is an independent variable.

z is an independent variable.

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

x is an independent variable.

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

x is an independent variable.

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

x is an independent variable.

e is a base of a natural logarithm, whose value is 2.718281828...

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

x is an independent variable.

y is an independent variable.

5.8 Structure Relational Operators

Definitions of structure relational operators should comply with provisions in Table 6.

Table 6 Definition of structure relational operator

Structure relational operator	Definition
->	For example, *a* -> *b* indicates that *a* is a structure and *b* is a member variable of *a*.

5.9 Bitstream Syntax Description Method

The description method of the bitstream syntax is similar to that of the C language. The syntax elements in a bitstream are represented by bold characters. Each syntax element is described based on its name, syntax, and semantics.

The syntax table describes the whole set of all bitstream syntaxes that comply with this document. Additional syntax restrictions are described in the corresponding sections.

Table 7 shows an example of pseudocode for describing the syntax. When a syntax element appears, it indicates that a data element is read from the bitstream.

Table 7 Pseudocode of syntax description

Pseudocode
/* A statement is a descriptor of a syntax element or indicates the existence, type, and quantity of the syntax elements, as in the following two examples. */
syntax_element
conditioning statement
/* A group of statements enclosed in curly brackets is a compound statement and is considered functionally as a single statement.*/
{
statement
...
}
/* A "while" statement tests whether a condition is TRUE. If TRUE, the loop body is repeatedly executed until the condition is no longer TRUE. */
while (condition)
statement

Pseudocode
/* A "do … while" statement first executes the loop body once, and then tests whether a condition is TRUE. If TRUE, the loop body is repeatedly executed until the condition is no longer TRUE. */
do
statement
while (condition)
/* An "if ... else" statement first tests a condition. If TRUE, a primary statement is executed, and otherwise, an alternative statement is executed. If the alternative statement does not need to be executed, the "else" part of the structure and the related alternative statement can be ignored. */
if (condition)
primary statement
else
alternative statement
/* A "for" statement first executes an initial statement, and then tests a condition. If the condition is TRUE, a primary statement and a subsequent statement are repeatedly executed until the condition is no longer TRUE. */
for (initial statement; condition; subsequent statement)
primary statement

The parsing process and the decoding process are described using text and pseudocode similar to that in C language.

5.10 Functions

5.10.1 byte_aligned( )

If the current position in the bitstream is byte-aligned, TRUE is returned. Otherwise, FALSE is returned.

5.10.2 next_start_code( )

Seeks the next start code in the bitstream, and points the bitstream pointer to the first binary bit of the start code prefix. The definition of the function should comply with provisions in Table 8.

Table 8 Definition of next_start_code function

Function definition	Value
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'
}	

5.10.3 read_bits(n)

Returns the next n binary bits in the bitstream in an MSB first manner, and advances the bitstream pointer by n binary bits. If n is equal to 0, read_bits(n) returns 0 and does not advance the bitstream pointer.

5.11 Descriptors

Descriptors indicate the parsing processes of different syntax elements and should comply with provisions in Table 9.

Table 9 Descriptor

Descriptor	Description
b(8)	A byte with an arbitrary value. The parsing process is specified by the returned value of the function read_bits(8).
f(n)	n consecutive binary bits with a specific value. The parsing process is specified by the returned value of the function read_bits(n).
r(n)	n consecutive '0's. The parsing process is specified by the returned value of the function read_bits(n).
u(n)	An n-bit unsigned integer. In the syntax table, if n is "v", the number of bits is determined by other syntax element values. The parsing process is specified by the returned value of the function read_bits(n). The returned value is indicated by an MSB-first binary number.

5.12 Reserved, Forbidden, and Marker Bit

In the bitstream syntax defined in this document, the values of some syntax elements are marked as "reserved" (reserved) or "forbidden" (forbidden).

The term "reserved" defines some particular syntax element values for future extensions of this document. These values should not appear in the bitstream that complies with this document.

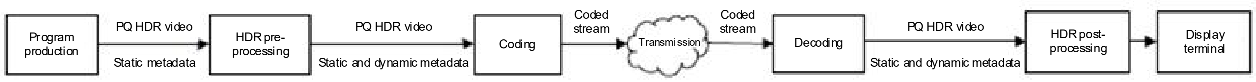
The term "forbidden" defines some particular syntax element values that should not appear in the bitstream that complies with this document. The "marker bit" (marker_bit) indicates that a value of this bit should be '1'.

"Reserved bits" (reserved_bits) in the bitstream indicate that some syntax units are reserved for future extensions of this document, and these bits should be ignored in decoding processing. "Reserved bits" should not have more than 21 consecutive '0's starting from any byte-aligned position.

6 End-to-end System Architecture

For the end-to-end system of the HDR video of the PQ curve, refer to Figure 1. The PQ HDR video and static metadata are obtained through program production. Technical parameters of the PQ HDR video should comply with provisions in GY/T 315-2018. HDR pre-processing is performed to extract dynamic metadata, to obtain the HDR video and metadata that are used for coding transmission. After coding and encapsulation are performed on the HDR video and metadata, the processed HDR video and metadata are transmitted over a network. At the receiver, HDR post-processing uses the transferred HDR metadata to implement a display adaptation function. The decoder performs decoding to obtain the PQ HDR video and metadata. An SDR display terminal reconstructs, based on the PQ HDR video and metadata, an SDR video for display. An HDR display terminal directly performs HDR display if a display capability of the terminal matches a luminance of the HDR video that is produced and transmitted. If the display capability of the terminal does not match the luminance of the HDR video that is produced and transmitted, the HDR display terminal performs adaptation based on the display capability of the terminal and the HDR video and metadata, and then performs display.

Figure 1 End-to-end System of HDR Video of PQ Curve



For a recommended end-to-end system of the HDR video of the HLG curve, refer to Figure 2. The HLG HDR video is obtained through program production. Technical parameters of the HLG HDR video should comply with provisions in GY/T 315-2018. The HLG HDR video is transmitted over the network after being encoded. At the receiver, a decoder performs decoding to obtain the HLG HDR video, and then an SDR terminal and an HDR terminal directly display the HLG HDR video.

If a decoding device that supports a PQ HDR post-processing display adaptation function has been deployed at the receiver, the method in Annex A may also be used for processing when the processing capability of the coding device is sufficient.

Figure 2 End-to-end System of HDR Video of HLG Curve

7 Syntax and Semantics of Metadata

7.1 Syntax of Static Metadata

The definition of the static metadata should comply with provisions in Table 10.

Table 10 Definition of static metadata

Definition of static metadata	Descriptor
hdr_static _metadata ( ) {	
for (c=0; c<3; c++) {	
display_primaries_x[c]	u(16)
display_primaries_y[c]	u(16)
}	
white_point_x	u(16)
white_point_y	u(16)
max_display_mastering_luminance	u(16)
min_display_mastering_luminance	u(16)
max_content_light_level	u(16)
max_picture_average_light_level	u(16)
}	

7.2 Semantics of Static Metadata

7.2.1 X coordinate of three primaries of the mastering display, Y coordinate of three primaries of the mastering display display_primaries_x[c], display_primaries_y[c]

A 16-bit unsigned integer. It indicates the normalized chrominance *x* coordinate and *y* coordinate of the primaries of the mastering display. The coordinates should comply with CIE 1931 specified in ISO 11664-1:2007/CIE S 014-1:2006, in unit of 0.00002, and ranges from 0 to 50000. The values of *c* that are 0, 1, and 2 respectively correspond to three colors: green, blue, and red.

7.2.2 X coordinate of the standard white light of the mastering display, Y coordinate of the standard white light of the mastering display white_point_x, white_point_y

A 16-bit unsigned integer. It indicates the normalized chrominance *x* coordinate and *y* coordinate of the standard white light of the display device, in unit of 0.00002, and ranges from 0 to 50000. The coordinates should comply with CIE 1931 specified in ISO 11664-1:2007/CIE S 014-1:2006. The standard white light coordinates are $x = 0.3127$ and $y = 0.3290$.

7.2.3 Maximum display luminance of the mastering display max_display_mastering_luminance

A 16-bit unsigned integer. It indicates the maximum display luminance of the mastering display, in unit of 1 cd/m^2, and ranges from 1 cd/m^2 to 65535 cd/m^2.

7.2.4 Minimum display luminance of the mastering display min_display_mastering_luminance

A 16-bit unsigned integer. It indicates the minimum display luminance of the mastering display, in unit of 0.0001 cd/m^2, and ranges from 0.0001 cd/m^2 to 6.5535 cd/m^2.

The value of *max_display_mastering_luminance* should be greater than that of the *min_display_mastering_luminance*.

7.2.5 Maximum luminance of the displayed content max_content_light_level

A 16-bit unsigned integer. It indicates the maximum luminance of the displayed content, in unit of 1 cd/m^2, and ranges from 1 cd/m^2 to 65535 cd/m^2.

The value of *max_content_light_level* is the maximum value of the maximum luminance *PictureMaxLightLevel* of all display pictures of certain display content. The maximum luminance *PictureMaxLightLevel* of a display picture is calculated as follows.

a) The maximum values *maxRGB* of the R, G, and B components of all pixels in the valid display area of the display picture are sequentially calculated. The valid display area is a rectangular area jointly defined by *display_horizontal_size* and *display_vertical_size*. *display_horizontal_size* indicates the number of samples in each row of the coded picture, and *display_vertical_size* indicates the number of rows of the coded picture.

   1) The non-linear (R', G', B') value of a pixel is converted to the linear (R, G, B) value, and calibrated to the value in unit of 1 cd/m^2.

   2) The maximum value *maxRGB* of the R, G, and B components of the pixel is calculated based on the calibrated (R, G, B) value of the pixel.

b) *PictureMaxLightLevel* of the display picture is equal to the maximum value of *maxRGB* of all pixels in the valid display area.

7.2.6 Maximum picture average luminance of the displayed content max_picture_average_light_level

A 16-bit unsigned integer. It indicates the maximum picture average luminance of the displayed content, in unit of 1 cd/m^2, and ranges from 1 cd/m^2 to 65535 cd/m^2.

The value of *max_picture_average_light_level* is the maximum value of the picture average luminance *PictureAverageLightLevel* of all display pictures of certain display content. The average luminance *PictureAverageLightLevel* of a display picture is calculated as follows.

a) The maximum values *maxRGB* of the R, G, and B components of all pixels in the valid display area of the display picture are sequentially calculated. The valid display area is a rectangular area jointly defined by *display_horizontal_size* and *display_vertical_size*. *display_horizontal_size* indicates the number of samples in each row of the coded picture, and *display_vertical_size* indicates the number of rows of the coded picture.

   1) The non-linear (R', G', B') value of a pixel is converted to the linear (R, G, B) value, and calibrated to the value in unit of 1 cd/m^2.

   2) The maximum value *maxRGB* of the R, G, and B components of the pixel is calculated based on the calibrated (R, G, B) value of the pixel.

b) *PictureAverageLightLevel* of the display picture is equal to the average value of *maxRGB* of all pixels in the valid display area.

7.3 Syntax of Dynamic Metadata

The definition of the dynamic metadata should comply with provisions in Table 11. Annex B shows a suggestion on a dynamic metadata extraction method.

Table 11 Definition of dynamic metadata

Definition of dynamic metadata	Descriptor
hdr_dynamic_metadata ( ) {	
system_start_code	u(8)
if(system_start_code==0x01){	
num_windows=1	
for(w = 0; w < num_windows; w++) {	
minimum_maxrgb_pq[w]	u(12)
average_maxrgb_pq[w]	u(12)
variance_maxrgb_pq[w]	u(12)
maximum_maxrgb_pq[w]	u(12)
}	
for(w = 0; w < num_windows; w++) {	
tone_mapping_enable_mode_flag[w]	u(1)
if(tone_mapping_enable_mode_flag [w]==1){	
tone_mapping_param_enable_num [w]	u(1)
tone_mapping_param_num [w]++	
for(i=0; i< tone_mapping_param_num [w]; i++){	

Definition of dynamic metadata	Descriptor
targeted_system_display_maximum_luminance_pq[i][w]	u(12)
base _enable_flag[i][w]	u(1)
if(base _enable_flag[i][w]){	
base_param_m_p[i][w]	u(14)
base_param_m_m[i][w]	u(6)
base_param_m_a[i][w]	u(10)
base_param_m_b[i][w]	u(10)
base_param_m_n[i][w]	u(6)
base_param_K1[i][w]	u(2)
base_param_K2[i][w]	u(2)
base_param_K3[i][w]	u(4)
base_param_Delta_enable_mode[i][w]	u(3)
base_param_enable_Delta[i][w]	u(7)
}	
3Spline _enable_flag[i][w]	u(1)
if(3Spline_enable_flag[i][w]){	
3Spline_enable_num[i][w]	u(1)
3Spline _num++;	
for(j = 0; j < 3Spline _num; j ++) {	
3Spline_TH_enable_mode[j] [i][w]	u(2)
if((3Spline_TH_mode[j][i] [w]==0)||(3Spline_TH_mode[j][i] [w]==2)){	
3Spline_TH_enable_MB [j][i][w]	f(8)
}	
3Spline_TH_enable[j][i][w]	f(12)
3Spline_TH_enable_Delta1 [j][i][w]	f(10)
3Spline_TH_enable_Delta2 [j][i][w]	f(10)
3Spline_enable_Strength[j][i][w]	f(8)
}	
}	
}	

Definition of dynamic metadata	Descriptor
}	
color_saturation_mapping_flag[w]	u(1)

Table 11 (continued)

Definition of dynamic metadata	Descriptor
if(color_saturation_mapping_flag[w]) {	
color_saturation_num[w]	u(3)
for(i = 0; i< color_saturation_num [w]; i++) {	
color_saturation_gain[i][w]	u(8)
}	
}	
}	
}	
}	

7.4 Semantics of Dynamic Metadata

7.4.1 System start code system_start_code

An 8-bit unsigned integer. It indicates the system version number.

7.4.2 Minimum value in the maximum RGB component values minimum_maxrgb_pq[w]

A 12-bit unsigned integer. It indicates the minimum luminance of the source picture, and ranges from 0 to 4095.

7.4.3 Average value of the maximum RGB component values average_maxrgb_pq[w]

A 12-bit unsigned integer. It indicates the average luminance of the source picture, and ranges from 0 to 4095.

7.4.4 Variance of the maximum RGB component values variance_maxrgb_pq[w]

A 12-bit unsigned integer. It indicates the luminance change range of the source picture, and ranges from 0 to 4095.

7.4.5 Maximum value in the maximum RGB component values maximum_maxrgb_pq[w]

A 12-bit unsigned integer. It indicates the maximum luminance of the source picture, and ranges from 0 to 4095.

7.4.6 Tone mapping flag tone_mapping_enable_mode_flag[w]

A binary variable. It indicates the transmission tone mapping flag. The value is 0 or 1.

7.4.7 Number of tone mapping parameter groups tone_mapping_param_enable_num[w]

A 1-bit unsigned integer. It indicates the number of tone mapping parameter groups. The value is 0 or 1.

7.4.8 Maximum luminance of reference target display targeted_system_display_maximum_luminance_pq[i][w]

A 12-bit unsigned integer. It indicates the maximum luminance of the reference target display corresponding to the metadata, and ranges from 0 to 4095.

7.4.9 Base curve flag base_enable_flag[i][w]

A binary variable. It indicates the transmission base curve flag. The value is 0 or 1.

7.4.10 Base curve parameter m_p base_param_m_p[i][w]

A 14-bit unsigned integer. It indicates the base curve parameter *m_p*, and ranges from 0 to 16383.

7.4.11 Base curve parameter m_m base_param_m_m[i][w]

A 6-bit unsigned integer. It indicates the base curve parameter *m_m*, and ranges from 0 to 63.

7.4.12 Base curve parameter m_a base_param_m_a[i][w]

A 10-bit unsigned integer. It indicates the base curve parameter *m_a*, and ranges from 0 to 1023.

7.4.13 Base curve parameter m_b base_param_m_b[i][w]

A 10-bit unsigned integer. It indicates the base curve parameter *m_b*, and ranges from 0 to 1023.

7.4.14 Base curve parameter m_n base_param_m_n[i][w]

A 6-bit unsigned integer. It indicates the base curve parameter *m_n*, and ranges from 0 to 63.

7.4.15 Base curve parameter K1 base_param_K1[i][w]

A 2-bit unsigned integer. It indicates the base curve parameter *K1*, and ranges from 0 to 3.

7.4.16 Base curve parameter K2 base_param_K2[i][w]

A 2-bit unsigned integer. It indicates the base curve parameter *K2*, and ranges from 0 to 3.

7.4.17 Base curve parameter K3 base_param_K3[i][w]

A 4-bit unsigned integer. It indicates the base curve parameter *K3*, and ranges from 0 to 15.

7.4.18 Base curve adjustment mode base_param_Delta_enable_mode[i][w]

A 3-bit unsigned integer. It indicates the adjustment coefficient mode of the base curve mapping parameter, and ranges from 0 to 7.

7.4.19 Base curve adjustment coefficient base_param_enable_Delta[i][w]

A 7-bit unsigned integer. It indicates the adjustment coefficient value of the base curve mapping parameter, and ranges from 0 to 127.

7.4.20 Cubic spline flag 3Spline_enable_flag[i][w]

A binary variable. It indicates the transmission cubic spline range. The value is 0 or 1.

7.4.21 Number of cubic spline interval groups 3Spline_enable_num[i][w]

A 1-bit unsigned integer. It indicates the number of cubic spline interval groups. The value is 0 or 1.

7.4.22 Cubic spline interval mode 3Spline_TH_enable_mode[j][i][w]

A 2-bit unsigned integer. It indicates the cubic spline interval mode, and ranges from 0 to 3.

7.4.23 Cubic spline interval slope and dark area offset parameter 3Spline_TH_enable_MB[j][i][w]

An 8-bit unsigned integer. It indicates the slope and dark area offset of the cubic spline interval parameter, and ranges from 0 to 255.

7.4.24 Cubic spline interval position parameter 3Spline_TH_enable[j][i][w]

A 12-bit unsigned integer. It indicates the cubic spline interval position parameter of tone mapping, and ranges from 0 to 4095.

7.4.25 Cubic spline interval 1 offset 3Spline_TH_enable_Delta1[j][i][w]

A 10-bit signed integer. It indicates the offset of the cubic spline interval 1, and ranges from 0 to 1023.

7.4.26 Cubic spline interval 2 offset 3Spline_TH_enable_Delta2[j][i][w]

A 10-bit signed integer. It indicates the offset of the cubic spline interval 2 of tone mapping, and ranges from 0 to 1023.

7.4.27 Cubic spline adjustment strength 3Spline_enable_Strength[j][i][w]

An 8-bit signed integer. It indicates the cubic spline adjustment strength, and ranges from 0 to 255.

7.4.28 Color saturation mapping flag color_saturation_mapping_enable_flag[w]

A binary variable. It indicates the flag of the color saturation mapping parameter. The value is 0 or 1.

7.4.29 Color saturation value color_saturation_enable_num[w]

A 3-bit unsigned integer. It indicates the color saturation value parameter, and ranges from 0 to 7.

7.4.30 Color saturation gain color_saturation_enable_gain[i][w]

An 8-bit unsigned integer. It indicates the color saturation gain parameter, and ranges from 0 to 255.

8 Encapsulation of Metadata in Coded Stream

8.1 Encapsulation of Metadata in AVS2 Coded Stream

The metadata is encapsulated in the extended data extension_data() in the AVS2 stream. The static metadata is encapsulated in mastering_display_and_content_metadata_extension() of the sequence header extension_data(), and a corresponding extension number is "1010". The dynamic metadata is encapsulated in hdr_dynamic_metadata_extension() of the picture header extension_data(), and a corresponding extension number is "0101".

The definition of the HDR static metadata extension in the AVS2 stream should comply with provisions in Table 12, and the definition of the HDR dynamic metadata extension should comply with provisions in Table 13.

Table 12 Definition of HDR static metadata extension in AVS2 stream

Definition of HDR static metadata extension in AVS2 stream	Descriptor
mastering_display_and_content_metadata_extension(){	
extension_id	f(4)
for (c=0; c<3; c++) {	
display_primaries_x[c]	u(16)
marker_bit	f(1)
display_primaries_y[c]	u(16)
marker_bit	f(1)
}	
white_point_x	u(16)
marker_bit	f(1)
white_point_y	u(16)
marker_bit	f(1)

Definition of HDR static metadata extension in AVS2 stream	**Descriptor**
max_display_mastering_luminance	u(16)

Table 12 (continued)

Definition of HDR static metadata extension in AVS2 stream	Descriptor
marker_bit	f(1)
min_display_mastering_luminance	u(16)
marker_bit	
max_content_light_level	u(16)
marker_bit	f(1)
max_picture_average_light_level	u(16)
marker_bit	f(1)
reserved_bits	r(16)
next_start_code( )	
}	

The video extension ID extension_id is a 4-bit binary number '1010'. It identifies the HDR static metadata extension.

Table 13 Definition of HDR dynamic metadata extension in AVS2 stream

Definition of HDR dynamic metadata extension in AVS2 stream	**Descriptor**
hdr_dynamic_metadata_extension( ) {	
extension_id	f(4)
hdr_dynamic_metadata_type	f(4)
itu_t_t35_country_code	0x26
itu_t_t35_terminal_provide_code	0x0004
itu_t_t35_terminal_provide_oriented_code	0x0005
if(system_start_code==0x01){	
num_windows=1	
for(w = 0; w < num_windows; w++) {	
minimum_maxrgb_pq[w]	u(12)
marker_bit	f(1)

Definition of HDR dynamic metadata extension in AVS2 stream	Descriptor
average_maxrgb_pq[w]	u(12)
marker_bit	f(1)
variance_maxrgb_pq[w]	u(12)
marker_bit	f(1)
maximum_maxrgb_pq[w]	u(12)
marker_bit	f(1)
}	
for(w = 0; w < num_windows; w++) {	
tone_mapping_enable_mode_flag[w]	u(1)
if(tone_mapping_enable_mode_flag [w]==1){	
tone_mapping_param_enable_num [w]	u(1)
tone_mapping_param_enable_num [w]++	
for(i=0; i< tone_mapping_param_enable_num [w]; i++){	
targeted_system_display_maximum_luminance_ pq[i][w]	u(12)
base _enable_flag[i][w]	u(1)
marker_bit	f(1)
if(base _enable_flag[i][w]){	
base_param_m_p[i][w]	u(14)
base_param_m_m[i][w]	u(6)
marker_bit	f(1)
base_param_m_a[i][w]	u(10)
base_param_m_b[i][w]	u(10)
marker_bit	f(1)
base_param_m_n[i][w]	u(6)

Table 13 (continued)

Definition of HDR dynamic metadata extension in AVS2 stream	Descriptor
base_param_K1[i][w]	u(2)
base_param_K2[i][w]	u(2)
base_param_K3[i][w]	u(4)

Definition of HDR dynamic metadata extension in AVS2 stream	Descriptor
base_param_Delta_enable_mode[i][w]	u(3)
marker_bit	f(1)
base_param_enable_Delta[i][w]	u(7)
}	
3Spline _enable_flag[i][w]	u(1)
if(3Spline_enable_flag[i][w]){	
3Spline_enable_num[i][w]	u(1)
3Spline_enable_num[i][w]++;	
for(j = 0; j < 3Spline_enable_num[i][w]; j ++) {	
3Spline_TH_enable_mode[j] [i][w]	u(2)
if((3Spline_TH_mode[j][i] [w]==0)|| (3Spline_TH_mode[j][i] [w]==2)){	
3Spline_TH_enable_MB [j][i][w]	f(8)
}	
marker_bit	f(1)
3Spline_TH_enable[j][i][w]	f(12)
marker_bit	f(1)
3Spline_TH_enable_Delta1 [j][i][w]	f(10)
3Spline_TH_enable_Delta2 [j][i][w]	f(10)
marker_bit	f(1)
3Spline_enable_Strength[j][i][w]	f(8)
}	
}	
}	
}	
}	
color_saturation_mapping_enable_flag[w]	u(1)
if(color_saturation_mapping_enable_flag[w]) {	
color_saturation_enable_num[w]	u(3)
for(i = 0; i< color_saturation_enable_num [w]; i++) {	
color_saturation_enable_gain[i][w]	u(8)

Definition of HDR dynamic metadata extension in AVS2 stream	Descriptor
marker_bit	f(1)
}	
}	
}	
}	
stuffing_bit	'1'
while(!byte_aligned())	
stuffing_bit	'0'
next_start_code()	
}	

The video extension ID extension_id is a bit string '0101'. It identifies the high dynamic range picture extension.

The high dynamic range picture metadata type hdr_dynamic_metadata_type is a 4-bit unsigned integer. It identifies the dynamic metadata type.

The ITU-T T.35 country code itu_t_t35_country_code is an 8-bit unsigned integer. It identifies the country identification code specified in ITU-T T.35.

The ITU-T T.35 terminal manufacturer code itu_t_t35_terminal_provide_code is a 16-bit unsigned integer. It identifies the terminal manufacturer code specified in ITU-T T.35.

The ITU-T T.35 terminal manufacturer oriented code itu_t_t35_terminal_provide_oriented_code is an 8-bit unsigned integer. It identifies the terminal manufacturer oriented code specified in ITU-T T.35.

8.2 Encapsulation of Metadata in ITU-T H.265 Coded Stream

The metadata is encapsulated in the ITU-T H.265 coded stream. Refer to Annex C.

9 Metadata Conversion During Display Adaptation

To implement display adaptation, the HDR metadata encapsulated in the AVS2 stream needs to be converted into the required variables. In addition, the variables required for display adaptation are calculated based on the maximum luminance (*MaxDisplay*) and minimum luminance (*MinDispla*) of the terminal display. The variables are defined and converted as follows.

—— *minimum_maxrgb*: a floating point number. *minimum_maxrgb* = *minimum_maxrgb_pq*[*w*] ÷ 4095. The unit is 0.00024, and the value ranges from 0.0000 to 1.00000.

—— *average_maxrgb*: a floating point number. *average_maxrgb* = *average_maxrgb_pq*[*w*] ÷ 4095. The unit is 0.00024, and the value ranges from 0.00000 to 1.00000.

—— *variance_maxrgb*: a floating point number. *variance_maxrgb* = *variance_maxrgb_pq*[*w*] ÷ 4095. The unit is 0.00024, and the value range is 0.00000 to 1.00000.

—— *maximum_maxrgb*: a floating point number. *maximum_maxrgb* = *maximum_maxrgb_pq*[*w*] ÷ 4095. The unit is 0.00024, and the value ranges from 0.00000 to 1.00000.

—— *tone_mapping_mode_flag*: a binary variable. *tone_mapping_mode_flag* == *tone_mapping_enable_mode_flag*[*w*]. The value is 0 or 1.

—— *tone_mapping_param_num*: a 1-bit unsigned integer. *tone_mapping_param_num* = *tone_mapping_param_enable_num*[*w*]. The value is 0 or 1.

—— *targeted_system_display_maximum_luminance*: a floating point number. *targeted_system_display_maximum_luminance* = *targeted_system_display_maximum_luminance_pq*[*i*][*w*] ÷ 4095. The unit is 0.00024, and the value ranges from 0.00024 to 1.0. When *targeted_system_display_maximum_luminance_pq*[*i*][*w*] is 2080, the *targeted_system_display_maximum_luminance* variable is used only in the SDR display adaptation of PQ HDR in chapter 11.

—— *base_flag*: a binary variable. *base_flag* = *base_enable_flag*[*i*][*w*]. The value is 0 or 1.

—— *m_p*_0: a floating point number. m_p_0 = 10.0 × *base_param_m_p*[*i*][*w*] ÷ 16383. The unit is 0.00061, and the value ranges from 0.00000 to 10.00000.

—— *m_m*_0: a floating point number. *m_m*_0 = *base_param_m_m*[*i*][*w*] ÷ 10.0. The unit is 0.1, and the value ranges from 0.0 to 6.3.

—— *m_a*_0: a floating point number. *m_a*_0 = *base_param_m_a*[*i*][*w*] ÷ 1023. The unit is 0.00098, and the value ranges from 0.00000 to 1.00000.

—— *m_b*_0: a floating point number. *m_b*_0 = *base_param_m_b*[*i*][*w*] × 0.25 ÷ 1023. The unit is 0.00024, and the value ranges from 0.00000 to 0.25000.

—— *m_n*_0: a floating point number. *m_n*_0 = *base_param_m_n*[*i*][*w*] ÷ 10. The unit is 0.1, and the value ranges from 0.0 to 6.3.

—— *k*1_0: an unsigned integer. *k*1_0 = Clip3 (0, 1, *base_param_K*1[*i*][*w*]). The value ranges from 0 to 1.

—— *k*2_0: an unsigned integer. It indicates *k*2_0 in the base curve mapping parameter. *k*2_0 = Clip3 (0, 1, *base_param_K*2[*i*][*w*]). The value ranges from 0 to 1.

—— *k*3_0: a floating point number. When *base_param_K*3[*i*][*w*] is 2, *k*3_0 = *maximum_maxrgb*. Otherwise, *k*3_0 = 1.0.

—— *base_param_Delta_mode*: an unsigned integer. *base_param_Delta_mode* = *base_param_Delta_enable_mode*[*i*][*w*].

—— *base_param_Delta*: a floating point number. When *base_param_Delta_mode* is equal to 2 or 6, *base_param_Delta* =– (*base_param_enable_Delta*[*i*][*w*]÷ 127). Otherwise, *base_param_Delta* = *base_param_enable_Delta*[*i*][*w*] ÷ 127.

——*3Spline_flag*: a binary variable. *3Spline_flag* = *3Spline_enable_flag*[*i*][*w*].

—— *3Spline_num*: an unsigned integer. When *3Spline_enable_flag*[*i*][*w*] is 1, *3Spline_num* = *3Spline_enable_num*[*i*][*w*] + 1. Otherwise, *3Splin _num* = 1.

—— *3Spline_TH_mode*: an unsigned integer. When *3Spline_enable_flag*[*i*][*w*] is 1, *3Spline_TH_mode* = *3Spline_TH_enable_mode*[*j*][*i*][*w*]. Otherwise, *3Spline_TH_mode* = 0.

—— *3Spline_TH_MB*0: a floating point number. *3Spline_TH_MB*0 =(*3Spline_TH_enable_MB*[*j*][*i*][*w*]&0xFC) ÷ 63.

—— *3Spline_TH_MB*1: a floating point number. *3Spline_TH_MB*1 = *3Spline_TH_enable_MB*[*j*][*i*][*w*] × 1.1 ÷ 255.

—— *base_offset*: a floating point number. *base_offset* = (*3Spline_TH_enable_MB*[*j*][*i*][*w*]&0x03) × 0.1 ÷ 3.

—— *3Spline_TH*0: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is 0, *3Spline_TH*0 = *3Spline_TH enable*[*j*][*i*][*w*] ÷ 4095. Otherwise, *3Spline_TH*0 = 0. The unit is 0.00024, and the value ranges from 0.00000 to 1.00000.

—— *3Splin _TH*1: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is not 0, *3Spline_TH*1 = *3Spline_TH enable*[*j*][*i*][*w*] ÷ 4095. Otherwise, *3Spline_TH*1 = 1.00000. The unit is 0.00024, and the value ranges from 0.00000 to 1.00000.

—— *3Spline_TH_Delta*10: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is 0, *3Spline_TH_Delta*10 = *3Spline_TH_enable_Delta*1[*j*][*i*][*w*] × 0.25 ÷ 1023. Otherwise, *3Spline_TH_Delta*10 = 0. The unit is 0.00024, and the value ranges from 0.00000 to 0.10000.

—— *3Spline_TH_Delta*11: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is not 0, *3Spline_TH_Delta*11 = *3Spline_TH_enable_Delta*1[*j*][*i*][*w*] × 0.25 ÷ 1023. Otherwise, *3Spline_TH_Delta*11 = 0. The unit is 0.00024, and the value ranges from 0.00000 to 0.10000.

—— *3Spline_TH_Delta*20: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is 0, *3Spline_TH_Delta*20 = *3Spline_TH_enable_Delta*2[*j*][*i*][*w*] × 0.25 ÷ 1023. Otherwise, *3Spline_TH_Delta*20 = 0. The unit is 0.00024, and the value ranges from 0.00000 to 1.00000.

—— *3Spline_TH_Delta21*: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is not 0, *3Spline_TH_Delta*21 = *3Spline_TH_enable_Delta*2[*j*][*i*][*w*] × 0.25 ÷ 1023. Otherwise, *3Spline_TH_Delta*21 = 0. The unit is 0.00024, and the value ranges from 0.00000 to 1.00000.

—— *3Spline_Strength*0: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is 0, *3Spline_Strength*0 = (*3Spline_enable_Strength*[*j*][*i*][*w*] – 127) ÷ 127. Otherwise, *3Spline_Strength*0 = 0. The unit is 0.0079, and the value ranges from –1.0000 to 1.0000.

—— *3Spline_Strength*1: a floating point number. When *3Spline_TH_enable_mode*[*j*][*i*][*w*] is not 0, *3Spline_Strength*1 = (*3Spline_enable_Strength*[*j*][*i*][*w*] – 127) ÷ 127. Otherwise, *3Spline_Strength*1 = 0. The unit is 0.0079, and the value ranges from –1.0000 to 1.0000.

—— *color_saturation_mapping_flag*: a binary variable. *color_saturation_mapping_flag* = *color_saturation_mapping_enable_flag*[*w*].

—— *color_saturation_num*: an unsigned integer. *color_saturation_num* = *color_saturation_enable_num*[*w*]. The unit is 1, and the value ranges from 0 to 7.

—— *color_saturation_gain*[0]: a floating point number. *color_saturation_gain*[0] = *color_saturation_enable_gain*[*i*][*w*] ÷ 128. The unit is 0.0078, and the value ranges from 0.0000 to 2.0000.

—— *color_saturation_gain*[1]: a floating point number. *color_saturation_gain*[1] = (*color_saturation_enable_gain*[*i*][*w*]&0xFC) ÷ 128. The unit is 0.0078, and the value ranges from 0.0000 to 2.0000.

—— *MaxDisplayPQ*: a 16-bit unsigned integer. It indicates the maximum luminance of the terminal display.

—— *MinDisplayPQ*: a 16-bit unsigned integer. It indicates the minimum luminance of the terminal display.

—— *MaxDisplayPQ*: a floating point number. *MaxDisplayPQ* = PQ_EOTF^{-1}(*MaxDisplay*). PQ_EOTF^{-1} should comply with requirements in GY/T 315-2018.

—— *MinDisplayPQ*: a floating point number. *MinDisplayPQ* = PQ_EOTF^{-1}(*MinDisplay*).

10 HDR Display Adaptation of PQ HDR Video

10.1 HDR Display Adaptation Process

This section describes the process of adapting PQ content for display on HDR terminals. The received metadata is converted into a metadata variable according to provisions in chapter 9. A base curve parameter and a cubic spline curve parameter are generated based on the metadata variable, and a corresponding tone mapping curve is generated. As shown in Figure 3, the tone mapping curve includes a linear spline curve, a first segment of a cubic spline curve, a base curve, and a second segment of the cubic spline curve. The HDR display adaptation process is completed through dynamic range conversion of color signals and color adjustment, as shown in Figure 4.

Figure 3 Schematic diagram of tone mapping curve

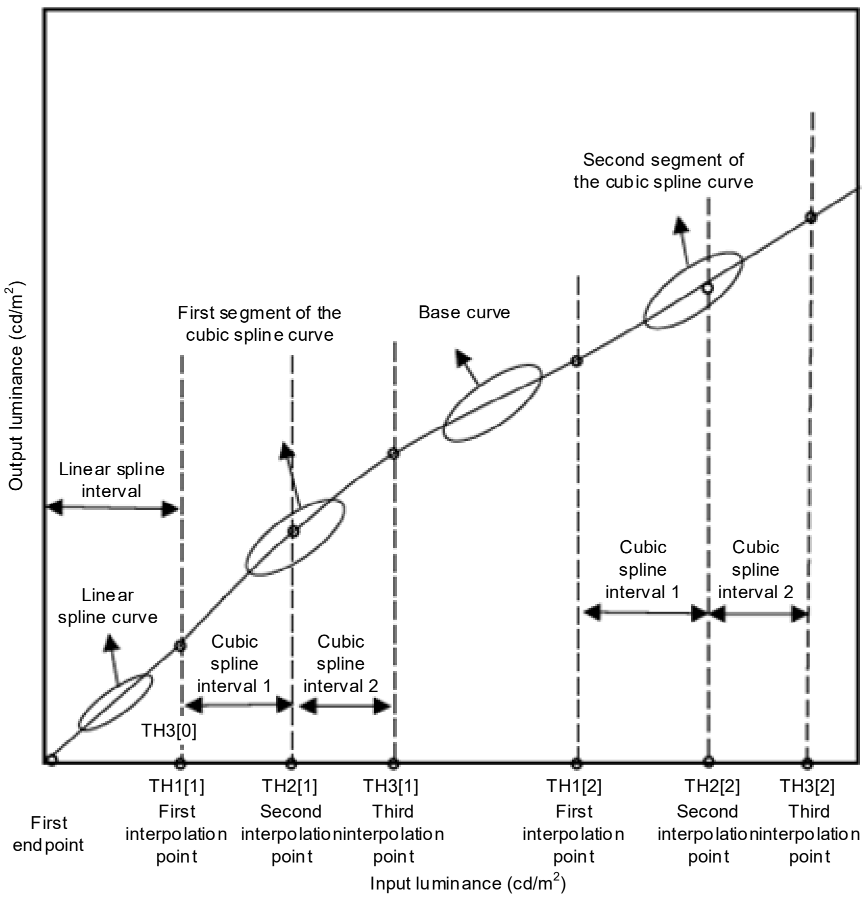
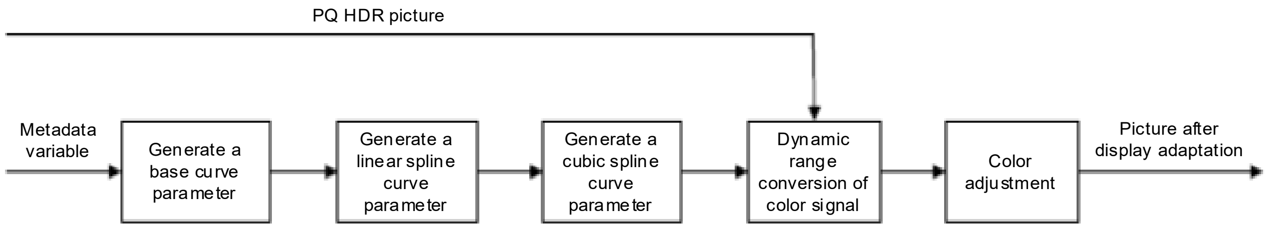


Figure 4 HDR display adaptation process of PQ content



Input: an RGB pixel buffer $f[N_{\text{frame}}][3]$ (N_{frame} indicates the total number of sampling points of the to-be-processed frame) and the metadata variable.

Output: an RGB pixel buffer $f_{\text{process}}[N_{\text{frame}}][3]$ after HDR display adaptation.

The HDR display adaptation process is as follows.

a) The base curve parameter is generated according to provisions in section 10.2.

b) The cubic spline curve parameter is generated according to provisions in section 10.3.

c) The RGB pixel buffer $f_{\text{TM}}[N_{\text{frame}}][3]$ after dynamic range conversion processing is generated according to provisions in section 10.4.

d) The $f_{\text{process}}[N_{\text{frame}}][3]$ is generated according to provisions in section 10.5.

10.2 Base Curve Parameter Obtaining Process

10.2.1 Overview

The base curve parameter obtaining process is as follows.

a) The minimum luminance correction value *min_lum* is calculated: *min_lum* = *minimum_maxrgb*.

b) The maximum luminance correction value *max_lum* is calculated according to section 10.2.2.

c) The base curve parameter is calculated.

1) If *tone_mapping_mode_flag* is 0, sections 10.2.3 and 10.2.6 are sequentially called to obtain the base curve parameter.

2) If *tone_mapping_mode_flag* is 1 and *base_flag* is 0, sections 10.2.3 and 10.2.6 are sequentially called to obtain the base curve parameter.

3) If *tone_mapping_mode_flag* is 1 and *base_flag* is 1:

If *targeted_system_display_maximum_luminance* is equal to *MaxDisplayPQ*, *m_p* = *m_p*_0, *m_a* = *m_a*_0, *m_m* = *m_m*_0, *m_n* = *m_n*_0, *m_b* = *m_b*_0, *K*1 = *k*1_0, *K*2 = *k*2_0, and *K*3 = *k*3_0;

If *base_param_Delta_mode* is 3, *m_p* = *m_p*_0, *m_a* = *m_a*_0, *m_m* = *m_m*_0, *m_n* = *m_n*_0, *m_b* = *m_b*_0, *K*1 = *k*1_0, *K*2 = *k*2_0, and *K*3 = *k*3_0;

If *base_param_Delta_mode* is 0, 2, 4, or 6, sections 10.2.4 and 10.2.6 are sequentially called to obtain the base curve parameter;

If *base_param_Delta_mode* is 1 or 5, sections 10.2.5 and 10.2.6 are sequentially called to obtain the base curve parameter.

10.2.2 Calculation process of maximum luminance correction value max_lum

Input: *MaxDisplayPQ*, *max_display_mastering_luminance*, *maximum_maxrgb*, *average_maxrgb*, and *variance_maxrgb*.

Output: the maximum luminance correction value *max_lum*.

The calculation process is as follows.

a) The display luminance value *MaxRefDisplay* of the reference mastering display is calculated.

$$MaxRefDisplay = \mathrm{PQ\_EOTF}^{-1}(max\_display\_mastering\_luminance).$$

b) For calculation of the reference maximum value *MAX*1, refer to formula (11).

$$MAX1 = B \times maximum\_maxrgb + A \times (2 \times average\_maxrgb) + (1 - A - B) \times (variance\_maxrgb) \quad \text{......(11)}$$

A and *B* are weight coefficients. *A* = (1 – *B*) × (1 – F(*average_maxrgb* ÷ *maximum_maxrgb*)), F(*x*) = 0.5, *A* = 0.4, *B* = 0.2.

c) The maximum luminance correction value *max_lum* is calculated according to formula (12).

$$max\_lum = \begin{cases} MaxRefDisplay & MAX1 > MaxRefDisplay \\ MAX1 & MIN \le MAX1 \le MaxRefDisplay \\ MIN & MAX1 < MIN \end{cases} \quad \text{......(12)}$$

MIN = 0.5081.

d) If *max_lum* < *MaxDisplayPQ*, *max_lum* = *MaxDisplayPQ*.

10.2.3 Base curve parameter obtaining process 0

Input: *MaxDisplayPQ*, *MinDisplayPQ*, *minimum_maxrgb*, *maximum_maxrgb*, *variance_maxrgb*, *average_maxrgb*, and *max_l* .

Output: dynamic range conversion parameter values *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, *K*1, *K*2, and *K*3 of the color signal.

The steps of the base curve parameter obtaining process 0 are as follows.

a) *m_m* = 2.4, *m_n* = 1, *K*1 = 1, *K*2 = 1, *K*3 = 1, and *m_b* = *MinDisplayPQ*.

b) The intermediate variable *m_p* 0 of *m_p* is calculated according to formula (13).

$$m\_p0 = \begin{cases} p_{\mathrm{valueH0}} & avgL > TPH0 \\ p_{\mathrm{valueH0}} \times \mathrm{g0}(w0) + p_{\mathrm{valueL0}} \times (1 - \mathrm{g0}(w0)) & TPL0 \le avgL \le TPH0 \\ p_{\mathrm{valueL0}} & avgL < TPL0 \end{cases} \quad \text{......(13)}$$

avgL is obtained according to formula (14), and *w*0 is obtained according to formula (15).

$$avgL = average\_maxrgb. \quad \text{......(14)}$$

$$w0 = \left(\frac{avgL - TPL0}{TPH0 - TPL0}\right) \quad \text{......(15)}$$

In formula (13) to formula (15), $p_{valueH0}$ = 3.5, $p_{valueL0}$ = 4.0, *TPH*0 = 0.6, *TPL*0 = 0.3, and g0(*x*) meets *y* = *x*.

c) The base curve parameter *m_p* is calculated according to formula (16).

$$m\_p = \begin{cases} m\_p0 + p_{\text{deltaH1}} & max\_lum > TPH1 \\ m\_p0 + p_{\text{deltaH1}} \times \text{g1}(w1) + p_{\text{deltaL1}} \times (1 - \text{g1}(w1)) & TPL1 \le max\_lum \le TPH1 \\ m\_p0 + p_{\text{deltaL1}} & max\_lum < TPL1 \end{cases} \quad \text{............(16)}$$

*w*1 is obtained according to formula (17).

$$w1 = \left(\frac{max\_lum - TPL1}{TPH1 - TPL1}\right) \quad \text{............(17)}$$

In formula (16) and formula (17), $p_{deltaH1}$ = 0.6, $p_{deltaL1}$ = 0.0, *TPH*1 = 0.9, *TPL*1 = 0.75, and g1(*x*) meets *y* = *x*.

d) The base curve parameter *m_a* is calculated according to formula (18).

$$m\_a = (MaxDisplayPQ - MinDisplayPQ) \div \left(\frac{m\_p \times max\_lum^{m\_n}}{(K1 \times m\_p - K2) \times max\_lum^{m\_n} + K3}\right)^{m\_m} \quad \text{....(18)}$$

10.2.4 Base curve parameter obtaining process 1

Input: *MaxDisplayPQ*, *MinDisplayPQ*, *m_p*_0, *m_m*_0, *m_n*_0, *m_a*_0, *m_b*_0, *k*1_0, *k*2_0, *k*3_0, *targeted_system_display_maximum_luminance*, and *base_param_Delta*.

Output: *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, *K*1, *K*2, and *K*3.

The obtaining process is as follows.

a) *m_m* = *m_m*_0, *m_n* = *m_n*_0, *K*1 = *k*1_0, *K*2 = *k*2_0, and *K*3 = *k*3_0;

b) *m_b* = *m_b*_0 × ((*MaxDisplayPQ* – *MinDisplayPQ*) ÷ *targeted_system_display_maximum_luminance*);

c) *m_a* = *m_a*_0 × ((*MaxDisplayPQ* – *MinDisplayPQ*) ÷ *targeted_system_display_maximum_luminance*);

d) *m_p*0 = *m_p*_0 + *base_param_Delta* × (Abs((PQ_EOTF(*MaxDisplayPQ*) – PQ_EOTF (*targeted_system_display_maximum_luminance*))) ÷ 100)^N, where *N* = 0.5, PQ_EOTF should comply with provisions in GY/T 315-2018;

e) *m_p* = Clip3(3.0, 7.5, *m_p*0).

10.2.5 Base curve parameter obtaining process 2

Input: *MaxDisplayPQ*, *MinDisplayPQ*, *m_p*_0, *m_m*_0, *m_n*_0, *m_a*_0, *m_b*_0, *k*1_0, *k*2_0, *k*3_0, *targeted_system_display_maximum_luminance*, *base_param_Delta*, *minimum_maxrgb_pq*, *maximum_maxrgb_pq*, *variance_maxrgb_pq*, *average_maxrgb_pq*, and *max_lum*.

Output: base curve parameters *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, *K*1, *K*2, and *K*3.

The obtaining process is as follows.

a) *m_p*_1, *m_m*_1, *m_n*_1, *m_a*_1, *m_b*_1, *K*1_1, *K*2_1, and *K*3_1 are obtained by calling 10.2.3 based on *MaxDisplayPQ*, *MinDisplayPQ*, *minimum_maxrgb_pq*, *maximum_maxrgb_pq*, *variance_maxrgb_pq* and *average_maxrgb_pq*.

b) $w0 = base\_param\_Delta \times (\text{Abs}((\text{PQ\_EOTF}(MaxDisplayPQ) - \text{PQ\_EOTF}(targeted\_system\_display\_maximum\_luminance))) \div 100)^{N}$, where $N = 0.5$.

c) $w = \text{Clip3}(0.0, 1.0, w0)$.

d) $m\_p = (1 - w) \times m\_p\_0 + w \times m\_p\_1$,

$m\_m = (1 - w) \times m\_m\_0 + w \times m\_m\_1$,

$m\_n = (1 - w) \times m\_n\_0 + w \times m\_n\_1$,

$K1 = (1 - w) \times k1\_0 + w \times K1\_1$,

$K2 = (1 - w) \times k2\_0 + w \times K2\_1$,

$K3 = (1 - w) \times k3\_0 + w \times K3\_1$.

e) $m\_b = MinDisplayPQ$.

f) The base curve parameter m_a is calculated according to formula (19).

$$m\_a = \frac{(MaxDisplayPQ - MinDisplayPQ)}{\left(\frac{m\_p \times MaxSource^{m\_n}}{(K1 \times m\_p - K2) \times MaxSource^{m\_n} + K3}\right)^{m\_m}} \quad (19)$$

$MaxSource = max\_lum$.

10.2.6 Base curve parameter obtaining process 3

Input: m_p, m_m, m_n, m_a, m_b, $K1$, $K2$, $K3$, $3Spline\_TH0$, $3Splin\ \_TH\_Delta10$, and $3Spline\_TH\_Delta20$.

Output: m_b.

The obtaining process is as follows.

a) m_b0 is calculated:

If $base\_param\_Delta\_modebase\_param\_Delta$ is greater than or equal to 3, or $base\_flag$ is equal to 0, the intermediate variable m_b0 of m_b is calculated according to formula (20).

$$m\_b0 = m\_b \quad (20)$$

Otherwise, m_b0 is calculated according to formula (21).

$$m\_b0 = \begin{cases} m\_b & m\_a \le m\_a\_T \\ (1 - WA) \times m\_b & \text{other} \end{cases} \quad (21)$$

WA is obtained according to formula (22), and m_a_T is obtained according to formula (23).

$$WA = \left(\frac{MaxDisplayPQ}{max\_lum} - \frac{\text{H}(ma\quad\_lum)}{max\_lum}\right) \div \left(1 - \frac{\text{H}(max\_lum)}{max\_lum}\right) \quad (22)$$

$$m\_a\_T = \begin{cases} 0.990 & m\_p < 2.5 \\ 0.990 - (m\_p - 2.5) \times 0.111 & 2.5 \le m\_p < 3.5 \\ 0.879 - (m\_p - 3.5) \times 0.102 & 3.5 \le m\_p < 4.5 \\ 0.777 - (m\_p - 4.5) \times 0.079 & 4.5 \le m\_p < 7.5 \\ 0.540 & m\_p \ge 7.5 \end{cases} \quad (23)$$

$\text{H}(max\_lum)$ is obtained according to formula (24).

$$\mathrm{H}(max\_lum) = m\_a\_T \times \left(\frac{m\_p \times max\_lum^{m\_n}}{(K1 \times m\_p - K2) \times max\_lum^{m\_n} + K3}\right)^{m\_m} \quad (24)$$

b) The third interpolation point $TH3[1]$ is calculated according to formula (25), and the output value $VA3$ of the third interpolation point $TH3[1]$ on the base curve is calculated according to formula (26).

$$TH3[1] = 3Spline\_TH0 + 3Spline\_TH\_Delta10 + 3Spline\_TH\_Delta20 \quad (25)$$

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m} + m\_b0 \quad (26)$$

c) *m_b* is calculated:

If $VA3 > TH3[1]$, $VA3 > 0$, and *base_param_Delta_modebase_param_Delta* is a value other than 2, 3, and 6, *m_b* is calculated according to formula (27).

$$m\_b = m\_b0 - (VA3 - TH3[1]) \quad (27)$$

Otherwise, *m_b* is calculated according to formula (28).

$$m\_b = m\_b0 \quad (28)$$

10.3 Cubic Spline Curve Parameter Obtaining Process

10.3.1 Overview

The cubic spline curve parameter obtaining process is as follows. Refer to Figure 3.

a) 3*Spline_num* and 3*Spline_TH_mode* are calculated:

If *tone_map ng_mode_flag* is 0, 3*Spline_num* = 1, and 3*Spline_TH_mode* = 0. Otherwise, 3*Spline_num* and 3*Spline_TH_mode* are obtained according to chapter 9.

b) The cubic spline curve parameter is calculated.

If *tone_mapping_mode_flag* is 0, sections 10.3.2.2 and 10.3.3.2 are sequentially called to obtain the cubic spline curve parameter values.

If *tone_mapp g_mode_flag* is 1,

- When 3*Spline_flag* is 0, sections 10.3.2.2, 10.3.2.4, and 10.3.3.2 are sequentially called to obtain the cubic spline mapping curve parameter values.
- When 3*Spline_flag* is 1 and 3*Spline_TH_mode* is 0, sections 10.3.2.3 and 10.3.2.4 are sequentially called to obtain the linear spline curve parameters, and section 10.3.3.3 is called to obtain the parameter value of the first segment of the cubic spline curve.
- When 3*Spline_flag* is 1, if 3*Spl e_TH_mode* is not 0, sections 10.3.2.2 and 10.3.2.4 are sequentially called to obtain the linear spline curve parameters, section 10.3.3.2 is called to obtain the parameter value of the first segment of the cubic spline curve, and then section 10.3.3.4 is called to obtain the parameter value of the second segment of the cubic spline curve.

c) If 3*Spline_num* is equal to 2, section 10.3.3.4 is called to obtain the parameter value of the second segment of the cubic spline curve.

10.3.2 Linear spline curve parameter obtaining process

10.3.2.1 Linear spline curve

The linear spline curve is a curve between the first endpoint and the first interpolation point *TH*3[0]. Refer to formula (29).

$$\mathrm{F}(L) = MB[0][0] \times L + base\_offset \quad (29)$$

10.3.2.2 Linear spline curve parameter obtaining process 0

Input: *average_maxrgb*.

Output: *TH*3[0], *MB*[0][0], and *base_offset*.

The steps of the linear spline curve parameter obtaining process 0 are as follows.

a) The first interpolation point *TH*3[0] is calculated according to formula (30).

$$TH3[0] = \begin{cases} T_{\mathrm{dmaxL2}} & avgL > HLMAXH2 \\ (T_{\mathrm{dmaxL2}} \times \mathrm{g2}(\mathrm{w2}) + \mathrm{T}_{\mathrm{dmaxH2}} \times (1 - \mathrm{g2}(\mathrm{w2}))) & HLMAXL2 \le avgL \le HLMAXH2 \\ T_{\mathrm{dmaxH2}} & avgL < HLMAXL2 \end{cases} \quad (30)$$

T_{dmaxH2} = 0.25, T_{dmaxL2} = 0.1, g2(x) = x^N, N = 1, *avgL* is obtained according to formula (31), and *w*2 is obtained according to formula (32).

$$avgL = average\_maxrgb \quad (31)$$

$$w2 = \left(\frac{avgL - HLMAXL2}{HLMAXH2 - HLMAXL2}\right) \quad (32)$$

In formula (31) and formula (32), *HLMAXH*2 = 0.6, and *HLMAXL*2 = 0.3.

b) *base_offset* = 0.

c) The slope *MB*[0][0] is obtained according to formula (33).

$$MB[0][0] = \begin{cases} S_{\mathrm{dmaxL3}} & avgL > AVMAXH3 \\ S_{\mathrm{dmaxL3}} \times \mathrm{g3}(W3) + S_{\mathrm{dmaxH3}} \times (1 - \mathrm{g3}(W3)) & AVMAXL3 \le avgL \le AVMAXH3 \\ S_{\mathrm{dmaxH3}} & avgL < AVMAXL3 \end{cases} \quad (33)$$

S_{dmaxH3} = 1.0, S_{dmaxL3} = 0.96, $g3(x) = x^N$, N = 1, *avgL* is obtained according to formula (34), and *w*3 is obtained according to formula (35).

$$avgL = average\_maxrgb. \quad (34)$$

$$w3 = \left(\frac{avgL - AVMAXL3}{AVMAXH3 - AVMAXL3}\right) \quad (35)$$

In formula (34) and formula (35), *AVMAXH*3 = 0.6, and *AVMAXL*3 = 0.3.

10.3.2.3 Linear spline curve parameter obtaining process 1

Input: 3*Spl e_TH*0, 3*Spline_TH_MB*0, and *base_offset*.

Output: *TH*3[0], *MB*[0][0], and *base_offset*.

The first interpolation point *TH*3[0] is calculated according to formula (36), the slope *MB*[0][0] is calculated according to formula (37), and the offset *base_offset* is calculated according to formula (38).

$$TH3[0] = 3Spline\_TH0 \quad (36)$$

$$MB[0][0] = 3Spline\_TH\_MB0 \quad (37)$$

$$base\_offset = base\_offset \quad (38)$$

10.3.2.4 Linear spline curve parameter obtaining process 0

Input: *MaxDisplayPQ*, *max_lum*, *MB*[0][0], *TH*3[0], *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, *K*1, *K*2, and *K*3.

Output: *MB*[0][0] and *TH*3[0].

The obtaining process is as follows.

a) If *base_param_Delta_mode* is greater than or equal to 3 or *base_flag* is 0, steps b) to e) are skipped.

b) MB_*mid*[0][0] = *MB*[0][0], and *TH*3_*mid*[0] = *TH*3[0].

c) *m_a_T* is calculated:

If *m_p* < 2.5, *m_a_T* = 0.990;
If 2.5 ≤ *m_p* < 3.5, *m_a_T* = 0.990 – (*m_p* – 2.5) × 0.111;
If 3.5 ≤ *m_p* < 4.5, *m_a_T* = 0.879 – (*m_p* – 3.5) × 0.102;
If 4.5 ≤ *m_p* < 7.5, *m_a_T* = 0.777 – (*m_p* – 4.5) × 0.079;
If *m_p* ≥ 7.5, *m_a_T* = 0.540.

d) If *m_a* is less than or equal to *m_a_T*, step e) is skipped.

e) The slope *MB*[0][0] is calculated according to formula (39), and the first interpolation point *TH*3[0] is calculated according to formula (40).

$$MB[0][0] = \mathrm{Min}(\mathrm{Max}(MB\_mid[0][0] + (1 - MB\_mid[0][0]) \times (WA)^{N1}, MB\_mid[0][0]), 1) \quad (39)$$

$$TH3[0] = \mathrm{Min}(\mathrm{Max}(TH3\_mid[0] + (max\_lum - TH3\_mid[0]) \times (WA)^{N2}, TH3\_mid[0]), 1) \quad (40)$$

In formula (39) and formula (40), *N*1 = 1.0, *N*2 = 1.0, and *WA* is calculated according to formula (41).

$$WA = \left(\frac{MaxDisplayPQ}{max\_lum} - \frac{\mathrm{H}(max\_lum)}{max\_lum}\right) \div \left(1 - \frac{\mathrm{H}(max\_lum)}{max\_lum}\right) \quad (41)$$

H(*max_lum*) is calculated according to formula (42).

$$\mathrm{H}(max\_lum) = m\_a\_T \times \left(\frac{m\_p \times max\_lum^{m\_n}}{(K1 \times m\_p - K2) \times max\_lum^{m\_n} + K3}\right)^{m\_m} \quad (42)$$

10.3.3 Cubic spline curve parameter obtaining process

10.3.3.1 Linear spline curve

A curve between the first interpolation point $TH1[n]$ and the second interpolation point $TH2[n]$ is a curve of a cubic spline interval 1. Refer to formula (43).

$$F(L) = MD[0][n] \times (L - TH1[n])^3 + MC[0][n] \times (L - TH1[n])^2 + MB[0][n] \times (L - TH1[n])^1 + MA[0][n] \quad (43)$$

L is an independent variable in an interval $[TH1[n], TH2[n]]$.

A curve between the second interpolation point $TH2[n]$ and the third interpolation point $TH3[n]$ is a curve of a cubic spline interval 2. Refer to formula (44).

$$F(L) = MD[1][n] \times (L - TH2[n])^3 + MC[1][n] \times (L - TH2[n])^2 + MB[1][n] \times (L - TH2[n])^1 + MA[1][n] \quad (44)$$

L is an independent variable in an interval $[TH2[n], TH3[n]]$, and $0 < n \leq 3Spline\_num$.

10.3.3.2 Cubic spline curve parameter obtaining process 0

Input: $TH3[0]$, $MB[0][0]$, $base\_offset$, m_p, m_m, m_n, m_a, m_b, $K1$, $K2$, and $K3$.

Output: $TH1[1]$, $TH2[1]$, $TH3[1]$, $MA[0][1]$, $MB[0][1]$, $MC[0][1]$, $MD[0][1]$, $MA[1][1]$, $MB[1][1]$, $MC[1][1]$, and $MD[1][1]$.

The obtaining process is as follows.

a) The first interpolation point $TH1[1]$ is calculated according to formula (45), the second interpolation point $TH2[1]$ is calculated according to formula (46), and the third interpolation point $TH3[1]$ is calculated according to formula (47).

$$TH1[1] = TH3[0] \quad (45)$$

$$TH2[1] = TH1[1] + B \quad (46)$$

$$TH3[1] = TH2[1] + C \times TH2[1] - D \times TH1[1] \quad (47)$$

In formula (46) and formula (47), $B = 0.15$, $C = 0.5$, and $D = 0.5$.

b) The process of calculating $MA[0][1]$, $MB[0][1]$, $MC[0][1]$, $MD[0][1]$, $MA[1][1]$, $MB[1][1]$, $MC[1][1]$, and $MD[1][1]$ is as follows.

1) The output value $VA1$ of the first interpolation point $TH1[1]$ on the linear spline curve is calculated according to formula (48), the output value $VA3$ of the third interpolation point $TH3[1]$ on the base curve is calculated according to formula (49), and the output value $VA2$ of the second interpolation point $TH2[1]$ on the curve is calculated according to formula (50).

$$VA1 = MB[0][0] \times TH1[1] + base\_offset \quad (48)$$

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad (49)$$

$$VA2 = VA1 + \frac{(TH2[1] - TH1[1]) \times (VA3 - VA1)}{TH3[1] - TH1[1]} \quad (50)$$

2) The curve parameter $MA[0][1]$ of the cubic spline interval 1 is calculated according to formula (51), and the curve parameter $MA[1][1]$ of the cubic spline interval 2 is calculated according to formula (52).

$$MA[0][1] = VA1 \tag{51}$$

$$MA[1][1] = VA2 \tag{52}$$

3) The slope $GD1$ of the first interpolation point $TH1[1]$ on the curve is calculated according to formula (53), the curve parameter $MB[0][1]$ of the cubic spline interval 1 is calculated according to formula (54), and the slope $GD3$ of the third interpolation point $TH3[1]$ on the curve is calculated according to formula (55).

$$GD1 = MB[0][0] \tag{53}$$

$$MB[0][1] = MB[0][0] \tag{54}$$

$$GD3 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH3[1]^{m\_n-1} \times DGD3(L) \tag{55}$$

$DGD3(L)$ is obtained according to formula (56).

$$\mathrm{DGD3}(L) = \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m+1} \times \left(\frac{1}{TH3[1]^{m\_n} \times m\_p}\right)^{2} \tag{56}$$

4) The curve parameter $MC[0][1]$ of the cubic spline interval 1 is calculated according to formula (57), the curve parameter $MD[0][1]$ of the cubic spline interval 1 is calculated according to formula (58), the curve parameter $MB[1][1]$ of the cubic spline interval 2 is calculated according to formula (59), the curve parameter $MC[1][1]$ of the cubic spline interval 2 is calculated according to formula (60), and the curve parameter $MD[1][1]$ of the cubic spline interval 2 is calculated according to formula (61).

$$MC[0][1] = \frac{3.0 \times VA2 - 2.0 \times GD1 \times h1 - 3.0 \times VA1 - MB[1][1] \times h1}{h1 \times h1} \tag{57}$$

$$MD[0][1] = \frac{h1 \times GD1 + h1 \times MB[1][1] + 2.0 \times VA1 - 2.0 \times VA2}{h1 \times h1 \times h1} \tag{58}$$

$$MB[1][1] = \frac{-3.0 \times VA1 \times h2 \times h2 - 3.0 \times VA2 \times h1 \times h1 + 3.0 \times VA3 \times h1 \times h1 + 3.0 \times h2 \times h2 \times VA2 - h1 \times h1 \times h2 \times GD3 - GD1 \times h1 \times h2 \times h2}{2.0 \times h2 \times (h1 \times h1 + h2 \times h1)} \tag{59}$$

$$MC[1][1] = MC[0][1] + 3.0 \times MD[0][1] \times h1 \tag{60}$$

$$MD[1][1] = -\frac{VA3 - VA2 - h2 \times GD3 + MC[0][1] \times h2 \times h2 + 3 \times MD[0][1] \times h1 \times h2 \times h2}{2 \times h2 \times h2 \times h2} \tag{61}$$

In formula (57) to formula (61), $h1$ is represented by a cubic spline interval 1. Refer to formula (62). $h2$ is represented by a cubic spline interval 2. Refer to formula (63).

$$h1 = TH2[1] - TH1[1] \tag{62}$$

$$h2 = TH3[1] - TH2[1] \tag{63}$$

10.3.3.3 Cubic spline curve parameter obtaining process 1

Input: $TH3[0]$, $MB[0][0]$, *base_offset*, 3*Spline_TH_Delta*10, 3*Spline_TH_Delta*20, 3*Spline_Strength*0, *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, $K1$, $K2$, and $K3$.

Output: *TH*1[1], *TH*2[1], *TH*3[1], *MA*[0][1], *MB*[0][1], *MC*[0][1], *MD*[0][1], *MA*[1][1], *MB*[1][1], *MC*[1][1], and *MD*[1][1].

The obtaining process is as follows.

a) The first interpolation point *TH*1[1] is calculated according to formula (64), the second interpolation point *TH*2[1] is calculated according to formula (65), and the third interpolation point *TH*3[1] is calculated according to formula (66).

$$TH1[1] = TH3[0] \tag{64}$$

$$TH2[1] = TH1[1] + 3Spline\_TH\_Delta10 \tag{65}$$

$$TH3[1] = TH1[1] + 3Spline\_TH\_Delta10 + 3Spline\_TH\_Delta20 \tag{66}$$

b) The process of calculating *MA*[0][1], *MB*[0][1], *MC*[0][1], *MD*[0][1], *MA*[1][1], *MB*[1][1], *MC*[1][1], and *MD*[1][1] is as follows.

1) The output value *VA*1 of the first interpolation point *TH*1[1] on the linear spline curve is calculated according to formula (67), and the output value *VA*3 of the third interpolation point *TH*3[1] on the base curve is calculated according to formula (68).

$$VA1 = MB[0][0] \times TH1[1] + base\_offset \tag{67}$$

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \tag{68}$$

2) If *VA*3 > *TH*3[1], and *base_param_Delta_mode* is a value other than 2, 3, and 6, *VA*3 is updated. Refer to formula (69).

$$VA3 = TH3[1] \tag{69}$$

3) The output value *VA*2 of the second interpolation point *TH*2[1] on the curve is calculated according to formula (70).

$$VA2 = VA1 + \frac{(TH2[1] - TH1[1]) \times (VA3 - VA1)}{TH3[1] - TH1[1]} + \frac{(VA3 - VA1) \times Spline\_Strength0}{2} \tag{70}$$

If *VA*2 > *TH*2[1], and *base_param_delta_mode* is a value other than 2, 3, and 6, *VA*2 is updated. Refer to formula (71).

$$VA2 = TH2[1] \tag{71}$$

4) The curve parameter *MA*[0][1] of the cubic spline interval 1 is calculated according to formula (72), and the curve parameter *MA*[1][1] of the cubic spline interval 2 is calculated according to formula (73).

$$MA[0][1] = VA1 \tag{72}$$

$$MA[1][1] = VA2 \tag{73}$$

5) The curve parameter *MB*[0][1] of the cubic spline interval 1 is calculated according to formula (74), the slope *GD*1 of the first interpolation point *TH*1[1] on the curve is calculated according to formula (75), and the slope *GD*3 of the third interpolation point *TH*3[1] on the curve is calculated according to formula (76).

$$MB[0][1] = MB[0][0] \qquad (74)$$

$$GD1 = MB[0][0] \qquad (75)$$

$$GD3 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH3[1]^{m\_n-1} \times DGD3(L) \qquad (76)$$

$DGD3(L)$ is obtained according to formula (77).

$$DGD3(L) = \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m+1} \times \left(\frac{1}{TH3[1]^{m\_n} \times m\_p}\right)^2 \qquad (77)$$

6) The curve parameter *MC*[0][1] of the cubic spline interval 1 is calculated according to formula (78), the curve parameter *MD*[0][1] of the cubic spline interval 1 is calculated according to formula (79), the curve parameter *MB*[1][1] of the cubic spline interval 2 is calculated according to formula (80), the curve parameter *MC*[1][1] of the cubic spline interval 2 is calculated according to formula (81), and the curve parameter *MD*[1][1] of the cubic spline interval 2 is calculated according to formula (82).

$$MC[0][1] = \frac{3.0 \times VA2 - 2.0 \times GD1 \times h1 - 3.0 \times VA1 - MB[1][1] \times h1}{h1 \times h1} \qquad (78)$$

$$MD[0][1] = \frac{h1 \times GD1 + h1 \times MB[1][1] + 2.0 \times VA1 - 2.0 \times VA2}{h1 \times h1 \times h1} \qquad (79)$$

$$MB[1][1] = \frac{-3.0 \times VA1 \times h2 \times h2 - 3.0 \times VA2 \times h1 \times h1 + 3.0 \times VA3 \times h1 \times h1 + 3.0 \times h2 \times h2 \times VA2 - h1 \times h1 \times h2 \times GD3 - GD1 \times h1 \times h2 \times h2}{2.0 \times h2 \times (h1 \times h1 + h2 \times h1)} \qquad (80)$$

$$MC[1][1] = MC[0][1] + 3.0 \times MD[0][1] \times h1 \qquad (81)$$

$$MD[1][1] = -\frac{VA3 - VA2 - h2 \times GD3 + MC[0][1] \times h2 \times h2 + 3 \times MD[0][1] \times h1 \times h2 \times h2}{2 \times h2 \times h2 \times h2} \qquad (82)$$

In formula (78) to formula (82), *h*1 is represented by a cubic spline interval 1. Refer to formula (83). *h*2 is represented by a cubic spline interval 2. Refer to formula (84).

$$h1 = TH2[1] - TH1[1] \qquad (83)$$

$$h2 = TH3[1] - TH2[1] \qquad (84)$$

10.3.3.4 Cubic spline curve parameter obtaining process 2

Input: 3*Spline_TH*1, 3*Spline_TH_MB*1, 3*Spline_TH_Delta*11, 3*Spline_TH_Delta*21, 3*Spline_Strength*1, *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, *K*1, *K*2, and *K*3.

Output: *TH*1[2], *TH*2[2], *TH*3[2], *MA*[0][2], *MB*[0][2], *MC*[0][2], *MD*[0][2], *MA*[1][2], *MB*[1][2], *MC*[1][2], *MD*[1][2], and 3*Spline_num*.

The obtaining process is as follows.

a) The first interpolation point *TH*1[2] is calculated according to formula (85), the second interpolation point *TH*2[2] is calculated according to formula (86), and the third interpolation point *TH*3[2] is calculated according to formula (87).

$$TH1[2] = Spline\_TH1 \qquad (85)$$

$$TH2[2] = 3Spline\_TH1 + 3Spline\_TH\_Delta11 \qquad (86)$$

$$TH3[2] = 3Spline\_TH1 + 3Spline\_TH\_Delta11 + 3Spline\_TH\_Delta21 \qquad (87)$$

If $TH3[2] < TH3[1]$, $3Spline\_num = 1$, and steps b) to j) are skipped;

If $TH1[2] < TH3[1]$, $TH1[2]$ is calculated according to formula (88), and $TH2[2]$ is calculated according to formula (89).

$$TH1[2] = TH3[1] \quad (88)$$

$$TH2[2] = (TH1[2] + TH3[2]) \div 2 \quad (89)$$

b) The output value *VA*1 of the first interpolation point *TH*1[2] on the base curve is calculated according to formula (90), and the output value *VA*3 of the third interpolation point *TH*3[2] on the base curve is calculated according to formula (91).

$$VA1 = m\_a \times \left(\frac{m\_p \times TH1[2]^{m\_n}}{(K1 \times m\_p - K2) \times TH1[2]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad (90)$$

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[2]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[2]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad (91)$$

c) *VA*3, *TH*3[2], and *TH*2[2] are updated:

If 3*Splin _TH_mode* is 1 or 2 and *base_param_Delta_mode* is not equal to 3, *VA*3 is calculated according to formula (92).

$$VA3 = MaxDisplayPQ \quad (92)$$

If updated $VA3 > TH3[2]$, and *base_param_Delta_mode* is neither 2 nor 6, *TH*3[2] is calculated according to formula (93), and *TH*2[2] is calculated according to formula (94).

$$TH3[2] = VA3 \quad (93)$$

$$TH2[2] = TH1[2] + (TH3[2] - TH1[2]) \div 2.0 \quad (94)$$

If 3*Spline_TH_mode* is 1 or 2 and *base_param_Delta_mode* is 3, *VA*3 is calculated according to formula (95).

$$VA3 = targeted\_system\_display\_maximum\_luminance \quad (95)$$

d) The output value *VA*2 of the second interpolation point *TH*2[2] on the curve is calculated according to formula (96).

$$VA2 = VA1 + \frac{(TH2[2] - TH1[2]) \times (VA3 - VA1)}{TH3[2] - TH1[2]} + \frac{(VA3 - VA1) \times Spline\_Strength1}{2} \quad (96)$$

e) If 3*Spline_TH_mode* is 1 or 2, $VA2 > TH2[2]$, and *base_param_Delta_mode* is a value other than 2, 3, and 6, refer to formula (97) for the updated *VA*2.

$$VA2 = TH2[2] \quad (97)$$

f) The curve parameter *MA*[0][2] of the cubic spline interval 1 is calculated according to formula (98), and the curve parameter *MA*[1][2] of the cubic spline interval 2 is calculated according to formula (99).

$$MA[0][2] = VA1 \quad (98)$$

$$MA[1][2] = VA2 \quad (99)$$

g) The slope *GD*1 of the first interpolation point *TH*1[2] on the curve is calculated according to formula (100), and the curve parameter *MB*[0][2] of the cubic spline interval 1 is calculated according to formula (101).

$$GD1 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH1[2]^{m\_n-1} \times DGD(L) \quad (100)$$

$$MB[0][2] = GD1 \quad (101)$$

DGD(*L*) is obtained according to formula (102).

$$\mathrm{DGD}(L) = \left(\frac{m\_p \times TH1[2]^{m\_n}}{(K1 \times m\_p - K2) \times TH1[2]^{m\_n} + K3}\right)^{m\_m+1} \times \left(\frac{1}{TH1[2]^{m\_n} \times m\_p}\right)^2 \quad (102)$$

h) *GD*3 is calculated:

If 3*S ine_TH_mode* is 1, the slope *GD*3 of the first interpolation point *TH*3[2] on the curve is calculated according to formula (103).

$$GD3 = \begin{cases} (down\_T \times (-TH\_str) + mid\_T \times (1 + TH\_str)), & TH\_str < 0 \\ (up\_T \times TH\_str + mid\_T \times (1 - TH\_str)), & TH\_str \geq 0 \end{cases} \quad (103)$$

TH_str is calculated according to formula (104), *mid_T* is calculated according to formula (105), *down_T* is calculated according to formula (106), and *up_T* is calculated according to formula (107).

$$TH\_str = Spline\_Strength[1] \quad (104)$$

$$mid\_T = (VA3 - VA1) \div (TH3[2] - TH1[2]) \quad (105)$$

$$down\_T = \max(GD1, down\_T1) \quad (106)$$

$$up\_T = \max(GD1, up\_T1) \quad (107)$$

In formula (106), *GD*1 is obtained according to formula (100), and *down_T*1 is obtained according to formula (108); in formula (107), *GD*1 is obtained according to formula (100), and *up_T*1 is obtained according to formula (109).

$$down\_T1 = (VA3 - VA1) \times 0.1 \div (TH3[2] - TH1[2]) \quad (108)$$

$$up\_T1 = (VA3 - VA1) \div (TH3[2] - TH2[2]) \quad (109)$$

If 3*Spline_TH_mode* is 2, *GD*3 is obtained according to formula (110).

$$GD3 = GD2 - 3Spline\_TH\_MB \quad (110)$$

If 3*Spline_TH_mode* is 3, *GD*3 = *GD*2, where *GD*2 is obtained according to formula (111).

$$GD2 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH3[2]^{m\_n-1} \times \mathrm{DGD3}(L) \quad (111)$$

*DGD*3(*L*) is obtained according to formula (112).

$$\mathrm{DGD3}(L) = \left(\frac{m\_p \times TH3[2]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[2]^{m\_n} + K3}\right)^{m\_m+1} \times \left(\frac{1}{TH3[2]^{m\_n} \times m\_p}\right)^2 \quad (112)$$

i) *GD*3 is updated: If 3*Spline_TH_mode* is 1 or 2, *VA*3 is *TH*3[2], and *base_param_Delta_mode* is a value other than 2, 3, and 6, *GD*3 = 1.0.

j) The curve parameter *MC*[0][2] of the cubic spline interval 1 is calculated according to formula (113), the curve parameter *MD*[0][2] of the cubic spline interval 1 is calculated according to formula (114), the curve parameter *MB*[1][2] of the cubic spline interval 2 is calculated according to formula (115), the curve parameter *MC*[1][2] of the cubic spline interval 2 is calculated according to formula (116), and the curve parameter *MD*[1][2] of the cubic spline interval 2 is calculated according to formula (117).

$$MC[0][2] = \frac{3.0 \times VA2 - 2.0 \times GD1 \times h1 - 3.0 \times VA1 - MB[1][2] \times h1}{h1 \times h1} \quad (113)$$

$$MD[0][2] = \frac{h1 \times GD1 + h1 \times MB[1][2] + 2 \times VA1 - 2.0 \times VA2}{h1 \times h1 \times h1} \quad (114)$$

$$MB[1][2] = \frac{-3.0 \times VA1 \times h2 \times h2 - 3.0 \times VA2 \times h1 \times h1 + 3.0 \times VA3 \times h1 \times h1 + 3.0 \times h2 \times h2 \times VA2 - h1 \times h1 \times h2 \times GD3 - GD1 \times h1 \times h2 \times h2}{2.0 \times h2 \times (h1 \times h1 + h2 \times h1)} \quad (115)$$

$$MC[1][2] = MC[0][2] + 3.0 \times MD[0][2] \times h1 \quad (116)$$

$$MD[1][2] = -\frac{VA3 - VA2 - h2 \times GD3 + MC[0][2] \times h2 \times h2 + 3 \times MD[0][2] \times h1 \times h2 \times h2}{2 \times h2 \times h2 \times h2} \quad (117)$$

In formula (113) to formula (117), *h*1 is represented by a cubic spline interval 1. Refer to formula (118). *h*2 is represented by a cubic spline interval 2. Refer to formula (119).

$$h1 = TH2[2] - TH1[2] \quad (118)$$

$$h2 = TH3[2] - TH2[2] \quad (119)$$

10.4 Dynamic Range Conversion Process of Color Signal

Input: an RGB pixel buffer $f[N_{frame}][3]$, 3*Spline_TH_mode*, *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, *K*1, *K*2, *K*3, *TH*3[0], *TH*2[1], *TH*3[1], *MA*[0][1], *MB*[0][1], *MC*[0][1], *MD*[0][1], *MA*[1][1], *MB*[1][1], *MC*[1][1], *MD*[1][1], *TH*1[2], *TH*2[2], *TH*3[2], *MA*[0][2], *MB*[0][2], *MC*[0][2], *MD*[0][2], *MA*[1][2], *MB*[1][2], *MC*[1][2], *MD*[1][2], and *base_offset*.

Output: an RGB color gamut pixel buffer $f_{TM}[N_{frame}][3]$ after dynamic range conversion.

The conversion process is as follows.

a) Calculate $f_{MAX}[i]$, $f_{MAX}[i]$ = Max(Max($f[i][0]$, $f[i][1]$), $f[i][2]$), where *i* is the pixel index.

b) $f_{MAX\_TM}[i]$ is calculated:

If 0 ≤ $f_{MAX}[i]$ < *TH*3[0], the output value $f_{MAX\_TM}[i]$ of $f_{MAX}[i]$ on the curve is calculated according to formula (120).

$$f_{MAX\_TM}[i] = MB[0][0] \times f_{MAX}[i] + base\_offset \quad (120)$$

If *TH*3[0] ≤ $f_{MAX}[i]$ < *TH*2[1], for calculation of $f_{MAX\_TM}[i]$, refer to formula (121).

$$f_{MAX\_TM}[i] = MD[0][0] \times (f_{MAX}[i] - TH3[0])^3 + MC[0][0] \times (f_{MAX}[i] - TH3[0])^2 + MB[0][0] \times (f_{MAX}[i] - TH3[0])^1 + MA[0][0] \quad (121)$$

If *TH*2[1] ≤ $f_{MAX}[i]$ < *TH*3[1], for calculation of $f_{MAX\_TM}[i]$, refer to formula (122).

$$f_{MAX\_TM}[i] = MD[1][0] \times (f_{MAX}[i] - TH2[1])^3 + MC[1][0] \times (f_{MAX}[i] - TH2[1])^2 + MB[1][0] \times (f_{MAX}[i] - TH2[1])^1 + MA[1][0] \quad (122)$$

If *TH*3[1] ≤ $f_{MAX}[i]$ ≤ *TH*1[2], for calculation of $f_{MAX\_TM}[i]$, refer to formula (123).

$$f_{\mathrm{MAX\_TM}}[i] = m\_a \times \left(\frac{m\_p \times (f_{\mathrm{MAX}}[i])^{m\_n}}{(K \times m\_p - K_2) \times (f_{\mathrm{MAX}}[i])^{m\_n} + K_3}\right)^{m\_m} + m\_b \quad (123)$$

If $TH1[2] < f_{\mathrm{MAX}}[i] < TH2[2]$, for calculation of $f_{\mathrm{MAX\_TM}}[i]$, refer to formula (124).

$$f_{\mathrm{MAX\_TM}}[i] = MD[0][2] \times (f_{\mathrm{MAX}}[i] - TH1[2])^3 + MC[0][2] \times (f_{\mathrm{MAX}}[i] - TH1[2])^2 + MB[0][2] \times (f_{\mathrm{MAX}}[i] - TH1[2])^1 + MA[0][2] \quad (124)$$

If $TH2[2] \le f_{\mathrm{MAX}}[i] < TH3[2]$, for calculation of $f_{\mathrm{MAX\_TM}}[i]$, refer to formula (125).

$$f_{\mathrm{MAX\_TM}}[i] = MD[1][2] \times (f_{\mathrm{MAX}}[i] - TH2[2])^3 + MC[1][2] \times (f_{\mathrm{MAX}}[i] - TH2[2])^2 + MB[1][2] \times (f_{\mathrm{MAX}}[i] - TH2[2])^1 + MA[1][2] \quad (125)$$

If $f_{\mathrm{MAX}}[i] \ge TH3[2]$:

If 3*Spline_TH_mode* is 1 or 2, for calculation of $f_{\mathrm{MAX\_TM}}[i]$, refer to formula (126).

$$f_{\mathrm{MAX\_TM}}[i] = MBH \times (f_{\mathrm{MAX}}[i] - TH3[2]) + BASEH \quad (126)$$

MBH is obtained according to formula (127), and *BASEH* is obtained according to formula (128).

$$MBH = 3 \times MD[1][2] \times H1^2 + 2 \times MC[1][2] \times H1 + MB[1][2] \quad (127)$$

$$BASEH = MD[1][2] \times H1^3 + MC[1][2] \times H1^2 + MB[1][2] \times H1^1 + MA[1][2] \quad (128)$$

*H*1 is obtained according to formula (129).

$$H1 = (TH3[2] - TH2[2]) \quad (129)$$

If 3*Spline_TH_mode* is neither 1 nor 2, for calculation of $f_{\mathrm{MAX\_TM}}[i]$, refer to formula (130).

$$f_{\mathrm{MAX\_TM}}[i] = m\_a \times \left(\frac{m\_p \times (f_{\mathrm{MAX}}[i])^{m\_n}}{(K_1 \times m\_p - K_2) \times (f_{\mathrm{MAX}}[i])^{m\_n} + K_3}\right)^{m\_m} + m\_b \quad (130)$$

c) The gain coefficient *K* is calculated according to formula (131).

$$K = \mathrm{PQ\_EOTF}(f_{\mathrm{MAX\_TM}}[i]) \div \mathrm{PQ\_EOTF}(f_{\mathrm{MAX}}[i]) \quad (131)$$

PQ_EOTF () should comply with the requirements in GY/T 315-2018.

d) Dynamic range conversion is performed:

The pixels $f_{\mathrm{TM}}[i][0]$, $f_{\mathrm{TM}}[i][1]$, and $f_{\mathrm{TM}}[i][2]$ after the dynamic range conversion are calculated according to formula (132).

$$\begin{aligned} f_{\mathrm{TM}}[i][0] &= \mathrm{PQ\_EOTF}(f[i][[0]) \times K \\ f_{\mathrm{TM}}[i][1] &= \mathrm{PQ\_EOTF}(f[i][[1]) \times K \\ f_{\mathrm{TM}}[i][2] &= \mathrm{PQ\_EOTF}(f[i][[2]) \times K \end{aligned} \quad (132)$$

10.5 Color Adjustment Process

Input: an RGB pixel buffer $f[N_{\mathrm{frame}}][3]$, $f_{\mathrm{TM}}[N_{\mathrm{frame}}][3]$, *color_saturation_gain*[0], *color_saturation_gain*[1], *MaxDisplayPQ*, *max_display_mastering_luminance*, *color_saturation_mapping_flag*, and *color_saturation_num*.

Output: an RGB pixel buffer $f_{\mathrm{process}}[N_{\mathrm{frame}}][3]$ after color adjustment process.

The specific color adjustment process is as follows.

a) If *color_saturation_mapping_flag* == 0, the pixels $f_{\mathrm{color}}[N_{\mathrm{frame}}][0]$, $f_{\mathrm{color}}[N_{\mathrm{frame}}][1]$, and $f_{\mathrm{color}}[N_{\mathrm{frame}}][2]$ after color adjustment are calculated according to formula (133).

$$\begin{aligned} f_{\mathrm{color}}[N_{\mathrm{frame}}][0] &= f_{\mathrm{TM}}[N_{\mathrm{frame}}][0] \\ f_{\mathrm{color}}[N_{\mathrm{frame}}][1] &= f_{\mathrm{TM}}[N_{\mathrm{frame}}][1] \\ f_{\mathrm{color}}[N_{\mathrm{frame}}][2] &= f_{\mathrm{TM}}[N_{\mathrm{frame}}][2] \end{aligned} \quad (133)$$

The color correction process ends.

Otherwise, the following steps are performed to calculate $f_{\mathrm{process}}[N_{\mathrm{frame}}][3]$.

b) The color correction parameter $C0$ is calculated according to formula (134), and the color correction parameter $C1$ is calculated according to formula (135).

$$C0 = color\_saturation\_gain[0] \quad (134)$$

$$C1 = color\_saturation\_gain[1] \quad (135)$$

For calculation of the conversion from a linear value to a non-linear PQ value, refer to formula (136).

$$\begin{aligned} f_{\mathrm{TM\_PQ}}[i][0] &= \mathrm{PQ\_EOTF}^{-1}(f_{\mathrm{TM}}[i][0]) \\ f_{\mathrm{TM\_PQ}}[i][1] &= \mathrm{PQ\_EOTF}^{-1}(f_{\mathrm{TM}}[i][1]) \\ f_{\mathrm{TM\_PQ}}[i][2] &= \mathrm{PQ\_EOTF}^{-1}(f_{\mathrm{TM}}[i][2]) \end{aligned} \quad (136)$$

For calculation of RGB-to-YC_bC_r conversion, refer to formula (137).

$$\begin{aligned} Y &= 0.2627 \times f_{\mathrm{TM\_PQ}}[i][0] + 0.6780 \times f_{\mathrm{TM\_PQ}}[i][1] + 0.0593 \times f_{\mathrm{TM\_PQ}}[i][2] \\ C_b &= -0.1396 \times f_{\mathrm{TM\_PQ}}[i][0] - 0.3604 \times f_{\mathrm{TM_{PQ}}}[i][1] + 0.5000 \times f_{\mathrm{TM\_PQ}}[i][2] \\ C_r &= 0.5000 \times f_{\mathrm{TM\_PQ}}[i][0] - 0.4598 \times f_{\mathrm{TM_{PQ}}}[i][1] - 0.0402 \times f_{\mathrm{TM\_PQ}}[i][2] \end{aligned} \quad (137)$$

c) For the maximum value ($f_{\mathrm{MAX}}[i]$) of $f[i][0]$, $f[i][1]$, and $f[i][2]$, refer to formula (138). For the maximum value ($f_{\mathrm{MAX\_TM\_PQ}}[i]$) of $f_{\mathrm{TM\_PQ}}[i][0]$, $f_{\mathrm{TM\_PQ}}[i][1]$, and $f_{\mathrm{TM\_PQ}}[i][2]$, refer to formula (139).

$$f_{\mathrm{MAX}}[i] = \mathrm{Max}(\mathrm{Max}(f[i][0], f[i][1]), f[i][2]) \quad (138)$$

$$f_{\mathrm{MAX\_TM\_PQ}}[i] = \mathrm{Max}(\mathrm{Max}(f_{\mathrm{TM\_PQ}}[i][0], f_{\mathrm{TM\_PQ}}[i][1]), f_{\mathrm{TM\_PQ}}[i][2]) \quad (139)$$

d) S_{ca} is calculated.

If $f_{\mathrm{MAX}}[i] > TML$ and *color_saturation_num* ≥ 2, the color adjustment coefficient S_{ca} is calculated according to formula (140).

$$S_{\mathrm{ca}} = \begin{cases} B - C1 \times SatR \times \left(\dfrac{f_{\mathrm{MAX}}[i] - A \times RML}{RML - A \times RML}\right)^{M} & TML < f_{\mathrm{MAX}}[i] < RML \\ B - C1 \times SatR & f_{\mathrm{MAX}}[i] \geq RML \end{cases} \quad (140)$$

TML = *MaxDisplayPQ*, *RML* = $\mathrm{PQ\_EOTF}^{-1}$(*max_display_mastering_luminance*), *SatR* = 0.4, *A* = *TML* ÷ *RML*, $M = 2^{(\mathrm{color\_saturation\_gain[1]\&0x3})}$, B = Clip3(0.8, 1.0, $(\frac{TML\_TM}{TML})^{C0}$) is a strength range coefficient, and ranges from 0.8 to 1.0, and the default value is $(\frac{TML\_TM}{TML})^{C0}$.

TML_TM = $f_{\mathrm{MAX\_TM}}[i]$. For calculation of $f_{\mathrm{MAX\_TM}}[i]$, refer to step b) in section 10.4, $f_{\mathrm{MAX}}[i]$ = *MaxDisplayPQ*. The updated S_{ca} is calculated according to formula (141).

$$S_{ca} = \mathrm{Clip3}(0.0, 1.0,\ S_{ca}) \quad (141)$$

Otherwise, the color adjustment coefficient S_{ca} is calculated according to formula (142).

$$S_{ca} = \mathrm{Clip3}\left(0.8, 1.0, \left(\frac{f_{MAX\_TM\_PQ}[i]}{f_{MAX}[i]}\right)^{C0}\right) \quad (142)$$

e) R_{ca}', G_{ca}', and B_{ca}' are calculated.

Y, C_b', and C_r' after saturation adjustment are calculated according to formula (143).

$$\begin{aligned} Y' &= Y \\ C_b' &= C_b \times S_{ca} \\ C_r' &= C_r \times S_{ca} \end{aligned} \quad (143)$$

Conversion from Y, C_b', and C_r' to R_{ca}', G_{ca}', and B_{ca}' are calculated according to formula (144).

$$\begin{aligned} R_{ca}' &= Y' + 0.0000 \times C_b' + 1.4746 \times C_r' \\ G_{ca}' &= Y' - 0.1645 \times C_b' - 0.5713 \times C_r' \\ B_{ca}' &= Y' + 1.8814 \times C_b' - 0.0001 \times C_r' \end{aligned} \quad (144)$$

f) For calculation of the linear values, refer to formula (145).

$$\begin{aligned} R_{color1} &= \mathrm{PQ\_EOTF}(R_{ca}') \\ G_{color1} &= \mathrm{PQ\_EOTF}(G_{ca}') \\ B_{color1} &= \mathrm{PQ\_EOTF}(B_{ca}') \end{aligned} \quad (145)$$

g) $f_{color}[N_{frame}][0]$, $f_{color}[N_{frame}][1]$, and $f_{color}[N_{frame}][2]$ are calculated: $f_{color}[N_{frame}][0] = R_{color1}$, $f_{color}[N_{frame}][1] = G_{color1}$, $f_{color}[N_{frame}][2] = B_{color1}$.

h) $f_{process}[N_{frame}][3]$ is calculated: $f_{process}[N_{frame}][0] = f_{color}[N_{frame}][0]$, $f_{process}[N_{frame}][1] = f_{color}[N_{frame}][1]$, $f_{process}[N_{frame}][2] = f_{color}[N_{frame}][2]$.

11 SDR Display Adaptation of PQ HDR Video

11.1 SDR Display Adaptation Process

Input: an RGB pixel buffer $f_{process}[N_{frame}][3]$ and a metadata variable.

Output: an RGB pixel buffer $f_{process}[N_{frame}][3]$ after SDR display adaptation.

The SDR display adaptation process is as follows.

a) The base curve parameter is generated by calling section 11.2.

b) The cubic spline curve parameter is generated by calling section 11.3.

c) The RGB pixel buffer $f_{\mathrm{TM}}[N_{\mathrm{frame}}][3]$ after dynamic range conversion processing is generated by calling section 10.4.

d) $f_{\mathrm{process}}[N_{\mathrm{frame}}][3]$ is generated by calling section 10.5.

11.2 Base Curve Parameter Obtaining Process

11.2.1 Overview

The base curve parameter obtaining process is as follows.

a) The minimum luminance correction value *min_lum* is calculated: *min_lum* = *minimum_maxrgb*.

b) The maximum luminance correction value *max_lum* is calculated according to section 10.2.2.

c) The base curve parameter is calculated.

1) If *tone_mapping_mode_flag* is 0, sections 11.2.2 and 10.2.6 are sequentially called to obtain the base curve parameter.

2) If *tone_mapping_mode_flag* is 1 and *base_flag* is 0, sections 11.2.2 and 10.2.6 are sequentially called to obtain the base curve parameter.

3) If *tone_mapping_mode_flag* is 1 and *base_flag* is 1:

If *targeted_system_display_maximum_luminance* is equal to *MaxDisplayPQ*, *m_p* = *m_p*_0, *m_a* = *m_a*_0, *m_m* = *m_m*_0, *m_n* = *m_n*_0, *m_b* = *m_b*_0, *K*1 = *k*1_0, *K*2 = *k*2_0, and *K*3 = *k*3_0;

If *base_param_Delta_mode* is 3, *m_p* = *m_p*_0, *m_a* = *m_a*_0, *m_m* = *m_m*_0, *m_n* = *m_n*_0, *m_b* = *m_b*_0, *K*1 = *k*1_0, *K*2 = *k*2_0, and *K*3 = *k*3_0;

If *base_param_Delta_mode* is 0, 2, 4, or 6, sections 10.2.4 and 10.2.6 are sequentially called to obtain the base curve parameter;

If *base_param_Delta_mode* is 1 or 5, sections 10.2.5 and 10.2.6 are sequentially called to obtain the base curve parameter.

11.2.2 Base curve parameter obtaining process 0

Input: *MaxDisplayPQ*, *MinDisplayPQ*, *minimum_maxrgb*, *maximum_maxrgb*, *variance_maxrgb*, *average_maxrgb*, and *max_lum*.

Output: *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, *K*1, *K*2, and *K*3.

The steps of the base curve parameter obtaining process 0 are as follows.

a) *m_m* = 2.4, *m_n* = 1, *K*1 = 1, *K*2 = 1, *K*3 = 1, and *m_b* = *MinDisplayPQ*.

b) The intermediate variable *m_p*0 of *m_p* is calculated according to formula (146).

$$m\_p0 = \begin{cases} p_{\mathrm{valueH4}} & avgL > TPH4 \\ p_{\mathrm{valueH4}} \times \mathrm{g4}(w4) + p_{\mathrm{valueL4}} \times (1 - \mathrm{g4}(w4)) & TPL4 \le avgL \le TPH4 \\ p_{\mathrm{valueL4}} & avgL < TPL4 \end{cases} \quad (146)$$

$avgL$ is obtained according to formula (147), and $w4$ is obtained according to formula (148).

$$avgL = average\_maxrgb \qquad (147)$$

$$w4 = \left(\frac{avgL - TPL4}{TPH4 - TPL4}\right) \qquad (148)$$

In formula (147) and formula (148), p_{valueL4} = 3.5, p_{valueL4} = 6.0, $TPH4$ = 0.6, $TPL4$ = 0.1, and g4(x) meets $y = x$.

c) The base curve parameter m_p is calculated according to formula (149).

$$m\_p = \begin{cases} m\_p0 + p_{\text{deltaH5}} & max\_lum > TPH5 \\ m\_p0 + p_{\text{deltaH5}} \times g5(w5) + p_{\text{deltaL5}} \times (1 - g5(w5)) & TPL5 \leq max\_lum \leq TPH5 \\ m\_p0 + p_{\text{deltaL5}} & max\_lum < TPL5 \end{cases} \qquad (149)$$

$w5$ is obtained according to formula (150).

$$w5 = \left(\frac{max\_lum - TPL5}{TPH5 - TPL5}\right) \qquad (150)$$

In formula (149) and formula (150), p_{deltaH5} = 0.6, p_{deltaL5} = 0.3, $TPH5$ = 0.75, $TPL5$ = 0.67; g5(x) meets $y = x$.

d) The base curve parameter m_a is calculated according to formula (151).

$$m\_a = (MaxDisplayPQ - MinDisplayPQ) \div \left(\frac{m\_p \times max\_lum^{m\_n}}{(K1 \times m\_p - K2) \times max\_lum^{m\_n} + K3}\right)^{m\_m} \qquad (151)$$

11.3 Cubic Spline Curve Parameter Obtaining Process

11.3.1 Overview

The cubic spline curve parameter obtaining process is as follows. Refer to Figure 3.

a) 3*Spline_num* and 3*Spline_TH_mode* are calculated:

If *tone_mapping_mode_flag* is 0, 3*Spline_num* = 1, and 3*Spline_TH_mode* = 0. Otherwise, 3*Spline_num* and 3*Spline_TH_mode* are obtained according to chapter 9.

b) The cubic spline curve parameter is calculated:

If *tone_mapping_mode_flag* is 0, sections 11.3.2.2 and 11.3.3.2 are sequentially called to obtain the cubic spline curve parameter values.

If *tone_mapping_mode_flag* is 1:

When 3*Spline_flag* is 0, sections 11.3.2.2, 10.3.2.4, and 11.3.3.2 are sequentially called to obtain the cubic spline mapping curve parameter values.

When 3*Spline_flag* is 1 and 3*Spline_TH_mode* is 0, sections 10.3.2.3 and 10.3.2.4 are sequentially called to obtain the linear spline curve parameters, and section 10.3.3.3 is called to obtain the parameter value of the first segment of the cubic spline curve.

When 3*Spline_flag* is 1, if 3*Spline_TH_mode* is not 0, sections 11.3.2.2 and 10.3.2.4 are sequentially called to obtain the linear spline curve parameters, section 11.3.3.2 is called to obtain the parameter value of the first segment of the cubic spline curve, and then section 10.3.3.4 is called to obtain the parameter value of the second segment of the cubic spline curve.

c) If 3*Spline_num* is equal to 2, section 10.3.3.4 is called to obtain the parameter value of the second segment of the cubic spline curve.

11.3.2 Linear spline curve parameter obtaining process

11.3.2.1 Linear spline curve

The linear spline curve is a curve between the first endpoint and the first interpolation point *TH*3[0]. Refer to formula (152).

$$F(L) = MB[0][0] \times L + base\_offset \quad (152)$$

11.3.2.2 Linear spline curve parameter obtaining process 0

Input: *average_maxrgb*.

Output: *TH*3[0], *MB*[0][0], and *base_offset*.

The steps of the linear spline curve parameter obtaining process 0 are as follows.

a) TH3[0] = 0;

b) *base_offset* = 0;

c) The slope *MB*[0][0] is obtained according to formula (153):

$$MB[0][0]=\begin{cases} S_{\mathrm{dmaxL6}} & avgL > AVMAXH6 \\ S_{\mathrm{dmaxL6}} \times g6(w6) + S_{\mathrm{dmaxH6}} \times (1 - g6(w6)) & AVMAXL3 \le avgL \le AVMAXH6 \\ S_{\mathrm{dmaxH6}} & avgL < AVMAXL6 \end{cases} \quad (153)$$

avgL is obtained according to formula (154), and *w*6 is obtained according to formula (155).

$$avgL = average\_maxrgb \quad (154)$$

$$w6 = \left(\frac{avgL - AVMAXL6}{AVMAXH6 - AVMAXL6}\right) \quad (155)$$

In formula (153) and formula (155), *AVMAXH*6 = 0.6, *AVMAXL*6 = 0.3, S_{dmaxH6} = 1.0, S_{dmaxL6} =0.9, and g6(*x*) meets *y* = *x*.

11.3.3 Cubic spline curve parameter obtaining process

11.3.3.1 Linear spline curve

A curve between the first interpolation point *TH*1[*n*] and the second interpolation point *TH*2[*n*] is a curve of a cubic spline interval 1. Refer to formula (156).

$$F(L) = MD[0][n] \times (L - TH1[n])^3 + MC[0][n] \times (L - TH1[n])^2 + MB[0][n] \times (L - TH1[n])^1 + MA[0][n] \quad (156)$$

L is an independent variable in an interval [$TH1[n]$, $TH2[n]$].

A curve between the second interpolation point $TH2[n]$ and the third interpolation point $TH3[n]$ is a curve of a cubic spline interval 2. Refer to formula (157).

$$F(L) = MD[1][n] \times (L - TH2[n])^3 + MC[1][n] \times (L - TH2[n])^2 + MB[1][n] \times (L - TH2[n])^1 + MA[1][n] \quad (157)$$

L is an independent variable in an interval [$TH2[n]$, $TH3[n]$], and $0 < n \le 3Spline\_num$.

11.3.3.2 Cubic spline curve parameter obtaining process 0

Input: $TH3[0]$, $MB[0][0]$, *base_offset*, *m_p*, *m_m*, *m_n*, *m_a*, *m_b*, $K1$, $K2$, and $K3$.

Output: $TH1[1]$, $TH2[1]$, $TH3[1]$, $MA[0][1]$, $MB[0][1]$, $MC[0][1]$, $MD[0][1]$, $MA[1][1]$, $MB[1][1]$, $MC[1][1]$, and $MD[1][1]$.

The obtaining process is as follows.

a) The first interpolation point $TH1[1]$ is calculated according to formula (158), the second interpolation point $TH2[1]$ is calculated according to formula (159), and the third interpolation point $TH3[1]$ is calculated according to formula (160).

$$TH1[1] = TH3[0] \quad (158)$$

$$TH2[1] = TH1[1] + B \quad (159)$$

$$TH3[1] = TH2[1] + C \times TH2[1] - D \times TH1[1] \quad (160)$$

In formula (159) and formula (160), $B = 0.15$, $C = 0.5$, and $D = 0.5$.

b) $MA[0][1]$, $MB[0][1]$, $MC[0][1]$, $MD[0][1]$, $MA[1][1]$, $MB[1][1]$, $MC[1][1]$, and $MD[1][1]$ are calculated.

1) The output value $VA1$ of the first interpolation point $TH1[1]$ on the linear spline curve is calculated according to formula (161), the output value $VA3$ of the third interpolation point $TH3[1]$ on the base curve is calculated according to formula (162), and the output value $VA2$ of the second interpolation point $TH2[1]$ on the curve is calculated according to formula (163).

$$VA1 = MB[0][0] \times TH1[1] + base\_offset \quad (161)$$

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad (162)$$

$$VA2 = m\_a \times \left(\frac{m\_p \times TH2[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH2[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad (163)$$

2) The curve parameter $MA[0][1]$ of the cubic spline interval 1 is calculated according to formula (164), and the curve parameter $MA[1][1]$ of the cubic spline interval 2 is calculated according to formula (165).

$$MA[0][1] = VA1 \quad (164)$$

$$MA[1][1] = VA2 \quad (165)$$

3) The slope $GD1$ of the first interpolation point $TH1[1]$ on the curve is calculated according to formula (166), the curve parameter $MB[0][1]$ of the cubic spline interval 1 is calculated according to formula (167), and the slope $GD3$ of the third interpolation point $TH3[1]$ on the curve is calculated according to formula (168).

$$GD1 = MB[0][0] \tag{166}$$

$$MB[0][1] = MB[0][0] \tag{167}$$

$$GD3 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH3[1]^{m\_n-1} \times \mathrm{DGD3}(L) \tag{168}$$

$DGD3(L)$ is obtained according to formula (169).

$$\mathrm{DGD3}(L) = \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m+1} \times \left(\frac{1}{TH3[1]^{m\_n} \times m\_p}\right)^2 \tag{169}$$

4) The curve parameter $MC[0][1]$ of the cubic spline interval 1 is calculated according to formula (170), the curve parameter $MD[0][1]$ of the cubic spline interval 1 is calculated according to formula (171), the curve parameter $MB[1][1]$ of the cubic spline interval 2 is calculated according to formula (172), the curve parameter $MC[1][1]$ of the cubic spline interval 2 is calculated according to formula (173), and the curve parameter $MD[1][1]$ of the cubic spline interval 2 is calculated according to formula (174).

$$MC[0][1] = \frac{3.0 \times VA2 - 2.0 \times GD1 \times h1 - 3.0 \times VA1 - MB[1][1] \times h1}{h1 \times h1} \tag{170}$$

$$MD[0][1] = \frac{h1 \times GD1 + h1 \times MB[1][1] + 2 \times VA1 - 2.0 \times VA2}{h1 \times h1 \times h1} \tag{171}$$

$$MB[1][1] = \frac{-3.0 \times VA1 \times h2 \times h2 - 3.0 \times VA2 \times h1 \times h1 + 3.0 \times VA3 \times h1 \times h1 + 3.0 \times h2 \times h2 \times VA2 - h1 \times h1 \times h2 \times GD3 - GD1 \times h1 \times h2 \times h2}{2.0 \times h2 \times (h1 \times h1 + h2 \times h1)} \tag{172}$$

$$MC[1][1] = MC[0][1] + 3.0 \times MD[0][1] \times h1 \tag{173}$$

$$MD[1][1] = -\frac{VA3 - VA2 - h2 \times GD3 + MC[0][1] \times h2 \times h2 + 3 \times MD[0][1] \times h1 \times h2 \times h2}{2 \times h2 \times h2 \times h2} \tag{174}$$

In formula (170) to formula (174), $h1$ is represented by a cubic spline interval 1. Refer to formula (175). $h2$ is represented by a cubic spline interval 2. Refer to formula (176).

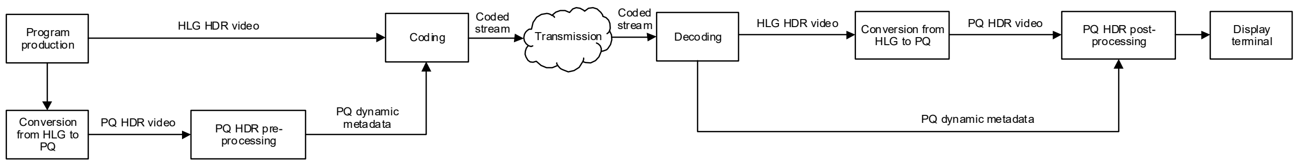
$$h1 = TH2[1] - TH1[1] \tag{175}$$

$$h2 = TH3[1] - TH2[1] \tag{176}$$

Annex A
(Informative)
HLG HDR Video Display Adaptation Method

The HLG HDR video can be converted into the PQ HDR video at the front end, and the PQ HDR dynamic metadata can be obtained according to the PQ HDR video pre-processing method. During video coding, the HLG HDR video is coded and the PQ HDR dynamic metadata is encapsulated. At the receiver, the decoder performs decoding to obtain the HLG HDR video and the PQ HDR dynamic metadata, converts the HLG HDR video into the PQ HDR video, and performs display adaptation processing on the PQ HDR video based on the PQ HDR video dynamic metadata. For the specific processing process, refer to Figure A.1.

Figure A.1 HLG HDR video display adaptation process



Annex B
(Informative)
Dynamic Metadata Extraction Method

B.1 Overview

This annex describes a method for extracting metadata in the HDR pre-processing phase.

The HDR metadata extraction process is as follows:

a) Metadata *minimum_maxrgb_pq*, *maximum_maxrgb_pq*, *average_maxrgb_pq*, and *variance_maxrgb_pq* are calculated by calling sections B.2, B.3, and B.4;

b) The base curve parameter metadata is generated by calling section B.5;

c) The cubic spline curve parameter metadata is generated by calling section B.6;

d) Time-domain filtering is performed on the dynamic metadata by calling section B.7;

e) Quality control is performed on metadata by calling section B.8.

B.2 Calculation of Dynamic Metadata minimum_maxrgb_pq[w] and maximum_maxrgb_pq[w]

The calculation process of *minimum_maxrgb_pq*[*w*] and *maxmum_maxrgb_pq*[*w*] is as follows:

a) The maximum value ($f_{MAX}[index]$) of $f[index][0]$, $f[index][1]$, and $f[index][2]$ is calculated according to formula (B.1).

$$f_{MAX}[index] = \mathrm{Max}(\mathrm{Max}(f[index][0], f[index][1]), f[index][2]) \quad \text{(B.1)}$$

index is a pixel index value, and $0 \le index < N_{frame}$.

b) f_{MAX_MIN} and f_{MAX_MAX} are calculated.

```
fMAX_MIN=1.0, fMAX_MAX=0.0;
for (i=0;i<Nframe;i++) {
    fMAX_MIN=Min(fMAX_MIN, fMAX[i])
    fMAX_MAX=Max(fMAX_MAX, fMAX[i])
}
```

c) The metadata *minimum_maxrgb_pq*[*w*] is calculated according to formula (B.2), and the metadata *maximum_maxrgb_pq*[*w*] is calculated according to formula (B.3).

$$minimum\_maxrgb\_pq[w] = \mathrm{Floor}(f_{MAX\_MIN} \times 4095) \quad \text{(B.2)}$$

$$maximum\_maxrgb\_pq[w] = \mathrm{Floor}(f_{MAX\_MAX} \times 4095) \quad \text{(B.3)}$$

B.3 Calculation of Dynamic Metadata average_maxrgb_pq[w]

The calculation process of *average_maxrgb_pq*[*w*] is as follows:

a) The maximum value ($f_{MAX}[index]$) of $f[index][0]$, $f[index][1]$, and $f[index][2]$ is calculated according to formula (B.4).

$$f_{MAX}[index] = \mathrm{Max}(\mathrm{Max}(f[index][0], f[index][1]), f[index][2]) \quad \text{(B.4)}$$

b) For calculation of the average value $f_{MAX\_LINE\_AVG}$, refer to formula (B.5).

$$f_{MAX\_LINE\_AVG} = \frac{\sum_{i=0}^{N_{frame}-1} \mathrm{PQ\_EOTF}(f_{MAX}[i])}{N_{frame}} \quad \text{(B.5)}$$

c) The metadata *average_maxrgb_pq*[*w*] is calculated according to formula (B.6).

$$average\_maxrgb\_pq[w] = \mathrm{Floor}(\mathrm{PQ\_EOTF}^{-1}(f_{MAX\_LINE\_AVG}) \times 4095) \quad \text{(B.6)}$$

B.4 Calculation of Dynamic Metadata variance_maxrgb_pq[w]

The calculation process of *variance_maxrgb_pq*[*w*] is as follows:

a) The maximum value ($f_{MAX}[index]$) of $f[index][0]$, $f[index][1]$, and $f[index][2]$ is calculated according to formula (B.7).

$$f_{MAX}[index] = \mathrm{Max}(\mathrm{Max}(f[index][0], f[index][1]), f[index][2]) \quad \text{(B.7)}$$

b) The $f_{MAX}[index]$ value f_{MAX_A} corresponding to 10% of the number is calculated according to formula (B.8).

$$\frac{\mathrm{N}(f_{MAX\_A})}{N_{frame}} = 0.1 \quad \text{(B.8)}$$

N(x) indicates the number of $f_{MAX}[N_{frame}]$ within the range of $f_{MAX}[N_{frame}] < x$.

c) The $f_{MAX}[index]$ value f_{MAX_B} corresponding to 90% of the number is calculated according to formula (B.9).

$$\frac{\mathrm{N}(f_{MAX\_B})}{N_{frame}} = 0.9 \quad \text{(B.9)}$$

N(x) indicates the number of $f_{MAX}[N_{frame}]$ within the range of $f_{MAX}[N_{frame}] < x$.

d) The metadata *variance_maxrgb_pq*[*w*] is calculated according to formula (B.10).

$$variance\_maxrgb\_pq[w] = \mathrm{Floor}((f_{MAX\_B} - f_{MAX\_A}) \times 4095) \quad \text{(B.10)}$$

B.5 Base Curve Parameter Metadata Generation Process

B.5.1 Overview

Input: an RGB pixel buffer $f[N_{frame}][3]$.

Output: *base_param_m_p*[*i*][*w*], *base_param_m_m*[*i*][*w*], *base_param_m_n*[*i*][*w*], *base_param_m_a*[*i*][*w*], *base_param_m_b*[*i*][*w*], *base_param_K*1[*i*][*w*], *base_param_K*2[*i*][*w*], and *base_param_K*3[*i*][*w*].

The base curve parameter generation process is as follows.

a) The maximum value ($f_{MAX}[index]$) of $f[index][0]$, $f[index][1]$, and $f[index][2]$ is calculated according to formula (B.11).

$$f_{\mathrm{MAX}}[index] = \mathrm{Max}(\mathrm{Max}(f[index][0], f[index][1]), f[index][2]) \quad \text{(B.11)}$$

index is a pixel index, and $0 \le index < N_{\mathrm{frame}}$.

b) The dark area variable R_{DARK} is calculated according to formula (B.12), and the dark area variable L_{DARK} is calculated according to formula (B.13).

$$R_{\mathrm{DARK}} = \frac{\mathrm{N_{DARK}}}{N_{\mathrm{frame}}} \quad \text{(B.12)}$$

$$L_{\mathrm{DARK}} = \frac{\mathrm{PQ\_EOTF}^{-1}(DARK)}{max\_lum} \quad \text{(B.13)}$$

In formula (B.12) and formula (B.13), $\mathrm{N_{DARK}}$ is the number of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ within the range of $0 \le f_{\mathrm{MAX}}[N_{\mathrm{frame}}] \le \mathrm{PQ\_EOTF}^{-1}(DARK)$, and *DARK* is the maximum luminance value of the dark area.

c) The bright area variable R_{BRIGHT} is calculated according to formula (B.14), and the bright area variable L_{BRIGHT} is calculated according to formula (B.15).

$$R_{\mathrm{BRIGHT}} = \frac{N_{\mathrm{BRIGHT}}}{N_{\mathrm{frame}}} \quad \text{(B.14)}$$

$$L_{\mathrm{BRIGHT}} = \mathrm{Clip3}\left(0.08, 1.0, \frac{max\_lum - targeted\_lum}{max\_lum}\right) \quad \text{(B.15)}$$

In formula (B.14) and formula (B.15), N_{BRIGHT} is the number of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ within the range of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}] \ge$ targeted_lum. The *targeted_lum* is calculated according to formula (B.16), and the *max_lum* is calculated according to formula (B.17).

$$targeted\_lum = targeted\_system\_display\_maximum\_luminance \quad \text{(B.16)}$$

The *targeted_system_display_maximum_luminance* is the reference target display luminance during production.

$$max\_lum = \begin{cases} MaxRefDisplay & MAX1 > MaxRefDisplay \\ MAX1 & MIN \le MAX1 \le MaxRefDisplay \\ MIN & MAX1 < MIN \end{cases} \quad \text{(B.17)}$$

$M = 0.5081$, $MaxRefDisplay = \mathrm{PQ\_EOTF}^{-1}(4000)$, and $MAX1 = 0.2 \times (maximum\_maxrgb\_pq \div 4095) + 0.8 \times (average\_maxrgb\_pq \div 4095) + 0.4 \times (variance\_maxrgb\_pq \div 4095)$. For calculation of *maximum_maxrgb_pq*, *average_maxrgb_pq*, and *variance_maxrgb_pq*, refer to sections B.2, B.3, and B.4.

d) If the current frame is a scene switching frame, the generation process of the base curve parameter metadata is as follows: Otherwise, the same base curve parameter metadata generation process as that of the previous frame is used.

1) If the source video is a PQ video:

If $R_{DARK} \geq q1 \times L_{DARK}$, $R_{BRIGHT} \geq w1 \times L_{BRIGHT}$, $q1 = 0.5$, and $w1 = 0.5$, the base curve parameter metadata is generated by calling section B.5.2.

If $R_{BRIGHT} \geq w2 \times N_{BRIGHT}$ and $w2 = 1.75$, the base curve parameter metadata is generated by calling section B.5.5.

If $R_{DARK} \geq q2 \times N_{DARK}$ and $q2 = 4.0$, the base curve parameter metadata is generated by calling section B.5.4.

Otherwise, the base curve parameter metadata is generated by calling section B.5.3.

2) If the source video is an HLG video:

If $R_{DARK} \geq q1 \times L_{DARK}$, $R_{BRIGHT} \geq w1 \times L_{BRIGHT}$, $q1 = 0.5$, and $w1 = 0.5$, the base curve parameter metadata is generated by calling section B.5.2.

If $R_{BRIGHT} \geq w2 \times N_{BRIGHT}$ and $w2 = 1.75$, the base curve parameter metadata is generated by calling section B.5.5.

If $R_{DARK} \geq q2 \times N_{DARK}$ and $q2 = 4.0$, the base curve parameter metadata is generated by calling section B.5.4.

Otherwise, the base curve parameter metadata is generated by calling section B.5.6.

B.5.2 Base curve parameter metadata generation process 1

Input: an RGB pixel buffer $f[N_{frame}][3]$.

Output: *base_param_m_p*[*i*][*w*], *base_param_m_m*[*i*][*w*], *base_param_m_n*[*i*][*w*], *base_param_m_a*[*i*][*w*], *base_param_m_b*[*i*][*w*], *base_param_K*1[*i*][*w*], *base_param_K*2[*i*][*w*], and *base_param_K*3[*i*][*w*].

The generation process is as follows:

a) *base_param_m_m*[*i*][*w*] = 24, *base_param_m_n*[*i*][*w*] = 10, *base_param_K*1[*i*][*w*] = 1, *base_param_K*2[*i*][*w*] = 1, *base_param_K*3[*i*][*w*] = 1, and *base_param_m_b*[*i*][*w*] = 0.

b) $L3$ and $N3$ are calculated according to formula (B.18) and formula (B.19).

$$L3 = \frac{\sum_{i=0}^{N_{frame}-1} q(i)}{Num} \quad \text{(B.18)}$$

$$N3 = L3 \quad \text{(B.19)}$$

In formula (B.18), $q(i)$ is obtained according to formula (B.20).

$$q(i) = \begin{cases} f_{MAX}[i] & 0.15 \leq f_{MAX}[i] \leq 0.35 \\ 0 & \text{other} \end{cases} \quad \text{(B.20)}$$

Num is the number of $f_{MAX}[N_{frame}]$ within the range of $0.15 \leq f_{MAX}[N_{frame}] \leq 0.35$.

c) *HISA_Length*[0], *HISA_Length*[1], *HISA_Length*[2], *HISA_Num*[0], *HISA_Num*[1], and *HISA_Num*[2] are calculated.

Half_Num is calculated according to formula (B.21).

$$Half\_Num = N(defusingLight) - N(midLight) \quad \text{(B.21)}$$

N(x) indicates the number of pixels within the range of $f_{MAX}[N_{frame}] < x$, and *defusingLight* is obtained according to formula (B.22).

$$defusingLight = 0.35 + (max\_lum - 0.35) \times Ratio \quad \text{(B.22)}$$

In formula (B.21) and formula (B.22), *midLight* = 0.35, *Ratio* = $\frac{2}{3}$, and *max_lum* is obtained according to formula (B.23).

$$max\_lum = \begin{cases} MaxRefDisplay & MAX1 > MaxRefDisplay \\ MAX1 & MIN \le MAX1 \le MaxRefDisplay \\ MIN & MAX1 < MIN \end{cases} \quad \text{(B.23)}$$

MIN = 0.5081, *MaxRefDisplay* = PQ_EOTF^{-1}(4000), and *MA* 1 = 0.2 × (*maximum_maxrgb_pq* ÷ 4095) + 0.8 × (*average_maxrgb_pq* ÷ 4095) + 0.4 × (*variance_maxrgb_pq* ÷ 4095). For calculation of *maximum_maxrgb_pq*, *average_maxrgb_pq*, and *variance_maxrgb_pq*, refer to sections B.2, B.3, and B.4.

When 0.15 to *max_lum* is evenly divided into six segments, the length *HISA_Length*[0] of each segment and the number *HISA_Num*[0] corresponding to *HISA_Length*[0] are calculated according to formula (B.24). When 0.15 to *max_lum* is evenly divided into three segments, the length *HISA_Length*[1] of each segment and the number *HISA_Num*[1] corresponding to *HISA_Length*[1] are calculated according to formula (B.25). When 0.15 to *max_lum* is evenly divided into two segments, the length *HISA_Length*[2] of each segment and the number *HISA_Num*[2] corresponding to *HISA_Length*[2] are calculated according to formula (B.26).

$$HISA\_Length[0] = \frac{(max\_lum - 0.15)}{6} \ ; \ HISA\_Num[0] = \mathrm{N}(HISA\_Length[0]) \quad \text{(B.24)}$$

$$HISA\_Length[1] = \frac{(max\_lum - 0.15)}{3} \ ; \ HISA\_Num[1] = \mathrm{N}(HISA\_Length[1]) \quad \text{(B.25)}$$

$$HISA\_Length[2] = \frac{(max\_lum - 0.15)}{2} \ ; \ HISA\_Num[2] = \mathrm{N}(HISA\_Length[2]) \quad \text{(B.26)}$$

d) *M*1 and *N*1 are calculated: The average value *M*1 within the range from *midLight* to *defusingLight* is calculated according to formula (B.27).

$$M1 = \frac{\sum_{i=0}^{N_{frame}-1} q(i)}{Num_1} \quad \text{(B.27)}$$

$q(i)$ is obtained according to formula (B.28).

$$q(i) = \begin{cases} f_{MAX}[i] & midLight \le f_{MAX}[i] \le defusingLight \\ 0 & \text{other} \end{cases} \quad \text{(B.28)}$$

Num_1 is the number of $f_{MAX}[N_{frame}]$ within the range of $midLight \le f_{MAX}[N_{frame}] \le defusingLight$, and *midLight* = 0.35.

If *HISA_Num*[0] > *Half_Num*, *HISA_Num*[1] > *Half_Num*, or *HISA_Num*[2] > *Half_Num*, the average value *N*1 from *midLight* to *defusingLight* is calculated according to formula (B.29).

$$N1 = \frac{\sum_{i=0}^{N_{frame}-1} q1(i)}{Num_1} \quad \text{(B.29)}$$

Num_1 is the number of $f_{MAX}[N_{frame}]$ within the range of $midLight \le f_{MAX}[N_{frame}] \le defusingLight$, and $q1(i)$ is obtained according to formula (B.30).

$$q1(i) = \begin{cases} f_{\mathrm{MAX}}[i] & midLight \le f_{\mathrm{MAX}}[i] \le defusingLight \\ 0 & \text{other} \end{cases} \quad \text{(B.30)}$$

$q1(i)$ is updated:

If $q1(i) \ge targeted\_lum$, $q1(i) = targeted\_lum$,

$targeted\_lum = targeted\_system\_display\_maximum\_luminance$, and $midLight = 0.35$.

If $HISA\_Num[0] \le Half\_Num$, $HISA\_Num[1] \le Half\_Num$, and $HISA\_Num[2] \le Half\_Num$, for calculation of $N1$, refer to formula (B.31).

$$N1 = \frac{\sum_{i=0}^{N_{\mathrm{frame}}-1} q2(i)}{Num_2} \quad \text{(B.31)}$$

$q2(i)$ is obtained according to formula (B.32).

$$q2(i) = \begin{cases} f_{\mathrm{MAX}}[i] & midLight \le f_{\mathrm{MAX}}[i] \le defusingLightH \\ 0 & \text{other} \end{cases} \quad \text{(B.32)}$$

$q2(i)$ is updated: If $q2(i) \ge targeted\_lum$, $q2(i) = targeted\_lum$.

Num_2 is the number of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ within the range of $midLight \le f_{\mathrm{MAX}}[N_{\mathrm{frame}}] \le defusingLightH$. $midLight = 0.35$. For calculation of $defusingLightH$, refer to formula (B.33).

$$defusingLightH = 0.35 + (max\_lum - 0.35) \times RatioH \quad \text{(B.33)}$$

$RatioH = \frac{5}{6}$.

e) $ratio[0]$, $ratio[1]$, and $ratio[2]$ are calculated.

The histogram $His[i]$ of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ is calculated, where $0 \le i < 1024$:

```
for(i=0; i<1024; i++)
{
    His[Floor(f_MAX[i] × 1023)]++;
}
```

$max\_content$ is calculated:

```
HisThrehold = N_frame×4÷(1024×10)
for(i=1024; i>=622; i-=4)
{
    max_content = i;
    if((His[i]+ His[i − 1]+ His[i − 2]+ His[i − 3])> HisThrehold) {
      break;
      }
}
```

$max\_content = max\_content \div 1024$;

$Num_3 = \mathrm{N}(L3)$, $Num_4 = \mathrm{N}(M1) - \mathrm{N}(L3)$, $Num_5 = \mathrm{N}(max\_content) - \mathrm{N}(M1)$;

$NumAll = Num_3 + Num_4 + Num_5$;

$ratio[0] = (targeted\_lum \div max\_content) \times (Num_3 \div (L3 \times NumAll \div max\_content))$;

$ratio[1] = (targeted\_lum \div max\_content) \times (Num_4 \div ((M1 - L3) \times NumAll \div max\_content))$;

$$ratio[2] = (\, targeted\_lum \div max\_content \,) \times (\, Num_5 \div ((\, max\_content - M1 \,) \times NumAll \div max\_lum));$$

f) *ratio*[0] and *ratio*[1] are updated.

$$MaxRatio = \mathrm{Max}(\mathrm{Max}(ratio[0], ratio[1]), ratio[2]);$$
$$adjust = (\,1 - (\, targeted\_lum \div max\_content \,)\,) \div (\, MaxRatio - (\, targeted\_lum \div max\_content));$$
$$adjust = \mathrm{Clip3}(0,1,adjust);$$
$$ratio[0] = (\, ratio[0] - (\, targeted\_lum \div max\_content \,)) \times adjust + (\, targeted\_lum \div max\_content);$$
$$ratio[1] = (\, ratio[1] - (\, targeted\_lum \div max\_content \,)) \times adjust + (\, targeted\_lum \div max\_content);$$

g) *N*3 and *N*1 are updated.

$$N3 = L3 \times ratio[0];$$
$$N1 = (M1 - L3) \times ratio[1] + N3;$$

h) *m_p* and *m_a* are calculated:

Equations are obtained based on (*M*1, *N*1) and (*L*3, *N*3). Refer to formula (B.34) and formula (B.35):

$$m\_a \times \left(\frac{m\_p \times M1^{m\_n}}{(m\_p - 1) \times M1^{m\_n} + 1}\right)^{m\_m} + m\_b = N1 \quad \text{(B.34)}$$

$$m\_a \times \left(\frac{m\_p \times L3^{m\_n}}{(m\_p - 1) \times L3^{m\_n} + 1}\right)^{m\_m} + m\_b = N3 \quad \text{(B.35)}$$

In formula (B.34) and formula (B.35), *m_m* = 2.4, *m_n* = 1.0, and *m_b* = 0.0.
m_p and *m_a* are obtained by solving the equations. Refer to formula (B.36).

$$m\_p = 1 + \left(\left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}} \times L3 - M1\right) \div \left(M1 \times L3 \times \left(1 - \left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}}\right)\right)$$
$$m\_a = \frac{N1}{\left(m\_p \times M1 \div \left((m\_p - 1) \times M1 + 1\right)\right)^{m\_m}} \quad \text{(B.36)}$$

i) *m_p* and *m_a* are updated.

The variable $f_{\mathrm{MAX\_997}}$ is calculated according to formula (B.37).

$$\frac{\mathrm{N}(f_{\mathrm{MAX\_997}})}{N_{\mathrm{frame}}} = 0.997 \quad \text{(B.37)}$$

$N(x)$ indicates the number of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ within the range of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}] < x$.

The variable *Threshold* is calculated according to formula (B.38).

$$Threshold = \begin{cases} 12.0 & f_{\mathrm{MAX\_997}} \geq 0.75 \\ 12.28 - (f_{\mathrm{MAX_{997}}} - 0.7) \div (0.75 - 0.7) \times (12.28 - 12.0) & 0.7 \leq f_{\mathrm{MAX_{997}}} < 0.75 \\ 12.28 & f_{\mathrm{MAX\_997}} < 0.7 \end{cases} \quad \text{(B.38)}$$

If m_p + 10 x m_a > *Threshold* and m_p > 3.5, the following steps are repeated:

$m\_p \mathrel{-}= \triangle, \triangle = 0.1;$

m_a is obtained according to formula (B.39).

$$m\_a = \frac{N1}{\left(m\_p \times M1 \div ((m\_p - 1) \times M1 + 1)\right)^{m\_m}} \quad \text{(B.39)}$$

If $m\_p \le 3.5$, m_a = (*Threshold* – m_p) ÷ 10.0, the repetition is stopped, and step j) is performed.

Alternatively, if m_p + 10 × $m\_a \le$ *Threshold*, the repetition is stopped, and step j) is performed.

j) If m_p + 10 x m_a > *Threshold*, m_a = (*Threshold* – m_p) ÷ 10.0.

k) The metadata *base_param_m_p*[*i*][*w*] is calculated according to formula (B.40) and the metadata *base_param_m_a*[*i*][*w*] is calculated according to formula (B.41).

$$base\_param\_m\_p[i][w] = \mathrm{Floor}(m\_p \times 16383 \div 10.0) \quad \text{(B.40)}$$

$$base\_param\_m\_a[i][w] = \mathrm{Floor}(m\_a \times 1023) \quad \text{(B.41)}$$

B.5.3 Base curve parameter metadata generation process 2

Input: an RGB pixel buffer $f[N_{\mathrm{frame}}][3]$.

Output: *base_param_m_p*[*i*][*w*], *base_param_m_m*[*i*][*w*], *base_param_m_n*[*i*][*w*], *base_param_m_a*[*i*][*w*], *base_param_m_b*[*i*][*w*], *base_param_K1*[*i*][*w*], *base_param_K2*[*i*][*w*], and *base_param_K3*[*i*][*w*].

The generation process is as follows:

a) *base_param_m_m*[*i*][*w*] = 24, *base_param_m_n*[*i*][*w*] = 10, *base_param_K1*[*i*][*w*] = 1, *base_param_K2*[*i*][*w*] = 1, *base_param_K3*[*i*][*w*] = 1, and *base_param_m_b*[*i*][*w*] = 0.

b) *L*3 and *N*3 are calculated.

The histogram *His*[*i*] of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ is calculated, where $0 \le i < 1024$:

```
for(i=0; i <1024; i ++)
{
    His[Floor(f_MAX[i] × 1023)]++;
}
```

max_content is calculated:

```
HisThrehold = N_frame×4÷(1024×10)
for(i=1024; i>=622; i−=4)
{
    max_content= i;
if((His[i]+ His[i − 1]+ His[i − 2]+ His[i − 3])> HisThrehold
    ){
        break;
```

}

}

$max\_content = max\_content \div 1024;$

The luminance value $L3$ is calculated according to formula (B.42), and the luminance value $N3$ is calculated according to formula (B.43).

$$L3 = max\_content \quad \text{(B.42)}$$

$$N3 = targeted\_system\_display\_maximum\_luminance \quad \text{(B.43)}$$

c) The luminance value $L2$ is calculated according to formula (B.44), and the luminance value $N2$ is calculated according to formula (B.45).

$$L2 = \frac{\sum_{i=0}^{N_{\text{frame}}-1} f_{\text{MAX}}[i]}{N_{\text{frame}}} \quad \text{(B.44)}$$

$$N2 = \frac{\sum_{i=0}^{N_{\text{frame}}-1} q(i)}{N_{\text{frame}}} \quad \text{(B.45)}$$

$q(i)$ is obtained according to formula (B.46).

$$q(i) = \begin{cases} N3 & f_{\text{MAX}}[i] \geq N3 \\ f_{\text{MAX}}[i] & \text{other} \end{cases} \quad \text{(B.46)}$$

d) The luminance value $L1$ is calculated according to formula (B.47), and the luminance value $F1$ is calculated according to formula (B.48).

$$L1 = Perceprual\_1nit \quad \text{(B.47)}$$

For calculation of *Perceprual*_1*nit*, refer to section B.5.7.

$$F1 = \text{PQ\_EOTF}^{-1}(1) \quad \text{(B.48)}$$

e) $M1$ and $N1$ are calculated:

If $L2 < \text{PQ\_EOTF}^{-1}(DARK)$ or $N2 < \text{PQ\_EOTF}^{-1}(DARK)$, $M1 = N1$, and $N1 = F1$. Otherwise, $M1 = L2$, and $N1 = N2$.

f) m_p and m_a are calculated:

Equations are obtained based on ($M1$, $N1$) and ($L3$, $N3$). Refer to formula (B.49) and formula (B.50).

$$m\_a \times \left(\frac{m\_p \times M1^{m\_n}}{(m\_p - 1) \times M1^{m\_n} + 1}\right)^{m\_m} + m\_b = N1 \quad \text{(B.49)}$$

$$m\_a \times \left(\frac{m\_p \times L3^{m\_n}}{(m\_p - 1) \times L3^{m\_n} + 1}\right)^{m\_m} + m\_b = N3 \quad \text{(B.50)}$$

$m\_m = 2.4$, $m\_n = 1.0$, and $m\_b = 0.0$.
m_p and m_a are obtained by solving the equations. Refer to formula (B.51).

$$m\_p = 1 + \left(\left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}} \times L3 - M1\right) \div \left(M1 \times L3 \times \left(1 - \left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}}\right)\right) \quad \text{(B.51)}$$

$$m\_a = \frac{N1}{\left(m\_p \times M1 \div ((m\_p - 1) \times M1 + 1)\right)^{m\_m}}$$

g) m_p and m_a are updated:

If m_p + 10 × m_a > *Threshold* and m_p > 3.5, where *Threshold* is calculated according to formula (B.37) and formula (B.38), the following steps are repeated:

$m\_p - \Delta$, Δ=0.1;

m_a is obtained according to formula (B.52).

$$m\_a = \frac{N1}{\left(m\_p \times M1 \div ((m\_p - 1) \times M1 + 1)\right)^{m\_m}} \quad \text{(B.52)}$$

If m_p ≤ 3.5, m_a = (*Threshold* – m_p) ÷ 10.0, the repetition is stopped, and step h) is performed.
Alternatively, if m_p + 10 × m_a ≤ *Threshold*, the repetition is stopped, and step h) is performed.

h) If m_p + 10 × m_a > *Threshold*, m_a is calculated according to formula (B.53).

$$m\_a = (Threshold - m_p) \div 10.0 \quad \text{(B.53)}$$

i) The metadata *base_param_m_p*[*i*][*w*] is calculated according to formula (B.54); the metadata *base_param_m_a*[*i*][*w*] is calculated according to formula (B.55).

$$base\_param\_m\_p[i][w] = \text{Floor}(m\_p \times 16383 \div 10.0) \quad \text{(B.54)}$$

$$base\_param\_m\_a[i][w] = \text{Floor}(m\_a \times 1023) \quad \text{(B.55)}$$

B.5.4 Base curve parameter metadata generation process 3

Input: an RGB pixel buffer $f[N_{frame}][3]$.

Output: *base_param_m_p*[*i*][*w*], *base_param_m_m*[*i*][*w*], *base_param_m_n*[*i*][*w*], *base_param_m_a*[*i*][*w*], *base_param_m_b*[*i*][*w*], *base_param_K*1[*i*][*w*], *base_param_K*2[*i*][*w*], and *base_param_K*3[*i*][*w*].

The generation process is as follows:

a) *base_param_m_m*[*i*][*w*] = 24, *base_param_m_n*[*i*][*w*] = 10, *base_param_K*1[*i*][*w*] = 1, *base_param_K*2[*i*][*w*] = 1, *base_param_K*3[*i*][*w*] = 1, *base_param_m_b*[*i*][*w*] = 0, and *base_param_m_a*[*i*][*w*] = *Floor*(*targeted_system_display_maximum_luminance* × 1023).

b) For calculation of the ratio *v* of *T*p' to *T*p in the total pixels, refer to formula (B.56).

$$v = \text{R}(T\text{p}') - \text{R}(T\text{p}) \quad \text{(B.56)}$$

In formula (B.56), R(x) indicates the ratio of the number of $f_{MAX}[N_{frame}]$ within the range of $f_{MAX}[N_{frame}] < x$ to the total number of pixels. For calculation of Tp, refer to formula (B.57). For calculation of Tp', refer to formula (B.58). For calculation of *max_lum* in formula (B.58), refer to formula (B.59).

$$Tp = \mathrm{PQ\_EOTF}^{-1}(1) \quad \text{(B.57)}$$

$$Tp' = T_p \times \frac{max\_lum}{targeted\_system\_display\_maximum\_luminance} \quad \text{(B.58)}$$

$$max\_lum = \begin{cases} MaxRefDisplay & MAX1 > MaxRefDisplay \\ MAX1 & MIN \leq MAX1 \leq MaxRefDisplay \\ MIN & MAX1 < MIN \end{cases} \quad \text{(B.59)}$$

MIN = 0.5081, $MaxRefDisplay$ = PQ_EOTF^{-1}(4000), and $MAX1$ = 0.2 × (*maximum_maxrgb_pq* ÷ 4095) + 0.8 × (*average_maxrgb_pq* ÷ 4095) + 0.4 × (*variance_maxrgb_pq* ÷ 4095). For calculation of *maximum_maxrgb_pq*, *average_maxrgb_pq*, and *variance_maxrgb_pq*, refer to sections B.2, B.3, and B.4.

c) The base curve parameter *m_p* is calculated according to formula (B.60), and the metadata *base_param_m_p*[*i*][*w*] is calculated according to formula (B.61).

$$m\_p = c \times v + d \quad \text{(B.60)}$$

$$base\_param\_m\_p[i][w] = \mathrm{Floor}(m\_p \times 16383 \div 10.0) \quad \text{(B.61)}$$

c and *d* can be changed frame by frame. The recommended values are as follows: *c* = 7 and *d* = 3.

B.5.5 Base curve parameter metadata generation process 4

Input: an RGB pixel buffer $f[N_{frame}][3]$.

Output: *base_param_m_p*[*i*][*w*], *base_param_m_m*[*i*][*w*], *base_param_m_n*[*i*][*w*], *base_param_m_a*[*i*][*w*], *base_param_m_b*[*i*][*w*], *base_param_K*1[*i*][*w*], *base_param_K*2[*i*][*w*], and *base_param_K*3[*i*][*w*].

The operations are as follows:

a) *base_param_m_m*[*i*][*w*] = 10, *base_param_m_n*[*i*][*w*] = 4, *base_param_K*1[*i*][*w*] = 1, *base_param_K*2[*i*][*w*] = 1, *base_param_K*3[*i*][*w*] = 1, and *base_param_m_b*[*i*][*w*] = 0.

b) *L*3 and *N*3 are calculated:

The histogram *His*[*i*] of $f_{MAX}[N_{frame}]$ is calculated, where $0 \leq i < 1024$:

```
for(i=0; i<1024; i++)
{
    His[Floor(f_MAX[i] × 1023)]++;
}
```

max_content is calculated:

```
HisThrehold = N_frame × 4 ÷(1024×10)
for(i=1024; i>=622; i−=4)
{
```

$max\_content = i;$

$\text{if}((His[i] + His[i-1] + His[i-2] + His[i-3]) > HisThrehold$

){

break;

}

}

$max\_content = max\_content \div 1024;$

The luminance value *L*3 is calculated according to formula (B.62), and the luminance value *N*3 is calculated according to formula (B.63).

$$L3 = max\_content \quad \text{(B.62)}$$

$$N3 = targeted\_system\_display\_maximum\_luminance \quad \text{(B.63)}$$

max_lum is obtained according to formula (B.64).

$$max\_lum = \begin{cases} MaxRefDisplay & MAX1 > MaxRefDisplay \\ MAX1 & MIN \le MAX1 \le MaxRefDisplay \\ MIN & MAX1 < MIN \end{cases} \quad \text{(B.64)}$$

MIN = 0.5081, *MaxRefDisplay* = PQ_EOTF^{-1}(4000), and *MAX*1 = 0.2 × (*maximum_maxrgb_pq* ÷ 4095) + 0.8 × (*average_m rgb_pq* ÷ 4095) + 0.4 × (*variance_maxrgb_pq* ÷ 4095). For calculation of *maximum_maxrgb_pq*, *average_maxrgb_pq*, and *variance_maxrgb_pq*, refer to sections B.2, B.3, and B.4.

c) The luminance value *L*2 is calculated according to formula (B.65), and the luminance value *N*3 is calculated according to formula (B.66).

$$L2 = \frac{\sum_{i=0}^{N_{\text{frame}}-1} f_{\text{MAX}}[i]}{N_{\text{frame}}} \quad \text{(B.65)}$$

$$N2 = \frac{\sum_{i=0}^{N_{\text{frame}}-1} q(i)}{N_{\text{frame}}} \quad \text{(B.66)}$$

q(*i*) is obtained according to formula (B.67).

$$q(i) = \begin{cases} N3 & f_{\text{MAX}}[i] \ge N3 \\ f_{\text{MAX}}[i] & \text{other} \end{cases} \quad \text{(B.67)}$$

d) The luminance value *L*1 is calculated according to formula (B.68), and the luminance value *F*1 is calculated according to formula (B.69).

$$L1 = Perceprual\_1nit \quad \text{(B.68)}$$

For calculation of *Perceprual_*1*nit*, refer to section B.5.7.

$$F1 = \text{PQ\_EOTF}^{-1}(1) \quad \text{(B.69)}$$

e) *M*1 and *N*1 are calculated:

If *L*2 < PQ_EOTF^{-1}(*DARK*) or *N*2 < PQ_EOTF^{-1}(*DARK*), *M*1 = *L*1, and *N*1 = *F*1. Otherwise, *M*1 = *L*2, and *N*1 = *N*2.

f) *base_param_m_p*[*i*][*w*] and *base_param_m_a*[*i*][*w*] are calculated.

Equations are obtained based on ($M1$, $N1$) and ($L3$, $N3$). Refer to formula (B.70) and formula (B.71).

$$m\_a \times \left(\frac{m\_p \times M1^{m\_n}}{(m\_p-1)\times M1^{m\_n}+1}\right)^{m\_m} + m\_b = N1 \quad \text{(B.70)}$$

$$m\_a \times \left(\frac{m\_p \times L3^{m\_n}}{(m\_p-1)\times L3^{m\_n}+1}\right)^{m\_m} + m\_b = N3 \quad \text{(B.71)}$$

m_m = 1.0, m_n = 0.4, and m_b = 0.0.
m_p and m_a are obtained by solving the equations. Refer to formula (B.72).

$$m\_p = 1 + \left(\left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}} \times L3 - M1\right) \div \left(M1 \times L3 \times \left(1 - \left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}}\right)\right)$$
$$m\_a = \frac{N1}{\left(m\_p \times M1 \div \left((m\_p-1)\times M1+1\right)\right)^{m\_m}} \quad \text{(B.72)}$$

The metadata *base_param_m_p*[*i*][*w*] is calculated according to formula (B.73); the metadata *base_param_m_a*[*i*][*w*] is calculated according to formula (B.74).

$$base\_param\_m\_p[i][w] = \mathrm{Floor}(m\_p \times 16383 \div 10.0) \quad \text{(B.73)}$$

$$base\_param\_m\_a[i][w] = \mathrm{Floor}(m\_a \times 1023) \quad \text{(B.74)}$$

B.5.6 Base curve parameter metadata generation process 5

Input: an RGB pixel buffer $f[N_{\mathrm{frame}}][3]$.

Output: *base_param_m_p*[*i*][*w*], *base_param_m_m*[*i*][*w*], *base_param_m_n*[*i*][*w*], *base_param_m_a*[*i*][*w*], *base_param_m_b*[*i*][*w*], *base_param_K*1[*i*][*w*], *base_param_K*2[*i*][*w*], and *base_param_K*3[*i*][*w*].

The generation process is as follows:

a) *base_param_m_m*[*i*][*w*] = 24, *base_param_m_n*[*i*][*w*] = 10, *base_param_K*1[*i*][*w*] = 1, *base_param_K*2[*i*][*w*] = 1, *base_param_K*3[*i*][*w*] = 1, and *base_param_m_b*[*i*][*w*] = 0.

b) $L3$ and $N3$ are calculated.

The histogram *His*[3][*i*] of $f[N_{\mathrm{frame}}][3]$ is calculated, where $0 \le i < 1024$:

```
for(i=0; i <1024; i ++)
{
    His[0][Floor(f[i][0] × 1023)]++;
    His[1][Floor(f[i][1] × 1023)]++;
    His[2][Floor(f[i][2] × 1023)]++;
}
```

max_content_RGB[0], *max_content_RGB*[1], and *max_content_RGB*[2] are calculated:

```
HisThrehold = N_frame × 4 ÷(1024×10)
for(i=1024; i>=622; i-=4)
{
```

```
for(k=0; k<3; k++)
{
    max_content_RGB[k]= i;
    if((His[k][i]+ His[k][i − 1]+ His[k][i − 2]+ His[k][i − 3])> HisThrehold ){
        count = 0 ;
         for(j = i; j < i + 5; j ++){
           if(His[k][j] > HisThrehold ÷ 3){
              count ++;
           } else {
              break;
           }
         }
    for(j = i – 1; j > i – 5; j −−){
     if(His[k][j] > HisThrehold ÷ 3){
        count ++;
     } else {
        break;
     } }
}
if(count >= 8){
  break;
 }
}
```

max_content is calculated:

$max\_content = \mathrm{Median}(max\_content\_RGB[0], max\_content\_RGB[1], max\_content\_RGB[2])$

$max\_content = max\_content \div 1024;$

The luminance value *L*3 is calculated according to formula (B.75), and the luminance value *N*3 is calculated according to formula (B.76).

$$L3 = max\_content \quad \text{(B.75)}$$

$$N3 = targeted\_system\_display\_maximum\_luminance \quad \text{(B.76)}$$

c) The luminance value *L*2 is calculated according to formula (B.77), and the luminance value *N*2 is calculated according to formula (B.78).

$$L2 = \frac{\sum_{i=0}^{N_{\mathrm{frame}}-1} f_{\mathrm{MAX}}[i]}{N_{\mathrm{frame}}} \quad \text{(B.77)}$$

$$N2 = \frac{\sum_{i=0}^{N_{\mathrm{frame}}-1} q(i)}{N_{\mathrm{frame}}} \quad \text{(B.78)}$$

q(*i*) is obtained according to formula (B.79).

$$q(i) = \begin{cases} N3 & f_{\mathrm{MAX}}[i] \geq N3 \\ f_{\mathrm{MAX}}[i] & \text{other} \end{cases} \quad \text{(B.79)}$$

d) The luminance value $L1$ is calculated according to formula (B.80), and the luminance value $F1$ is calculated according to formula (B.81).

$$L1 = Perceprual\_1nit \qquad (B.80)$$

For calculation of *Perceprual_1nit*, refer to section B.5.7.

$$F1 = \mathrm{PQ\_EOTF}^{-1}(1) \qquad (B.81)$$

e) $M1$ and $N1$ are calculated:

If $L2 < \mathrm{PQ\_EOTF}^{-1}(DARK)$ or $N2 < \mathrm{PQ\_EOTF}^{-1}(DARK)$, $M1 = N1$, and $N1 = F1$. Otherwise, $M1 = L2$, and $N1 = N2$.

f) m_p and m_a are calculated:

Equations are obtained based on ($M1$, $N1$) and ($L3$, $N3$). Refer to formula (B.82) and formula (B.83).

$$m\_a \times \left(\frac{m\_p \times M1^{m\_n}}{(m\_p-1) \times M1^{m\_n}+1}\right)^{m\_m} + m\_b = N1 \qquad (B.82)$$

$$m\_a \times \left(\frac{m\_p \times L3^{m\_n}}{(m\_p-1) \times L3^{m\_n}+1}\right)^{m\_m} + m\_b = N3 \qquad (B.83)$$

$m\_m = 2.4$, $m\_n = 1.0$, and $m\_b = 0.0$.
m_p and m_a are obtained by solving the equations. Refer to formula (B.84).

$$m\_p = 1 + \left(\left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}} \times L3 - M1\right) \div \left(M1 \times L3 \times \left(1 - \left(\frac{N1}{N3}\right)^{\frac{1}{m\_m}}\right)\right) \qquad (B.84)$$

$$m\_a = \frac{N1}{\left(m\_p \times M1 \div \left((m\_p-1) \times M1+1\right)\right)^{m\_m}}$$

g) m_p and m_a are updated.

If $m\_p + 10 \times m\_a > Threshold$ and $m\_p > 3.5$, where *Threshold* is calculated according to formula (B.37) and formula (B.38), the following steps are repeated:

$m\_p \mathrel{-}= \Delta$, $\Delta = 0.1$;

m_a is obtained according to formula (B.85).

$$m\_a = \frac{N1}{\left(m\_p \times M1 \div \left((m\_p-1) \times M1+1\right)\right)^{m\_m}} \qquad (B.85)$$

If $m\_p \le 3.5$, $m\_a = (Threshold - m\_p) \div 10.0$, the repetition is stopped, and step h) is performed.

Alternatively, if $m\_p + 10 \times m\_a \le Threshold$, the repetition is stopped, and step h) is performed.

h) If $m\_p + 10 \times m\_a > Threshold$, m_a is calculated according to formula (B.86).

$$m\_a = (Threshold - m\_p) \div 10.0) \qquad (B.86)$$

i) *m_a* and *m_p* are updated:

$numTH1TH3 = 0;$

$\text{for}(i = 0; i < 0.225 \times 1025; i{+}{+})\ \{$

$\quad numTH1TH3 \mathrel{+}= his[i];$

$\}$

$darkratio = numTH1TH3 \div N_{\text{frame}};$

$mprange = m\_p > 5.5\ ?\ 2.0 : (m\_p - 3.5);$

$marange = m\_p > 5.5\ ?\ 0.1 : (mprange \div 2.0 \times 0.1);$

$\text{if}(darkratio > 0.05\ \&\&\ darkratio <= 0.4)\ \{$

$\quad m\_p \mathrel{-}= (darkratio - 0.4) \div (0.05 - 0.4) \times mprange;$

$\quad m\_a \mathrel{+}= (darkratio - 0.4) \div (0.05 - 0.4) \times marange;$

$\}$

j) The metadata *base_param_m_p*[*i*][*w*] is calculated according to formula (B.87); the metadata *base_param_m_a*[*i*][*w*] is calculated according to formula (B.88).

$$base\_param\_m\_p[i][w] = \text{Floor}(m\_p \times 16383 \div 10.0) \quad \text{(B.87)}$$

$$base\_param\_m\_a[i][w] = \text{Floor}(m\_a \times 1023) \quad \text{(B.88)}$$

B.5.7 Perceptual_1nit calculation method

The *Perceptual_1nit* calculation method is as follows:

a) The luminance value *Lp* is calculated according to formula (B.89).

$$\text{N}(Lp) = (\text{N}(L0) - \text{N}(1)) \times Rate + \text{N}(1) \quad \text{(B.89)}$$

N(x) indicates the number of $f_{\text{MAX}}[N_{\text{frame}}]$ within the range of $f_{\text{MAX}}[N_{\text{frame}}] < x$, $L0$ = 5, and *Rate* = 0.3.

b) The luminance value *Perceptual_1nit* is calculated according to formula (B.90).

$$Perceptual\_1nit = \text{PQ\_EOTF}^{-1}(Lp) \quad \text{(B.90)}$$

B.6 Cubic Spline Parameter Metadata Generation Process

B.6.1 Overview

The cubic spline parameter metadata generation process is as follows:

a) If the source video is a PQ video, the cubic spline parameter metadata is generated by calling section B.6.2.

b) If the source video is an HLG video, the cubic spline parameter metadata is generated by calling section B.6.3.

B.6.2 Cubic spline parameter metadata generation process 1

Input: an RGB pixel buffer $f[N_{frame}][3]$.

Output: 3*Spline_TH_enable*[0][*i*][*w*], *3Spline_TH_enable_Delta*1[0][*i*][*w*], 3*Spline_TH_enable_Delta*2[0][*i*][*w*], 3*Spline_enable_strength*[0][*i*][*w*], 3*Spline_TH_enable*[1][*i*][*w*], 3*Spline_TH_enable_Delta*1[1][*i*][*w*], 3*Spline_TH_enable_Delta*2[1][*i*][*w*], and 3*Spline_enable_strength*[1][*i*][*w*].

The generation process is as follows:

a) *TH*1[1] = 0.15, and *TH*3[1] = 0.35.

b) The maximum value ($f_{MAX}[index]$) of *f*[*index*][0], *f*[*index*][1], and *f*[*index*][2] is calculated according to formula (B.91).

$$f_{\text{MAX}}[index] = \text{Max}(\text{Max}(f[index][0], f[index][1]), f[index][2]) \quad \text{(B.91)}$$

c) The second interpolation point *TH*2[1] is calculated according to formula (B.92).

$$TH2[1] = \frac{\sum_{i=0}^{N_{\text{frame}}-1} q(i)}{Num} \quad \text{(B.92)}$$

q(*i*) is obtained according to formula (B.93).

$$q(i) = \begin{cases} f_{\text{MAX}}[i] & TH1[1] \le f_{\text{MAX}}[i] \le TH3[1] \\ 0 & \text{other} \end{cases} \quad \text{(B.93)}$$

Num is the number of $f_{MAX}[N_{frame}]$ within the range of $TH1[1] \le f_{MAX}[N_{frame}] \le TH3[1]$.

d) The metadata 3*Spline_TH_enable*[0][*i*][*w*] is calculated according to formula (B.94); the metadata 3*Spline_TH_enable_Delta*1[0][*i*][*w*] is calculated according to formula (B.95); the metadata 3*Spline_TH_enable_Delta*2[0][*i*][*w*] is calculated according to formula (B.96).

$$3Spline\_TH\_enable[0][i][w] = \text{Floor}(TH1[1] \times 4095) \quad \text{(B.94)}$$

$$3Spline\_TH\_enable\_Delta1[0][i][w] = \text{Floor}((TH2[1] - TH1[1]) \times 4.0 \times 1023) \quad \text{(B.95)}$$

$$3Spline\_TH\_enable\_Delta2[0][i][w] = \text{Floor}((TH3[1] - TH2[1]) \times 4.0 \times 1023) \quad \text{(B.96)}$$

e) The number Num_{11} is calculated according to formula (B.97), and the number Num_{12} is calculated according to formula (B.98).

$$Num_{11} = \text{N}(TH2[1]) - \text{N}(TH1[1]) \quad \text{(B.97)}$$

$$Num_{12} = \text{N}(TH3[1]) - \text{N}(TH2[1]) \quad \text{(B.98)}$$

N(*x*) indicates the number of $f_{MAX}[N_{frame}]$ within the range of $f_{MAX}[N_{frame}] < x$.

f) 3*Spline_enable_strength*[0][*i*][*w*] is calculated.

$Spline\_Strength_1 = 0$;

If $Num_{11} < Num_{12}$, $Splien\_Strength_1$ += Δ

If 2 x $Num_{11} < Num_{12}$, $Splien\_Strength_1$ += 2 x Δ

Otherwise, $Splien\_Strength_1$ is not updated

$\Delta = -0.1$.

3*Spline_enable_strength*[0][*i*][*w*] is calculated according to formula (B.99).

$$3Spline\_enable\_strength[0][i][w] = \mathrm{Floor}((Splien\_Strength_1 + 1.0) \times (255 \div 2)) \quad \text{(B.99)}$$

g) *TH*1[2] and *TH*3[2] are calculated.

The histogram *His*[*i*] of $f_{MAX}[N_{frame}]$ is calculated, where $0 \le i < 1024$:

```
for(i=0; i<1024; i++)
{
    His[Floor(fMAX[i] × 1023)]++;
}
```

max_content is calculated:

```
HisThrehold = Nframe × 4 ÷(1024×10)
for(i = 1024; i >=622; i −=4)
{
    max_content = i ;
if((His[i]+ His[i − 1]+ His[i − 2]+ His[i − 3])> HisThrehold
    ) {
    break;
  }
}
max_content = max_content ÷1024;
```

The first interpolation point *TH*1[2] is calculated according to formula (B.100), and the third interpolation point is calculated according to formula (B.101).

$$TH1[2] = TH3[1] + ((max\_content - TH3[1]) \div U) \times (U - 2) \quad \text{(B.100)}$$

$$TH3[2] = max\_content \quad \text{(B.101)}$$

$U = 6$.

h) The intermediate variable value *highRatio* is calculated according to formula (B.102), and the intermediate variable value *wholeRatio* is calculated according to formula (B.103).

$$highRatio = \mathrm{R}(TH3[2]) - \mathrm{R}(TH1[2]) \quad \text{(B.102)}$$

$R(x)$ indicates the number of $f_{MAX}[N_{frame}]$ within the range of $f_{MAX}[N_{frame}] < x$ to the total number of pixels.

$$wholeRatio = (TH3[2] - TH1[2]) \div max\_content \quad \text{(B.103)}$$

i) *TH*1[2] is updated through calculation according to formula (B.104).

$$TH1[2] = TH1[2] - \mathrm{pow}\,(highRatio \div wholeRatio, 0.5) \times ((max\_content - TH3[1]) \div U) \quad \text{(B.104)}$$

The metadata 3*Spline_TH_enable*[1][*i*][*w*] is calculated according to formula (B.105), the metadata 3*Spline_TH_enable_Delta*1[1][*i*][*w*] is calculated according to formula (B.106), and the metadata 3*Spline_TH_enable_Delta*2[1][*i*][*w*] is calculated according to formula (B.107).

$$3Spline\_TH\_enable[1][i][w] = \text{Floor}(TH1[2] \times 4095) \quad (B.105)$$

$$3Spline\_TH\_enable\_Delta1[1][i][w] = \text{Floor}((TH2[2] - TH1[2]) \times 4.0 \times 1023) \quad (B.106)$$

$$3Spline\_TH\_enable\_Delta2[1][i][w] = \text{Floor}((TH3[2] - TH2[2]) \times 4.0 \times 1023) \quad (B.107)$$

j) [*TH*1[2], *TH*3[2]] is divided into eight subintervals of an equal size, and a subinterval *n_min* that includes the minimum number of pixels in subintervals 2, 3, 4, 5, and 6 is calculated.

*TH*2[2] is calculated according to formula (B.108).

$$TH2[2] = TH1[2] + (TH3[2] - TH1[2]) \times n\_min \div N + (TH3[2] - TH1[2]) \div (2 \times N) \quad (B.108)$$

k) The updated metadata 3*Spline_TH_enable_Delta*1[1][*i*][*w*] is calculated according to formula (B.109).

$$3Spline\_TH\_enable\_Delta1[1][i][w] = (TH2[2] - TH1[2]) \times 4.0 \times 1023 \quad (B.109)$$

l) The number Num_{21} is calculated according to formula (B.110), and the number Num_{22} is calculated according to formula (B.111).

$$Num_{21} = \text{N}(TH2[2]) - \text{N}(TH1[2]) \quad (B.110)$$

$$Num_{22} = \text{N}(TH3[2]) - \text{N}(TH2[2]) \quad (B.111)$$

N(*x*) indicates the number of $f_{MAX}[N_{frame}]$ within the range of $f_{MAX}[N_{frame}] < x$.

m) 3*Spline_enable_strength*[1][*i*][*w*] is calculated.

$Splien\_Strength_2 = 0$;

If $Num_{21} < Num_{22}$, $Splien\_Strength_2 \mathrel{+}= \Delta$;

If $2 \times Num_{21} < Num_{22}$, $Splien\_Strength_2 \mathrel{+}= 2 \times \Delta$;

$\Delta = -0.1$.

Otherwise, $Splien\_Strength_2$ is not updated.

The metadata 3*Spline_enable_strength*[1][*i*][*w*] is calculated according to formula (B.112).

$$3Spline\_enable\_Strength[1][i][w] = \text{Floor}((Spline\_Strength + 1.0) \times (255 \div 2)) \quad (B.112)$$

B.6.3 Cubic spline parameter metadata generation process 2

Input: an RGB pixel buffer $f[N_{frame}][3]$.

Output: 3*Spline_TH_enable*[0][*i*][*w*], *3Spline_TH_enable_Delta*1[0][*i*][*w*], 3*Spline_TH_enable_Delta*2[0][*i*][*w*], 3*Spline_enable_strength*[0][*i*][*w*], 3*Spline_TH_enable*[1][*i*][*w*], 3*Spline_TH_enable_Delta*1[1][*i*][*w*], 3*Spline_TH_enable_Delta*2[1][*i*][*w*], and 3*Spline_enable_strength*[1][*i*][*w*].

The generation process is as follows:

a) *TH*1[1] = 0.15, and *TH*3[1] = 0.35.

b) The maximum value ($f_{MAX}[index]$) of $f[index][0]$, $f[index][1]$, and $f[index][2]$ is calculated according to formula (B.113).

$$f_{MAX}[index] = \mathrm{Max}(\mathrm{Max}(f[index][0], f[index][1]), f[index][2]) \quad \text{(B.113)}$$

c) The second interpolation point $TH2[1]$ is calculated according to formula (B.114).

$$TH2[1] = \frac{\sum_{i=0}^{N_{\mathrm{frame}}-1} q(i)}{Num} \quad \text{(B.114)}$$

$q(i)$ is obtained according to formula (B.115).

$$q(i) = \begin{cases} f_{MAX}[i] & TH1[1] \le f_{MAX}[i] \le TH3[1] \\ 0 & \text{other} \end{cases} \quad \text{(B.115)}$$

Num is the number of $f_{MAX}[N_{\mathrm{frame}}]$ within the range of $TH1[1] \le f_{MAX}[N_{\mathrm{frame}}] \le TH3[1]$.

d) The metadata 3*Spline_TH_enable*[0][*i*][*w*] is calculated according to formula (B.116); the metadata 3*Spline_TH_enable_Delta*1[0][*i*][*w*] is calculated according to formula (B.117); the metadata 3*Spline_TH_enable_Delta*2[0][*i*][*w*] is calculated according to formula (B.118).

$$3Spline\_TH\_enable[0][i][w] = \mathrm{Floor}(TH1[1] \times 4095) \quad \text{(B.116)}$$

$$3Spline\_TH\_enable\_Delta1[0][i][w] = \mathrm{Floor}((TH2[1] - TH1[1]) \times 4.0 \times 1023) \quad \text{(B.117)}$$

$$3Spline\_TH\_enable\_Delta2[0][i][w] = \mathrm{Floor}((TH3[1] - TH2[1]) \times 4.0 \times 1023) \quad \text{(B.118)}$$

e) The number Num_{11} is calculated according to formula (B.119), and the number Num_{12} is calculated according to formula (B.120).

$$Num_{11} = \mathrm{N}(TH2[1]) - \mathrm{N}(TH1[1]) \quad \text{(B.119)}$$

$$Num_{12} = \mathrm{N}(TH3[1]) - \mathrm{N}(TH2[1]) \quad \text{(B.120)}$$

$N(x)$ indicates the number of $f_{MAX}[N_{\mathrm{frame}}]$ within the range of $f_{MAX}[N_{\mathrm{frame}}] < x$.

f) 3*Spline_enable_strength*[0][*i*][*w*] is calculated.

$Splien\_Strength_1 = 0$;

If $Num_{11} < Num_{12}$, $Splien\_Strength_1$ += Δ;

If 2 x $Num_{11} < Num_{12}$, $Splien\_Strength_1$ += 2 x Δ;

Otherwise, $Splien\_Strength_1$ is not updated.

$\Delta = -0.1$.

3*Spline_enable_strength*[0][*i*][*w*] is calculated according to formula (B.121).

$$3Spline\_enable\_strength[0][i][w] = \mathrm{Floor}((Splien\_Strength_1 + 1.0) \times (255 \div 2)) \quad \text{(B.121)}$$

g) $TH1[2]$ and $TH3[2]$ are calculated.

The histogram $His[i]$ of $f_{MAX}[N_{\mathrm{frame}}]$ is calculated, where $0 \le i < 1024$:

```
for(i=0; i<1024; i++)
{
```

```
    His[Floor(f_MAX[i] × 1023)]++;
}
```

max_content is calculated:

max_content_RGB[0], *max_content_RGB*[1], and *max_content_RGB*[2] are calculated according to step b) in section B5.6; *max_content_mid* = Median(*max_content_RGB*[0], *max_content_RGB*[1], *max_content_RGB*[2])

```
cutoff = max_content_mid;
sum=0;
 for(i = 0; i < 1024; i ++)
 {
    sum += His[i] ;
    if(sum >= 0.999×N_frame){
        his999 = i;
        break;
    }
    else if(sum >= 0.998×N_frame){
        his999 = i;
        his998 = i;
    }
    else if(sum >= 0.997×N_frame){
        his999 = i;
        his998 = i;
        his997 = i;
    }
}
if(cutoff < his997){
numexp = 0;
over997= 0;
over998 = 0;
for(i = cutoff; i <= maximum_maxrgb_pq÷4095×1024; i++)
{
  numexp += His[i];
}
for(i = his997; i <= maximum_maxrgb_pq÷4095×1024; i++)
{
  over997 += His[i];
}
for(i = his998; i <= maximum_maxrgb_pq÷4095×1024; i++)
{
  over998 += His[i];
}
```

$$
\begin{aligned}
&\text{if}((over997 \div numexp) >= 0.2 \ \&\& \ (over998 \div numexp) >= 0.2)\{\\
&\quad cutoff = 1.015 \times his999;\\
&\} \text{ else } \{\\
&\quad cutoff = his997;\\
&\}\\
&\}
\end{aligned}
$$

$max\_content = cutoff \div 1024;$

The first interpolation point $TH1[2]$ is calculated according to formula (B.122), and the third interpolation point is calculated according to formula (B.123).

$$TH1[2] = TH3[1] + ((max\_content - TH3[1]) \div U) \times (U - 2) \quad \text{(B.122)}$$

$$TH3[2] = max\_content \quad \text{(B.123)}$$

U = 6.

h) The intermediate variable value *highRatio* is calculated according to formula (B.124), and the intermediate variable value *wholeRatio* is calculated according to formula (B.125).

$$highRatio = \mathrm{R}(TH3[2]) - \mathrm{R}(TH1[2]) \quad \text{(B.124)}$$

$R(x)$ indicates the number of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ within the range of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}] < x$ to the total number of pixels.

$$wholeRatio = (TH3[2] - TH1[2]) \div max\_content \quad \text{(B.125)}$$

i) $TH1[2]$ is updated through calculation according to formula (B.126).

$$TH1[2] = TH1[2] - \mathrm{pow}\,(highRatio \div wholeRatio, 0.5) \times ((max\_content - TH3[1]) \div U) \quad \text{(B.126)}$$

The metadata 3*Spline_TH_enable*[1][*i*][*w*] is calculated according to formula (B.127); the metadata 3*Spline_TH_enable_Delta*1[1][*i*][*w*] is calculated based on formula (B.128); the metadata 3*Spline_TH_enable_Delta*2[1][*i*][*w*] is calculated according to formula (B.129).

$$3Spline\_TH\_enable[1][i][w] = \mathrm{Floor}(TH1[2] \times 4095) \quad \text{(B.127)}$$

$$3Spline\_TH\_enable\_Delta1[1][i][w] = \mathrm{Floor}((TH2[2] - TH1[2]) \times 4.0 \times 1023) \quad \text{(B.128)}$$

$$3Spline\_TH\_enable\_Delta2[1][i][w] = \mathrm{Floor}((TH3[2] - TH2[2]) \times 4.0 \times 1023) \quad \text{(B.129)}$$

j) [$TH1[2]$, $TH3[2]$] is divided into eight subintervals of an equal size, and a subinterval *n_min* that includes the minimum number of pixels in subintervals 2, 3, 4, 5, and 6 is calculated.

$TH2[2]$ is calculated according to formula (B.130).

$$TH2[2] = TH1[2] + (TH3[2] - TH1[2]) \times n\_min \div N + (TH3[2] - TH1[2]) \div (2 \times N) \quad \text{(B.130)}$$

k) The updated metadata 3*Spline_TH_enable_Delta*1[1][*i*][*w*] is calculated according to formula (B.131).

$$3Spline\_TH\_enable\_Delta1[1][i][w] = (TH2[2] - TH1[2]) \times 4.0 \times 1023 \quad \text{(B.131)}$$

l) The number Num_{21} is calculated according to formula (B.132), and the number Num_{22} is calculated according to formula (B.133).

$$Num_{21} = \mathrm{N}(TH2[2]) - \mathrm{N}(TH1[2]) \quad \text{(B.132)}$$

$$Num_{22} = \mathrm{N}(TH3[2]) - \mathrm{N}(TH2[2]) \quad \text{(B.133)}$$

$N(x)$ indicates the number of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}]$ within the range of $f_{\mathrm{MAX}}[N_{\mathrm{frame}}] < x$.

m) 3*Spline _enable_strength*[1][*i*][*w*] is calculated.

$Splien\_Strength_2 = 0$;

If $Num_{21} < Num_{22}$, $Splien\_Strength_2$ += *Δ;*

If 2 x $Num_{21} < Num_{22}$, $Splien\_Strength_2$ += 2 x *Δ;*

Δ = –0.1.

Otherwise, $Splien\_Strength_2$ is not updated.

The metadata 3*Spline_enable_strength*[1][*i*][*w*] is calculated according to formula (B.134).

$$3Spline\_enable\_Strength[1][i][w] = \mathrm{Floor}((Spline\_Strength + 1.0) \times (255 \div 2)) \quad \text{(B.134)}$$

B.7 Time-domain Filtering of Dynamic Metadata

A process of performing dynamic metadata time-domain filtering on the dynamic metadata extracted from the current frame includes the following steps:

a) A dynamic metadata queue *hdr_dynamic_metadata_fifo* is created. The queue length is M, and M is 32. *hdr_dynamic_metadata_fifo*[*hdr_dynamic_metadata_fifo_Num*] indicates the (*hdr_dynamic_metadata_fifo_Num*)-th piece of metadata in the queue, *hdr_dynamic_metadata_fifo_Num* is the number of pieces of valid data in the queue. The initial value is 0.

b) The current Nth frame of dynamic metadata *hdr_dynamic_metadata_org* is generated by calling sections B.2 to B.6. *N* is the frame sequence number and N ≥ 0.

c) If *N* is equal to 0 or the current frame is a scene switching frame, $hdr\_dynamic\_metadata\_fifo[0] = hdr\_dynamic\_metadata\_org$, $hdr\_dynamic\_metadata\_fifo\_Num = 1$. Otherwise:

If *hdr_dynamic_metadata_fifo_Num* is less than *M*:

$hdr\_dynamic\_metadata\_fifo[hdr\_dynamic\_metadata\_fifo\_Num] = hdr\_dynamic\_metadata\_org$; $hdr\_dynamic\_metadata\_fifo\_Num = hdr\_dynamic\_metadata\_fifo\_Num + 1$.

If *hdr_dynamic_metadata_fifo_Num* is equal to *M*:

```
for (n = 0; n < M−1; n ++) {
    hdr_dynamic_metadata_fifo[n + 1]= hdr_dynamic_metadata_fifo[n];
}
```

$hdr\_dynamic\_metadata\_fifo[M-1] = hdr\_dynamic\_metadata\_org$;

d) The metadata *hdr_dynamic_metadata_fliter* after time-domain filtering is output. Refer to formula (B.135).

$$hdr\_dynamic\_metadata\_filter = \frac{\sum_{i=0}^{hdr\_dynamic\_metadata\_fifo\_Num-1} hdr\_dynamic\_metadata\_fifo[i]}{hdr\_dynamic\_metadata\_fifo\_Num} \quad (B.135)$$

B.8 Time-domain Quality Intra-loop Adjustment and Feedback for Dynamic Metadata

A process of time-domain quality intra-loop adjustment and feedback for dynamic metadata is as follows:

a) The dynamic metadata queue *hdr_dynamic_metadata_fifo* is created by calling a) in section B.7.

b) Three subjective distortion queues *diff_tmhdr1_fifo*, *diff_tmhdr2_fifo,* and *diff_tmsdr_fifo* with the same length as *hdr_dynamic_metadata_fifo* are created.

c) The current Nth frame of dynamic metadata *hdr_dynamic_metadata_org* is generated by calling sections B.2 to B.6, and the *hdr_dynamic_metadata_org* is placed into the queue *hdr_dynamic_metadata_fifo* according to b) and c) in section B.7. The location of the *hdr_dynamic_metadata_org* in the queue is *Num*1.

d) The metadata conversion is performed on *hdr_dynamic_metadata_org* by calling chapter 9. The output frame $f_{TM1}(N)$ during display adaptation is obtained by calling chapter 10, where *MaxDisplayPQ* = $PQ\_EOTF^{-1}(1000)$, and *MinDisplayPQ* = 0. The output frame $f_{TM2}(N)$ during display adaptation is obtained by calling chapter 10, where *MaxDisplayPQ* = $PQ\_EOTF^{-1}(500)$, and *MinDisplayPQ* = 0. The output frame $f_{TMSDR}(N)$ during display adaptation is obtained by calling chapter 11, where *MaxDisplayPQ* = $PQ\_EOTF^{-1}(100)$, and *MinDisplayPQ* = 0.

e) The subjective distortion D_{TM1}, D_{TM2}, and D_{TMSDR} corresponding to $f_{TM1}(N)$, $f_{TM2}(N)$, and $f_{TMSDR}(N)$ are calculated according to the quality evaluation algorithm in *HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions*, and are placed into the queue *diff_tmhdr*1*_fifo*, *diff_tmhdr*2*_fifo* and *diff_tmsdr_fifo* according to c) and d) in section B.7. The position of D_{TM1}, D_{TM2}, and D_{TMSDR} in the queue is Num_2.

f) *n* is calculated:

```
Diff_min=1.0;
for (i = Num2; i >= 0; i --) {
    Diff_total=0.3×D_TM1[i]+0.4×D_TM2[i] +0.3×D_TMSDR[i] ;
    if(Diff_total < Diff_min){
        n = i ;
    }
    Diff_min = Min(Diff_total,Diff_min);
}
```

g) *m* is calculated:

$Diff_{\min}$=1.0;
for (i= Num_2; i >= 0; i --) {
    if(i ==n) {
    break;
    }
$Diff_{\text{total}}=0.3\times D_{\text{TM1}}[i] +0.4\times D_{\text{TM2}}[i] +0.3\times D_{\text{TMSDR}}[i]$;
if($Diff_{\text{total}}< Diff_{\min}$){
    $m=i$;
    }
$Diff_{\min} = \text{Min}(Diff_{\text{total}}, Diff_{\min})$;
}

h) *deltaC* is calculated:

If n is not equal to *Num*2, *deltaC* = (*hdr_dynamic_metadata_fifo*[*n*] + *hdr_dynamic_metadata_fifo*[*n* – 1]) ÷ 2 – *hdr_dynamic_metadata_org*.
Otherwise,
deltaC = 2 × *hdr_dynamic_metadata_fifo*[*n*] – *hdr_dynamic_metadata_fifo*[*m*] – *hdr_dynamic_metadata_org*.

i) If $D_{\text{TM1}} \le DT$ && $D_{\text{TM2}} \le DT$ && $D_{\text{TMSDR}} \le DT$, and a value of DT is 0.05, the dynamic metadata *hdr_dynamic_metadata* = *hdr_dynamic_metadata_org* is output.

j) If $D_{\text{TM1}} > DT$ | | $D_{\text{TM21}} > DT$ | | $D_{\text{TMSDR1}} > DT$ and the adjusted metadata *hdr_dynamic_metadata_modified* = *hdr_dynamic_metadata_org* + *deltaC*, *hdr_dynamic_metadata_modified* is placed in the *Num*1 position in the queue *hdr_dynamic_metadata_fifo*. Metadata conversion is performed on *hdr_dynamic_metadata_modified* by calling chapter 9. The output frame $f'_{\text{TM1}}(N)$ during display adaptation is obtained by calling chapter 10, where *MaxDisplayPQ* = $\text{PQ\_EOTF}^{-1}(1000)$ and *MinDisplayPQ* = 0. The output frame $f'_{\text{TM2}}(N)$ during display adaptation is obtained by calling chapter 10, where *MaxDisplayPQ* = $\text{PQ\_EOTF}^{-1}(500)$ and *MinDisplayPQ* = 0. The output frame $f'_{\text{TMSDR}}(N)$ during display adaptation is obtained by calling chapter 11, where *MaxDisplayPQ* = $\text{PQ\_EOTF}^{-1}(100)$ and *MinDisplayPQ* = 0. The subjective distortion D_{TM1}, D_{TM2}, and D_{TMSDR} of $f'_{\text{TM1}}(N)$, $f'_{\text{TM2}}(N)$, and $f'_{\text{TMSDR}}(N)$ are evaluated according to the quality evaluation algorithm, and are placed in position *Num*2 of the queue *diff_tmhdr1_fifo*, *diff_tmhdr2_fifo* and *diff_tmsdr_fifo*. The dynamic metadata *hdr_dynamic_metadata* = *hdr_dynamic_metadata_modified* is output.

Annex C
(Informative)
Encapsulation of Metadata in ITU-T H.265 Coded Stream

For encapsulation of metadata in an ITU-T H.265 coded stream, refer to ITU-T H.274. Static metadata and dynamic metadata are encapsulated in supplemental enhancement information (SEI).

The static metadata is encapsulated in the mastering_display_colour_volume() and content_light_level_info() of the SEI. Refer to Table C.1 and Table C.2. For the related syntax definition, refer to section 7.2.

Table C.1 Definition of HDR static metadata extension in H.265 coded stream

mastering_display_colour_volume(payloadSize) {	**Descriptor**
for(c = 0; c < 3; c++) {	
display_primaries_x[c]	u(16)
display_primaries_y[c]	u(16)
}	
white_point_x	u(16)
white_point_y	u(16)
max_display_mastering_luminance	u(32)
min_display_mastering_luminance	u(32)
}	

Table C.2 Definition 2 of HDR static metadata extension in H.265 coded stream

content_light_level_info(payloadSize) {	**Descriptor**
max_content_light_level	u(16)
max_pic_average_light_level	u(16)
}	

The dynamic metadata is encapsulated in user_data_registered_itu_t_t35(). Refer to Table C.3.

Table C.3 Definition of HDR dynamic metadata extension in H.265 coded stream

user_data_registered_itu_t_t35(payloadSize) {	**Descriptor**
itu_t_t35_country_code	b(8)
if(itu_t_t35_country_code != 0xFF){	

user_data_registered_itu_t_t35(payloadSize) {	**Descriptor**
i = 1	
}	
else {	
itu_t_t35_country_code_extension_byte	b(8)
i = 2	
}	
do {	
itu_t_t35_payload_byte	b(8)
i++	

Table C.3 (continued)

} while(i < payloadSize)	
}	

For the syntax and semantics of hdr_dynamic_metadata(), refer to sections 7.3 and 7.4. Other syntax and semantics are as follows:

—— The ITU-T T.35 country code itu_t_t35_country_code is an 8-bit unsigned integer. It identifies the country identification code specified in ITU-T T.35.

—— The ITU-T T.35 terminal manufacturer code itu_t_t35_country_code_extension_byte is an 8-bit unsigned integer. It identifies the country identification code extension specified in ITU-T T.35.

References

[1] ITU-T H.274 Versatile supplemental enhancement information messages for cod

[2] Rafal Mantiuk, Kil Joong Kim, Allan G.Rempel and Wolfgang Heidrich. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions,In: ACM Transactions on Graphics (Proc. of SIGGRAPH'11), 30(4), article no.40, 2011