HDR Video Technology Part 1

Metadata and Tone Mapping

（Version NO. 1.0）

Release Time
2022-02-10

UHD World Association (UWA)
T/UWA 005.1-2022

Contents

About This Document

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 Part 1 of UWA 005 *High Dynamic Range Video Technology*, which consists of the following parts:

Part 1: Metadata and Tone Mapping;

Part 2-1: Application Guide to System Integration;

Part 2-2: Post-production Requirements and Processes;

Part 3-1: Technical Requirements and Test Methods – Display Device;

Part 3-2: Technical Requirements and Test Methods – Mobile Display Device;

Part 3-3: Technical Requirements and Test Methods – Playback Device;

Part 3-4: Technical Requirements and Test Methods – Playback Software.

This document replaces UWA 005-2020 *High Dynamic Range Video Technology Part 1: Metadata and Tone Mapping*. Apart from structural adjustment and editing, this document includes the following technical changes:

a) The definitions of AVS2 and AVS3 HDR picture metadata extension have been modified based on the latest contributions in AVS.

b) In 7.2.3, the range of maximum luminance of the mastering display has been changed to 1000 cd/m^2 to 10000 cd/m^2. When no static metadata is transmitted, the default value of 4000 cd/m^2.

c) In 7.3 Dynamic Metadata Syntax, 3Spline_enable_num has been changed to 3Spline_num.

d) In 9.1, maximum display luminance in the display luminance range of the mastering display has been renamed from MaxDisplay to MaxDisplayPQ (value in the PQ gamut); and the minimum display luminance in the display luminance range of the mastering display has been renamed from MinDisplay to MinDisplayPQ.

e) Five curve modes are added: 3, 4, 5, 6, and 7 are added under base_param_Delta_mode. The IF conditions have been changed accordingly.

f) In 9.3.2.3, a new IF condition has been added: If base_param_Delta_mode ≥ 3 or if base_flag = 0, skip step b and step c.

g) In 9.3.3.2, TH2[1] and TH3[1] expressions have been updated. The logic has been changed: If VA3>TH3, and base_param_Delta_mode ≠ 3 and ≠ 2 and ≠ 6, then m_b is changed to m_b-(-TH3), and VA3 is changed to TH3. If VA2>TH2[1], and base_param_Delta_mode ≠ 3 and ≠ 2 and ≠ 6, then VA2 is changed to TH2[1].

h) In 9.3.3.3, the suffix has been changed from 1 to 2, representing the cubic spline interval parameter in paragraph 2. In process a), a new If-Else statement has been added: If TH3[2]<TH3[1], then set

3Spline_num to 1, skip processes b-g, and end Cubic Spline Interval Parameter Acquisition Process 2; else if TH1[2]<TH3[1], then TH1[2]=TH3[1], TH2[2]=(TH1[2]+TH3[2])/2.

Introduction

Please note the fact that the following related patents may be used when declaring compliance with this document:

Video Signal Processing Method and Device (CN201810799589.4)

Video Signal Processing Method and Device (CN201810797968.X)

Method and Apparatus for Processing Video Signals (CN201810799603.0)

Image Signal Conversion Method and Apparatus, and Terminal Device (CN201580085178.3)

Image Processing Method and Apparatus (CN201680087579.7)

Image Processing System and Method for Generating High Dynamic Range Image (CN201480076068.6)

Image Encoding/Decoding Method and Apparatus (CN201580077269.2)

Image Processing Method and Device, and Terminal Equipment (CN201710266941.3)

Photography Method, Related Equipment and Computer Storage Medium (CN201810257073.7)

Method and Device for High-dynamic-range Image Synthesis (CN201410101591.1)

UWA has no position on the authenticity, validity and scope of the above patents.

The holder of above patents has assured UWA that he is willing to negotiate with any applicant on patent licensing under reasonable and non-discriminatory terms and conditions. The holder's statement has been filed with UWA. Relevant information can be obtained by any of the following contact methods:

Contact person: Ding Chengfei

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

Postcode: 518129

Email: dingchengfei@huawei.com

Tel: 13922880564

Fax: +86-755-36674842

Website: www.huawei.com

Please note that in addition to the above patents, some content of this document may involve other patents. UWA is not responsible for identifying these patents.

High Dynamic Range Video Technology
Part 1: Metadata and Tone Mapping

1 Scope

This document stipulates the tone mapping process of high quality High Dynamic Range (HDR) end-to-end transmission and display. The tone mapping process is suitable for various types of HDR Optical-Electro Transfer Functions (OETF) and levels of terminal display brightness.
This document applies to multiple display application scenarios, such as television broadcast, digital film, network television, network video, video surveillance, real-time and instant communication, digital storage media and still images.

2 Normative References

The content of the following documents is normatively referenced as indispensable clauses in this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.
GY/T 315-2018 Image parameter values for high dynamic range television for use in production and programme exchange
ISO 11664-1/CIE S 014-1 Colorimetry - Part 1: Standard Colorimetric Observers
ISO 11664-3/CIE S 014-3 Colorimetry - Part 3: CIE Tristimulus Values
ISO/IEC 9899:2011 Information technology--Programming languages--C
ITU-R BT.2020-2 Parameter values for ultra-high definition television systems for production and programme exchange
ITU-R BT.2100-1 Image parameter values for high dynamic rang television for use in production and programme exchange
ITU-R BT.2408 HDR Operational practices in HDR television production
ITU-R BT.2390 High dynamic range television for production and international programme exchange
SMPTE ST 2086 Mastering Display Color Volume Metadata Supporting High Luminance And Wide Color Gamut Images
CIE S 015 Lighting of Outdoor Workplaces

3 Terms and Definitions

For the purpose of this document, the terms and definitions below apply.

3.1 Metadata

A set of data that describes key information or features needed in video or image processing.

3.2 Static Metadata

Metadata that stays unchanged in videos.

3.3 Dynamic Metadata

Metadata that changes with videos or images.

3.4 PQ HDR Video

Video transmitted in PQ format specified in GY/T 315-2018.

3.5 HLG HDR Video

Video transmitted in HLG format specified in GY/T 315-2018.

4 Abbreviations

For the purpose of this document, the abbreviations below apply.

EOTF	Electro-Optical Transfer Function
HDR	High Dynamic Range
HLG	Hybrid Log-Gamma
HEVC	High Efficiency Video Coding
MSB	Most Significant Bit
OETF	Optical-Electro Transfer Function
PQ	Perception Quantization
SDR	Standard Dynamic Range

5 End-to-end System Requirements

5.1 General Requirements

The parameters of HDR content production and exchange shall comply with GY/T 315-2018.
Static metadata program shall comply with SMPTE ST 2086.
Dynamic metadata program shall comply with this document.

5.2 System Framework Diagram

See Figure 1 for HDR end-to-end system framework. It consists mainly of HDR preprocessing, encoding transmission, decoding and HDR/SDR display module.

HDR preprocessing module: Used to extract static and dynamic metadata from HDR videos. Chapter 7 introduces the syntax and semantics of static and dynamic metadata. Appendix A introduces the method for extracting dynamic metadata. Appendix B introduces methods for processing input sources in various formats.

Encoding transmission module: Used to encode HDR videos and dynamic metadata and to output elementary streams (ESs). Chapter 8 introduces the method for encapsulating metadata in ESs.

Decoding module: Used to decode the ES and output decoded HDR videos and metadata. Chapter 8 introduces the method for encapsulating metadata in ESs.

HDR and SDR display module: Used to process decoded HDR videos with static and dynamic metadata and target display terminal parameters and display the processed videos on the display terminal. The target display terminal parameter is the display capability of the target display terminal. Chapter 9 describes the HDR display mapping process of PQ HDR. Chapter 10 describes the SDR display mapping process of PQ HDR. Chapter 11 describes the display mapping process of HLG HDR. Appendix C describes the display process of processed videos in the display terminal.

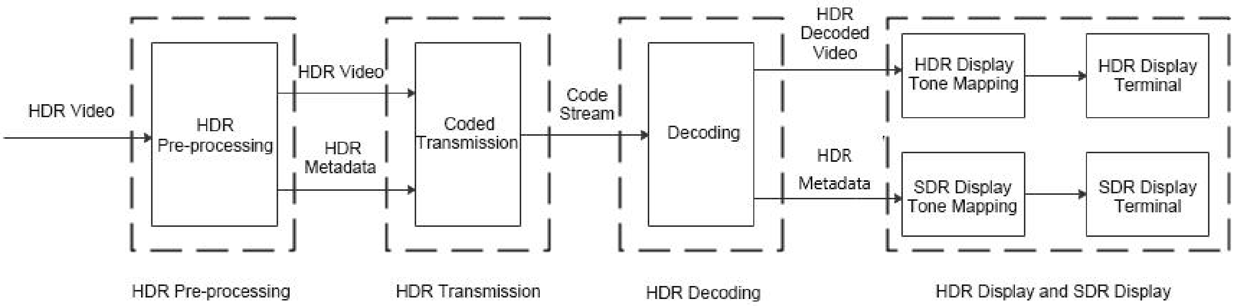

Figure 1 HDR End-to-end System

6 Notations and Operations

6.1 General Requirement

Mathematic operators and priority used in this document refer to the C programming language and comply with ISO/IEC 9899:2011. There are special definitions for integer division and arithmetic shift. Numbers and counting start from "0", unless otherwise specified.

6.2 Arithmetic Operators

See Table 1 for definitions of arithmetic operators.

Table 1 Definitions of Arithmetic Operators

Arithmetic Operators	Definition
+	Addition
-	Subtraction (binary operator) or negation (unary prefix operator)
×	Multiplication
a^b	Exponentiation. It means *a* to the power of *b*. It can also refer to superscript.
/	Integer division. Results are truncated toward 0. For example, 7/4 and -7/-4 are truncated to 1 while -7/4 and 7/-4 are truncated to -1.
÷	Division without truncation or round-off.
$\frac{a}{b}$	Division without truncation or round-off.
$\sum_{i=a}^{b} f(i)$	Cumulative sum of function $f(i)$ when independent variable *i* has all integer values from *a* to *b* (including *b*).
a % *b*	Modulus operation. It refers to the remainder returned after *a* is divided by *b* (both *a* and *b* are positive integers).
$\lceil \cdot \rceil$	Round up

6.3 Logical Operators

See Table 2 for definitions of logical operators.

Table 2 Definitions of Logical Operators

Logical Operators	Definition
a && *b*	AND logical operation between *a* and *b*.
$a \Vert b$	OR logical operation between *a* and *b*.
!	Not logical operation

6.4 Relational Operators

See Table 3 for definitions of relational operators.

Table 3 Definitions of Relational Operators

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

6.5 Bitwise Operators

See Table 4 for definitions of bitwise operators.

Table 4 Definitions of Bitwise Operators

Bitwise Operators	Definition
&	AND operation
$\vert$	OR operation
~	Negation operation
a >> *b*	Shift *a* to the right by *b* bits in form of 2's complement integer. Define this operation only when *b* is positive.
a << *b*	Shift *a* to the left by *b* bits in form of 2's complement integer. Define this operation only when *b* is positive.

6.6 Assignment

See Table 5 for definitions of assignment operations.

Table 5 Definitions of Assignment Operations

Assignment Operation	Definition
=	Assignment operator
++	Ascending. *x*++ is shorthand for $x = x + 1$. Evaluate variables before self-addition when ++ is used as array index.
--	Descending. *x*-- is shorthand for $x = x - 1$. Evaluate variables before self-subtraction when ++ is used as array index.

+=	Add a designated value to a number itself, for example, x += 3 is equal to x = x + 3, x += (-3) is shorthand for $x = x + (-3)$.
-=	Subtract a designated value from a number itself, for example, x -= 3 is equal to x = x – 3, x -= (-3) is shorthand for $x = x - (-3)$.

6.7 Mathematical Functions

See Formula (1) - (11) for definitions of mathematical functions.

$$\text{Abs}(x) = \begin{cases} x & ; \quad x >= 0 \\ -x & ; \quad x < 0 \end{cases} \tag{1}$$

In this formula:

x —— independent variable x.

$$\text{Ceil}(x) = \lceil x \rceil \tag{2}$$

In this formula:

x —— independent variable x.

$$\text{Clip1}(x) = \text{Clip3}(0, 2^{BitDepth} - 1, x) \tag{3}$$

In this formula:

x —— independent variable x;

$BitDepth$ —— accuracy of encoding sample.

$$Clip3\,(i, j, x) = \begin{cases} i & ; \quad x < i \\ j & ; \quad x > j \\ x & ; \quad others \end{cases} \tag{4}$$

In this formula:

x —— independent variable x;

i —— lower bound;

j —— upper bound.

$$\text{Median}(x, y, z) = x + y + z - \text{Min}(x, \text{Min}(y, z)) - \text{Max}(x, \text{Max}(y, z)) \tag{5}$$

In this formula:

x —— independent variable x;

y —— independent variable y;

z —— independent variable z.

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x <= y \\ y & ; \quad x > y \end{cases} \tag{6}$$

In this formula:

x —— independent variable x;

y —— independent variable y.

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x >= y \\ y & ; \quad x < y \end{cases} \tag{7}$$

In this formula:
x —— independent variable x;
y —— independent variable y.

$$\mathrm{Sign}(x) = \begin{cases} 1 & ; \quad x >= 0 \\ -1 & ; \quad x < 0 \end{cases} \tag{8}$$

In this formula:
x —— independent variable *x*.

$$\mathrm{Log}(x) = \log_2 x \tag{9}$$

In this formula:
x —— independent variable *x*.

$$\mathrm{Ln}(x) = \log_e x \tag{10}$$

In this formula:
x —— independent variable x;
e —— base of natural logarithm with the value of 2.718281828….

$$pow(x, y) = x^y \tag{11}$$

In this formula:
x —— independent variable x;
y —— independent variable y.

6.8 Transition Functions

See Formula (12) - (15) for definitions of transition functions.

$$PQ\_EOTF^{-1}(L) = \left(\frac{c_1 + c_2 L^{m_1}}{1 + c_3 L^{m_1}}\right)^{m_2} \text{ ....} (12)$$

$$PQ\_EOTF(L) = \left(\frac{max\left[\left(L^{1/m_2} - c_1\right), 0\right]}{c_2 - c_3 L^{1/m_2}}\right)^{1/m_1} \text{ ....} (13)$$

In this formula:
L —— independent variable *L*.

$$m_1 = \frac{2610}{4096} \times \frac{1}{4} = 0.1593017578125$$

$$m_2 = \frac{2523}{4096} \times 128 = 78.84375$$

$$c_1 = c_3 - c_2 + 1 = \frac{3424}{4096} = 0.8359375$$

$$c_2 = \frac{2413}{4096} \times 32 = 18.8515625$$

$$c_3 = \frac{2392}{4096} \times 32 = 18.6875$$

$$HLG\_OETF(L) = \begin{cases} \sqrt{3 \times L} & 0 < L \leq 1.0/12 \\ a * log(12 \times L - b) + c & 1.0/12 < L \leq 1 \end{cases} \qquad \text{.........................(14)}$$

$$HLG\_OETF^{-1}(L) = \begin{cases} \frac{L^2}{3} & 0 \leq L \leq 0.5 \\ \frac{\left(exp\left(\frac{(L-c)}{a}\right)+b\right)}{12} & 0.5 < L \leq 1 \end{cases} \qquad \text{..............(15)}$$

In this formula:

L —— independent variable L.

$$a = 0.17883277$$
$$b = 1 - 4 \times a$$
$$c = 0.5 - a \times ln(4 \times a)$$

6.9 Structural Relationship Notation

See Table 6 for definition of structural relationship notation.

Table 6 Structural Relationship Notation

Structural Relationship Notation	Definition
->	For example: *a*->*b* means *a* is a structure and *b* is a member variable of *a*.

6.10 Description of Bitstream Syntax

The description of bitstream syntax is similar to C programming language. Syntax elements of bitstreams are boldfaced. Description of each syntax element includes name (English lowercase letter groups divided by underscores), syntax and semantics. Values of syntax elements in the syntax list and the text are in normal font style.

In some cases, other variables derived from syntax elements can be applied to the syntax list. In the syntax list or the text, such variables are named with lowercase letters between underscores, or a mixture of lowercase and uppercase letters. Variables with names beginning with an uppercase letter can be applied to the decoding of the current and follow-up related syntax structure. Variables with names beginning with a lowercase letter are applicable only to the sections in which they appear.

In the body of this document, the relationship between mnemonics of syntax elements and their variable values, and between mnemonics of variable values and their values is explain. In some cases, values and mnemonics are equivalent.

If the length of a bit string is in multiples of 4, the string can be represented using the hexadecimal notation. The prefix of hexadecimal notation is “0x”, for example, “0x1*a*” means the bit string “0001 1010”.

0 indicates FALSE and non-0 indicates TRUE in conditional statements.

Syntax list describes all supersets of bitstream syntax that conform to this document. Additional syntax limitation is specified in relevant items.

Table 7 provides pseudo-code examples for describing syntax. The appearance of a syntax element indicates that a data unit is read from a bitstream.

Table 7 Pseudo-codes for Syntax Description

Pseudo-code	Descriptor
/* Statement is the descriptor of a syntax element or the explanation of its existence, type and value. Here are two examples. */	
syntax_element	
conditioning statement	
/* The statement group in braces is a compound statement and equal to a single statement in terms of functions. */	
{	
statement	
…	
}	
/* The "while" statement tests whether condition is TRUE or not. If TRUE, the statement keeps executing loop body repeatedly till condition is not TRUE. */	
while (condition)	
statement	
/* The "do…while" statement executes loop body once first and then tests whether condition is TRUE or not. If TRUE, the statement keeps executing loop body repeatedly till condition is not TRUE. */	
do	
statement	
while (condition)	
/* The "if…else" statement tests whether condition is TRUE or not first. If TRUE, the statement then executes the primary statement. If not, it executes the alternative statement. If the alternative statement requires no execution, the execution process can ignore "else"part of the "if…else" statement and the alternative statement. */	
if (condition)	
primary statement	
else	
alternative statement	
/* The "for" statement executes the initial statement first and then tests whether condition is TRUE or not. If TRUE, the statement keeps executing the primary statement and the subsequent statement repeatedly till condition is not TRUE. */	
for (initial statement； condition； subsequent statement)	
primary statement	

Table 7 Pseudo-codes for Syntax Description (continued)

The parsing and decoding processes are described in words and pseudo-codes similar to C programming language.

6.11 *read_bits(n)*

This function returns *n* binary digits next to the bitstream with MSB in front, and the bitstream pointer moves forward by *n* binary digits. If *n* is equal to 0, the function returns 0 binary digits and the bitstream pointer does not move forward.
The description of parsing and decoding processes applies to this function as well.

6.12 Descriptor

Descriptor refers to the parsing process of different syntax elements. See Table 8.

Table 8 Descriptor

Descriptor	Explanation
b(8)	It refers to a byte with an arbitrary value. The returned value of function read_bits (8) stipulates its parsing process.
f(n)	It refers to n sequential binary digits with specific values. The returned value of function read_bits(n) stipulates its parsing process.
r(n)	It refers to n sequential zeros. The returned value of function read_bits(n) stipulates its parsing process.
u(n)	It refers to n signless integers. In the syntax list, other syntax elements determine the value of n if n is "v". The returned value of function read_bits(n) stipulates its parsing process. It has a form of binary digits with high bits in front.

6.13 Reserved, Forbidden and Marker_bit

Values of some syntax elements are marked as "reserved" or "forbidden" in the bitstream syntax defined in this document.
"Reserved" means the values will be used for future supplement of this document. These values shall not appear in bitstreams of this document.
"Forbidden" means the values shall not appear in bitstreams of this document.
"Marker_bit" means the value of the bit is '1'.
"Reserved_bits" in bitstreams mean some syntax units are reserved for future supplement of this document, but they shall be ignored in the decoding process. "Reserved_bits" shall not contain over 21 sequential zeros starting from any byte alignment position.

7 Metadata

7.1 Static Metadata Syntax

Table 9 describes the specific syntax for static metadata.

Table 9 Static Metadata Syntax

Definition of Mastering Display and Content Metadata Extension	Descriptor
static _metadata ( ) {	
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)
max_display_mastering_luminance	u(16)
marker_bit	f(1)
min_display_mastering_luminance	u(16)
marker_bit	f(1)
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( )	
}	

7.2 Static Metadata Semantics

7.2.1 display_primaries_x[c], display_primaries_y[c]

16-bit unsigned integers that respectively specify the normalized x and y chromaticity coordinates of the color primary component of the mastering display in increments of 0.00002 within the range of 0 to 50 000. The coordinates comply with the CIE 1931 definition of x and y (as specified in ISO 11664-1/CIE S 014-1 and ISO 11664-3/CIE S 014-3 and CIE S 015). The value of c may be 0, 1, or 2, values which respectively correspond to green, blue, and red.

7.2.2 white_point_x, white_point_y

16-bit unsigned integers that respectively specify the normalized x and y chromaticity coordinates of the standard white point component of the mastering display in increments of 0.00002 within a range of 0 to 50 000. The coordinates comply with the CIE 1931 definition of x and y (as specified in ISO 11664-1/CIE S 014-1 and ISO 11664-3/CIE S 014-3 and CIE S 015).

7.2.3 max_display_mastering_luminance

A 16-bit unsigned integer that specifies the maximum display luminance of the mastering display in increments of 1 cd/m^2 within a range of 1 000 cd/m^2 to 10 000 cd/m^2.

7.2.4 min_display_mastering_luminance

A 16-bit unsigned integer that specifies the minimum display luminance of the mastering display in increments of 0.0001 cd/m^2 within a range of 0.0001 cd/m^2 to 6.5535 cd/m^2.

The value of max_display_mastering_luminance must be greater than min_display_mastering_luminance.

7.2.5 max_content_light_level

A 16-bit unsigned integer that specifies the highest content light level in increments of 1 cd/m^2 within a range of 1 cd/m^2 to 65,535 cd/m^2.
The value of max_content_light_level is the highest value among the highest light levels (PictureMaxLightLevels) of all display pictures in a piece of display content. PictureMaxLightLevel of a display picture is calculated as follows:

——Calculate the highest value of the R, G, and B components (maxRGB) of each pixel in the active area of the display picture. The active area is a rectangular area defined by display_horizontal_size and display_vertical_size.

1) The nonlinear (R', G', B') value of the pixel is converted into a linear (R, G, B) value by formula (13) and calibrated to a value which is an integral multiple of 1 cd/m^2.
2) maxRGB is calculated based on the calibrated (R, G, B) value.

——PictureMaxLightLevel of the display picture refers to the highest value among the maxRGBs of all the pixels in the active area.

7.2.6 max_picture_average_light_level

A 16-bit unsigned integer that specifies the highest picture average light level in increments of 1 cd/m^2 within a range of 1 cd/m^2 to 65,535 cd/m^2.
The value of max_picture_average_light_level is the highest value in the picture average light levels (PictureAverageLightLevels) of all display pictures in a piece of display content. PictureAverageLightLevel of a display picture is calculated as follows:

——Calculate maxRGB of each pixel in the active area of the display picture. The active area is a rectangular area defined by display_horizontal_size and display_vertical_size.

1) The nonlinear (R', G', B') value of the pixel is converted into a linear (R, G, B) value by formula (13) and calibrated to a value which is an integral multiple of 1 cd/m^2.
2) maxRGB is calculated based on the calibrated (R, G, B) value.

——PictureAverageLightLevel of the display picture refers to the average value of maxRGBs of all the pixels in the active area.

7.2.7 extension_id

A "1010" bitstring that identifies the target display device and metadata extension.

7.3 Dynamic Metadata Syntax

Table 10 describes the specific syntax for dynamic metadata.

Table 10 Dynamic metadata syntax

Definition of dynamic metadata	Descriptor
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++){	
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 ++) {	

Table 10 Dynamic metadata syntax (continued)

Definition of dynamic metadata	Descriptor
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)
}	
}	
}	
}	
color_saturation_mapping_ enable_flag[w]	u(1)
if(color_saturation_mapping_flag[w]) {	
color_saturation_ enable_num[w]	u(3)
for(i = 0； i< color_saturation_num [w]； i++) {	
color_saturation_ enable_gain[i][w]	u(8)
}	
}	
}	
}	
}	

7.4 Dynamic Metadata Semantics

7.4.1 system_start_code

An 8-bit unsigned integer that specifies the system version number.

7.4.2 minimum_maxrgb_pq[w]

A 12-bit unsigned integer that specifies the lowest luminance of the display content. It is the lowest value among the highest nonlinear RGB values of all pixels in a scenario or a frame. The value of minimum_maxrgb is equal to minimum_maxrgb_pq[w]/4095 in increments of 1.0/4095 within a range of 0.0 to 1.0.

7.4.3 average_maxrgb_pq[w]

A 12-bit unsigned integer that specifies the average luminance of the display content. For the calculation process, see Appendix A.6. The value of average_maxrgb is equal to average_maxrgb_pq[w]/4095 in increments of 1.0/4095 within a range of 0.0 to 1.0.

7.4.4 variance_maxrgb_pq[w]

A 12-bit unsigned integer that specifies the variation range of the display content. For the calculation process, see Appendix A.7. The value of variance_maxrgb is equal to variance_maxrgb_pq[w]/4095 in increments of 1.0/4095 within a range of 0.0 to 1.0.

7.4.5 maximum_maxrgb_pq[w]

A 12-bit unsigned integer that specifies the highest luminance of the display content. It is the highest value among the highest nonlinear RGB values of all pixels in a scenario or a frame. The value of maximum_maxrgb is equal to maximum_maxrgb_pq[w]/4095 in increments of 1.0/4095 within a range of 0.0 to 1.0.

7.4.6 tone_mapping_enable_mode_flag[w]

A 1-bit unsigned integer that specifies whether to enable tone mapping. The value is 0 or 1. tone_mapping_mode_flag is equal to tone_mapping_enable_mode_flag[w]. If tone_mapping_mode_flag is 0, curve parameters are not transferred; and if tone_mapping_mode_flag is 1, curve parameters are transferred.

7.4.7 tone_mapping_param_enable_num[w]

A 1-bit unsigned integer that specifies the current number of tone mapping parameter sets minus 1. tone_mapping_param_num is equal to tone_mapping_param_enable_num[w]. When tone_mapping_param_num is 0, the number of tone mapping parameter sets is 1; and when tone_mapping_param_num is 1, the number of tone mapping parameter sets is 2.

7.4.8 targeted_system_display_maximum_luminance_pq[i][w]

A 12-bit unsigned integer that specifies the maximum luminance of the metadata reference display. targeted_system_display_maximum_luminance is equal to targeted_system_display_maximum_luminance_pq[i][w]/4095 and is in increments of 1.0/4095 within a range of 0.0 to 1.0. When targeted_system_display_maximum_luminance_pq[i][w] is equal to 2080, it indicates that the current tone mapping curve parameters are intended for SDR and is used only when SDR signals are output; and when targeted_system_display_maximum_luminance_pq[i][w] is not equal to 2080, it indicates that the current tone mapping curve parameters are intended for HDR.

7.4.9 base_enable_flag[i][w]

A 1-bit unsigned integer that specifies whether to enable the base curve. The value is 0 or 1. base_flag is equal to base_enable_flag[i][w]. If base_flag is 0, base curve parameters are not transferred. When base_flag is 1, the base curve parameters are transferred.

7.4.10 base_param_m_p[i][w]

A 14-bit unsigned integer that specifies m_p_0 in the base curve parameters. The value of m_p_0 is 10.0*base_param_m_p[i][w]/16383 in increments of 10.0/16383 within a range of 0.0 to 10.0.

7.4.11 base_param_m_m[i][w]

A 6-bit unsigned integer that specifies m_m_0 in the base curve parameters. The value of m_m_0 is base_param_m_m[i][w]/10.0 in increments of 0.1 within a range of 0.0 to 6.3.

7.4.12 base_param_m_a[i][w]

A 10-bit unsigned integer that specifies m_a_0 in the base curve parameters. The value of m_a_0 is base_param_m_a[i][w]/1023 in increments of 1.0/1023 within a range of 0.0 to 1.0.

7.4.13 base_param_m_b[i][w]

A 10-bit unsigned integer that specifies m_b_0 in the base curve parameters. The value of m_b_0 is base_param_m_b[i][w] x 0.25/1023 in increments of 0.25/1023 within a range of 0.0 to 0.25.

7.4.14 base_param_m_n[i][w]

A 6-bit unsigned integer that specifies m_n_0 in the base curve parameters. The value of m_n_0 is base_param_m_n[i][w]/10 in increments of 0.1 within a range of 0.0 to 6.3.

7.4.15 base_param_K1[i][w]

A 2-bit unsigned integer that specifies k1_0 in the base curve parameters. When base_param_K1[i][w] is less than or equal to 1, the value of k1_0 is equal to that of base_param_K1[i][w]; and values of base_param_K1[i][w] greater than 1 are reserved.

7.4.16 base_param_K2[i][w]

A 2-bit unsigned integer that specifies k2_0 in the base curve parameters. When base_param_K2[i][w] is less than or equal to 1, the value of k2_0 is equal to that of base_param_K2[i][w]; and values of base_param_K2 greater than 1 are reserved.

7.4.17 base_param_K3[i][w]

A 4-bit unsigned integer that specifies k3_0 in the base curve parameters. When base_param_K3[i][w] is equal to 1, the value of k3_0 is equal to that of base_param_K3[i][w]; when base_param_K3[i][w] is 2, the value of k3_0 is that of maximum_maxrgb; and the rest of base_param_K3[i][w] values are reserved.

7.4.18 base_param_Delta_enable_mode[i][w]

A 3-bit unsigned integer that specifies the delta mode of current base curve parameters. base_param_Delta_mode is equal to base_param_Delta_enable_mode[i][w].

7.4.19 base_param_enable_Delta[i][w]

A 7-bit unsigned integer that specifies the parameter adjustment factor value of current base curve parameters. The value of base_param_Delta is base_param_enable_Delta[i][w]/127 in increments of 1.0/127 within a range of 0.0 to 1.0. When base_param_Delta_mode is equal to 2 or 6, base_param_Delta is set to -base_param_Delta.

7.4.20 3Spline_enable_flag[i][w]

A binary variable. When the value is 1, it indicates that cubic spline parameters need to be transferred. 3Spline_flag is equal to 3Spline_enable_flag[i][w]. When the value is 0, it indicates that cubic spline parameters do not need to be transferred. If 3Spline_flag is 0, cubic spline parameters 3Spline_num[w] and 3Spline_TH_mode are not transferred, and 3Spline_num[w] is set to 1 and 3Spline_TH_mode is set to 0.

7.4.21 3Spline_enable_num[i][w]

A 1-bit unsigned integer that specifies the number of cubic spline interval sets used in tone mapping. The value is 0 or 1. 3Spline_num is equal to 3Spline_enable_num[i][w] plus 1, that is, 1 or 2.

7.4.22 3Spline_TH_enable_mode[j][i][w]

A 2-bit unsigned integer that specifies the cubic spline mode in tone mapping. The value range is 0 to 3. 3Spline_TH_mode is equal to 3Spline_TH_enable_mode[j][i][w].

7.4.23 3Spline_TH_enable_MB[j][i][w]

An 8-bit unsigned integer that specifies the slope and dark-area offset of the cubic spline interval parameter used in tone mapping. When 3Spline_TH_mode is 0, the value of 3Spline_TH_MB is six MSBs of 3Spline_TH_enable_MB[j][i][w]/63 within a range of 0.0 to 1.0, and the value of base_offset is two LSBs of 3Spline_TH_enable_MB[j][i][w] x 0.1/3. When 3Spline_TH_mode is not 0, the value of 3Spline_TH_MB is 3Spline_TH_enable_MB[j][i][w] x 1.1/255 in increments of 1.1/255 within a range of 0.0 to 1.1.

7.4.24 3Spline_TH enable[j][i][w]

A 12-bit unsigned integer that specifies the cubic spline interval parameter in tone mapping. The value of 3Spline_TH is 3Spline_TH enable[j][i][w]/4095 in increments of 1.0/4095 within a range of 0.0 to 1.0.

7.4.25 3Spline_TH_enable_Delta1[j][i][w]

A 10-bit unsigned integer that specifies the offset of a cubic spline interval parameter in tone mapping. The value of 3Spline_TH_Delta1 is 3Spline_TH_enable_Delta1[j][i][w] x 0.25/1023 in increments of 0.25/1023 within a range of 0.0 to 0.25. The value of the cubic spline interval parameter TH2 is the cubic spline interval plus the offset of the cubic spline interval 1, that is, 3Spline_TH + 3Spline_TH_Delta1.

7.4.26 3Spline_TH_enable_Delta2[j][i][w]

A 10-bit unsigned integer that specifies the offset of a cubic spline interval parameter used in tone mapping. The value of 3Spline_TH_Delta2 is 3Spline_TH_enable_Delta2[j][i][w] x 0.25/1023 in increments of 0.25/1023 within a range of 0.0 to 0.25. The value of the cubic spline interval parameter TH3 is the cubic spline interval 1 plus the offset of the cubic spline interval 2, that is, 3Spline_TH + 3Spline_TH_Delta1 + 3Spline_TH_Delta2.

7.4.27 3Spline_enable_Strength[j][i][w]

An 8-bit unsigned integer that specifies the correction strength parameter of cubic spline intervals in tone mapping. The value of 3Spline_Strength is 2.0*3Spline_enable_Strength[j][i][w]/255-1.0 in increments of 2.0/255 within a range of -1.0 to 1.0. When 3Spline_Strength does not exist in the bitstream, the value of 3Spline_Strength is 0.

7.4.28 color_saturation_mapping_enable_flag[w]

A binary variable. color_saturation_mapping_flag is equal to color_saturation_mapping_enable_flag[w]. When the value is 1, it indicates that the color saturation parameter needs to be transferred, and when the value is 0, it indicates that the color saturation parameter does not need to be transferred.

7.4.29 color_saturation_enable_num[w]

A 3-bit unsigned integer that specifies the number of color saturation parameters. color_saturation_num is equal to color_saturation_enable_num[w] and is within a range of 0 to 7.

7.4.30 color_saturation_enable_gain[i][w]

An 8-bit unsigned integer that specifies the color saturation strength. The value of color_saturation_gain[i] is color_saturation_enable_gain[i][w]/128 in increments of 1.0/128 within a range of 0.0 to 2.0. The value of color_saturation_gain[1] is (color_saturation_enable_gain[i][w]&0xFC)/128 in increments of 1.0/128, within a range of 0.0 to 2.0.

8 Metadata Encapsulation

8.1 Encapsulation of Metadata in AVS2 and AVS3 ESs

Metadata is encapsulated in second-generation Audio Video coding Standard (AVS2) elementary streams (ESs), as shown in Table 11.

Table 11 Definitions of AVS2 extension data and user data

<table>
<tr><th>Definitions of Extension Data and User Data</th><th>Descriptor</th></tr>
<tr><td>extension_and_user_data(i) {</td><td></td></tr>
<tr><td>while ((next_bits(32) extension_start_code) || (next_bits(32) user_data_start_code)) {</td><td></td></tr>
<tr><td>if (next_bits(32) extension_start_code)</td><td></td></tr>
<tr><td>extension_data(i)</td><td></td></tr>
<tr><td>if (next_bits(32) user_data_start_code)</td><td></td></tr>
<tr><td>user_data()</td><td></td></tr>
<tr><td>}</td><td></td></tr>
<tr><td>}</td><td></td></tr>
</table>

extension_data(i) is extension data in which i is 0. The extension data is transmitted only in sequence headers and is encapsulated as shown in Table 12.

Table 12 Definition of AVS2 extension data

Definition of Extension Data	Descriptor
extension_data(i) {	

while (next_bits(32) == extension_start_code) {	
extension_start_code	f(32)
if (i == 0) { /* After the sequence header */	
if (next_bits(4) == '0010') /* Sequence display extension */	
sequence_display_extension( )	
else if (next_bits(4) == '0011') /* Scalable in temporal domain */	
temporal_scalability_extension( )	
else if (next_bits(4) == '0100') /* Copyright extension */	
copyright_extension()	
else if (next_bits(4) == '1010') /* Mastering display and content metadata extension */	
mastering_display_and_content_metadata_extension( )	
else if (next_bits(4) == '1011') /* Camera parameter extension */	
camera_parameters_extension( )	
else if (next_bits(4) == '1101') /* Reference library picture extension */	
cross_random_access_point_reference_extension( )	
else	
while (next_bits(24) != '0000 0000 0000 0000 0000 0001')	
reserved_extension_data_byte	u(8)
}	
}	
else { /* After the picture header */	
if (next_bits(4) == '0100') /* Copyright extension */	
copyright_extension( )	
else if (next_bits(4) == '0111') /* Picture display extension */	
picture_display_extension( )	
else if (next_bits(4) == '1011') /* Camera parameter extension */	
camera_parameters_extension( )	
else if (next_bits(4) == '1100') /* Region of interest (ROI) parameter extension */	
roi_parameters_extension( )	
else {	
while (next_bits(24) != '0000 0000 0000 0000 0000 0001')	
reserved_extension_data_byte	u(8)
}	
}	
}	
}	

Static metadata is encapsulated in mastering_display_and_content_metadata_extension(), as shown in Table 12. For the syntax, see section 7.1. For the semantics, see section 7.2. Dynamic metadata is encapsulated in hdr_dynamic_metadata_extension( ), see Table 14.

AVS3 ES bitstream is encapsulated as shown in Table 13 and Table 14.

Table 13 Definition of AVS3 extension data

Definition of Extension Data	Descriptor
extension_data(i) {	
while (next_bits(32) == extension_start_code) {	
extension_start_code	f(32)
if (i == 0) { /* After the sequence header */	
if (next_bits(4) == '0010') /* Sequence display extension */	
sequence_display_extension( )	
else if (next_bits(4) == '0011') /* Scalable in temporal domain */	
temporal_scalability_extension( )	
else if (next_bits(4) == '0100') /* Copyright extension */	
copyright_extension()	
else if (next_bits(4) == '0110') /* Content encryption extension */	
cei_extension( )	
else if (next_bits(4) == '1010') /* Mastering display and content metadata extension */	
mastering_display_and_content_metadata_extension( )	
else if (next_bits(4) == '1011') /* Camera parameter extension */	
camera_parameters_extension( )	
else if (next_bits(4) == '1101') /* Reference library picture extension */	
cross_random_access_point_reference_extension( )	
else	
while (next_bits(24) != '0000 0000 0000 0000 0000 0001')	
reserved_extension_data_byte	u(8)
}	
}	
else { /* After the picture header */	
if (next_bits(4) == '0100') /* Copyright extension */	
copyright_extension( )	
else if (next_bits(4) == '0101') /* HDR picture metadata extension */	
hdr_dynamic_metadata_extension( )	
else if (next_bits(4) == '0111') /* Picture display extension */	
picture_display_extension( )	
else if (next_bits(4) == '1011') /* Camera parameter extension */	
camera_parameters_extension( )	
else if (next_bits(4) == '1100') /* ROI parameter extension */	
roi_parameters_extension( )	
else {	
while (next_bits(24) != '0000 0000 0000 0000 0000 0001')	
reserved_extension_data_byte	u(8)
}	
}	
}	
}	

Static metadata is encapsulated in mastering_display_and_content_metadata_extension(), as shown in in Table 13. For the syntax, see section 7.1. For the semantics, see section 7.2.

Table 14 Definition of AVS2 and AVS3 HDR picture extension

Definition of HDR Picture Metadata Extension	Descriptor
hdr_dynamic_metadata_extension( ) {	
extension_id	f(4)
hdr_dynamic_metadata_type	f(4)
while (next_bits(24) ! '0000 0000 0000 0000 0000 0001') {	
extension_data_byte	u(8)
}	
next_start_code()	
}	

The dynamic metadata is encapsulated in hdr_dynamic_metadata_extension(), as shown in Table 14. extension_id is a "0101" bit string that indenties HDR picture extension. hdr_dynamic_metadata_type is a 4-bit unsigned integer that identifies the type of dynamic metadata. extension_data_byte is an 8-bit unsigned integer. Video extension data bytes cannot contain over 21 consecutive 0s starting from any byte alignment location.

8.2 Encapsulation of Metadata in HEVC and VVC ESs

The metadata is encapsulated in High Efficiency Video Coding (HEVC) ESs, as shown in Table 15.

Table 15 Encapsulation of dynamic metadata in HEVC ESs

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

The static metadata is encapsulated in mastering_display_colour_volume() and content_light_level_info() in Table 16. Its syntax is defined in Rec.ITU-T H.265.

The dynamic metadata is encapsulated in itu_t_t35_payload_byte of user_data_registered_itu_t_t35(), as shown in Table 15. The encapsulation structure is shown in Table 17.

itu_t_t35_country_code

An 8-bit unsigned integer in ITU T35 that identifies the country code of the organization that the self-defined content block belongs to.

itu_t_t35_country_code_extension_byte

An 8-bit unsigned integer in ITU T35 that identifies the country extension code of the organization that the self-defined content block belongs to.

The metadata is encapsulated in Versatile Video Coding (VVC) ESs, as shown in Table 16.

Table 16 Encapsulation of dynamic metadata in VVC ESs

sei_payload(payloadType, payloadSize) {	**Descriptor**
if(nal_unit_type == PREFIX_SEI_NUT)	
if(payloadType == 0)	
buffering_period(payloadSize)	
else if(payloadType == 1)	
pic_timing(payloadSize)	
else if(payloadType == 3)	
filler_payload(payloadSize) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
else if(payloadType == 4) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
user_data_registered_itu_t_t35(payloadSize)	
else if(payloadType == 5) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
user_data_unregistered(payloadSize)	
else if(payloadType == 19) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
film_grain_characteristics(payloadSize)	
else if(payloadType == 45) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
frame_packing_arrangement(payloadSize)	
else if(payloadType == 129) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
parameter_sets_inclusion_indication(payloadSize)	
else if(payloadType == 130)	
decoding_unit_info(payloadSize)	
else if(payloadType == 133)	
scalable_nesting(payloadSize)	
else if(payloadType == 137) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
mastering_display_colour_volume(payloadSize)	
else if(payloadType == 144) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
content_light_level_info(payloadSize)	
else if(payloadType == 145) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
dependent_rap_indication(payloadSize)	
else if(payloadType == 147) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
alternative_transfer_characteristics(payloadSize)	
else if(payloadType == 148) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
ambient_viewing_environment(payloadSize)	
else if(payloadType == 149) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
content_colour_volume(payloadSize)	
else if(payloadType == 150) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	

Table 16 Encapsulation of dynamic metadata in VVC ESs (continued)

sei_payload(payloadType, payloadSize) {	**Descriptor**
equirectangular_projection(payloadSize)	
else if(payloadType = = 153) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
generalized_cubemap_projection(payloadSize)	
else if(payloadType = = 154) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
sphere_rotation(payloadSize)	
else if(payloadType = = 155) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
regionwise_packing(payloadSize)	
else if(payloadType = = 156) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
omni_viewport(payloadSize)	
else if(payloadType = = 168) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
frame_field_info(payloadSize)	
else if(payloadType = = 203)	
subpic_level_info(payloadSize)	
else if(payloadType = = 204) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
sample_aspect_ratio_info(payloadSize)	
else /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
reserved_message(payloadSize)	
else /* nal_unit_type = = SUFFIX_SEI_NUT */	
if(payloadType = = 3) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
filler_payload(payloadSize)	
if(payloadType = = 132) /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
decoded_picture_hash(payloadSize)	
else if(payloadType = = 133)	
scalable_nesting(payloadSize)	
else /* Specified in ITU-T H.274 | ISO/IEC 23002-7 */	
reserved_message(payloadSize)	
if(more_data_in_payload( )) {	
if(payload_extension_present( ))	
sei_reserved_payload_extension_data	u(v)
sei_payload_bit_equal_to_one /* equal to 1 */	f(1)
while(!byte_aligned( ))	
sei_payload_bit_equal_to_zero /* equal to 0 */	f(1)
}	
}	

The static metadata is encapsulated in mastering_display_colour_volume() and content_light_level_info(), as shown in Table 16. Its syntax is defined in Rec.ITU-T H.265.

The dynamic metadata is encapsulated in itu_t_t35_payload_byte of user_data_registered_itu_t_t35(), as shown in Table 16. The encapsulation structure is shown in Table 17.

terminal_provide_code

A 16-bit unsigned integer in ITU T35 that identifies the code of the organization that the self-defined content block belongs to.

terminal_provide_oriented_code

A 16-bit unsigned integer that identifies the code of the content block as defined by the organization it belongs to in ITU T35.

Table 17 Encapsulation structure of itu_t_t35_payload_byte

itu_t_t35_payload(){	**Descriptor**
terminal_provide_code	f(16)
terminal_provide_oriented_code	f(16)
dynamic_metadata ()	
}	

Syntax definition of dynamic_metadata() in Table 17 can be found in section 7.3, and semantic description can be found in section 7.4.

9 HDR Display Tone Mapping of PQ HDR

9.1 HDR Display Tone Mapping Process

Input: MaxDisplayPQ (maximum display luminance in the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (minimum display luminance in the display luminance range of the mastering display in the PQ gamut, which is 0 by default), $f[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed), and metadata information

Output: $f_{process}[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed, once the HDR display tone mapping is complete)

The HDR display tone mapping process is as follows:

1).... Apply the base curve parameter acquisition process described in section 9.2 to generate base curve parameters.
2).... Apply the cubic spline parameter acquisition process described in section 9.3 to generate cubic spline interval parameters.
3).... Apply the color signal dynamic range conversion process described in section 9.4 to generate $f_{TM}[N_{frame}][3]$, which will be the RGB gamut pixel buffer for the frame to be processed once dynamic range conversion is complete.
4).... Apply the color correction process described in section 9.5 to generate $f_{color}[N_{frame}][3]$, which will be the processed RGB gamut pixel buffer for the frame to be processed.
5).... Apply the post-processing process described in section 9.6 to generate $f_{process}[N_{frame}][3]$, which will be the RGB gamut pixel buffer for the frame to be processed once HDR display tone mapping is complete. N_{frame} is the total sampling points for the frame to be processed.

9.2 Base Curve Parameter Acquisition Process

9.2.1 Overview

Input: metadata information, MaxDisplayPQ (maximum display luminance in the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (minimum display luminance in the display luminance range of the mastering display in the PQ gamut, which is 0 by default)

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3, which form the base curve

$$H(L) = m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \text{……(16)}$$

The base curve parameter acquisition process is as follows:

1).... From the metadata information, obtain the tone mapping mode flag (tone_mapping_mode_flag) and the base curve flag (base_flag) corresponding to target_system_display_maximum_luminance.

2).... From the metadata information, obtain the highest luminance correction value (max_lum) and the lowest luminance (min_lum) of the frame to be processed:

i.........The highest luminance correction value (max_lum) = maximum_maxrgb in the metadata.

ii.........The lowest luminance (min_lum) = minimum_maxrgb in the metadata.

3).... Update the highest luminance correction value (max_lum) of the frame to be processed, in accordance with the process described in section 9.2.3.

4).... Calculate the base curve parameter set $P_{tone\_mapping}$:

a) If tone_mapping_mode_flag is 0, or tone_mapping_mode_flag is 1 and base_flag is 0, apply the base curve parameter acquisition process 0 described in section 9.2.2 to update the parameters.

b) If tone_mapping_mode_flag is 1 and base_flag is 1:

If targeted_system_display_maximum_luminance is equal to MaxDisplayPQ or if base_param_Delta_mode is 3, map m_p, m_a, m_m, m_n, m_b, K1, K2, and K3 to m_p_0, m_a_0, m_m_0, m_n_0, m_b_0, K1_0, K2_0, and K3_0, respectively.

Otherwise:

If base_param_Delta_mode is 0 or 2 or 4 or 6, apply the base curve parameter adjustment process 1 described in section 9.2.4 to update the parameters.

If base_param_Delta_mode is 1 or 5, apply the base curve parameter adjustment process 2 described in section 9.2.5 to update the parameters.

9.2.2 Base Curve Parameter Acquisition Process 0

Input: MaxDisplayPQ (maximum display luminance in the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (minimum display luminance in the display luminance range of the mastering display in the PQ gamut, which is 0 by default), metadata information

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The base curve parameter acquisition process 0 is as follows:

1) Set m_m, m_n, K1, K2, and K3 to the values 2.4, 1, 1, 1, and 1, respectively. The curve then becomes

$$\left(\frac{m\_p \times L}{(m\_p-1) \times L+1}\right)^{2.4} \quad \ldots\ldots \quad (17)$$

2) Set m_b to MinDisplay

3) Based on the average_maxrgb (avgL) in the metadata, calculate m_p:

$$\begin{cases} p_{valueH0} & avgL > TPH0 \\ p_{valueH0} \times g0(w0) + p_{valueL0} \times (1 - g0(w0)) & avgL \geq TPL, avgL \leq TPH0 \\ p_{valueL0} & avgL < TPL0 \end{cases} \quad \ldots\ldots(18)$$

Where:

$w0 = \left(\frac{avgL-TPL0}{TPH0-TPL0}\right)$, $P_{valueH0}$, $P_{valueL0}$, TPH0 and TPL0 are preset values, and the default values are 3.5, 4.0, 0.6, and 0.3.

g0() is $y=x^N$, which defaults to y=x.

4) Update m_p based on the highest luminance correction value (max_lum)

$$\begin{cases} m\_p + p_{deltaH1} & max\_lum > TPH1 \\ m\_p + p_{deltaH1} \times g1(w1) + p_{deltaL1} \times (1 - g1(w1)) & max\_lum \geq TPL1, max\_lum \leq TPH1 \\ m\_p + p_{deltaL1} & max\_lum < TPL1 \end{cases} \quad \text{……(19)}$$

Where:

$w1 = \left(\frac{max\_lum - TPL1}{TPH1 - TPL1}\right)$, $P_{deltaH1}$, $P_{deltaL1}$, TPH1 and TPL1 are preset values, and the default values are 0.6, 0.0, 0.9, and 0.75.

g1() is $y=x^N$, which defaults to y=x.

5) Based on m_p, m_m, m_n, m_b, K1, K2, and K3, calculate m_a

$$f(L) = \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} \quad \text{……(20)}$$

Accordingly,

$$m\_a = (MaxDisplayPQ - MinDisplayPQ)/f(MaxSource) \quad \text{……(21)}$$

Where MaxSource is equal to the max_lum (PQ gamut) of the frame to be processed.

9.2.3 The Process for Updating the Highest Luminance Correction Value (Max_Lum) of the Frame to Be Processed

Input: MaxDisplayPQ (maximum display luminance in the display luminance range of the mastering display in the PQ gamut), maximum display luminance value max_display_mastering_luminance, metadata information
Output: the highest luminance correction value (max_lum) of the frame to be processed
The update process is as follows:

1) Convert the maximum display luminance value (max_display_mastering_luminance) to the PQ gamut to get the display luminance value MaxRefDisplay for the reference display device.
2) Based on maximum_maxrgb, average_maxrgb and variance_maxrgb in the metadata, calculate the reference maximum value MAX1 for the frame to be processed

$$MAX1 = \begin{matrix} B \times maximum\_maxrgb + A \times (2 \times average\_maxrgb) \\ +(1 - A - B) * (variance\_maxrgb) \end{matrix} \quad \text{……(22)}$$

Where A and B are weighting coefficients. A is a function of average_maxrgb, A=(1-B)*(1-F(average_maxrgb/maximum_maxrgb)), F(x) is a constant function, A is 0.4 by default, and B is 0.2 by default.

3) Determine the final max_lum based on MaxRefDisplay, the preset minimum value MIN (the default is 0.5081), and MAX1:

$$max\_lum = \begin{cases} MaxRefDisplay & MAX1 > MaxRefDisplay \\ MAX1 & MAX1 \geq MIN \&\& MAX1 \leq MaxRefDisplay \\ MIN & MAX1 < MIN \end{cases} \quad \text{……(23)}$$

If max_lum is smaller than MaxDisplayPQ, set max_lum to MaxDisplayPQ.

9.2.4 Base Curve Parameter Adjustment Process 1

Input: MaxDisplayPQ (maximum display luminance of the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (minimum display luminance of the display luminance range of the mastering display in the PQ gamut, which is 0 by default), metadata information m_p_0, m_m_0, m_n_0, m_a_0, m_b_0, K1_0, K2_0, and K3_0, targeted_system_display_maximum_luminance, base_param_Delta
Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The adjustment process is as follows:

1) Set m_m, m_n, K1, K2, and K3 to m_m_0, m_n_0, K1_0, K2_0, and K3_0, respectively.
2) Set m_b to m_b_0*((MaxDisplayPQ-MinDisplayPQ)/targeted_system_display_maximum_luminance).
3) Set m_a to m_a_0*((MaxDisplayPQ-MinDisplayPQ)/targeted_system_display_maximum_luminance).
4) Set m_p to m_p_0+base_param_Delta*(10000*Abs((PQ_EOTF (MaxDisplayPQ)- PQ_EOTF (targeted_system_display_maximum_luminance)))/ 100)^N, the default value of N is 0.5. m_p is contrained in the range [3.0, 7.5].

9.2.5 Base Curve Parameter Adjustment Process 2

Input: MaxDisplayPQ (maximum display luminance of the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (minimum display luminance of the display luminance range of the mastering display in the PQ gamut, which is 0 by default), metadata information m_p_0, m_m_0, m_n_0, m_a_0, m_b_0, K1_0, K2_0, and K3_0, targeted_system_display_maximum_luminance, base_param_Delta

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The adjustment process is as follows:

1) As described in 9.2.2, generate the base curve parameter set $P_{tone\_mapping}$, consisting of m_p_1, m_m_1, m_n_1, m_a_1, m_b_1, K1_1, K2_1, and K3_1
2) Calculate w=base_param_Delta *(10000*Abs((PQ_EOTF(MaxDisplayPQ)-PQ_EOTF(targeted_system_display_maximum_luminance)))/ 100)^N, where N is a number greater than 0, and it uses 0.5 as the default; w is greater than or equal to 0, and less than or equal to 1.
3) Set m_p, m_m, m_n, K1, K2, and K3 to (1-w)*m_p_0+w*m_p_1, (1-w)*m_m_0+w*m_m_1, (1-w)*m_n_0+w*m_n_1, (1-w)*K1_0+w*K1_1, (1-w)*K2_0+w*K2_1, (1-w)*K3_0+w*K3_1
4) Set m_b to MinDisplayPQ
5) To get:

$$f(L) = \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} \quad ......(24)$$

a. Assign L=MaxSource to the following formula to get m_a:

$$m\_a = \frac{(MaxDisplayPQ - MinDisplayPQ)}{\left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m}} \quad ......(25)$$

Where MaxSource is equal to the max_lum (PQ gamut) of the frame to be processed.

9.3 Cubic Spline Parameter Acquisition Process

9.3.1 Overview

Input: metadata information

Output: cubic spline mapping curve parameter set $P_{3spline}$

The process is as follows:

1) Based on the tone mapping mode flag tone_mapping_mode_flag in the metadata information and the cubic spline mode flag 3Spline_flag corresponding to targeted_system_display_maximum_luminance, set the cubic spline interval parameter. If tone_mapping_mode_flag is 0 or tone_mapping_mode_flag is 1 and 3Spline_flag is 0, the cubic spline parameters 3Spline_num[w] and 3Spline_TH_mode are not passed in the code stream, 3Spline_num is set to 1 and 3Spline_TH_mode is set to 0; if

tone_mapping_mode_flag is 1 and 3Spline_flag is 1, get the corresponding parameters from the code stream.

2) In accordance with 3Spline_num, traverse every cubic spline interval to get cubic spline mapping curve parameter set $P_{3spline}$

   a) If tone_mapping_mode_flag is 0 or tone_mapping_mode_flag is 1 and 3Spline_flag is 0 and 3Spline_TH_mode is 0, or if tone_mapping_mode_flag is 1, 3Spline_flag is 1 and 3Spline_TH_mode is not 0, apply the linear spline curve parameter acquisition process 0 described in section 9.3.2.1, the linear spline curve parameter adjustment process 0 in section 9.3.2.3 and the cubic spline interval parameter acquisition process 0 in section 9.3.3.1 to get the parameters. The parameter values are the default values.

   If tone_mapping_mode_flag is 1, 3Spline_flag is 1 and 3Spline_TH_mode is 0, apply the linear spline curve parameter acquisition process 1 described in section 9.3.2.2 and the linear spline curve parameter adjustment process 0 in section 9.3.2.3 and the cubic spline interval parameter acquisition process 1 in section 9.3.3.2 to get the parameters. The parameter values are obtained from the code stream.

   b) If tone_mapping_mode_flag is 1, 3Spline_flag is 1 and 3Spline_TH_mode is 1/2/3, apply the cubic spline interval parameter acquisition process 2 described in section 9.3.3.3 to get the parameter. The parameter value is obtained from the code stream.

   When 3spline_num is 1, set 3Spline_num to 2, 3Spline_TH_mode[1]=3Spline_TH_mode[0], and stop traversing b.

9.3.2 Linear Spline Interval Parameter Acquisition Process

Input: metadata information

Output: the linear spline curve parameter set $P_{1spline}$, TH3[0], MB[0][0], and base_offset

Get the linear spline curve between the first endpoint and the first interpolation point TH3[0]:

$$F(L)=MB[0][0]\times L+base\_offset \quad \text{……(26)}$$

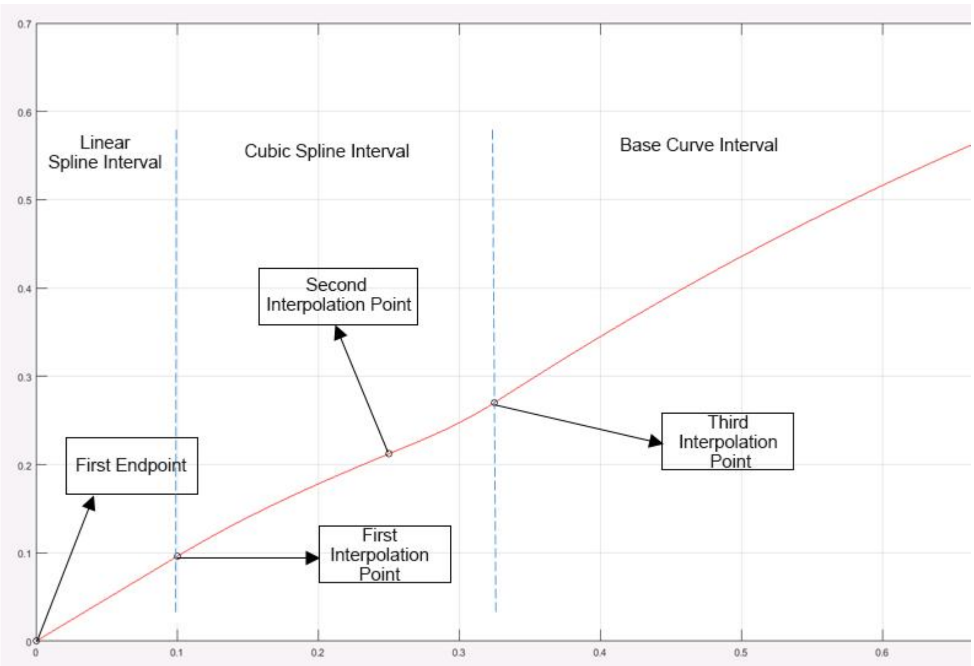

Figure 2. Diagram of the Linear Spline and the Cubic Splines

9.3.2.1 Linear Spline Curve Parameter Acquisition Process 0

Input: metadata information

Output: the linear spline curve parameter set $P_{1spline}$, TH3[0], MB[0][0], and base_offset

The process is as follows:

1) Based on the average_maxrgb (avgL) in the metadata, get the linear spline parameter TH3[0]:

$$TH3[0] = \begin{cases} T_{dmaxL2} & avgL > HLMAXH2 \\ (T_{dmaxL2} \times g2(w2) + T_{dmaxH2} \times (1 - g2(w2))) & avgL \geq HLMAXL2 \&\& avgL \leq HLMAXH2 \\ T_{dmaxH2} & avgL < HLMAXL2 \end{cases} \quad \ldots\ldots(27)$$

Where:

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

$HLMAXH2$ and $HLMAXL2$ are the thresholds corresponding to the luminance value of the pixel in the bright area. $HLMAXH2$ is the upper luminance threshold of the pixel in the bright area, the default is 0.6; $HLMAXL2$ is the lower luminance threshold of the pixel in the bright area, the default is 0.3.

T_{dmaxL2} and T_{dmaxH2} are the thresholds corresponding to the luminance value of the pixel in the dark area. T_{dmaxL2} is the lower luminance threshold of the pixel in the dark area, the default is 0.1; T_{dmaxH2} is the upper luminance threshold of the pixel in the dark area, the default is 0.25; g2() is $y=x^N$, and the default is y=x.

2) Based on the average_maxrgb（avgL）in the metadata, get the linear spline parameter MB[0][0] and *base_offset*:

$$MB[0][0] = \begin{cases} S_{dmaxL3} & avgL > AVMAXH3 \\ S_{dmaxL3} \times g3(W3) + S_{dmaxH3} \times (1 - g3(W3)) & avgL \geq AVMAXL3, avgL \leq AVMAXH3 \\ S_{dmaxH3} & avgL < AVMAXL3 \end{cases} \quad \ldots\ldots(16)$$

base_offset=0

Where:

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

AVMAXH3 and AVMAXL3 are the thresholds corresponding to the average luminance value of the pixel in the bright area.AVMAXH3 is the upper threshold of the average luminance value of the pixel in the bright area, the default is 0.6; AVMAXL3 is the lower threshold of the average luminance value of the pixel in the bright area, the default is 0.3.

S_{dmaxL3} and S_{dmaxH3} are the thresholds for the slope at which the luminance value of pixel in the dark area is increased. S_{dmaxL3} is the lower threshold for the slope at which the luminance value of pixel in the dark region is increased, the default is 0.96; S_{dmaxH3} is the upper threshold for the slope at which the luminance value of pixel in the dark region is increased, the default is 1.0; g3() is $y=x^N$, and the default is y=x.

9.3.2.2 Linear Spline Curve Parameter Acquisition Process 1

Input: metadata information

Output: the linear spline curve parameter set $P_{1spline}$, TH3[0], MB[0][0], and base_offset

Based on the 3Spline_TH_MB and 3Spline_TH in the metadata, get the linear spline interpolation point TH3[0], the linear spline parameter MB[0][0] and base_offset:

$$TH3[0] = 3Spline\_TH[0] \quad \text{......(29)}$$

$$MB[0][0] = 3Spline\_TH\_MB \quad \text{......(30)}$$

$$\text{base\_offset} = base\_offset \quad \text{......(31)}$$

9.3.2.3 Linear Spline Curve Parameter Adjustment Process 0

Input: the MaxDisplayPQ (maximum display luminance of the display luminance range of the mastering display in the PQ gamut), max_lum (the highest luminance correction value of the frame to be processed), targeted_system_display_maximum_luminance in the metadata, if there is no targeted_system_display_maximum_luminance in the metadata, then targeted_system_display_maximum_luminance is equal to MaxDisplayPQ; the original linear spline curve parameter set $P_{1spline}$, MB[0][0], TH3[0]; the tone mapping curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

Output: the linear spline curve parameter set $P_{1spline}$, MB[0][0], TH3[0]

The adjustment process is as follows:

1) If base_param_Delta_mode is larger than or equal to 3, or if base_flag is equal to 0, skip step b and c;
2) MB_mid[0][0]= MB[0][0], TH3_mid[0]=TH3[0].
3) If m_a is smaller than or equal to Tm_AP (m_p), step c does not need to be executed, and the adjustment process is complete.

   If m_a is greater than Tm_ap (m_p) and Tm_ap is the lookup table (m_p_T, m_a_T) (when m_m=2.4, m_n=1, m_b=0, k1=k2=k3=1, the specific values are (2.5, 0.990), (3.5, 0.879), (4.5, 0.777), (7.5, 0.540), and the remaining values are obtained by linear interpolation), then adjust MB[0][0] and TH3[0] based on max_lum/MaxDisplayPQ, where the default values of N1 and N2 are both 1.0; use m_p to obtain m_a_T from the lookup table Tm_ap; and set m_b value to (1-WA)*m_b, to get the new linear spline tone mapping curve parameters MB[0][0] and TH3[0] (the output luminance of the new tone mapping curve will be less than or equal to the input luminance of the new tone mapping curve):

$$MB[0][0] = Min\,(Max\,(MB\_mid[0][0] + (1 - MB\_mid[0][0]) \times (WA)^{N1}, MB\_mid[0][0]), 1) \quad \text{......(32)}$$

$$TH3[0] = Min\,(Max\,(TH3\_mid[0] + (max\_lum - TH3\_mid[0]) \times (WA)^{N2}, TH3\_mid[0]), 1) \quad \text{......(33)}$$

   Where:

$$WA = \frac{\frac{MaxDisplayPQ}{max\_lum} - \frac{H(\text{max}\_lum, m\_a\_T)}{max\_lum}}{1 - \frac{H(max\_lum, m\_a\_T)}{max\_lum}}$$

$$H(L) = m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} + m\_b$$

   H(L, m_a_T) represents the H(L) value corresponding to the input variable L when the parameter m_a in H(L) takes the value m_a_T, and WA is the weighting coefficient.

9.3.3 Cubic Spline Interval Parameter Acquisition Process

Get the first cubic spline curve between the first interpolation point TH1[n] and the second interpolation point TH2[n]:

$$F(L) = MD[0][n] \times (L - TH1[n])^3 + MC[0][n] \times (L - TH1[n])^2 + MB[0][n] \times (L - TH1[n])^1 + MA[0][n] \quad ......(34)$$

Where L is the independent variable within the interval [TH1[n], TH2[n]],

And the second cubic spline curve between the second interpolation point TH2[n] and the third interpolation point TH3[n]:

$$F(L) = MD[1][n] \times (L - TH2[n])^3 + MC[1][n] \times (L - TH2[n])^2 + MB[1][n] \times (L - TH2[n])^1 + MA[1][n] \quad ......(35)$$

Where L is the independent variable within the interval [TH2[n], TH3[n]], 0<n<=3Spline_num.

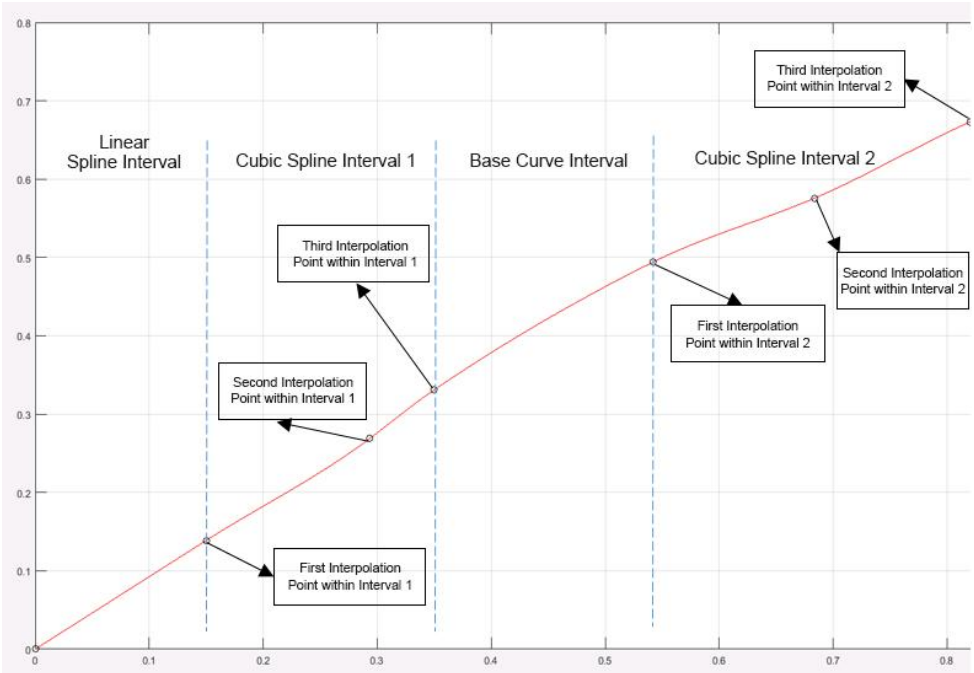

Figure 3. Diagram of the Cubic Spline and the Base Curve

9.3.3.1 Cubic Spline Interval Parameter Acquisition Process 0

Input: linear spline curve parameter set $P_{1spline}$, consisting of TH3[0], MB[0][0], and base_offset, color signal mapping curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3.

Output: the cubic spline interval parameter $P_{3spline}$, consisting of 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 process is as follows:

1) Based on TH3[0], get the three interpolation points (TH1[1], TH2[1], and TH3[1]) within the current cubic spline interval:

$$TH1[1] = TH3[0] \quad ......(36)$$

$$TH2[1] = TH1[1] + B \quad ......(37)$$

Where B is the offset corresponding to the luminance value of the pixel in the dark transition area, and the default value is 0.15.

$$TH3[1] = TH2[1] + C * TH2[1] - D * TH1[1] \quad ......(38)$$

Where C and D are the weighting coefficients corresponding to the luminance value of the pixel in the luminance area, and the default value is 0.5.

2) Get these eight parameters 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] from TH1[1], TH2[1], and TH3[1]:

i.........Calculate the second coordinates (Y coordinates) VA1, VA2 and VA3 corresponding to TH1[1], TH2[1], and TH3[1].

First, based on the linear spline function in TH1[0] through TH3[0] in a), get the following

$$F(L) = MB[0][0] \times L + base\_offset \quad \ldots\ldots(39)$$

Then, set L to TH1[1] to calculate TH1[1]'s second coordinate VA1 using the above formula

$$VA1 = MB[0][0] \times TH1[1] + base\_offset \quad \ldots\ldots(40)$$

And, based on the base curve parameter set $P_{tone\_mapping}$, get the following:

$$H(L) = m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \ldots\ldots(41)$$

Then, set L to TH3[1] to calculate TH3[1]'s second coordinate VA3 using the above formula

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \ldots \quad (42)$$

Then, calculate TH2[1]'s second coordinate VA2; VA2 is the second coordinate of the point TH2[1] on the line connecting the point (TH1[1], VA1) and the point (TH3[1], VA3):

$$VA2 = VA1 + \frac{(TH2[1] - TH1[1]) \times (VA3 - VA1)}{TH3[1] - TH1[1]} \quad \ldots\ldots(43)$$

And VA1<VA2<VA3

ii.........Calculate MA[0][1] and MA[1][1]: (let the value of the first cubic spline at TH1[1] be the value of VA1, and let the value of the second cubic spline at TH2[1] be the value of VA2)

$$MA[0][1]=VA1 \quad \ldots\ldots(44)$$

$$MA[1][1]=VA2 \quad \ldots\ldots(45)$$

iii.........Calculate MB[0][1]: calculate GD1, which is the first derivative of the formula (39)at TH1[1], so that MB[0][1]=GD1; Calculate GD3, which is the first derivative of the formula (41) at TH3[1]

$$MB[0][1] = GD1 = MB[0][0] \quad \ldots\ldots(46)$$

$$GD3 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH3[1]^{m\_n-1} \times DGD3(L) \quad \ldots\ldots(47)$$

Where:

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

iv.........Calculate MC[0][1], MD[0][1], MB[1][1], MC[1][1], and MD[1][1]:

Calculate VA2[0], the value of the first cubic spline curve at TH2[1], so that VA2[0]=VA2: calculate VA3[0], the value of the second cubic spline curve at TH3[1], so that VA3[0]=VA3.

Calculate GD3[0], the derivative of the second cubic spline curve at TH3[1], so that GD3[0]=GD3.

Calculate GD2[0] and GD2[1], the first derivatives of the two cubic splines at TH2[1], so that GD2[0]=GD2[1]; calculate GGD2[0] and GGD2[1], the second derivatives of the two cubic splines at TH2[1], so that GGD2[0]=GGD2[1]:

$$\begin{cases} MD[0][1]\times(DTH2)^3+MC[0][1]\times(DTH2)^2+MB[0][1]\times(DTH2)^1+MA[0][1]=VA2 \\ MD[1][1]\times(DTH3)^3+MC[1][1]\times(DTH3)^2+MB[1][1]\times(DTH3)^1+MA[1][1]=VA3 \\ 3\times MD[1][1]\times(DTH3)^2+2\times MC[1][1]\times(DTH3)^1+MB[1][1]=GD3 \\ 3\times MD[0][1]\times(DTH2)^2+2\times MC[0][1]\times(DTH2)^1+MB[0][1]=GD2[0] \\ 3\times MD[1][1]\times(DTH3)^2+2\times MC[1][1]\times(DTH3)^1+MB[1][1]=GD2[1] \\ 6\times MD[0][1]\times(DTH2)^1+2\times MC[0][1]=GGD2[0] \\ 6\times MD[1][1]\times(DTH3)^1+2\times MC[1][1]=GGD2[1] \end{cases} \quad \text{……(48)}$$

Where:

$DTH2 = (TH2[1]-TH1[1])$

$DTH3 = (TH3[1]-TH2[1])$

By analyzing and solving the above equations, get the values of MB[1][1], MC[0][1], MD[0][1], MC[1][1], and MD[1][1]:

$$\begin{cases} MB[1][1]=\dfrac{-(3.0*VA1*h2*h2+3.0*VA2*h1*h1-3.0*VA3*h1*h1-3.0*h2*h2*VA2+h1*h1*h2*GD3+GD1*h1*h2*h2)}{(2.0*h2*(h1*h1+h2*h1))} \\ MC[0][1]=\dfrac{(3.0*VA2-2.0*GD1*h1-3.0*VA1-MB[1][1]*h1)}{h1*h1} \\ MD[0][1]=\dfrac{(h1*GD1+h1*MB[1][1]+2*VA1-2.0*VA2)}{h1*h1*h1} \\ MC[1][1]=MC[0][1]+3.0*MD[0][1]*h1 \\ MD[1][1]=-\dfrac{(VA3-VA2-h2*GD3+MC[0][1]*h2*h2+3*MD[0][1]*h1*h2*h2)}{2*h2*h2*h2} \end{cases}$$

……(49)

Where:

$$h1 = DTH2 = (TH2[1]-TH1[1])$$
$$h2 = DTH3 = (TH3[1]-TH2[1])$$

9.3.3.2 Cubic Spline Interval Parameter Acquisition Process 1

Input: metadata information, base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

Output: the cubic spline interval parameter set $P_{3spline}$, consisting of 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 process is as follows:

1) Based on the data information in the metadata, get the three interpolation points (TH1[1], TH2[1], and TH3[1]) within the current cubic spline interval:

$$TH1[1]= TH3[0] \quad \text{……(50)}$$

Where TH1[1] is the lowest luminance value of the pixel in the first interval of the adjustment area.

$$TH2[1]= TH1[1] + 3Spline\_TH\_Delta1 \quad \text{……(51)}$$

Where TH2[1] is the highest luminance value of the pixel in the first interval of the adjustment area and the lowest luminance value of the pixel in the second interval of the adjustment area.

$$TH3[1]= TH1[1] + 3Spline\_TH\_Delta1+ 3Spline\_TH\_Delta2 \quad \text{……(52)}$$

Where TH3[1] is the highest luminance value of the pixel in the second interval of the adjustment area.

2) Get the eight parameters 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] from TH1[1], TH2[1], and TH3[1]:

   i. Calculate the second coordinates (Y coordinates) VA1, VA2 and VA3 corresponding to TH1[1], TH2[1], and TH3[1].

   First, based on the linear spline function in TH1[0] through TH3[0] in 1, get the following:

$$F(L) = MB[0][0] \times L + base\_offset \quad \ldots\ldots(53)$$

Then, set L to TH1[1] to calculate TH1[1]'s second coordinate VA1 using the above formula.

$$VA1 = MB[0][0] \times TH1[1] + base\_offst \quad \ldots\ldots(54)$$

And based on the base curve parameter set $P_{tone\_mapping}$, get the following:

$$H(L) = m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \ldots\ldots(55)$$

Set L to TH3[1] to calculate TH3[1]'s second coordinate (Y point) VA3 using the above formula

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \ldots\ldots(56)$$

If VA3 is larger than TH3[1] and base_param_Delta_mode is not equal to 3, 2, or 6, m_b is set to m_b-(VA3-TH3[1]), and VA3 is set to TH3[1];

Then, calculate TH2[1]'s second coordinate VA2, where VA2 is equal to the second coordinate of the point TH2[1] on the line connecting the point (TH1[1], VA1) and the point (TH3[1], VA3) plus the offset value of TH2[1] obtained from the metadata Spline_Strength[0]information:

$$VA2 = VA1 + \frac{(TH2[1] - TH1[1]) \times (VA3 - VA1)}{TH3[1] - TH1[1]} + \frac{(VA3 - VA1) \times Spline\_Strength[0]}{2} \quad \ldots\ldots(57)$$

And VA1<VA2<VA3. If VA2 is larger than TH2[1], and base_param_Delta_mode is not equal to 2, or 3, or 6, set VA2 to TH2[1].

ii. Calculate MA[0][1] and MA[1][1]: (let the value of the first cubic spline at TH1[1] be the value of VA1, and let the value of the second cubic spline at TH2[1] be the value of VA2)

$$MA[0][1]=VA1 \quad \ldots\ldots(58)$$

$$MA[1][1]=VA2 \quad \ldots\ldots(59)$$

iii. Calculate MB[0][1]: calculate GD1, which is the first derivative of the formula (53) at TH1[1], so that MB[0][1]=GD1; Calculate GD3, the first derivative of the formula (55) at TH3[1]

$$MB[0][1] = GD1 = MB[0][0] \quad \ldots\ldots(60)$$

$$GD3 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH3[1]^{m\_n-1} \times DGD3(L) \quad \ldots\ldots(61)$$

Where:

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

iv. Calculate MC[0][1], MD[0][1], MB[1][1], MC[1][1], and MD[1][1]:

Calculate VA2[0], the value of the first cubic spline curve at TH2[1], so that VA2[0]=VA2: calculate VA3[0], the value of the second cubic spline curve at TH3[1], so that VA3[0]=VA3.

Calculate GD3[0], the derivative of the second cubic spline curve at TH3[1], so that GD3[0]=GD3.

Calculate GD2[0] and GD2[1], the first derivatives of the two cubic splines at TH2[1], so that GD2[0]=GD2[1]; calculate GGD2[0] and GGD2[1], the second derivatives of the two cubic splines at TH2[1], so that GGD2[0]=GGD2[1].

$$\begin{cases} MD[0][1] \times (DTH2)^3 + MC[0][1] \times (DTH2)^2 + MB[0][1] \times (DTH2)^1 + MA[0][1] = VA2 \\ MD[1][1] \times (DTH3)^3 + MC[1][1] \times (DTH3)^2 + MB[1][1] \times (DTH3)^1 + MA[1][1] = VA3 \\ 3 \times MD[1][1] \times (DTH3)^2 + 2 \times MC[1][1] \times (DTH3)^1 + MB[1][1] = GD3 \\ 3 \times MD[0][1] \times (DTH2)^2 + 2 \times MC[0][1] \times (DTH2)^1 + MB[0][1] = GD2[0] \\ 3 \times MD[1][1] \times (DTH3)^2 + 2 \times MC[1][1] \times (DTH3)^1 + MB[1][1] = GD2[1] \\ 6 \times MD[0][1] \times (DTH2)^1 + 2 \times MC[0][1] = GGD2[0] \\ 6 \times MD[1][1] \times (DTH3)^1 + 2 \times MC[1][1] = GGD2[1] \end{cases} \quad \ldots\ldots(62)$$

Where:

$$DTH2 = (TH2[1] - TH1[1])$$
$$DTH3 = (TH3[1] - TH2[1])$$

By analyzing and solving the above equations, get the values of MB[0][1], MC[0][1], MD[0][1], MC[1][1], and MD[1][1]:

$$\begin{cases} MB[1][1] = \frac{-(3.0*VA1*h2*h2 + 3.0*VA2*h1*h1 - 3.0*VA3*h1*h1 - 3.0*h2*h2*VA2 + h1*h1*h2*GD3 + GD1*h1*h2*h2)}{(2.0*h2*(h1*h1 + h2*h1))} \\ MC[0][1] = \frac{(3.0*VA2 - 2.0*GD1*h1 - 3.0*VA1 - MB[1][1]*h1)}{h1*h1} \\ MD[0][1] = \frac{(h1*GD1 + h1*MB[1][1] + 2*VA1 - 2.0*VA2)}{h1*h1*h1} \\ MC[1][1] = MC[0][1] + 3.0*MD[0][1]*h1 \\ MD[1][1] = -\frac{(VA3 - VA2 - h2*GD3 + MC[0][1]*h2*h2 + 3*MD[0][1]*h1*h2*h2)}{2*h2*h2*h2} \end{cases}$$

……(63)

Where:

$h1 = DTH2 = (TH2[1] - TH1[1])$

$h2 = DTH3 = (TH3[1] - TH2[1])$

9.3.3.3 Cubic Spline Interval Parameter Acquisition Process 2

Input: metadata information, base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, k1, k2, and k3

Output: the cubic spline interval parameter set $P_{3spline}$, consisting of TH1[2], TH2[2], TH3[2], MA[0][2], MB[0][2], MC[0][2], MD [0][2], MA[1][2], MB[1][2], MC[1][2], and MD[1][2]

The process is as follows:

1) Based on the data information in the metadata, get the three interpolation points (TH1[2], TH2[2], and TH3[2]) within the current cubic spline interval:

$$TH1[2] = 3Spline\_TH \quad ……(64)$$

Where TH1[2] is the lowest luminance value of the pixel in the first interval of the adjustment area.

$$TH2[2] = 3Spline\_TH + 3Spline\_TH\_Delta1 \quad ……(65)$$

Where TH2[2] is the highest luminance value of the pixel in the first interval of the adjustment area and the lowest luminance value of the pixel in the second interval of the adjustment area.

$$TH3[2] = 3Spline\_TH + 3Spline\_TH\_Delta1 + 3Spline\_TH\_Delta2 \quad ……(66)$$

Where TH3[2] is the highest luminance value of the pixel in the second interval of the adjustment area.

If TH3[2] < TH3[1], set 3Spline_num to 1, do not execute process b-g, and end Cubic Spline Interval Parameter Acquisition Process 2;

Otherwise, if TH1[2]<TH3[1], then TH1[2] = TH3[1], TH2[2] = (TH1[2]+TH3[2])/2.

2) Calculate the second coordinates (Y coordinates) VA1 (y1), VA2 and VA3 (y2) corresponding to TH1[2] (threshold1), TH2[2] (threshold2), and TH3[2] (threshold3).

First, based on the base curve parameter set $P_{tone\_mapping}$, get the following:

$$H(L) = m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} + m\_b \quad ……(67)$$

Set L to TH1[2] and TH3[2], and calculate the second coordinates (y points) VA1 and VA3 of TH1[2] and TH3[2] by using the above formula, the point (TH1[2], VA1) and the point (TH3[2], VA3) are located respectively at the two ends of the base curve.

$$\begin{cases} VA1 = m\_a \times \left(\frac{m\_p \times TH1[2]^{m\_n}}{(K1 \times m\_p - K2) \times TH1[2]^{m\_n} + K3}\right)^{m\_m} + m\_b \\ VA3 = m\_a \times \left(\frac{m\_p \times TH3[2]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[2]^{m\_n} + K3}\right)^{m\_m} + m\_b \end{cases} \quad \text{……(68)}$$

If 3Spline_TH_mode is 1 or 2, and base_param_Delta_mode is not equal to 3, set VA3 to MaxDisplayPQ. If VA3 is now larger than TH3[2], and base_param_Delta_mode is not equal to 2 or 6, then TH3[2] = VA3, and TH2[2] = TH1[2]+ (TH3[2]- TH1[2])/2.0;

If 3Spline_TH_mode is 1 or 2, and base_param_Delta_mode is 3, then set VA3 to targeted_system_display_maximum_luminance.

If 3Spline_TH_mode is 3, keep VA3 as it is.

Then, calculate TH2[2]'s second coordinate VA2; VA2 is calculated by adding the second coordinate of the point TH2[2] on the line connecting the point (TH1[2], VA1) and the point (TH3[2], VA3), to the offset value of TH2[2] from the metadata Spline_Strength[1]information:

$$VA2 = VA1 + \frac{(TH2[2] - TH1[2]) \times (VA3 - VA1)}{TH3[2] - TH1[2]} + \frac{(VA3 - VA1) \times Spline\_Strength[1]}{2} \quad \text{……(69)}$$

And VA1<VA2<VA3. If 3Spline_TH_mode is 1 or 2, VA2 > TH2[2], and base_param_Delta_mode is not equal to 2 or 3 or 6, then set VA2 to TH2[2].

3) Calculate MA[0][2] and MA[1][2]:

$$MA[0][2]=VA1 \quad \text{……(70)}$$

$$MA[1][2]=VA2 \quad \text{……(71)}$$

4) Calculate MB[0][2]: calculate GD1, the first derivative of the formula (67) at TH1[2], so that MB[0][2]=GD1.

$$MB[0][2] = GD1 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH1[2]^{m\_n-1} \times DGD(L) \quad \text{……(72)}$$

Where:

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

5) If 3Spline_TH_mode is 1 or 2, VA3 = TH3[2], and base_param_Delta_mode is not equal to 2 or 3 or 6, the first derivative of GD3 at TH3[2] is 1.0.

If 3Spline_TH_mode is 1, let TH_str = Spline_Strength[1] , and calculate GD3, the first derivative at TH3[2]; GD3 will be less than or equal to 1:

$$\mathrm{GD3} = \begin{cases} (\mathrm{down\_T} * (-\mathrm{TH\_str}) + \mathrm{mid\_T} * (1 + \mathrm{TH\_str})), & \mathrm{TH\_str} < 0 \\ (\mathrm{up\_T} * \mathrm{TH\_str} + \mathrm{mid\_T} * (1 - \mathrm{TH\_str})), & \mathrm{TH\_str} \geq 0 \end{cases} \quad \text{……(73)}$$

Where:

$$up\_T1 = (y2 - y1) / (threshold3 - threshold2)$$

$mid\_T = (y2 - y1) / (threshold3 - threshold1)$,

$down\_T1 = (y2 - y1) * 0.1 / (threshold3 - threshold1)$,

$down\_T = down\_T1 < GD1\ ?\ GD1 : down\_T1$,

$up\_T = up\_T1 < GD1\ ?\ GD1 : up\_T1$

If 3Spline_TH_mode is 2, let $GD3=GD2 - 3Spline\_TH\_MB$

If 3Spline_TH_mode is 3, let GD3=GD2

Where:

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

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

6) Calculate MC[0][2], MD[0][2], MB[1][2], MC[1][2], and MD[1][2]:
Calculate VA2[0], the value of the first cubic spline curve at TH2[2], so that VA2[0]=VA2; calculate VA3[0], the value of the second cubic spline curve at TH3[2], so that VA3[0]=VA3.
Calculate GD3[0], the derivative of the second cubic spline curve at TH3[2], so that GD3[0]=GD3.
Calculate GD2[0] and GD2[2], the first derivatives of the two cubic splines at TH2[1], so that GD2[0]=GD2[1]; calculate GGD2[0] and GGD2[2], the second derivatives of the two cubic splines at TH2[1], so that GGD2[0]=GGD2[1].

$$\begin{cases} MD[0][2] \times (DTH2)^3 + MC[0][2] \times (DTH2)^2 + MB[0][2] \times (DTH2)^1 + MA[0][2] = VA2 \\ MD[1][2] \times (DTH3)^3 + MC[1][2] \times (DTH3)^2 + MB[1][2] \times (DTH3)^1 + MA[1][2] = VA3 \\ 3 \times MD[1][2] \times (DTH3)^2 + 2 \times MC[1][2] \times (DTH3)^1 + MB[1][2] = GD3 \\ 3 \times MD[0][2] \times (DTH2)^2 + 2 \times MC[0][2] \times (DTH2)^1 + MB[0][2] = GD2[0] \\ 3 \times MD[1][2] \times (DTH3)^2 + 2 \times MC[1][2] \times (DTH3)^1 + MB[1][2] = GD2[1] \\ 6 \times MD[0][2] \times (DTH2)^1 + 2 \times MC[0][2] = GGD2[0] \\ 6 \times MD[1][2] \times (DTH3)^1 + 2 \times MC[1][2] = GGD2[1] \end{cases} \quad \ldots\ldots(74)$$

Where:
$DTH2 = (TH2[2] - TH1[2])$
$DTH3 = (TH3[2] - TH2[2])$

7) By analyzing and solving the above equations, get the values of MB[1][2], MC[0][2], MD[0][2], MC[1][2], and MD[1][2]:

$$\begin{cases} MB[1][2] = \frac{-(3.0*VA1*h2*h2 + 3.0*VA2*h1*h1 - 3.0*VA3*h1*h1 - 3.0*h2*h2*VA2 + h1*h1*h2*GD3 + GD1*h1*h2*h2)}{(2.0*h2*(h1*h1 + h2*h1)} \\ MC[0][2] = \frac{(3.0*VA2 - 2.0*GD1*h1 - 3.0*VA1 - MB[1][2]*h1)}{h1*h1} \\ MD[0][2] = \frac{(h1*GD1 + h1*MB[1][2] + 2*VA1 - 2.0*VA2)}{h1*h1*h1} \\ MC[1][2] = MC[0][2] + 3.0 * MD[0][2] * h1 \\ MD[1][2] = -\frac{(VA3 - VA2 - h2*GD3 + MC[0][2]*h2*h2 + 3*MD[0][2]*h1*h2*h2)}{2*h2*h2*h2} \end{cases}$$

$$\ldots\ldots(75)$$

Where:
$h1 = DTH2 = (TH2[2] - TH1[2])$
$h2 = DTH3 = (TH3[2] - TH2[2])$

9.4 Color Signal Dynamic Range Conversion Process

Input: $f[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed), the corrected tone mapping curve; MinDisplayPQ (minimum display luminance in the display luminance range of the mastering display in the PQ gamut), base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3; linear spline parameter set $P_{1spline}$, consisting of MB[0][0] and TH3[0]; cubic spline parameter set $P_{3spline}$, consisting of TH1[3Spline_num], TH2[3Spline_num], TH3[3Spline_num], MA[2][3Spline_num], MB[2][3Spline_num], MC[2][3Spline_num], and MD[2][3Spline_num]
Output: $f_{TM}[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed, once dynamic range conversion is complete)
The conversion process is as follows:

1) Calculate $f_{MAX}[i]$, the highest value among all color gamut components of the sample value f[i][3] in the frames to be processed. i is the pixel index of the frames to be processed, in the range of [0, N_{frame}).
2) Perform color signal dynamic range conversion on $f_{MAX}[i]$

a) If 3Spline_TH_mode in the range of [TH1[n], TH3[n]] is 1 or 2, and $f_{MAX}[i]$> TH3[n], then use the following formula to get $f_{MAX\_TM}[i]$:

$$f_{MAX_{TM}}[i] = \mathrm{MBH} \times (f_{\mathrm{MAX}}[\mathrm{i}] - TH3[\mathrm{n}]) + \mathrm{BASEH}$$
$$\mathrm{MBH} = 3 \times \mathrm{MD}[1][\mathrm{n}] \times H1^2 + 2 \times \mathrm{MC}[1][\mathrm{n}] \times H1 + \mathrm{MB}[1][\mathrm{n}] \qquad ......(76)$$
$$\mathrm{BASEH} = \mathrm{MD}[1][\mathrm{n}] \times H1^3 + \mathrm{MC}[1][\mathrm{n}] \times H1^2 + \mathrm{MB}[1][\mathrm{n}] \times H1^1 + \mathrm{MA}[1][\mathrm{n}]$$

b) If 3Spline_num >0 and fMAX[i] falls into [0, TH3[0]], then use the following dynamic range conversion model to get fMAX_TM[i]:

$$f_{\mathrm{MAX\_TM}}[\mathrm{i}] = \mathrm{MB}[0][0] \times f_{\mathrm{MAX}}[\mathrm{i}] + base\_offset \qquad ......(77)$$

c) If there is n that satisfies 0<n<=3Spline_num, so that TH1[n]<fMAX[i]<TH3[n], then use the following dynamic range conversion model to get fMAX_TM[i]:

$$L1 = (f_{MAX}[i] - TH1[n])$$
$$L2 = (f_{MAX}[i] - TH2[n])$$
$$\begin{cases} \mathrm{MD}[0][\mathrm{n}] \times L1^3 + \mathrm{MC}[0][\mathrm{n}] \times L1^2 + \mathrm{MB}[0][\mathrm{n}] \times L1^1 + \mathrm{MA}[0][\mathrm{n}] & \mathrm{TH1}[\mathrm{n}] < f_{\mathrm{MAX}}[\mathrm{i}] \le \mathrm{TH2}[\mathrm{n}] \\ \mathrm{MD}[1][\mathrm{n}] \times L2^3 + \mathrm{MC}[1][\mathrm{n}] \times L2^2 + \mathrm{MB}[1][\mathrm{n}] \times L2^1 + \mathrm{MA}[1][\mathrm{n}] & \mathrm{TH2}[\mathrm{n}] < f_{\mathrm{MAX}}[\mathrm{i}] < \mathrm{TH3}[\mathrm{n}] \end{cases} \qquad ......(78)$$

d) Else, use the following dynamic range conversion model to get $f_{MAX\_TM}[i]$

$$f_{MAX\_TM}[i] = m\_a \times \left(\frac{m\_p \times (f_{MAX}[i])^{m\_n}}{(k_1 \times m\_p - k_2) \times (f_{MAX}[i])^{m\_n} + k_3}\right)^{m\_m} + m\_b \qquad ......(79)$$

3) Perform color signal dynamic range conversion on all color gamut components of the current sample value

Calculate the gain K:

$$K = PQ\_EOTF(f_{MAX\_TM}[i]) / PQ\_EOTF(f_{MAX}[i]) \qquad ......(80)$$

Convert all the components:

$$f_{TM}[i][0] = PQ\_EOTF(f[i][[0]) \times K$$
$$f_{TM}[i][1] = PQ\_EOTF(f[i][[1]) \times K \qquad ......(81)$$
$$f_{TM}[i][2] = PQ\_EOTF(f[i][[2]) \times K$$

9.5 Color Correction Process

Input: $f[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed), $f_{TM}[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed, once dynamic range conversion is complete), metadata information, TML (value in PQ gamut) of MaxDisplay (maximum display luminance of the mastering display), and RML (value of max_display_mastering_luminance in PQ gamut).

Output: $f_{color}[N_{frame}][3]$ (the processed RGB gamut pixel buffer for the frame to be processed)

The color correction process is as follows:

1) If the color correction flag color_saturation_mapping_flag is 0, then

$$f_{color}[N_{frame}][0] = f_{TM}[N_{frame}][0]$$
$$f_{color}[N_{frame}][1] = f_{TM}[N_{frame}][1] \qquad ......(82)$$
$$f_{color}[N_{frame}][2] = f_{TM}[N_{frame}][2]$$

and end the color correction process.

Else, calculate color correction intensity C0 and C1.

$$C0 = color\_saturation\_gain[0] \qquad ......(83)$$
$$C1 = color\_saturation\_gain[1] \qquad ......(84)$$

2) Convert the output RGB linear signal $f_{TM}[i][3]$ described in section 9.4 into a nonlinear signal $f_{TM\_PQ}[i][3]$ in the PQ gamut.

$$f_{TM\_PQ}[i][3] = PQ\_EOTF^{-1}(f_{TM}[i][3]) \qquad \text{……(85)}$$

Convert $f_{TM\_PQ}[i][3]$ into YCbCr color signal YC_bC_r in accordance with ITU-BT2020:

$$\begin{bmatrix} Y_{in} \\ C_{b\_in} \\ C_{r\_in} \end{bmatrix} = \begin{bmatrix} 0.2627 & 0.6780 & 0.0593 \\ -0.1396 & -0.3604 & 0.5000 \\ 0.5000 & -0.4598 & -0.0402 \end{bmatrix} \cdot \begin{bmatrix} f[i][0] \\ f[i][1] \\ f[i][2] \end{bmatrix} \qquad \text{……(86)}$$

3) Calculate $f_{MAX}[i]$, the highest value among all color gamut components of the sample value f[i][3] in the frames to be processed, and calculate $f_{MAX\_TM\_PQ}[i]$, the highest value among all color gamut components of the sample value $f_{TM\_PQ}[i][3]$ in the non-linear signals of the frame to be processed, once dynamic range conversion is complete. i is the pixel index of the frames to be processed, in the range of [0, N_{frame}).
4) Calculate the color adjustment coefficient S_{ca}:
   a) If $f_{MAX}[i]$ > TML, and color_saturation_num ≥ 2, then

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

……(86)

Where S_{ca} is in the range of 0.0-1.0, SatR is the saturation correction coefficient, with a default value of 0.4. $M = 2^{(color\_saturation\_enable\_gain[1][w]\&0x3)}$. A is the adjustment range coefficient, with a default value of TML/RML. B is the intensity range coefficient in the range of 0.8-1.0, with a default value of $\left(\frac{TML\_TM}{TML}\right)^{C0}$. TML_TM is the nonlinear signal of TML after dynamic range conversion.

   b) Otherwise, the color adjustment coefficient S_{ca} is related to the the tone mapping curve, in the range of 0.8-1.0. The formula is as follows:

$$S_{ca} = \left(\frac{f_{MAX\_TM\_PQ}[i]}{f_{MAX}[i]}\right)^{C0} \qquad (87)$$

5) Adjust the color of the YCbCr signal.

Multiply Cb and Cr respectively by the same color adjustment coefficient S_{ca}:

$$Y' = Y$$
$$C_b' = C_b \cdot S_{ca} \qquad \text{…… (88)}$$
$$C_r' = C_r \cdot S_{ca}$$

Convert YCbCr signal back into RGB signal of the PQ gamut.

$$\begin{bmatrix} R_{ca}' \\ G_{ca}' \\ B_{ca}' \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.0000 & 1.4746 \\ 1.0000 & -0.1645 & -0.5713 \\ 1.0000 & 1.8814 & -0.0001 \end{bmatrix} \cdot \begin{bmatrix} Y' \\ C_b' \\ C_r' \end{bmatrix} \qquad \text{……(89)}$$

6) Convert the RGB signal in the PQ gamut back to the linear gamut to get (R_{color1}, G_{color1}, B_{color1})

$$R_{color1} = PQ\_EOTF(R_{ca}')$$
$$G_{color1} = PQ\_EOTF(G_{ca}') \qquad \text{……(90)}$$
$$B_{color1} = PQ\_EOTF(B_{ca}')$$

7) $f_{color}[N_{frame}][0]= R_{color1}$, $f_{color}[N_{frame}][1]= G_{color1}$, $f_{color}[N_{frame}][2]= B_{color1}$

9.6 Post-Processing Process

Input: $f_{color}[N_{frame}][3]$ (the processed RGB gamut pixel buffer for the frame to be processed)
Output: $f_{process}[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed, once the HDR display tone mapping is complete)

Assign $f_{color}[N_{frame}][3]$, the processed RGB gamut pixel buffer for the frame to be processed, to $f_{process}[N_{frame}][3]$, the RGB gamut pixel buffer for the frame to be processed, once HDR display tone mapping is complete; $f_{process}[N_{frame}][0]=f_{color}[N_{frame}][0]$, $f_{process}[N_{frame}][1]=f_{color}[N_{frame}][1]$, $f_{process}[N_{frame}][2]=f_{color}[N_{frame}][2]$.

10 SDR Tone Mapping of PQ HDR

10.1 SDR Tone Mapping Process

Input: MaxDisplayPQ (maximum display luminance in the display luminance range of the mastering display in the PQ gamut, default value is 0.5081), MinDisplayPQ (minimum display luminance in the display luminance range of the mastering display in the PQ gamut, default value is 0), $f[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed), and metadata information

Output: $f_{process}[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed, once SDR tone mapping is complete)

The tone mapping process is as follows:

1) To generate base curve parameters, apply the base curve parameter acquisition process described in section 10.2.
2) To generate the cubic spline interval parameters, apply the cubic spline parameter acquisition process described in section 10.3.
3) Apply the color signal dynamic range conversion process described in section 9.4 to generate fTM[Nframe][3], which will be the RGB gamut pixel buffer for the frame to be processed, once dynamic range conversion is complete.
4) Apply the color correction process described in section 9.5 to generate fcolor[Nframe][3], which will be the processed RGB gamut pixel buffer for the frame to be processed.
5) Apply the SDR post-processing process described in section 10.4 to generate fprocess[Nframe][3] (the RGB gamut pixel buffer for the frame to be processed, once SDR tone mapping is complete). Nframe is the total sampling points for the frame to be processed.

10.2 Base Curve Parameter Acquisition Process

Input: metadata information

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The process is as follows:

1) From the metadata information, obtain the tone mapping mode flag tone_mapping_mode_flag and the base curve flag base_flag corresponding to target_system_display_maximum_luminance.
2) From the metadata information, obtain the max_lum and min_lum of the frame to be processed:
3) Based on maximum_maxrgb in the metadata information, obtain the max_lum.
4) Based on minimum_maxrgb in the metadata information, obtain the min_lum.
5) Update the max_lum of the frame to be processed in accordance with the procedure described in section 9.2.3
6) Calculate the base curve parameter set Ptone_mapping
   a).... If tone_mapping_mode_flag is 0, or tone_mapping_mode_flag is 1 and base_flag is 0, apply the base curve parameter acquisition process 0 described in section 9.2.2 to update the parameters, but you need to change the preset values $P_{valueH0}$, $P_{valueL0}$, TPH0 and TPL0 to 3.5 , 6.0, 0.6, and 0.1,

respectively, and change the preset values $P_{deltaH1}$, $P_{deltaL1}$, TPH1 and TPL1 to 0.6, 0.3, 0.75, and 0.67 respectively.

b) If tone_mapping_mode_flag is 1 and base_flag is 1:

If targeted_system_display_maximum_luminance = MaxDisplayPQ or if base_param_Delta_mode = 3, set m_p, m_a, m_m, m_n, m_b, K1, K2, and K3 to m_p_0, m_a_0, m_m_0, m_n_0, m_b, K1_0, K2_0, and K3_0, respectively.

Otherwise:

If base_param_Delta_mode is 0, or 2, or 4, or 6, apply the base curve parameter adjustment process 1 described in section 9.2.4 to update the parameters.

If base_param_Delta_mode is 1 or 5, apply the base curve parameter adjustment process 2 described in section 9.2.5 to update the parameters.

10.3 Cubic Spline Parameter Acquisition Process

Input: metadata information

Output: cubic spline mapping curve parameter set $P_{3spline}$.

The process is as follows:

1) Based on the tone mapping mode flag tone_mapping_mode_flag in the metadata information and the cubic spline mode flag 3Spline_flag corresponding to targeted_system_display_maximum_luminance, set the cubic spline interval parameter. If tone_mapping_mode_flag is 0 or tone_mapping_mode_flag is 1 and 3Spline_flag is 0, the cubic spline parameters 3Spline_num and 3Spline_TH_mode are not passed in the code stream, 3Spline_num is set to 1 and 3Spline_TH_mode is set to 0; if tone_mapping_mode_flag is 1 and 3Spline_flag is 1, get the corresponding parameters from the code stream.

2) Based on 3Spline_num, traverse every cubic spline interval to get cubic spline mapping curve parameter set $P_{3spline}$

   a) If tone_mapping_mode_flag is 0 or tone_mapping_mode_flag is 1 and 3Spline_flag is 0 and 3Spline_TH_mode is 0, or if tone_mapping_mode_flag is 1, 3Spline_flag is 1 and 3Spline_TH_mode is not 0, apply the linear spline curve parameter acquisition process 0 described in section 10.3.1.1, the linear spline curve parameter adjustment process 0 in section 9.3.2.3 and the cubic spline interval parameter acquisition process 0 in section 10.3.2.1 to get the parameter. The parameter value is the default value.

   If tone_mapping_mode_flag is 1, 3Spline_flag is 1 and 3Spline_TH_mode is 0, apply the linear spline curve parameter acquisition process 1 described in section 9.3.2.2 and the linear spline curve parameter adjustment process 0 in section 9.3.2.3 and the cubic spline interval parameter acquisition process 1 in section 9.3.3.2 to get the parameters. The parameter values are obtained from the code stream.

   b) If tone_mapping_mode_flag is 1, 3Spline_flag is 1 and 3Spline_TH_mode is 1/2/3, apply the cubic spline interval parameter acquisition process 2 described in section 9.3.3.3 to get the parameter. The parameter value is obtained from the code stream.

   If 3spline_num = 1, set 3Spline_num to 2, 3Spline_TH_mode[1]=3Spline_TH_mode[0] and stop traversing b.

10.3.1 Linear Spline Interval Parameter Acquisition Process

Input: metadata information

Output: the linear spline curve parameter set $P_{1spline}$, TH3[0], MB[0][0], and base_offset

Get the linear spline curve between the first endpoint and the first interpolation point TH3[0]:

$$F(L)=MB[0][0]\times L+base\_offset \quad \text{……(91)}$$

Where L is the independent variable within the interval [0, TH3[0]].

10.3.1.1 **Linear Spline Curve Parameter Acquisition Process 0**

Input: metadata information

Output: the linear spline curve parameter set $P_{1spline}$, TH3[0], MB[0][0], and base_offset

The process is as follows:

1) The linear spline parameter TH3[0]:

$$TH3[0] = 0 \quad \text{……(92)}$$

2) Based on the average_maxrgb (avgL) in the metadata, get the linear spline parameter MB[0][0]:

$$MB[0][0]=\begin{cases} S_{dmaxL6} & avgL > AVMAXH6 \\ S_{dmaxL6}\times g6(W6)+S_{dmaxH6}\times(1-g6(W6)) & avgL \geq AVMAXL3, avgL \leq AVMAXH6 \\ S_{dmaxH6} & avgL < AVMAXL6 \end{cases} \quad \text{……(93)}$$

base_offset=0

Where:

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

AVMAXH6 and AVMAXL6 are the thresholds corresponding to the average luminance value of the pixel in the bright area.AVMAXH6 is the upper threshold of the average luminance value of the pixel in the bright area, the default is 0.6; AVMAXL6 is the lower threshold of the average luminance value of the pixel in the bright area, the default is 0.3.

S_{dmaxL6} and S_{dmaxH6} are the thresholds for the slope at which the luminance value of pixel in the dark area is increased. S_{dmaxL6} is the lower threshold for the slope at which the luminance value of pixel in the dark area is increased, the default is 0.9; S_{dmaxH6} is the upper threshold for the slope at which the luminance value of pixel in the dark area is increased, the default is 1.0.g6() is $y=x^N$, and the default is y=x.

10.3.2 Cubic Spline Interval Parameter Acquisition Process

Get the first cubic spline curve between the first interpolation point TH1[n] and the second interpolation point TH2[n]:

$$F(L)=MD[0][1]\times(L-TH1[n])^3+MC[0][1]\times(L-TH1[n])^2+MB[0][1]\times(L-TH1[n])^1+MA[0][1] \quad \text{……(94)}$$

Where L is the independent variable within the interval [TH1[n], TH2[n]],

And the second cubic spline curve between the second interpolation point TH2[n] and the third interpolation point TH3[n]:

$$F(L)=MD[1][1]\times(L-TH2[n])^3+MC[1][1]\times(L-TH2[n])^2+MB[1][1]\times(L-TH2[n])^1+MA[1][1] \quad \text{……(95)}$$

Where L is the independent variable within the interval [TH2[n], TH3[n]], 0<n<=3Spline_num.

10.3.2.1 Cubic Spline Curve Parameter Acquisition Process 0

Input: metadata information, base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, k1, k2, and k3, linear spline curve parameter set $P_{1spline}$, consisting of TH3[0] and MB[0][0]

Output: the cubic spline interval parameter set $P_{3spline}$, consisting of 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 process is as follows:

1) Based on the data information in the metadata, get the three interpolation points (TH1[1], TH2[1], and TH3[1]) within the current cubic spline interval:

$$TH1\,[1] = TH3\,[0] \quad \text{……(96)}$$

$$TH2\,[1] = TH1\,[1] + B \quad \text{……(97)}$$

Where B is the offset corresponding to the luminance value of the pixel in the dark transition area, and the default value is 0.15.

$$TH3\,[1] = TH2\,[1] + C^{*}\,TH2\,[1] - D^{*}TH1\,[1] \quad \text{……(98)}$$

Where C and D are the weighting coefficients corresponding to the luminance value of the pixel in the luminance area, and the default value is 0.5.

2) Get the eight parameters 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] from TH1[1], TH2[1], and TH3[1]:

i. Calculate the second coordinates (Y coordinates) VA1, VA2 and VA3 corresponding to TH1[1], TH2[1], and TH3[1].

First, based on the linear spline function in TH1[0] through TH3[0] in a), get the following

$$F(L) = MB[0][0] \times L + base\_offset \quad \text{……(99)}$$

Then, set L to TH1[1] to calculate TH1[1]'s second coordinate VA1 using the above formula

$$VA1 = MB[0][0] \times TH1[1] + base\_offset \quad \text{……(100)}$$

And, based on the base curve parameter set $P_{tone\_mapping}$, get the following

$$H(L) = m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \text{……(101)}$$

Then, set L to TH3[1] to calculate TH3[1]'s second coordinate VA3 using the above formula

$$VA3 = m\_a \times \left(\frac{m\_p \times TH3[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH3[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \text{…… (102)}$$

Then, calculate TH2[1]'s second coordinate VA2:

$$VA2 = m\_a \times \left(\frac{m\_p \times TH2[1]^{m\_n}}{(K1 \times m\_p - K2) \times TH2[1]^{m\_n} + K3}\right)^{m\_m} + m\_b \quad \text{……(103)}$$

And VA1<VA2<VA3.

ii. Calculate MA[0][1] and MA[1][1]:

$$MA\,[0]\,[1] = VA1 \quad \text{……(104)}$$

$$MA\,[1]\,[1] = VA2 \quad \text{……(105)}$$

iii. Calculate MB[0][1]: calculate GD1, the first derivative of the formula (99) at TH1[1], so that MB[0][1]=GD1; Calculate GD3, the first derivative of the formula (101) at TH3[1]

$$MB[0][1] = GD1 = MB[0][0] \quad \text{……(106)}$$

$$GD3 = m\_a \times m\_m \times m\_p \times K3 \times m\_n \times TH3[1]^{m\_n-1} \times DGD3(L) \quad \text{……(107)}$$

Where:

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

iv. Calculate MC[0][1], MD[0][1], MB[1][1], MC[1][1], and MD[1][1]:

Calculate VA2[0], the value of the first cubic spline curve at TH2[1], so that VA2[0]=VA2: calculate VA3[0], the value of the second cubic spline curve at TH3[1], so that VA3[0]=VA3.

Calculate GD3[0], the derivative of the second cubic spline curve at TH3[1], so that GD3[0]=GD3.

Calculate GD2[0] and GD2[1], the first derivatives of the two cubic splines at TH2[1], so that GD2[0]=GD2[1]; calculate GGD2[0] and GGD2[1], the second derivatives of the two cubic splines at TH2[1], so that GGD2[0]=GGD2[1]:

$$\begin{cases} MD[0][1]\times(DTH2)^3+MC[0][1]\times(DTH2)^2+MB[0][1]\times(DTH2)^1+MA[0][1]=VA2 \\ MD[1][1]\times(DTH3)^3+MC[1][1]\times(DTH3)^2+MB[1][1]\times(DTH3)^1+MA[1][1]=VA3 \\ 3\times MD[1][1]\times(DTH3)^2+2\times MC[1][1]\times(DTH3)^1+MB[1][1]=GD3 \\ 3\times MD[0][1]\times(DTH2)^2+2\times MC[0][1]\times(DTH2)^1+MB[0][1]=GD2[0] \\ 3\times MD[1][1]\times(DTH3)^2+2\times MC[1][1]\times(DTH3)^1+MB[1][1]=GD2[1] \\ 6\times MD[0][1]\times(DTH2)^1+2\times MC[0][1]=GGD2[0] \\ 6\times MD[1][1]\times(DTH3)^1+2\times MC[1][1]=GGD2[1] \end{cases} \quad \ldots\ldots(108)$$

By analyzing and solving the above equations, get the values of MB[1][1], MC[0][1], MD[0][1], MC[1][1], and MD[1][1]:

$$\begin{cases} MB[1][1]=\frac{-(3.0*VA1*h2*h2+3.0*VA2*h1*h1-3.0*VA3*h1*h1-3.0*h2*h2*VA2+h1*h1*h2*GD3+GD1*h1*h2*h2)}{(2.0*h2*(h1*h1+h2*h1)} \\ MC[0][1]=\frac{(3.0*VA2-2.0*GD1*h1-3.0*VA1-MB[1][1]*h1)}{h1*h1} \\ MD[0][1]=\frac{(h1*GD1+h1*MB[1][1]+2*VA1-2.0*VA2)}{h1*h1*h1} \\ MC[1][1]=MC[0][1]+3.0*MD[0][1]*h1 \\ MD[1][1]=-\frac{(VA3-VA2-h2*GD3+MC[0][1]*h2*h2+3*MD[0][1]*h1*h2*h2)}{2*h2*h2*h2} \end{cases}$$

……(109)

Where:

$$h1=DTH2=(TH2[1]-TH1[1])$$
$$h2=DTH3=(TH3[1]-TH2[1])$$

10.4 SDR Post-Processing Process

Input: $f_{color}[N_{frame}][3]$ (the processed RGB gamut pixel buffer for the frame to be processed)
Output: $f_{process}[N_{frame}][3]$ (the RGB gamut pixel buffer for the frame to be processed after SDR tone mapping)
Perform reverse gamma processing (a gamma value of 2.2 is recommended) on $f_{color}[N_{frame}][3]$, the processed RGB gamut pixel buffer for the frame to be processed, and assign it to $f_{process}[N_{frame}][3]$, the RGB gamut pixel buffer for the frame to be processed, once SDR tone mapping is complete; $f_{process}[N_{frame}][0]=f_{color}[N_{frame}][0]$, $f_{process}[N_{frame}][1]=f_{color}[N_{frame}][1]$, $f_{process}[N_{frame}][2]=f_{color}[N_{frame}][2]$.

11 HLG Tone Mapping

11.1 Process of Tone Mapping of HLG HDR to a PQ HDR Display Device

It is recommended that tone mapping of HLG HDR to a PQ HDR display device be done with reference to section 7.2 of ITU-R BT.2390-8: *Conversion concepts using a reference condition at 1 000 cd/m²*.

11.2 Process of Tone Mapping of HLG HDR to a SDR Display Device

The input signal of the device is a 4:4:4 $Y_iCb_iCr_i$ non-linear video signal obtained by decoding and refactoring on the terminal and chroma upsampling. Each component is a digital coding value within the 10-bit limit range, where the value of Y_i should be within the interval [64, 940], while the values of Cb_i and Cr_i should be within the interval [64, 960]. The tone mapping process is shown in the figure below:

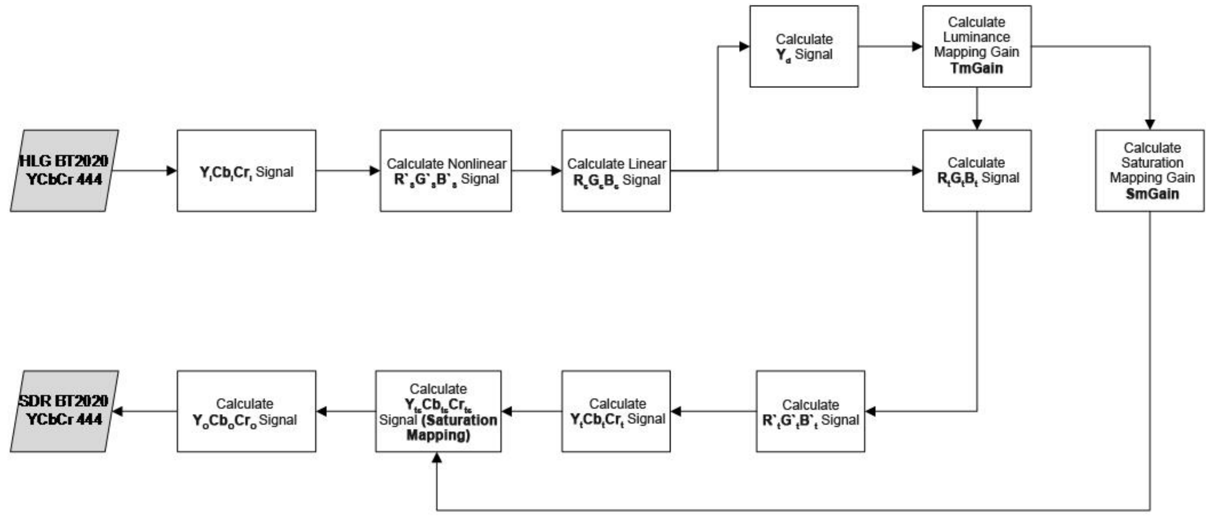

Figure 4. Process of compatible tone mapping of HDR HLG signal to SDR TV

The specific steps are as follows:

1) Calculate the nonlinear $R`_sG`_sB`_s$ signal

$$\begin{pmatrix} Y_{sf} \\ Cb_{sf} \\ Cr_{sf} \end{pmatrix} = \begin{pmatrix} \frac{1}{876} & 0 & 0 \\ 0 & \frac{1}{896} & 0 \\ 0 & 0 & \frac{1}{896} \end{pmatrix} \times \begin{pmatrix} Y_i - 64 \\ Cb_i - 512 \\ Cr_i - 512 \end{pmatrix} \quad \ldots\ldots(110)$$

$$\begin{pmatrix} R`_s \\ G`_s \\ B`_s \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.4746 \\ 1 & -0.1645 & -0.5713 \\ 1 & 1.8814 & 0 \end{pmatrix} \times \begin{pmatrix} Y_{sf} \\ Cb_{sf} \\ Cr_{sf} \end{pmatrix} \quad \ldots\ldots(111)$$

Where the $Y_iCb_iCr_i$ signal is the digital coding value within the 10-bit limit range, the resulting $R`_sG`_sB`_s$ after this processing is the floating-point nonlinear primary color value, and the value should be clipped to the interval [0, 1.0].

2) Calculate the linear $R_sG_sB_s$ signal and calculate the linear luminance Y_s of the input signal.

$$E_s = HLG\_OETF^{-1}(E`_s) \quad \ldots\ldots(112)$$

E_s in the equation represents the linear primary color value of any component of the $R_sG_sB_s$ signal, and its value is within the interval [0, 1]; $E`_s$ refers to the nonlinear primary color value of any component of the $R`_sG`_sB`_s$ signal. The function $HLG\_OETF^{-1}()$ is defined as follows, in accordance with ITU BT.2100:

$$HLG\_OETF^{-1}(E`) = \begin{cases} \frac{E`^2}{3} & 0 \le E` \le 1/2 \\ \frac{\left(exp\left(\frac{(E`-c)}{a}\right)+b\right)}{12} & 1/2 < E` \le 1 \end{cases} \quad \ldots\ldots(113)$$

Where a = 0.17883277, b =1-4a, c=0.5-a*ln(4a)

The linear luminance Y_s is calculated as follows:

$$Y_s = 0.2627R_s + 0.6780G_s + 0.0593B_s \quad \ldots\ldots(114)$$

Y_s is a real number, and its value is within the interval [0, 1.0].

3) Calculate the Y_t signal.

i. Calculate the display luminance Y_d:

$$Y_d = 1000(Y_s)^{1.2} \qquad \text{…… (115)}$$

ii. Calculate the nonlinear luminance Y_{dPQ}:

$$Y_{dPQ} = PQ\_EOTF^{-1}(Y_d) \qquad \text{…… (116)}$$

iii. Perform luminance mapping to get Y_{tPQ}:

$$Y_{tPQ} = f_{tm}(Y_{dPQ}) \qquad \text{…… (117)}$$

$f_{tm}(\ )$ in the equation is defined as follows:

$$f_{tm}(e) = \begin{cases} e, & \text{当} e \le KP1 \\ hmt(e), & \text{当} KP1 < e \le KP2 \\ maxDL, & \text{当} e > KP2 \end{cases} \qquad \text{…… (118)}$$

Where, $hmt(x) = y_0 \times \alpha_0(x) + y_1 \times \alpha_1(x) + y_0^{'}\beta_0(x) + y_1^{'}\beta_1(x)$

$$\begin{cases} \alpha_0(x) = \dfrac{(x_1 - 3x_0 + 2x)(x_1 - x)^2}{(x_1 - x_0)^3} \\ \alpha_1(x) = \dfrac{(3x_1 - x_0 - 2x)(x - x_0)^2}{(x_1 - x_0)^3} \\ \beta_0(x) = \dfrac{(x - x_0)(x - x_1)^2}{(x_1 - x_0)^2} \\ \beta_1(x) = \dfrac{(x - x_0)^2(x - x_1)}{(x_1 - x_0)^2} \end{cases}$$

e is the normalized original nonlinear luminance signal, $f_{tm}(e)$ is the normalized display nonlinear luminance signal, KP1 is the first threshold, the value of which is 0.5247, KP2 is the second threshold, the value of which is 0.7518, maxDL is the maximum nonlinear display luminance value of the mastering display and the value is 0.638285 (its corresponding linear luminance is 350cd/m^2), and maxSL is the highest nonlinear source luminance value, which is specified in this standard as 0.7518, x_0=KP1, x_1=maxSL, y_0=KP1, y_1=maxDL, $y_{0'}$=1, $y_{1'}$=0.

iv. Calculate the linear luminance Y_t after normalized luminance mapping:

$$Y_t = PQ\_EOTF\left(Y_{tPQ}\right) \qquad \text{…… (119)}$$

Therefore, the formula of Y_t is:

$$Y_t = PQ\_EOTF(f_{tm}(PQ\_EOTF^{-1}(1000(Y_s)^{1.2})) \qquad \text{…… (120)}$$

Y_t is a real number, and its value should be clipped to the interval [0, 350].

4) Calculate the luminance mapping gain TmGain.

The formula of the luminance mapping gain TmGain is as follows:

$$TmGain = \begin{cases} \frac{Y_t}{Y_s}, \ Y_s \neq 0 \\ 0, \ Y_s = 0 \end{cases} \quad (121)$$

5) Calculate the saturation gain SmGain.

The formula of the saturation mapping gain SmGain is as follows:

$$SmGain = \begin{cases} (\frac{Y_t}{1000*Y_s})^{smlift}, \ Y_s \neq 0 \\ 0, \ Y_s = 0 \end{cases} \quad ……(122)$$

Where the value of the variable smlift can be selected from the range of [0, 1.0], and the default value is 0.2.

6) Calculate SDR display $R_tG_tB_t$ signal.

$$E_t = (E_s \times TmGain)/350 \quad ……(123)$$

In the equation, E_s represents any component of the $R_sG_sB_s$ signal, and E_t represents any component of the $R_tG_tB_t$ signal.

The resulting $R_tG_tB_t$s after this processing is the floating-point linear primary color value, and the value should be clipped to the interval [0, 1.0], corresponding to the relative luminance value range of the SDR display device.

7) Calculate the $R`_tG`_tB`_t$ signal.

$$E`_t = (E_t)^{1/2.2} \quad ……(124)$$

In the equation, E_t represents any component of the $R_tG_tB_t$ signal, and $E`_t$ represents any component of the $R`_tG`_tB`_t$ signal.

The resulting $R`_tG`_tB`_t$ after this processing is the floating-point nonlinear primary color value, and the value should be clipped to the interval [0, 1.0].

8) Calculate the $Y_tCb_tCr_t$ signal.

$$\begin{pmatrix} Y_t \\ Cb_t \\ Cr_t \end{pmatrix} = \begin{pmatrix} 0.2627 & 0.6780 & 0.0593 \\ -0.1396 & -0.3604 & 0.5 \\ 0.5 & -0.4598 & -0.0402 \end{pmatrix} \times \begin{pmatrix} R`_t \\ G`_t \\ B`_t \end{pmatrix} \quad ……(125)$$

The value of the luminance signal Y_t should be within [0, 1.0], while the value of the chroma signal Cb_tCr_t should be within [-0.5, +0.5], and $R`_tG`_tB`_t$ is the floating-point nonlinear RGB primary color value obtained above.

9) Adjust the saturation, and calculate the $Y_{ts}Cb_{ts}Cr_{ts}$ signal.

$$\begin{pmatrix} Y_{ts} \\ Cb_{ts} \\ Cr_{ts} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & SmGain & 0 \\ 0 & 0 & SmGain \end{pmatrix} \times \begin{pmatrix} Y_t \\ Cb_t \\ Cr_t \end{pmatrix} \quad ……(126)$$

The value of the luminance signal Y_{ts} should be within [0, 1.0], while the value of the chroma signals Cb_{ts} and Cr_{ts} should be within [-0.5, +0.5], and $Y_tCb_tCr_t$ is the luminance and chroma signal obtained above.

10) Calculate the $Y_oCb_oCr_o$ signal.

$$\begin{pmatrix} Y_o \\ Cb_o \\ Cr_o \end{pmatrix} = \text{ROUND}\left(\begin{pmatrix} 876 & 0 & 0 \\ 0 & 896 & 0 \\ 0 & 0 & 896 \end{pmatrix} \times \begin{pmatrix} Y_{ts} \\ Cb_{ts} \\ Cr_{ts} \end{pmatrix} + \begin{pmatrix} 64 \\ 512 \\ 512 \end{pmatrix}\right) \qquad \text{……(127)}$$

The resulting $Y_oCb_oCr_o$ signal after this processing is a digital coding value within the 10-bit limit range, where the value of Y_o should be within the interval [64, 940], while the values of Cb_o and Cr_o should be within the interval [64, 960]. $Y_oCb_oCr_o$ is the YCbCr 444 video signal output of this proposal. ROUND denotes the round-off operation.

Appendix A
(Informative)
Dynamic Metadata Extraction Method

A.1 Overview

This appendix describes the recommended methods for the extraction of metadata in HDR pre-processing stage.

A.2: Tone mapping mode analysis for selecting the base curve flag.

A.3: Cubic spline parameter generation process for seting the cubic spline parameter.

A.4: Dynamic metadata time-domain filtering process for dynamic metadata pre-processing.

A.5: Time-domain quality intra-loop feedback of dynamic metadata.

A.6: Dynamic metadata variance_maxrgb_pq calculation.

A.2 Tone Mapping Mode Analysis

In the pre-processing stage, it is recommended to select the appropriate base curve flag tone_mapping_mode_flag based on the following scene analysis process and whether manual intervention is required or not:

1) Get the highest gamut pixel buffer f_{MAX} $[N_{frame}]$ of the current frame from the gamut pixel buffer $f[N_{frame}][3]$ for the frame to be processed.
2) Find the highest gamut pixel MaxSource of the whole frame from the gamut maximum pixel buffer $f_{MAX}[N_{frame}]$.
3) Obtain the histogram H[N] from f_{MAX} $[N_{frame}]$, where N is the number of gray intervals of the histogram. The total gray interval of the histogram is in the range from 0 to 1.0 in the PQ gamut.
4) Calculate the ratio of the gray interval LDARK of the histogram between 0 and 5 nits to the total pixels RDARK.
5) Calculate the ratio of the gray interval LBRIGHT of the histogram above the reference display luminance level to the total pixels RBRIGHT.
6) Use director mode, the tone mapping mode flag tone_mapping_mode_flag is 1.
7) If the current frame is a scene-change frame, and if tone mapping mode flag tone_mapping_mode_flag is 1, then, based on histogram information, choose a generation process. Otherwise, use the same generation process as the last frame:

   If $R_{DARK}>=q\times L_{DARK}$, $R_{BRIGHT}>=w\times L_{BRIGHT}$, where q and w are gain coefficients, then apply the process described in A.2.1 to set the model parameters.

   Otherwise, if $R_{BRIGHT}>=w\times L_{BRIGHT}$, where w is a gain coefficient, then apply the process described in A.2.4 to set the model parameters.

   Otherwise, if $R_{DARK}>=q\times L_{DARK}$, where q is a gain coefficient, then apply the process described in A.2.3 to set the model parameters.

   Otherwise, apply the process described in A.2.2 to set the model parameters.

A.2.1 Base curve mode process 1

Input: MaxDisplayPQ (the maximum display luminance of the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (the minimum display luminance of the display luminance range of the mastering display in the PQ gamut), f[Nframe][3] (the RGB gamut pixel buffer for the frame to be processed)

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The steps are as follows:

1) Set m_m, m_n, K1, K2, and K3 to the values 2.4, 1, 1, 1, 1.
2) Set m_b to 0.
3) Based on m_m, m_n, m_b, K1, K2, and K3, get

$$m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(m\_p - 1) \times L^{m\_n} + 1}\right)^{m\_m} + m\_b$$

After the above parameters are set, the curve becomes

$$m\_a \times \left(\frac{m\_p \times L}{(m\_p - 1) \times L + 1}\right)^{2.4} + m\_b$$

4) L3 is equal to average luminance value (the value in PQ gamut) in the metadata information, and F3N is equal to the average target luminance value (the value in PQ gamut) on the target display. The calculation is:

$$L3 = F3N = \frac{\sum_{i=0}^{N_{frame}} f(i)}{N_{frame}}$$

Where f(i) is the highest value among f[i][0], f[i][1] and f[i][2], which is greater than dark vision threshold (minCone, currently set to 0.15, for which reference to the lower limit of the color sensing cone cells of human eyes is recommended) and smaller than medium light threshold (midLight, currently set to 0.35, for which reference to the lower limit of the skin color range in the BT2408 standard is recommended).

5) M1 is is the average value in medium light region – average_midLight (the value in PQ gamut) in the metadata information, and N1N is the perceived luminance Perceprual_midLight (the value in PQ gamut) of the frame to be processed displayed on the target display. The calculation is:

First, the resulting pixel histogram distribution by calculation is greater than the interval HISA of Half_Num. The size of the HISA interval is (MaxSource-MinCone)*V/U, where U and V are positive integers. It is recommended that U be set to 6, V be less than or equal to 3, and Half_Num be the current number of pixels covering the MAXRGB value (midLight, defusingLight).

Then calculate F3, the average value of the pixels in the HISA interval displayed on the target display, and make it the perceived luminance Perceprual_midLight (the value in PQ gamut) of the frame to be processed displayed on the target display:

$$F3 = \frac{\sum_{i=0}^{N_{frame}} q1(i)}{N_{frame}}$$

Where

$$q1(i) = \begin{cases} \text{MaxDisplayPQ} & f(i) \geq \text{MaxDisplayPQ} \\ f(i) & other \end{cases}$$

Where f(i) is the highest value among f[i][0], f[i][1] and f[i][2], which is greater than midLight (currently set to 0.35, for which reference to the lower limit of the skin color range in the BT2408 standard is recommended) and less than defusingLight (set to midLight +(MaxSource-midLight)*ratio, where the ratio is set to 4/6, which is approximately 0.62 to 0.72 for sources from 1000 nits to 4000 nits. Reference to the lower limit of the diffuse reflection white range in the BT2408 standard is recommended).

Then, if HISA is not found, calculate the average value F4 of the pixels of the current frame within (midflight, defusingLightH) displayed on the target display, and make it the perceived luminance Perceprual_midLight (the value in PQ gamut) of the frame to be processed displayed on the target display:

$$F4 = \frac{\sum_{i=0}^{N_{frame}} q2(i)}{N_{frame}}$$

Where

$$q2(i) = \begin{cases} \text{MaxDisplayPQ} & \text{f(i)} \geq \text{MaxDisplayPQ} \\ \text{f(i)} & other \end{cases}$$

Where f(i) is the highet value among f[i][0], f[i][1] and f[i][2], f(i) is greater than midLight (currently set to 0.35, for which reference to the lower limit of the skin color range in the BT2408 standard is recommended) and f(i) is less than defusingLightH (set to midLight +(MaxSource-midLight)*ratioH, where the ratioH is set to 5/6, which is approximately 0.68 to 0.81 for sources from 1000 nits to 4000 nits).

6) Based on (M1, N1N), (L3, F3N), (MinSource, MinDisplayPQ), (MAXSource, MAXDisplayPQ), and the obtained (M1, N1N-M1*MaxDisplay*E/(MaxSource*10)) and (L3, F3N*MaxDisplayPQ*F/(MaxSource*10), construct curve segments with MinSource, M1, and L3 as their starting points respectively to estimate the local contrast change of the current frame, and calculate the final (M1, N1) and (L3, F3) using E and F, which have the maximum contrast.
7) Substitute (M1, N1) and (L3, F3) to get

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

$$m\_a \times \left(\frac{m\_p \times L3^{m\_n}}{(m\_p - 1) \times L3^{m\_n} + 1}\right)^{m\_m} + m\_b = \text{F3}$$

Expansion equation:

$$m\_p = 1 + \left(\left((N1/F3)^{\frac{1}{m\_m}} \times L3 - M1\right) \middle/ \left(M1 \times L3 \times \left(1 - (N1/F3)^{\frac{1}{m\_m}}\right)\right)\right)$$

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

A.2.2 Base Curve Mode Process 2

Input: MaxDisplayPQ (the maximum display luminance of the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (the minimum display luminance of the display luminance range of the mastering display in the PQ gamut), f[Nframe][3] (the RGB gamut pixel buffer for the frame to be processed)

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The steps are as follows:

1) Set m_m, m_n, K1, K2, and K3 to the values 2.4, 1, 1, 1, 1.
2) Set m_b to PQ (MinDisplayPQ)
3) Based on m_p, m_m, m_n, m_b, K1, K2, and K3, get

$$m\_a \times \left(\frac{m\_p \times L^{m\_n}}{(m\_p - 1) \times L^{m\_n} + 1}\right)^{m\_m} + m\_b$$

4) L3 is equal to MaxSource, which is the highest RGB correction value of the frame to be processed, and F3 is equal to MaxDisplayPQ (the maximum display luminance of the display luminance range of the mastering display in the PQ gamut).
5) L2 is equal to average_maxrgb in the metadata information, and F2 is equal to the average luminance (the value in PQ gamut) of the frame to be processed displayed on the target display. The calculation is:

$$F2 = \frac{\sum_{i=0}^{N_{frame}} q(i)}{N_{frame}}$$

Where

$$q(i) = \begin{cases} \text{MaxDisplayPQ} & \text{f(i)} \geq \text{MaxDisplayPQ} \\ \text{MinDisplayPQ} & \text{f(i)} \leq \text{MinDisplayPQ} \\ \text{f(i)} & \text{Else} \end{cases}$$

Where f(i) is the highest value among f[i][0], f[i][1] and f[i][2].
6) L1 is equal to Perceprual_1nit in the metadata information (refer to A.2.5), F1 is equal to PQ_EOTF^{-1} (0.001).
7) If L2<PQ_EOTF^{-1} (0.005), then M1=L1, N1=F1, otherwise M1=L2, N1=F2.
8) Substitute (M1, N1) and (L3, F3) to get

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

$$m\_a \times \left(\frac{m\_p \times L3^{m\_n}}{(m\_p - 1) \times L3^{m\_n} + 1}\right)^{m\_m} + m\_b = \text{F3}$$

Expansion equation:

$$m\_p = 1 + \left(\left((N1/F3)^{\frac{1}{m\_m}} \times L3 - M1\right) \Big/ \left(M1 \times L3 \times \left(1 - (N1/F3)^{\frac{1}{m\_m}}\right)\right)\right)$$

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

A.2.3 Base Curve Mode Process 3

Input: MaxDisplayPQ (the maximum display luminance of the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (the minimum display luminance of the display luminance range of the mastering display in the PQ gamut), f[Nframe][3] (the RGB gamut pixel buffer for the frame to be processed), metadata information

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The steps are as follows:

1) Set m_m, m_n, K1, and K2, to the values 2.4, 1, 1, 1
2) Set m_b to PQ (MinDisplayPQ)
3) Set K3 to MaxSource, where MaxSource is equal to the max_lum (the value in PQ gamut) of the frame to be processed
4) Set m_a to MaxDisplayPQ − MinDisplayPQ
5) J(x) represents the cumulative histogram. Tp represents the threshold for the detailed luminance loss in the low luminance area (the value in PQ gamut), and the default value of Tp is 0.15 (the value in PQ gamut for 1nit). Use the following formula to get Tp' (if the tone mapping input is Tp', then the output is Tp)

$$T_p^{'} = \mathrm{T_p}\frac{\mathrm{MaxSource}}{MaxDisplay}$$

Then estimate the tone loss caused by tone mapping

v = J(Tp') − J(Tp)

Finally calculate the value of m_p:

m_p = cv+d

c and d can be determined dynamically; the preset value is c=7, d=3.

A.2.4 Base Curve Mode Process 4

Input: MaxDisplayPQ (the maximum display luminance of the display luminance range of the mastering display in the PQ gamut), MinDisplayPQ (the minimum display luminance of the display luminance range of the mastering display in the PQ gamut), f[Nframe][3] (the RGB gamut pixel buffer for the frame to be processed)

Output: the base curve parameter set $P_{tone\_mapping}$, consisting of m_p, m_m, m_n, m_a, m_b, K1, K2, and K3

The steps are as follows:

1) Set m_m, m_n, K1, K2, and K3 to the values 1, 1/2.4, 1, 1, and 1, respectively.
2) Set m_b to MinDisplayPQ
3) Based on m_p, m_m, m_n, m_b, K1, K2, and K3, get

$$m\_\mathrm{a} \times \left(\frac{m\_p \times L^{\mathrm{m\_}n}}{(m\_p - 1) \times L^{m\_n} + 1}\right)^{m\_m} + m\_b$$

4) L3 is equal to MaxSource (Qp gamut), which is the highest RGB correction value for the frame to be processed, F3 is equal to the maximum display luminance MaxDisplayPQ of the display luminance range of the mastering display.
5) L2 is equal to average_maxrgb in the metadata information (the value in PQ gamut), F2 is equal to the average luminance (the value in PQ gamut) of the frame to be processed displayed on the target display. The calculation is:

$$F2 = \frac{\sum_{\mathrm{i}=0}^{N_{frame}} q(i)}{N_{frame}}$$

Where

$$q(i) = \begin{cases} \mathrm{MaxDisplayPQ} & \mathrm{f(i)} \geq \mathrm{MaxDisplayPQ} \\ \mathrm{MinDisplayPQ} & \mathrm{f(i)} \leq \mathrm{MinDisplayPQ} \\ \mathrm{f(i)} & Else \end{cases}$$

Where f(i) is the highest value among f[i][0], f[i][1] and f[i][2].

6) L3 is equal to Perceprual_1nit in the metadata information (refer to A.2.5), F1 is equal to $\mathrm{PQ\_EOTF}^{-1}$ (0.001)
7) If L2<$\mathrm{PQ\_EOTF}^{-1}(5)$, then M1=L1, N1=F1, otherwise M1=L2, N1=F2
8) Substitute (M1, N1) and (L3, F3) to get

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

$$\mathrm{m\_a} \times \left(\frac{\mathrm{m\_p} \times \mathrm{L3}^{\mathrm{m\_n}}}{(\mathrm{m\_p} - 1) \times \mathrm{L3}^{\mathrm{m\_n}} + 1}\right)^{\mathrm{m\_m}} + \mathrm{m\_b} = \mathrm{F3}$$

A.2.5 The Calculation Method for Perceprual_1nit

Calculated by J(Lp)–J(1nit) = （J(L0)-J(1nit))* Rate, where Lp is the perceived luminance value, L0 is the preset luminance value, Rate is the preset ratio, J(x) is the ratio of pixels of which the luminance is less than x nits in the frame of the source image signal to be processed.
L0 is set to 5 nits by default, and the preset ratio Rate is set to 30%.

A.3 Cubic Spline Parameter Generation

A.3.1 Cubic Spline Parameter Generation Mode 1

Input: His[Ngray] (the RGB gamut pixel histogram buffer for the frame to be processed)
Output: the cubic spline interval parameters TH1, TH2, TH3, TH_strength, TH1_high, TH2_high, TH3_high and TH_high_strength
The steps are as follows:

1) Set TH1, TH2, and TH3 to 0.15, 0.25, and 0.35.
2) TH2 is equal to the average value within the interval (TH1, TH3) (the value in PQ gamut):

$$L3 = F3N = \frac{\sum_{i=0}^{N_{frame}} f(i)}{N_{frame}}$$

3) Set TH1_high, TH2_high and TH3_high to (TH3+(MaxSource-TH3)*(U-2)/U), (TH3+(MaxSource-TH3)*(U-1)/U) and MaxSource.
4) Calculate highRatio, the ratio of the number of pixels within the interval (TH1_high, TH3_high) to the number of total pixels, calculate wholeRatio, the ratio of the length of the interval (TH1_high, TH3_high) to the MaxSource, and update TH1_high to (TH3+((MaxSource-TH3)*(U-2)/U)-(pow(highRatio/wholeRatio, 0.5)*(MaxSource-TH3)/U)).
5) Apply the A3.2 cubic spline parameter generation optimization process, take TH1_high and TH3_high as the input parameters, and output TH2_high and TH_high_strength.

A.3.2 Cubic Spline Parameter Generation Optimization Process

Input: His[Ngray] (the cubic spline interval parameters TH1 and TH3, the RGB gamut pixel histogram buffer for the frame to be processed)
Output: the cubic spline interval parameters TH2, TH_strength
The steps are as follows:

1) Divide (TH1, TH3) into N intervals of equal size. It is recommended that N be set to 8.
2) Based on the RGB gamut pixel histogram buffer His[Ngray] for the frame to be processed, where n is within N/4 through N*3/4, calculate the interval n_min, the n interval which has the smallest number of pixels.
3) Set TH2 to TH1+(TH3-TH1)*n_min/8+(TH3-TH1)/16.
4) Based on the RGB gamut pixel histogram buffer His[Ngray] for the frame to be processed, calculate Num1 and Num2, the number of pixels in the intervals (TH1, TH2) and (TH2, TH3), respectively. If Num1 < Num2, TH_strength+=-0.2. If 2*Num1 < Num2, TH_strength+=-0.4.

A.4 Time-domain Filtering of Dynamic Metadata

The process for time-domain filtering of the dynamic metadata extracted from the current frame includes the following steps:

1) Get the original dynamic metadata of the current frame from the gamut pixel buffer f[Nframe][3] for the frame to be processed.
2) Perform scene change detection for the current frame. If the scene is changed, clear the dynamic metadata queue.
3) Save the original dynamic metadata of the current frame in the dynamic metadata queue metadata_fifo. This dynamic metadata queue is managed in a first-in first-out way.
4) Calculate the average value of the dynamic metadata elements in each dynamic metadata candidate in the dynamic metadata queue, and take each average value of the dynamic metadata elements as the updated dynamic metadata respectively.

A.5 Time-domain Quality Intra Loop Adjustment Feedback of Dynamic Metadata

The frame-level quality evaluation algorithm is embedded in the process of generating metadata on the content generation side. The specific process is as follows:

1) For the Nth frame f(N) in the scene, use the dynamic metadata generation algorithm to generate metadata.
2) Apply the processes described in chapter 9, chapter 10, and chapter 11 to perform tone mapping against the preset configuration (the recommended configurations include 1000 nits, 500 nits, SDR), and get fTM1(N), fTM2(N), and fTMsdr(N).
3) Using the quality evaluation algorithm a, evaluate DTM1, DTM2 and DTMSDR, the subjective distortions of fTM1(N), fTM2(N) and fTMsdr(N), respectively .
4) If N>0, use the metadata of the frame at (N-M, N) in the M frame window, the curve parameters used for tone mapping, and reduced contrast and abnormal enhancement in the subjective distortion DTM1, DTM2 and DTMSDR to adjust the metadata generation process and the curve parameters, denoted as deltaC[N].
5) If the subjective distortion > DT, pass the curve parameters of the current frame using the director mode; if the subjective distortion is less than or equal to DT, pass the curve parameters of the current frame using the automatic mode.
6) For the Nth frame in the scene, if the director mode is used, use the dynamic metadata generation algorithm and deltaC[N] to generate metadata.

Note: For the quality evaluation algorithm in c), refer to Rafał Mantiuk, Kil Joong Kim, Allan G. Rempel and Wolfgang Heidrich. In: ACM Transactions on Graphics (Proc. of SIGGRAPH'11), 30(4), article no. 40, 2011.

A.6 The Calculation Method For Average_maxrgb_pq of Dynamic Metadata

Average_maxrgb_pq is mainly used to indicate the average value of luminance of the main content of the current frame or the current scene. The specific calculation is as follows:

1) Calculate the MAXRGB, which is the highest value among the R, G, and B values of all pixels in the current frame or the current scene, and convert it to the linear MAXRGB_LIN using formula (13).
2) Calculate the average value MAXRGB_LIN_AVG of the MAXRGB_LIN values of all pixels in the current frame or the current scene.
3) Convert MAXRGB_LIN_AVG to the nonlinear value average_maxrgb_pq using formula (12).

A.7 The Calculation Method For Variance_maxrgb_pq of Dynamic Metadata

Variance_maxrgb_pq is mainly used to indicate the luminance variation range of the main content of the current frame or the current scene. The specific calculation is as follows:

1) Calculate the MAXRGB, which is the highest value among the R, G, and B values of all pixels in the current frame or the current scene.
2) Based on the MAXRGB values of all pixels in the current frame or the current scene, get MAXRGB_A, so that 10% of the pixels' MAXRGB in the current frame or scene is less than or equal to the MAXRGB_A.
3) Based on the MAXRGB values of all pixels in the current frame or the current scene, get MAXRGB_B so that 90% of the pixels' MAXRGB in the current frame or scene is less than or equal to the MAXRGB_B.
4) variance_maxrgb_pq = (MAXRGB_B - MAXRGB_A).
5) If MAXRGB_B and MAXRGB_A are linear RGB values, convert variance_maxrgb_pq to a nonlinear value using formula (12).

Appendix B
(Informative)
The Methods for Processing Input Source

B.1 Overview

This appendix describes the recommended methods for processing input content in different formats.
If the source is PQ and the PQ data transmission scheme is adopted in the system, the processing is performed as described in B.2; if the source is HLG and the HLG data transmission scheme is adopted in the system, the processing is performed as described in B.3.

B.2 HDR10

Pre-processing: if the input content is an HDR10 video, only the dynamic metadata is extracted, the original static metadata in the HDR10 video can be used directly for the static metadata needed.
Transmission: the PQ HDR video and the metadata
Tone mapping: based on the PQ HDR video and the static metadata contained in the HDR10 signal as well as the dynamic metadata extracted from the pre-processing, perform the HDR display tone mapping of the PQ video in accordance with the procedure described in chapter 9, and perform the SDR display tone mapping of the PQ video as described in chapter 10.

B.3 HLG

Pre-processing: When the input content is HLG video, two processing methods are available:
a) When static display is used for processing, it is not necessary to extract the dynamic metadata.
b) Convert the content in HLG format into PQ format in accordance with BT.2390, then extract the dynamic metadata, but the content in the original HLG format is still sent during transmission.

Transmission: HLG HDR video and metadata/HLG HDR video
Tone mapping: If dynamic metadata is received, and system_start_code is 0x2, first convert the content in HLG format to PQ format in accordance with BT.2390, and then perform the HDR display tone mapping of the PQ video in accordance with the procedure described in chapter 9, and perform the SDR display tone mapping of the PQ video as described in chapter 10. If no dynamic metadata is received, perform the PQ tone mapping of the HLG video as described in section 11.1, and perform SDR tone mapping of the HLG video as described in section 11.2. HLG display devices use HLG video directly.

Appendix C
(Informative)
Gamut Mapping for Terminal Tone Mapping

This appendix describes the recommended methods for gamut mapping for terminal tone mapping.

When the terminal device needs to convert the HDR video signal of the BT.2020 gamut to the video signal output of the BT.709 gamut, this can be handled in accordance with the methods listed in the report ITU-R BT.2407-0 (2 Simple conversion from BT.2020 to BT. 709 based on linear matrix transformation).

Appendix D
References

[1] GB/T 33475.2-2016 Information technology—High efficiency media coding—Part 2: Video
[2] GY/T 307-2017 Parameter values for ultra-high definition television systems for production and programme exchange
[3] SMPTE ST 2084 High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays