ETSI TS 103 433 V1.1.1 (2016-08)

TECHNICAL SPECIFICATION

High-Performance Single Layer Directly Standard Dynamic Range (SDR) Compatible High Dynamic Range (HDR) System for use in Consumer Electronics devices (SL-HDR1)

EBU

OPERATING EUROVISION

Reference

DTS/JTC-036

Keywords

broadcasting, content, digital, distribution, HDR, HDTV, UHDTV, video

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

Motivation

Today High Efficiency Video Coding (HEVC) enables first Ultra HD broadcast services (also referred as "4K" resolution) via existing DVB specifications. Recently some High Dynamic Range (HDR) standards have been released by SMPTE (SMPTE ST 2084 [1] and SMPTE ST 2086 [2]). However, they define an HDR video signal that is not directly compatible with Standard Dynamic Range (SDR) Consumer Electronics (CE) devices. Thus, these devices require upstream external processing adapting the HDR video signal to a supported video format in order to render the video signal. Additionally, existing production and distribution infrastructures as well as play out may not be compatible with the SMPTE HDR standards with respect to carriage and signalling of the metadata in these standards.

The HDR system specified in the present document addresses direct backwards compatibility i.e. it leverages SDR distribution networks and services already in place and that enables high quality HDR rendering on HDR-enabled CE devices including high quality SDR rendering on SDR CE devices. Requirement for the present solution is that it is single layer to ensure that bit rate overhead for HDR and implementation complexity in CE devices will be low.

Pre-processing

At the distribution stage, an incoming HDR signal is decomposed in an SDR signal and content-dependent dynamic metadata. This stage is called "HDR-to-SDR decomposition", "HDR decomposition" or simply "decomposition". The SDR signal is encoded with any distribution codec (e.g. HEVC or AVC as respectively specified in Annex A and Annex B) and carried throughout the existing SDR distribution network with accompanying metadata conveyed on a specific channel or embedded in an SDR bitstream. The dynamic metadata can for instance be carried in an SEI message when used in conjunction with an HEVC or AVC codec. The HDR-to-SDR pre-processor that produces dynamic metadata is not a normative requirement of the present document. Nonetheless, the pre-processor is expected to produce a dynamic metadata stream matching the syntax described in clause 6.

Post-processing

The post-processing stage is functionally the inverse of the pre-processing stage and is called "SDR-to-HDR reconstruction", "HDR reconstruction" or just "reconstruction". It occurs just after SDR bitstream decoding. The post-processing takes as input an SDR video frame and associated dynamic metadata in order to reconstruct an HDR picture, as specified in clause 6, to be presented to the HDR compliant rendering device.

Structure of the present document

The present document is structured as follows. Clause 1 provides the scope of the current document. Clause 2 provides references used in the present document. Clause 3 gives essential definitions, symbols and abbreviations used in the present document. Clause 4 provides information on the end to end system. Clause 5 details the architecture of the HDR system. Clause 6 specifies the format of the content-based dynamic metadata that are produced during the HDR-to-SDR decomposition stage and that enable reconstruction of the HDR signal from the decoded SDR signal and those metadata. Clause 7 specifies the reconstruction process of the HDR signal. The dynamic metadata format specified in clause 6 is normatively mapped from HEVC and AVC specifications respectively in Annex A and Annex B. Finally, informative Annex C, Annex D and Annex E provide information on an HDR-to-SDR decomposition process, an inverse gamut mapping process and display adaptation.

1 Scope

The present document specifies the SDR-to-HDR content-based dynamic metadata and the post-decoding process enabling reconstruction of an HDR signal from an SDR signal and the specified metadata. This reconstruction process is typically invoked in a Consumer Electronics device such as a TV set, a smartphone, a tablet, or a Set Top Box. Besides, it provides information and recommendations on the usage of the described HDR system.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] SMPTE ST 2084:2014: "High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays".

[2] SMPTE ST 2086:2014: "Mastering Display Color Volume Metadata Supporting High Luminance and Wide Color Gamut Images".

[3] Recommendation ITU-T H.264 (02-2016): "Advanced video coding for generic audiovisual services".

[4] Recommendation ITU-T H.265 (04-2015): "High efficiency video coding".

[5] SMPTE ST 2094-30:2016: "Dynamic Metadata for Color Volume Transform - Application #3".

[6] SMPTE ST 2094-20:2016: "Dynamic Metadata for Color Volume Transform - Application #2".

[7] SMPTE RP 2077:2013: "Full Range Image Mapping".

[8] SMPTE RP 431-2:2011: "D-Cinema Quality - Reference Projector and Environment".

[9] Recommendation ITU-R BT.709-6 (06-2015): "Parameter values for HDTV standards for production and international programme exchange".

[10] Recommendation ITU-R BT.2020-2 (10-2015): "Parameter values for ultra-high definition television systems for production and international programme exchange".

[11] Recommendation ITU-R BT.1886 (03-2011): "Reference electro-optical transfer function for flat panel displays used in HDTV studio production".

[12] ISO 11664-1:2007 (CIE S 014-1/E:2006): "Colorimetry - Part 1: CIE standard colorimetric observers".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] CTA Standard CTA-861.3, January 2015: "HDR Static Metadata extensions".

[i.2] Recommendation ITU-R BT.2035.

3 Definitions, symbols, abbreviations and conventions

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

colour correction: adjustment of the luma and chroma components of a signal derived from the HDR signal in order to avoid hue shift and preserve the colour look of the HDR signal in the SDR signal

colour volume: solid in colorimetric space containing all possible colours a display can produce

display adaptation: adaptation of a video signal to the characteristics of the targeted Consumer Electronics display (e.g. maximum luminance of the CE display)

dynamic metadata: metadata that can be different for different portions of the video and can change at each associated picture

High Dynamic Range (HDR) system: system specified and designed for capturing, processing, and reproducing a scene, conveying the full range of perceptible shadow and highlight detail, with sufficient precision and acceptable artifacts, including sufficient separation of diffuse white and specular highlights

luminance mapping: adjustment of the luminance representative of a source signal to the luminance of a targeted system

Standard Dynamic Range (SDR) system: system having a reference reproduction using a luminance range constrained by Recommendation ITU-R BT.2035 Section 3.2

NOTE: Typically no more than 10 stops.

Supplemental Enhancement Information (SEI) message: carriage mechanism defined in Recommendation ITU-T H.264 and Recommendation ITU-T H.265 that is intended to assist in processes related to decoding, display or other purposes

Wide Colour Gamut (WCG): chromaticity gamut larger than the chromaticity gamut defined by Recommendation ITU-R BT.709-6

3.2 Symbols

3.2.1 Arithmetic operators

For the purposes of the present document, the following arithmetic operators apply:

+	Addition
−	Subtraction (as a two-argument operator) or negation (as a unary prefix operator)

$\times$	Multiplication, including matrix multiplication
x^y	Exponentiation. Specifies x to the power of y. In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.
/	Integer division with truncation of the result toward zero. For example, 7/4 and - 7/- 4 are truncated to 1 and - 7/4 and 7/- 4 are truncated to - 1.
$\div$	Used to denote division in mathematical equations where no truncation or rounding is intended.
$\frac{x}{y}$	Used to denote division in mathematical equations where no truncation or rounding is intended.
$x \% y$	Modulus. Remainder of x divided by y, defined only for integers x and y with x ≥ 0 and y > 0.

3.2.2 Bit-wise operators

For the purposes of the present document, the following bit-wise operators are defined as follows:

$x >> y$	Arithmetic right shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values y. Bits shifted into the most significant bits (MSBs) as a result of the right shift have a value equal to the MSB of x prior to the shift operation.
$x << y$	Arithmetic left shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values y. Bits shifted into the least significant bits as a result of the left shift have a value equal to 0.

3.2.3 Mathematical functions

For the purposes of the present document, the following mathematical functions apply:

Abs(x) $\begin{cases} x & , \quad x \geq 0 \\ -x & , \quad x < 0 \end{cases}$

Clip3(x, y, z) $\begin{cases} x & , \quad z < x \\ y & , \quad z > y \\ z & , \quad \text{otherwise} \end{cases}$

Flab(x) $\begin{cases} x^{\frac{1}{3}} & , \quad x > \left(\frac{6}{29}\right)^3 \\ \left(\frac{x}{3}\right) \times \left(\frac{29}{6}\right)^2 + \left(\frac{4}{29}\right) & , \quad x \leq \left(\frac{6}{29}\right)^3 \end{cases}$

Floor(x) the largest integer less than or equal to x.

log10(x) the base-10 logarithm of x.

Min(x ; y) $\begin{cases} x & , \quad x \leq y \\ y & , \quad x > y \end{cases}$

Max(x ; y) $\begin{cases} x & , \quad x \geq y \\ y & , \quad x < y \end{cases}$

Sqrt(x) $\sqrt{x}$

x = y..z x takes on integer values starting from y to z, inclusive, with x, y, and z being integer numbers and z being greater than y.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AVC	Advanced Video Coding
CE	Consumer Electronics
CIE	Commission Internationale de l'Eclairage
CLVS	Coded Layer-wise Video Sequence
CRI	Colour Remapping Information
CVRI	Colour Volume Reconstruction Information
EOTF	Electro-Optical Transfer Function
GBR	Green Blue Red colour model
HDMI	High-Definition Multimedia Interface
HDR	High Dynamic Range
HEVC	High Efficiency Video Coding
IRD	Integrated Receiver Decoder
LUT	Look-Up Table
MDCV	Mastering Display Colour Volume
MSB	Most Significant Bit
OETF	Opto-Electrical Transfer Function
RGB	Red Green Blue colour model
SDR	Standard Dynamic Range
SEI	Supplemental Enhancement Information (as in AVC and HEVC)
SMPTE	Society of Motion Picture and Television Engineers
STB	Set Top Box
VUI	Video Usability Information

3.4 Conventions

Unless otherwise stated, the following convention regarding the notation is used:

- Variables specified in the present document are indicated by bold Arial font 9 points camel case style e.g. **camelCase**. All those variables are described in clause 6.
- Internal variables of the present document are indicated by italic Cambria math font 10 points style e.g. $variable$.
- Syntactic elements structures or variables structures are indicated by Arial font 9 points C-style with parentheses e.g. structure_of_variables( ). Those structures are defined in clause 6, Annex A and Annex B.
- Bitstream syntactic elements are indicated by bold Arial font 9 points C-style e.g. **syntactic_element**. All those variables are defined in Annex A and Annex B.
- Functions are indicated as $func(\ x\ )$.
- Tables are indicated as $table[\ idx\ ]$.

4 End-to-end system

Figure 1 shows an end-to-end workflow supporting content production and delivery to HDR and legacy SDR displays. This HDR provision workflow primary goal is to provide direct SDR backward compatible services i.e. services which associated streams are directly compatible with SDR Consumer Electronics devices. This workflow is based on technologies and standards that facilitate an open approach.

It includes a single-layer SDR/HDR encoding-decoding, and uses standardized metadata:

- Mastering Display Colour Volume (MDCV) standardized in AVC [3], HEVC [4] and SMPTE ST 2086 [2] specifications;

- Colour Remapping Information (CRI) standardized in HEVC specification [4] and that conforms to SMPTE ST 2094-30 [5] specification; and
- Colour Volume Reconstruction Information (CVRI) based on SMPTE ST 2094-20 [6] specification.

Single-layer encoding/decoding requires only one encoder instance at HDR encoding side, and one decoder instance at player/display side. It supports the real-time workflow requirements of broadcast applications. The system uses dynamic metadata conveyed either in:

- CVRI, offering a very low payload or bandwidth usage; or
- CRI, offering higher level of adaptation.

The elements specifically addressed in the present document are related to the HDR reconstruction process and the associated dynamic metadata format.

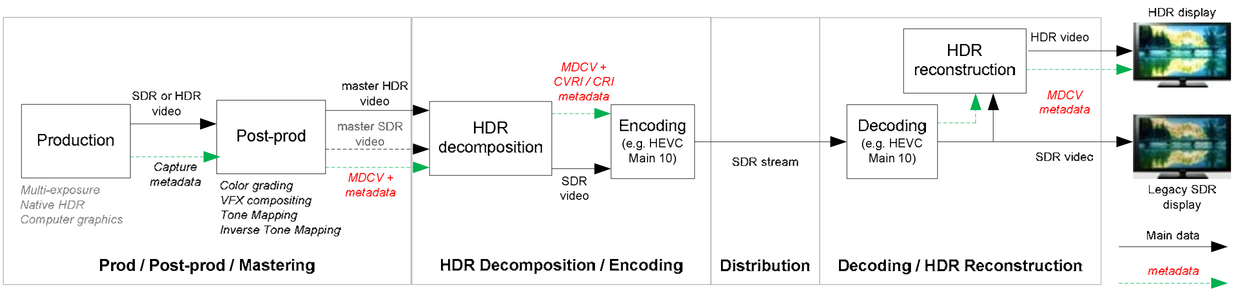

Figure 1: Example of HDR end-to-end system

5 HDR system architecture

The block diagram in Figure 2 depicts in more detail the HDR decomposition and reconstruction processes. The centre block included in dash-red box corresponds to the distribution encoding and decoding stages (e.g. based on HEVC or AVC video coding specifications). The two left and right grey-coloured boxes respectively enable format adaptation to the input video signal of the HDR system and to the targeted system (e.g. a STB, a connected TV, etc.) connected with the HDR system. The black solid line boxes show the HDR specific processing. The additional HDR dynamic metadata are transmitted on distribution networks typically by way of the SEI messaging mechanism. The present document relates to both the HDR signal reconstruction process and the HDR metadata format. The core component of the HDR decomposition stage is the HDR-to-SDR decomposition that generates an SDR video from the HDR signal. Optionally, a block of gamut mapping may be used when the input HDR and output SDR signals are represented with different colour gamut or colour spaces. The decoder side implements the inverse processes, in particular the SDR-to-HDR reconstruction step that goes back to HDR from the SDR video provided by the decoder.

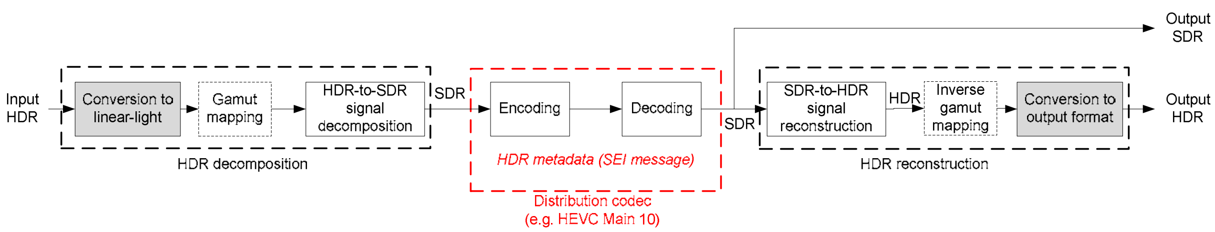

Figure 2: HDR system architecture overview

6 Dynamic metadata format for SDR-to-HDR reconstruction

6.1 Introduction

Clause 6 specifies the dynamic metadata format for SDR-to-HDR reconstruction.

Clause 6.2 specifies the syntax of the reconstruction metadata using pseudocode. The pseudocode is based on C language, but is simplified for ease of understanding. When the number of bits used to represent a variable is not fixed and not bound by constants, it is indicated as VAR. SDR-to-HDR reconstruction dynamic metadata are also called reconstruction metadata for the sake of conciseness.

Clause 6.3 specifies the semantics of the reconstruction metadata.

6.2 Reconstruction metadata syntax

6.2.1 Introduction

This clause specifies the format of the dynamic metadata used for SDR-to-HDR reconstruction. This format supports two mutually exclusive carriage modes: parameter-based mode and table-based mode. Both SDR picture and reconstructed HDR picture characteristics are common to both modes. The SDR-to-HDR-reconstruction process, specified in clause 7 for both modes relies on luminance mapping and colour correction curves produced from the dynamic reconstruction metadata associated with each mode. The reconstruction metadata are carried in HEVC or AVC video coding specifications thanks to a mapping process respectively described in Annex A and Annex B.

6.2.2 HDR reconstruction information

The syntax of hdr_reconstruction_info( ) is specified in Table 1.

Table 1: Syntax of hdr_reconstruction_info( )

Syntax	No. of bits
hdr_reconstruction_info( )	
{	
specVersion	4
payloadMode	4
hdr_characteristics( )	68
sdr_characteristics( )	4
switch (payloadMode) {	
case 0:	
luminance_mapping_variables( )	VAR
colour_correction_adjustment( )	VAR
break;	
case 1:	
luminance_mapping_table( )	VAR
colour_correction_table( )	VAR
chromaToLumaInjectionMuA	14
chromaToLumaInjectionMuB	14
break;	
}	
}	

6.2.3 Reconstructed HDR picture characteristics

HDR picture characteristics (output HDR picture format and characteristics of the reconstruction process) are specified by syntax elements present in Table 2. From the IRD viewpoint, those variables can be mapped from HEVC and AVC Mastering Display Colour Volume (SMPTE ST 2086 [2]) SEI message syntax elements as respectively specified in normative Annex A and Annex B.

Table 2: Syntax of hdr_characteristics( )

Syntax	No. of bits
hdr_characteristics( )	
{	
hdrPicColourSpace	2
hdrMasterDisplayColourSpace	2
hdrMasterDisplayMaxLuminance	32
hdrMasterDisplayMinLuminance	32
}	

6.2.4 SDR picture characteristics

SDR picture characteristics (input SDR format of the reconstruction process) are defined in Table 3.

Table 3: Syntax of sdr_characteristics( )

Syntax	No. of bits
sdr_characteristics( )	
{	
sdrPicColourSpace	2
sdrMasterDisplayColourSpace	2
}	

6.2.5 Luminance mapping variables

The luminance mapping variables are specified by syntax elements present in Table 4. Luminance mapping variables are invoked when **payloadMode** is equal to 0.

Table 4: Syntax of luminance_mapping_variables( )

Syntax	No. of bits
luminance_mapping_variables( )	
{	
tmInputSignalBlackLevelOffset	8
tmInputSignalWhiteLevelOffset	8
shadowGain	8
highlightGain	8
midToneWidthAdjFactor	8
tmOutputFineTuningNumVal	4
for(i = 0; i < tmOutputFineTuningNumVal; i++) {	
tmOutputFineTuningX[i]	8
tmOutputFineTuningY[i]	8
}	
}	

6.2.6 Colour correction adjustment variables

The colour correction adjustment variables are specified by syntax elements present in Table 5. Colour correction adjustment variables are invoked when **payloadMode** is equal to 0.

Table 5: Syntax of colour_correction_adjustment( )

Syntax	No. of bits
colour_correction_adjustment( )	
{	
saturationGainNumVal	4
for(i = 0; i < saturationGainNumVal; i++) {	
saturationGainX[i]	8
saturationGainY[i]	8
}	
}	

6.2.7 Luminance mapping table

The luminance mapping variables are specified by syntax elements present in Table 6. Luminance mapping table is invoked when **payloadMode** is equal to 1.

Table 6: Syntax of luminance_mapping_table( )

Syntax	No. of bits
luminance_mapping_table( )	
{	
luminanceMappingNumVal	6
for(i = 0; i < luminanceMappingNumVal; i++) {	
luminanceMappingX[i]	14
luminanceMappingY[i]	14
}	
}	

6.2.8 Colour correction table

The colour correction table is specified by syntax elements present in Table 7. Colour correction table is invoked when **payloadMode** is equal to 1.

Table 7: Syntax of colour_correction_table( )

Syntax	No. of bits
colour_correction_table( )	
{	
colourCorrectionNumVal	6
for(i = 0; i < colourCorrectionNumVal; i++) {	
colourCorrectionX[i]	11
colourCorrectionY[i]	11
}	
}	

6.3 Reconstruction metadata semantics

6.3.1 HDR reconstruction information

6.3.1.1 Introduction

The hdr_reconstruction_info contains the dynamic metadata that enables reconstruction of an HDR picture (as described in clause 7) when combined with the associated SDR picture.

6.3.1.2 specVersion - Specification version

This 4-bit code indicates the specification version to which the current bitstream conforms to. Dynamic metadata bitstreams that conform to the present document shall be equal to '0b0000'.

6.3.1.3 payloadMode - Payload mode

This variable indicates the carriage mode used to implement the dynamic metadata. Dynamic metadata bitstreams that conform to the present document shall be equal to 0b0000 or 0b0001 only, see Table 8.

Table 8: Payload mode

Value of payloadMode	Carriage mode
0b0000	Parameter-based
0b0001	Table-based
0b0010 - 0b1111	Reserved for future use

Parameter-based mode consists of few variables enabling the construction of luminance mapping and colour correction curves that are required as input of the HDR reconstruction process.

Alternatively, table-based mode consists of look-up tables that are representative of luminance mapping and colour correction curves. Look-up tables values shall be interpolated by piece-wise linear sections, see clause 7.3.

NOTE: Parameter-based mode may be of interest for distribution workflows which primary goal is to provide direct SDR backward compatible services with very low additional payload or bandwidth usage for carrying the dynamic metadata. Table-based mode may be of interest for workflows equipped with low-end terminals or when a higher level of adaptation is desired for representing both HDR and SDR streams.

6.3.1.4 chromaToLumaInjectionMuA

This variable shall be present when **payloadMode** is equal to 1. This variable indicates the ratio of the blue colour-difference component injection into the luma component. The value shall be in the bounded range [0 to 0,5] and in multiples of (1÷16 384).

The variable muA shall be derived as follows:

- muA = **chromaToLumaInjectionMuA**.

6.3.1.5 chromaToLumaInjectionMuB

This variable shall be present when **payloadMode** is equal to 1. This variable indicates the ratio of the red colour-difference component injection into the luma component. The value shall be in the bounded range [0 to 0,5] and in multiples of (1÷16 384).

The variable muB shall be derived as follows:

- muB = **chromaToLumaInjectionMuB**.

6.3.2 Reconstructed HDR picture characteristics

6.3.2.1 Introduction

The hdr_characteristics( ) contains reconstructed HDR picture signal characteristics namely: an indication on the colour space in which the HDR picture is represented, an indication on the primaries of the HDR display used to master the HDR picture and the minimum and maximum luminance of the mastering display used to grade the associated HDR pictures.

6.3.2.2 hdrPicColourSpace

This variable indicates the white point and primaries chromaticity coordinates of the HDR picture colour space in terms of CIE 1931 definitions of x and y as specified in ISO 11664-1 [12] and defined in Table 9.

Table 9: HDR video colour primaries and white point

Value of hdrPicColourSpace	Primaries			NOTE
0	primary	x	y	as defined in Recommendation ITU-R BT.709-6 [9]
	green	0,300	0,600	
	blue	0,150	0,060	
	red	0,640	0,330	
	white D65	0,3127	0,3290	
1	primary	x	y	as defined in Recommendation ITU-R BT.2020-2 [10]
	green	0,170	0,797	
	blue	0,131	0,046	
	red	0,708	0,292	
	white D65	0,3127	0,3290	
2 - 3	Reserved for future use			

The default value of **hdrPicColourSpace** is equal to **sdrPicColourSpace**.

The following constraints shall be respected:

- When **hdrMasterDisplayColourSpace** is greater than 0, **hdrPicColourSpace** shall be equal to 1.
- When **hdrPicColourSpace** is greater than **sdrPicColourSpace**, an inverse gamut mapping procedure shall be invoked after the HDR reconstruction process. An informative inverse gamut mapping process is provided in Annex D.

6.3.2.3 hdrMasterDisplayColourSpace

This variable indicates the white point and chromaticity coordinates of the HDR mastering display primaries in terms of CIE 1931 definitions of x and y as specified in ISO 11664-1 [12] and defined in Table 10.

Table 10: HDR mastering display colour primaries and white point

Value of hdrMasterDisplayColourSpace	Primaries			NOTE
0	primary	x	y	as defined in Recommendation ITU-R BT.709-6 [9]
	green	0,300	0,600	
	blue	0,150	0,060	
	red	0,640	0,330	
	white D65	0,3127	0,3290	
1	primary	x	y	as defined in Recommendation ITU-R BT.2020-2 [10]
	green	0,170	0,797	
	blue	0,131	0,046	
	red	0,708	0,292	
	white D65	0,3127	0,3290	
2	primary	x	y	primaries as defined in RP 431-2 [8] (DCI-P3) white point as defined in Recommendation ITU-R BT.2020-2 [10]
	green	0,265	0,690	
	blue	0,150	0,060	
	red	0,680	0,320	
	white D65	0,3127	0,3290	
3	Reserved for future use			

The default value of **hdrMasterDisplayColourSpace** is equal to **sdrMasterDisplayColourSpace**.

6.3.2.4 hdrMasterDisplayMaxLuminance

This variable specifies the nominal maximum display luminance of the mastering display used to grade the HDR picture in candela per square metre and in multiples of 50 candelas per square metre. **hdrMasterDisplayMaxLuminance** shall be greater than **hdrMasterDisplayMinLuminance**. The proper value of this variable shall be present in associated bitstreams that conform to this version of the present document.

6.3.2.5 hdrMasterDisplayMinLuminance

This variable specifies the nominal minimum display luminance of the mastering display used to grade the HDR picture in candela per square metre. **hdrMasterDisplayMinLuminance** shall be less than **hdrMasterDisplayMaxLuminance**. If the proper value of this variable is unknown, it is recommended that it is set to 0.

6.3.3 SDR picture characteristics

6.3.3.1 Introduction

The sdr_characteristics( ) contains SDR picture (coming from the HDR-to-SDR signal decomposition stage non normatively but informatively documented in the present document) signal characteristics namely: an indication on the colour space in which the SDR picture is represented and an indication on the primaries of the SDR mastering display that was used to master the SDR picture. If no SDR mastering display has been employed due to an automatic derivation process, a value is inferred by the present document.

6.3.3.2 sdrPicColourSpace

This variable indicates the white point and primaries chromaticity coordinates of the SDR picture colour space in terms of CIE 1931 definitions of x and y as specified in ISO 11664-1 [12] and defined in Table 11.

Table 11: SDR video colour primaries and white point

Value of sdrPicColourSpace	Primaries			NOTE
0	primary green blue red white D65	x 0,300 0,150 0,640 0,3127	y 0,600 0,060 0,330 0,3290	as defined in Recommendation ITU-R BT.709-6 [9]
1	primary green blue red white D65	x 0,170 0,131 0,708 0,3127	y 0,797 0,046 0,292 0,3290	as defined in Recommendation ITU-R BT.2020-2 [10]
2 - 3	Reserved for future use			

sdrPicColourSpace shall be present in associated bitstreams that conform to this version of the present document.

sdrPicColourSpace default value shall be 1.

sdrPicColourSpace shall be equal to or less than **hdrPicColourSpace**.

When **sdrPicColourSpace** is less than **hdrPicColourSpace**, an inverse gamut mapping procedure shall be invoked after the HDR reconstruction process. An informative inverse gamut mapping process is provided in Annex D.

NOTE: It is recommended that **sdrPicColourSpace** is equal to **hdrPicColourSpace**.

6.3.3.3 sdrMasterDisplayColourSpace

This variable indicates the white point and chromaticity coordinates of the SDR mastering display primaries in terms of CIE 1931 definitions of x and y as specified in ISO 11664-1 [12] and defined in Table 12.

Table 12: SDR mastering display colour primaries and white point

Value of sdrMasterDisplayColourSpace	Primaries			NOTE
0	primary green blue red white D65	x 0,300 0,150 0,640 0,3127	y 0,600 0,060 0,330 0,3290	as defined in Recommendation ITU-R BT.709-6 [9]
1	primary green blue red white D65	x 0,170 0,131 0,708 0,3127	y 0,797 0,046 0,292 0,3290	as defined in Recommendation ITU-R BT.2020-2 [10]
2	primary green blue red white D65	x 0,265 0,150 0,680 0,3127	y 0,690 0,060 0,320 0,3290	primaries as defined in SMPTE RP 431-2 [8] (DCI-P3) white point as defined in Recommendation ITU-R BT.2020-2 [10]
3	Reserved for future use			

sdrMasterDisplayColourSpace shall be present in associated bitstreams that conform to this version of the present document.

sdrMasterDisplayColourSpace default value shall be 2.

When **sdrPicColourSpace** is equal to **hdrPicColourSpace**, **sdrMasterDisplayColourSpace** shall be equal to **hdrMasterDisplayColourSpace** for bitstreams that conform to this version of the present document.

When **sdrPicColourSpace** is less than **hdrPicColourSpace**, **sdrMasterDisplayColourSpace** shall be equal to 0.

NOTE: It is recommended that **sdrMasterDisplayColourSpace** is equal to **hdrMasterDisplayColourSpace**.

6.3.4 Luminance mapping variables

6.3.4.1 Introduction

The luminance mapping variables defined in this clause are representative of the square-root of the inverse luminance mapping curve and are used to construct the look-up table $lutMapY$. This look-up table is taken as one input of the SDR-to-HDR reconstruction process specified in clause 7. The variables specified in this clause are present when **payloadMode** is equal to 0 (i.e. parameter-based carriage of dynamic metadata). The range and precisions defined in the semantics of the present clause are consistent with the range and precisions defined in SMPTE ST 2094-20 [6]. Those variables are used in the HDR reconstruction process specified in clause 7.

6.3.4.2 tmInputSignalBlackLevelOffset - Tone Mapping Input Signal Black Level Offset

This variable indicates the offset to be subtracted from the signal and is used to calculate the gain of the signal as a first step in the luminance mapping curve reconstruction process. The value shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 255).

6.3.4.3 tmInputSignalWhiteLevelOffset - Tone Mapping Input Signal White Level Offset

This variable is used to calculate the gain of the signal as a second step in the luminance mapping curve reconstruction process. The value shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 255).

6.3.4.4 shadowGain - Shadow Gain Control

This variable indicates the gain that is used to adjust the steepness of the luminance mapping curve in its shadow (darker) region. The value shall be in the bounded range [0 to 2] and in multiples of (2 ÷ 255).

6.3.4.5 highlightGain - Highlight Gain Control

This variable indicates the gain that is used to adjust the steepness of the luminance mapping curve in its highlight (brighter) region. The value shall be in the bounded range [0 to 2] and in multiples of (2 ÷ 255).

6.3.4.6 midToneWidthAdjFactor - Mid-Tone Width Adjustment Factor

This variable indicates the gain that is used to adjust the width of the luminance mapping curve in their mid-tone region. The value shall be in the bounded range [0 to 2] and in multiples of (2 ÷ 255).

6.3.4.7 tmOutputFineTuningNumVal - Number of Tone Mapping Output Fine Tuning Function Curve Points

This variable specifies the number of pivot points in the piece-wise linear tone mapping output fine tuning function $f_{ftlum}(\,)$, see clause 7.3, that maps a local tone mapping input value to an adjusted one. The value of **tmOutputFineTuningNumVal** shall be in the bounded range [0 to 10]. The first pair shall be { 0,0 ; 0,0 } and last pair shall be { 1,0 ; 1,0 }.

6.3.4.8 tmOutputFineTuningX - Tone Mapping Output Fine Tuning Function x values

This variable indicates the x_i values of the tone mapping output fine tuning function. The value shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 255). The value of **tmOutputFineTuningX**[i + 1] shall be greater than the value of **tmOutputFineTuningX**[i], for i in the range of 0 to **tmOutputFineTuningNumVal** - 2, inclusive.

6.3.4.9 tmOutputFineTuningY - Tone Mapping Output Fine Tuning Function y values

This variable indicates the y_i values of the tone mapping output fine tuning function, $f_{ftlum}(\ )$. The value shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 255).

When **tmOutputFineTuningX**[0] is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **tmOutputFineTuningX**[0], inclusive, to target values ranging from 0 to **tmOutputFineTuningY**[0], inclusive.

When **tmOutputFineTuningX**[**tmOutputFineTuningNumVal** - 1] is not equal to 1, a final linear segment shall be inferred that maps input values ranging from **tmOutputFineTuningX**[**tmOutputFineTuningNumVal** - 1] to 1, inclusive, to target values ranging from **tmOutputFineTuningY**[**tmOutputFineTuningNumVal** - 1] to 1, inclusive.

6.3.5 Colour correction adjustment variables

6.3.5.1 Introduction

The colour correction variables defined in this clause are used to adjust the default colour correction curve, implemented by the look-up table $lutCC$. This look-up table is derived from the square-root of the inverse luminance mapping curve. The colour correction curve is a required input of the SDR-to-HDR reconstruction process. The variables specified in this clause are present when **payloadMode** is equal to 0 (i.e. parameter-based carriage of dynamic metadata). The range and precisions defined in the semantics of the present clause are consistent with the range and precisions defined in SMPTE ST 2094-20 [6]. Those variables are used in the HDR reconstruction process specified in clause 7.

6.3.5.2 saturationGainNumVal - Number of Saturation Gain Function Curve Points

This variable specifies the number of pivot points in the piece-wise linear saturation gain function, $f_{sgf}(\ )$, that maps a colour correction input value to a saturation scaling factor. The value of **saturationGainNumVal** shall be in the bounded range [0 to 6]. See clause 7.3 for the computation of $f_{sgf}(\ )$ from the list of pivot points.

6.3.5.3 saturationGainX - Saturation Gain Function x values

This variable indicates the x_i values of the saturation gain function. The value shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 255). The value of **saturationGainX**[i + 1] shall be greater than the value of **saturationGainX**[i], for i in the range of 0 to **saturationGainNumVal** - 2, inclusive.

6.3.5.4 saturationGainY - Saturation Gain Function y values

This variable indicates the y_i values of the saturation gain function. The value shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 255).

When **saturationGainX**[0] is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **saturationGainX**[0], inclusive, to target values ranging from 0 to **saturationGainY**[0], inclusive.

When **saturationGainX**[**saturationGainNumVal** - 1] is not equal to 1, a final linear segment shall be inferred that maps input values ranging from **saturationGainX**[**saturationGainNumVal** - 1] to 1, inclusive, to target values ranging from **saturationGainY**[**saturationGainNumVal** - 1] to 1, inclusive.

NOTE: It is recommended that **saturationGainX**[0] is equal to 0 and **saturationGainX**[**saturationGainNumVal** - 1] is equal to 1.

6.3.6 Luminance mapping table

6.3.6.1 Introduction

The variables defined in this clause are piece-wise linear pivot points representative of the square-root of the inverse luminance mapping curve, implemented by the look-up table *lutMapY*. This look-up table is taken as one input of the SDR-to-HDR reconstruction process. The variables specified in this clause are present when **payloadMode** is set to 1 (i.e. table-based carriage of dynamic metadata). The range and precisions defined in the semantics of the present clause are consistent with the range and precisions defined in SMPTE ST 2094-30 [5]. Those variables are used in the HDR reconstruction process specified in clause 7.

6.3.6.2 luminanceMappingNumVal - Number of Luminance Mapping Curve Points

This variable specifies the number of pivot points in the piece-wise linear luminance mapping curve. The value of **luminanceMappingNumVal** shall be in the bounded range [0 to 33].

6.3.6.3 luminanceMappingX - Luminance Mapping x values

This variable indicates the x_i values of the luminance mapping curve. It shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 16 383). The value of **luminanceMappingX**[i + 1] shall be greater than the value of **luminanceMappingX**[i], for i in the range of 0 to **luminanceMappingNumVal** - 2, inclusive.

6.3.6.4 luminanceMappingY - Luminance Mapping y values

This variable indicates the y_i values of the luminance mapping curve. It shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 16 383).

When **luminanceMappingX**[0] is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **luminanceMappingX**[0], inclusive, to target values ranging from 0 to **luminanceMappingY**[0], inclusive.

When **luminanceMappingX**[**luminanceMappingNumVal** - 1] is not equal to 1, a final linear segment shall be inferred that maps input values ranging from **luminanceMappingX**[**luminanceMappingNumVal** - 1] to 1, inclusive, to target values ranging from **luminanceMappingY**[**luminanceMappingNumVal** - 1] to 1, inclusive.

6.3.7 Colour correction table

6.3.7.1 Introduction

The variables defined in this clause are piece-wise linear pivot points representative of colour correction curve, implemented by the look-up table *lutCC*. This look-up table is taken as one input of the SDR-to-HDR reconstruction process. The variables specified in this clause are present when **payloadMode** is set to 1 (i.e. table-based carriage of dynamic metadata). The range defined in the semantics of the present clause is consistent with the range defined in SMPTE ST 2094-30 [5]. Those variables are used in the HDR reconstruction process specified in clause 7.

6.3.7.2 colourCorrectionNumVal - Number of Colour Correction Curve Points

This variable specifies the number of pivot points in the piece-wise linear colour correction curve. The value of **colourCorrectionNumVal** shall be in the bounded range [0 to 33].

6.3.7.3 colourCorrectionX - Colour Correction x values

This variable indicates the x_i values of the colour correction curve. It shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 2 047). The value of **colourCorrectionX**[i + 1] shall be greater than the value of **colourCorrectionX**[i], for i in the range of 0 to **colourCorrectionNumVal** - 2, inclusive.

6.3.7.4 colourCorrectionY - Colour Correction y values

This variable indicates the y_i values of the colour correction curve. It shall be in the bounded range [0 to 1] and in multiples of (1 ÷ 2 047).

When **colourCorrectionX**[0] is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **colourCorrectionX**[0], inclusive, to target values ranging from 0 to **colourCorrectionY**[0], inclusive.

When **colourCorrectionX**[**colourCorrectionNumVal** - 1] is not equal to 1, a final linear segment shall be inferred that maps input values ranging from **colourCorrectionX**[**colourCorrectionNumVal** - 1] to 1, inclusive, to target values ranging from **colourCorrectionY**[**colourCorrectionNumVal** - 1] to 1, inclusive.

7 HDR signal reconstruction process

7.1 Input streams

The input stream is composed of a decoded SDR video stream and associated dynamic metadata that are combined to reconstruct an HDR video signal. The dynamic metadata can be conveyed thanks to two mutually exclusive modes: a parameter-based mode (**payloadMode** 0) and a table-based mode (**payloadMode** 1). Concerning ITU-T or ISO/IEC based video codecs, parameter-based mode is carried by Colour Volume Reconstruction Information SEI message that is embedded in a User Data Registered SEI message while table-based mode is carried by Colour Remapping Information SEI message. The HDR reconstruction process is described in this clause. This process employs syntax element specified in clause 6.2 and retrieved from parsed dynamic metadata stream. Semantics attached to the syntax elements is provided in clause 6.3.

7.2 Reconstruction process of the HDR stream

7.2.1 Introduction

This clause specifies the reconstruction process enabling the generation of an HDR picture from an SDR picture with associated dynamic metadata.

This process is defined for full range SDR picture signal (as defined in SMPTE RP 2077 [7]). For SDR picture defined as narrow-range signal, an (unspecified) conversion to full range process shall be applied first (e.g. as specified in SMPTE RP 2077 [7]). This process assumes that the SDR picture signal is represented with a bitdepth of 10-bit per component. For SDR picture represented with a different bitdepth, an (unspecified) conversion to 10-bit signal shall be applied first.

The process depicted in Figure 3 can be summarized as follows:

- From the input metadata conveyed in either **payloadMode** 0 or 1, a luma-related look-up table, $lutMapY$, is derived (see clause 7.2.3.1).
- Similarly, from the input metadata conveyed in either **payloadMode** 0 or 1, a colour correction look-up table, $lutCC$, is derived (see clause 7.2.3.2).
- The next step, described in clause 7.2.4, consists of applying the SDR-to-HDR reconstruction from the input SDR picture, the derived luma-related look-up table and colour correction look-up table. This process produces an output linear-light HDR picture.
- An optional inverse gamut mapping can be applied when the colour gamut and/or colour space of the SDR picture (as specified by the variable **sdrPicColourSpace**) and of the HDR picture (as specified by the variable **hdrPicColourSpace**) are different.

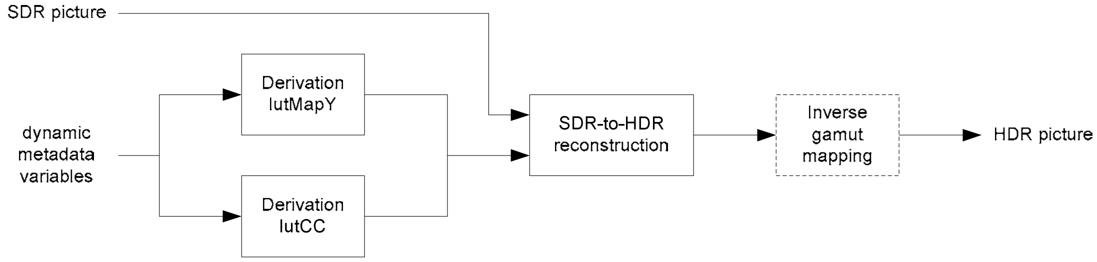

Figure 3: Overview of the HDR reconstruction process.

In the next clauses, the variables *picWidth, picHeight, maxSampleVal*, *midSampleVal* are defined as follows:

- *picWidth* and *picHeight* are the width and height, respectively, of the SDR picture (e.g. as specified by the syntax elements **pic_width_in_luma_samples** and **pic_height_in_luma_samples** in the HEVC specification [4]);
- *maxSampleVal* is equal to 2^{10} i.e. 1 024.
- *midSampleVal* is equal to 2^9 i.e. 512.

7.2.2 Selecting a reconstruction mode

Clause 7.2.3 describes the processing step to construct luminance mapping and colour correction tables that are used as inputs to the HDR stream reconstruction process. The HDR reconstruction process operates on look-up tables reconstructed from variables (**payloadMode** 0) specified in clauses 7.2.3.1 and 7.2.3.2 or derived from coded look-up tables (**payloadMode** 1) specified in clauses 7.2.3.3 and 7.2.3.4. The HDR picture reconstruction process specified in clause 7.2.4 is common to both modes (**payloadMode** 0 and 1).

7.2.3 Luminance mapping and colour correction tables construction

7.2.3.1 Luminance mapping table construction from variables (payloadMode 0)

7.2.3.1.1 Introduction

The luminance mapping table construction for **payloadMode** 0 derives a 1D look-up table *lutMapY* from the luminance mapping variables as described in clause 6.2.5.

This process takes as inputs:

- the HDR picture characteristics variable **hdrMasterDisplayMaxLuminance**;
- the luminance mapping variables **tmInputSignalBlackLevelOffset**, **tmInputSignalWhiteLevelOffset**, **shadowGain**, **highlightGain**, **midToneWidthAdjFactor**, **tmOutputFineTuningNumVal**, **tmOutputFineTuningX**[i] and **tmOutputFineTuningY**[i].

The process generates as output:

- the luminance mapping look-up table *lutMapY* of *maxSampleVal* entries.

7.2.3.1.2 Overview of the computation of lutMapY

The look-up table $lutMapY[\,Y_{post1}\,]$, for luma values $Y_{post1} = 0..(\ maxSampleVal - 1\ )$, implements an inverse tone mapping function. The inverse tone mapping process is shown in Figure 4.

For any Y_{post1} in 0..(*maxSampleVal* - 1), the $lutMapY[\,Y_{post1}\,]$ is derived by applying the following steps:

- Y_{post1} is converted via the linear-light domain to the perceptually uniform domain (uniform lightness), based on the assumption that SDR picture is graded on a mastering display with maximum display mastering luminance equal to 100 cd/m², by invoking clause 7.2.3.1.3, with Y_{post1} as input and Y_{pus} as output.

- the inverse fine tuning process is applied by invoking clause 7.2.3.1.4, with Y_{pus}, the variables **tmOutputFineTuningNumVal**, **tmOutputFineTuningX**[i] and **tmOutputFineTuningY**[i], for i=0..(**tmOutputFineTuningNumVal** - 1) as inputs and Y_{ft} as output.
- The inverse tone mapping curve is applied by invoking clause 7.2.3.1.5, with Y_{ft}, the variables **shadowGain**, **highlightGain**, **midToneWidthAdjFactor** and **hdrMasterDisplayMaxLuminance** as inputs, and Y_{adj} as output.
- The inverse black and white level offsets are applied by invoking clause 7.2.3.1.6, with Y_{adj}, the variables **tmInputSignalBlackLevelOffset** and **tmInputSignalWhiteLevelOffset** as inputs, and Y_{bw} as output.
- The signal Y_{bw} is processed through a gain limiter by invoking clause 7.2.3.1.7, with Y_{bw} and Y_{pus} as inputs, and Y_{glim} as output. A choice is made between limiting Y_{bw} or passing it on unchanged, based on the value of the variable **tmInputSignalBlackLevelOffset**.
- The signal Y_{glim} is converted back to the linear-light domain based on the maximum display mastering luminance, by invoking clause 7.2.3.1.8, with Y_{bw} and the variable **hdrMasterDisplayMaxLuminance** as inputs, and Y_{ll} as output.
- The final output $lutMapY[\ Y_{post1}\ ]$ is derived from the variable Y_{ll} by invoking clause 7.2.3.1.9.

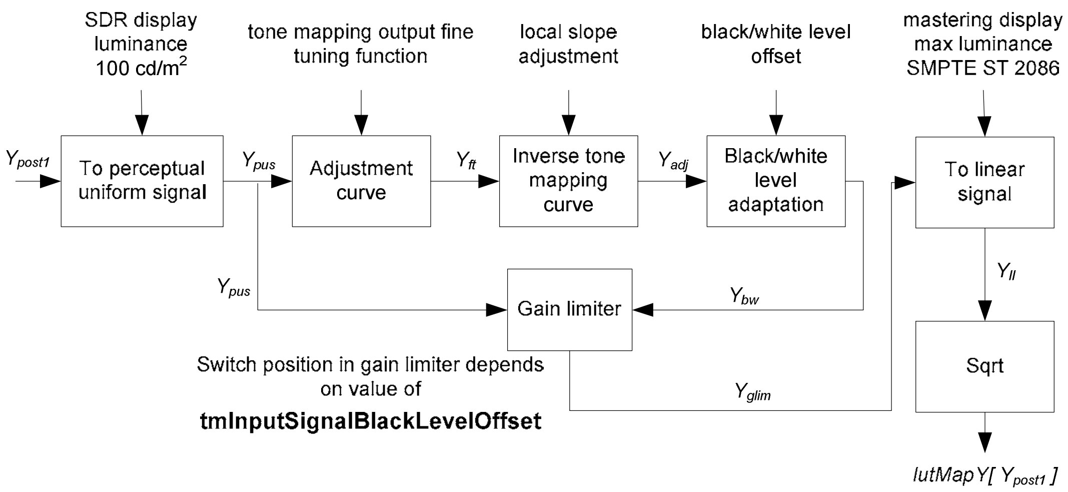

Figure 4: Inverse tone mapping process

The blocks shown in Figure 4 are specified in detail in the clauses 7.2.3.1.3 to 7.2.3.1.9.

7.2.3.1.3 Block "To perceptual uniform signal"

This process takes as inputs:

- the luma value Y_{post1}.

The process generates as output:

- the converted linear-light value Y_{pus}.

In the first step, Y_{post1}, which is a Recommendation ITU-R BT.1886 [11] compatible luma signal, shall be taken to the power $\gamma = 2{,}4$ to yield the linear-light signal Y_2.

$$Y_2 = \left(\frac{Y_{post1}}{maxSampleVal - 1}\right)^{2,4} \quad (1)$$

In the second step, the inverse EOTF, $v(x, y)$, shall be performed on $x = Y_2$, where $v(x, y)$ is the perceptually uniform colour component, when applied to the linear components, x, normalized to 0..1, where 1 corresponds to a maximum display mastering luminance of an SDR mastering display L_{SDR} of 100 cd/m^2, and using $\gamma = 2{,}4$, in order to get the perceptually uniform signal Y_{pus}.

$$v(x,y) = \frac{\log_{10}\left(1 + (\rho(y)-1)\times x^{\frac{1}{2,4}}\right)}{\log_{10}(\rho(y))} \quad (2)$$

$$\rho(y) = 1 + (33-1)\times\left(\frac{y}{10000}\right)^{\frac{1}{2,4}} \quad (3)$$

$$Y_{pus} = v(Y_2, L_{SDR}) \quad (4)$$

7.2.3.1.4 Block "Adjustment curve"

This process takes as inputs:

- the converted linear-light value Y_{pus};
- the variables **tmOutputFineTuningNumVal**, **tmOutputFineTuningX**[i] and **tmOutputFineTuningY**[i], for i=0..(**tmOutputFineTuningNumVal** - 1).

The process generates as output:

- the corrected value Y_{ft}.

In this block, the input signal Y_{pus} shall be corrected by the inverse of the *ToneMappingOutputFineTuningFunction* function which is derived by invoking clause 6.2.5 with the parameters **tmOutputFineTuningNumVal**, **tmOutputFineTuningX**[i] and **tmOutputFineTuningY**[i] as inputs, in order to get Y_{ft}.

The *ToneMappingOutputFineTuningFunction* function f_{ftlum}( ), is a piecewise linear function; see clause 7.3 for the computation of f_{ftlum}( ) from the list of points.

The samples explicitly defining the *ToneMappingOutputFineTuningFunction* function shall be the pairs **tmOutputFineTuningX**[i], **tmOutputFineTuningY**[i], in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5, possibly extended with a point at the start and/or at the end, as specified in clause 6.3.4.9.

$$Y_{ft} = \begin{cases} {f_{ftlum}}^{-1}(Y_{pus}), & 0 \le Y_{pus} \le 1 \\ Y_{pus}, & otherwise \end{cases} \quad (5)$$

NOTE: The inverse of the *ToneMappingOutputFineTuningFunction* function, f_{ftlum}^{-1}( ), can be obtained by swapping the x and y values of the given { x_i, y_i } pairs in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5.

7.2.3.1.5 Block "Inverse tone mapping curve"

This process takes as inputs:

- the corrected value Y_{ft};
- the variables **shadowGain**, **highlightGain**, **midToneWidthAdjFactor**;
- the variable **hdrMasterDisplayMaxLuminance**.

The process generates as output:

- the inverse tone-mapped value, in linear-light domain, Y_{adj}.

In this block, the input signal Y_{ft} shall be converted by an inverse tone mapping curve to the output signal Y_{adj} according to equation (6).

$$Y_{adj} = TMO_{inv}(Y_{ft}) \quad (6)$$

The inverse tone mapping curve TMO_{inv} is built from variables **shadowGain** (= base gain), **midToneWidthAdjFactor** (= parabola part), and **highlightGain** (= differential gain at the end). The basics of the curve for TMO_{inv} are explained below.

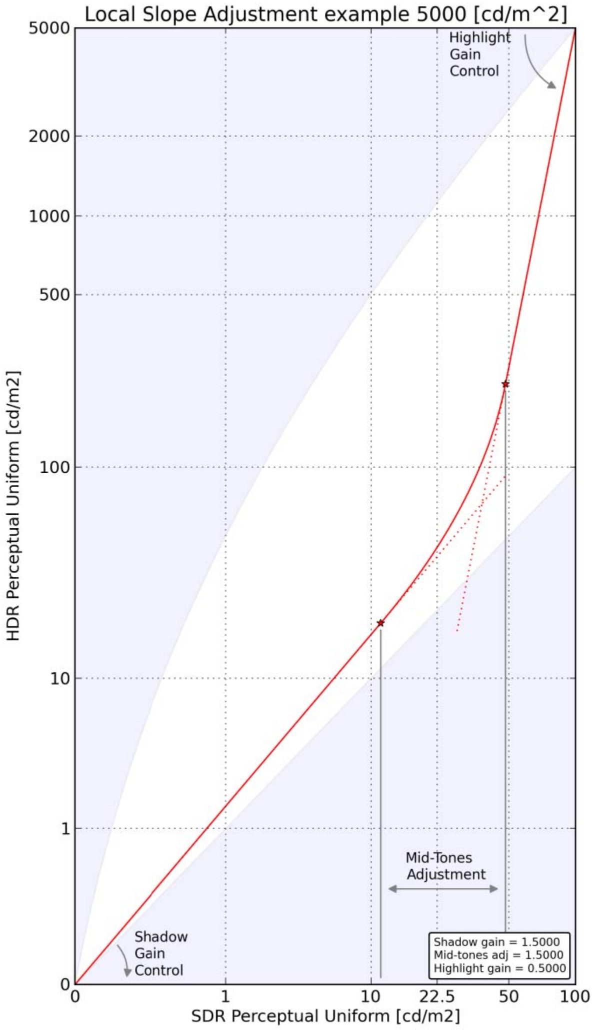

Figure 5: Inverse tone mapping curve shape

The curve has 3 shape parameters.

Parameter #1 is the base gain. This determines the brightness for most of the image except the highlights. It shall be determined by the variable **shadowGain** in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.5).

Parameter #2 is the highlight differential gain. This determines how much of the details in highlights is preserved, at the cost of the peak brightness. It shall be determined by the variable **highlightGain** in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.5).

Lines #1 and #2 intersect, and together they form a hard clipping curve. If this is not desired then a parabola segment can be inserted, and this is symmetrical with respect to the original intersection point of the 2 lines.

Parameter #3 is the width of the parabolic segment. It shall be determined by the variable **midToneWidthAdjFactor** in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.5).

$$TMO_{inv}(x) = \begin{cases} \frac{1}{SGC} \times x, & 0 \le x \le x_{SGC} \\ -\frac{b}{2a} + \frac{\sqrt{b^2 - 4a \times (c-x)}}{2a}, & x_{SGC} < x < x_{HGC} \\ \frac{1}{HGC} \times (x-1) + 1, & x_{HGC} \le x \le 1 \text{ and } HGC \ne 0 \\ 1, & x = 1 \text{ and } HGC = 0 \end{cases} \quad (7)$$

NOTE 1: Due to the limitation on **shadowGain**, see clause 6.3.4, $SGC > 1{,}0$ for $L_{SDR} \ge 100$ cd/m².

$$\begin{cases} a = -0{,}5 \times \frac{SGC-HGC}{para} \\ b = \frac{1-HGC}{para} + \frac{SGC+HGC}{2} \\ c = -\frac{\left((SGC-HGC)\times para-2\times(1-HGC)\right)^2}{8\times(SGC-HGC)\times para} \end{cases} \quad (8)$$

NOTE 2: If $para$ equals 0, x_{SGC} will equal x_{HGC} and the values of a, b and c are not needed.

NOTE 3: Due to the limitations on **highlightGain** and **shadowGain**, see clause 6.3.4, $HGC \leq 0{,}5$ and $SGC > 1{,}0$ for $L_{SDR} \geq 100$ cd/m².

$$\begin{aligned} x_{SGC} &= SGC \times \left(\frac{1-HGC}{SGC-HGC} - \frac{para}{2}\right) \\ x_{HGC} &= HGC \times \left(\frac{1-HGC}{SGC-HGC} + \frac{para}{2} - 1\right) + 1 \end{aligned} \quad (9)$$

$$exposure = \frac{\textbf{shadowGain}}{4} + 0{,}5 \quad (10)$$

$$expgain = v\left(\frac{L_{HDR}}{L_{SDR}}, L_{SDR}\right) \quad (11)$$

where:

- L_{SDR} shall be taken as 100 cd/m²; and
- L_{HDR}, shall be the maximum display mastering luminance, equal to the variable **hdrMasterDisplayMaxLuminance** in the structure hdr_characteristics( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.3).

$$SGC = expgain \times exposure \quad (12)$$

$$HGC = \frac{\textbf{highlightGain}}{4} \quad (13)$$

$$para = \frac{\textbf{midToneWidthAdjFactor}}{2} \quad (14)$$

7.2.3.1.6 Block "Black/white level adaptation"

This process takes as inputs:

- the inverse tone-mapped value, in linear-light domain, Y_{adj};
- the variables **tmInputSignalBlackLevelOffset** and **tmInputSignalWhiteLevelOffset**.

The process generates as output:

- the stretched value Y_{bw}.

In this block, the input signal Y_{adj} shall be adapted by the black and white stretch in order to derive the output signal Y_{bw}.

$$Y_{bw} = \left(1 - \frac{255\times wlo}{510} - \frac{255\times blo}{2040}\right) \times Y_{adj} + \frac{255\times blo}{2040} \quad (15)$$

where:

- blo shall be equal to the variable **tmInputSignalBlackLevelOffset** in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.5); and
- wlo shall be equal to the variable **tmInputSignalWhiteLevelOffset** in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.5).

7.2.3.1.7 Block "Gain limiter"

This process takes as inputs:

- the value Y_{bw} from clause 7.2.3.1.6;
- the value Y_{pus} from clause 7.2.3.1.3;
- the variable **tmInputSignalBlackLevelOffset**;
- the variable **hdrMasterDisplayMaxLuminance** in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.5).

The process generates as output:

- the value Y_{glim}.

In this block, a choice is made between limiting Y_{bw} or passing it on unchanged, based on the value of the variable **tmInputSignalBlackLevelOffset**.

When the value of the variable **tmInputSignalBlackLevelOffset** is equal to 0, the output Y_{glim} of this block shall be the value Y_{bw}.

When the value of the variable **tmInputSignalBlackLevelOffset** is not equal to 0, the value Y_{bw} shall be corrected for minimum gain based on the maximum display mastering luminance L_{HDR}, which is equal to the variable **hdrMasterDisplayMaxLuminance** in the structure hdr_characteristics( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.3), the maximum display mastering luminance of an SDR mastering display L_{SDR} of 100 cd/m^2, and using Y_{pus} from clause 7.2.3.1.3.

$$Y_{glim} = Min(Y_{bw} \,;\, Y_{pus} \div g) \tag{16}$$

$$g = v(0{,}1 \div L_{SDR}, L_{SDR}) \div v(1 \div L_{HDR}, L_{HDR}) \tag{17}$$

with the inverse EOTF, $v(x, y)$, taken from equation (2).

7.2.3.1.8 Block "To linear signal"

This process takes as inputs:

- the gain limited value Y_{glim};
- the variable **hdrMasterDisplayMaxLuminance**.

The process generates as output:

- the linear-light value Y_{ll}.

In this block the computation of the value Y_{ll}, the input signal Y_{glim} shall be converted from the perceptually uniform domain to the linear-light domain output value Y_{ll}, using the EOTF, $v_{inv}(x,y)$ and shall be based on the maximum display mastering luminance L_{HDR}, which is equal to the variable **hdrMasterDisplayMaxLuminance** in the structure hdr_characteristics( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.3).

$$v_{inv}(x, y) = \left(\frac{\rho(y)^{x} - 1}{\rho(y) - 1}\right)^{2,4} \tag{18}$$

$$Y_{ll} = v_{inv}(Y_{glim}, L_{HDR}) \tag{19}$$

7.2.3.1.9 Block "Sqrt"

This process takes as inputs:

- the linear-light value Y_{ll} ;
- the variable **hdrDisplayMasterMaxLuminance**.

The process generates as output:

- the value $lutMapY[\,Y_{post1}\,]$.

L_{HDR} is equal to **hdrDisplayMasterMaxLuminance.**

In this block, the square root of the value Y_{ll} is taken in order to compute the value of $lutMapY[\,Y_{post1}\,]$:

$$lutMapY\left[\,Y_{post1}\,\right] = \sqrt{Y_{ll} \times L_{HDR}} \tag{20}$$

7.2.3.2 Colour correction table construction from parameter-based mode (payloadMode 0)

The colour correction table construction for payload mode 0 derives a 1D look-up table $lutCC$ from the colour correction adjustment variables specified in clause 6.2.6.

This process takes as inputs:

- the colour correction adjustment variables **saturationGainNumVal**, **saturationGainX**[i] and **saturationGainY**[i].

The process generates as output:

- the colour correction look-up table $lutCC$ of $maxSampleVal$ entries.

For each luma value Y in 0..($maxSampleVal$ - 1), $lutCC$[Y] is derived as follows:

$$lutCC[Y] = Min\left(R_{chroma}\ ;\ \frac{1}{Max\left(1\ ;\ R_{sgf} \times f_{sgf}(Y_n)\right)} \times L(64 \times Y_n)\right) \tag{21}$$

where:

- $Y_n = \frac{Y}{maxSampleVal-1}$
- The saturation gain function $f_{sgf}(\,)$ is derived from the piece-wise linear pivot points defined by the variables **saturationGainX**[i] and **saturationGainY**[i], for i=0..(**saturationGainNumVal** - 1), see clause 7.3.
- The piece-wise linear function $L(x)$ is derived from specified { x_i, y_i } pairs in Table 13, see clause 7.3.
- R_{chroma} = 0,125 and R_{sgf} = (4×2^{14}) in this version of the present document.

Table 13: default colour correction look-up table representative of function *L(x)*.

x_i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
y_i	2 047	1 236	577	414	300	245	200	173	153	135	122	111	102	94	87	82	76
x_i	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
y_i	72	68	64	61	58	55	53	51	49	47	45	44	42	41	39	38	37
x_i	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
y_i	36	35	34	33	32	31	30	30	29	28	28	27	27	26	25	25	24
x_i	51	52	53	54	55	56	57	58	59	60	61	62	63	64			
y_i	24	23	23	23	22	22	21	21	21	20	20	20	19	19			

The default value of variable $valNu$ is equal to 0,5.

When **saturationGainNumVal** is greater than 2, $valNu$ shall be derived as follows:

- $valNu$ is equal to 0
- for i=1..(**saturationGainNumVal** - 2), the following applies:
  - $valNu$ = 2 × $valNu$ + ((255 × **saturationGainX**[i]) % 2)

- $valNu = 0{,}5 \times (\ valNu\ \div (\ 2^{\mathbf{saturationGainNumVal}\ -\ 2} - 1\ )\ )$

The variable muA shall be equal to $valNu$.

The variable muB shall be equal to ($ratioMuBmuA \times valNu$).

where $ratioMuBmuA$ is equal to 0,970738 if **sdrPicColourSpace** is equal to 0,

otherwise $ratioMuBmuA$ is equal to 0,884343.

7.2.3.3 Luminance mapping table retrieval (payloadMode 1)

This process derives, for payload mode 1, a 1D look-up table $lutMapY$ from the luminance mapping variables specified in clause 6.3.6.

This process takes as inputs:

- the luminance mapping table variables **luminanceMappingNumVal**, **luminanceMappingX**[i] and **luminanceMappingY**[i];
- the mastering display max luminance **hdrMasterDisplayMaxLuminance**.

The process generates as output:

- the luminance mapping table $lutMapY$ of $maxSampleVal$ entries.

The variables **luminanceMappingX**[i] and **luminanceMappingY**[i], for i=0..(**luminanceMappingNumVal** - 1), correspond to piece-wise linear pivot points representative of the curve $f_{luma}(\ )$ used to derive the look-up table $lutMapY$. See clause 7.3 for the computation of $f_{luma}(\ )$ from the list of points.

For any Y in 0..($maxSampleVal$ - 1), $lutMapY[\ Y]$ is derived as follows:

$$lutMapY[\ Y\ ] = \sqrt{L_{HDR}} \times f_{luma}\left(\frac{Y}{maxSampleVal-1}\right) \tag{22}$$

7.2.3.4 Colour correction table retrieval (payloadMode 1)

The process derives, for payload mode 1, a 1D look-up table $lutCC$ from the colour correction table as described in clause 6.3.7.

This process takes as inputs:

- the colour correction table variables **colourCorrectionNumVal**, **colourCorrectionX**[i] and **colourCorrectionY**[i].

The process generates as output:

- the colour correction table $lutCC$ of $maxSampleVal$ entries.

The variables **colourCorrectionX**[i] and **colourCorrectionY**[i], for i=0..(**colourCorrectionNumVal** - 1), correspond to piece-wise linear pivot points representative of the curve $f_{chroma}(\ )$ used to derive the look-up table $lutCC$. See clause 7.3 for the computation of $f_{chroma}(\ )$ from the list of points.

For any Y in 0..($maxSampleVal$ - 1), $lutCC[\ Y]$ is derived as follows:

$$lutCC[\ Y\ ] = R_{chroma} \times f_{chroma}\left(\frac{Y}{maxSampleVal - 1}\right) \tag{23}$$

with R_{chroma} equal to 0,125 in the present document.

7.2.4 HDR picture reconstruction from look-up tables and SDR picture

The HDR reconstruction process generates the reconstructed HDR picture from the decoded SDR picture and the luminance mapping and colour correction tables.

This process takes as inputs:

- an SDR picture made of two-dimensional arrays SDR_Y, SDR_{Cb}, SDR_{Cr} of width *picWidth* and height *picHeight*, after applying on the decoded picture an (unspecified) upsampling conversion process to the 4:4:4 colour sampling format, an (unspecified) samples conversion to full range and possibly an (unspecified) bitdepth conversion to 10 bits per component;
- the luminance mapping table *lutMapY* of *maxSampleVal* entries;
- the colour correction table *lutCC* of *maxSampleVal* entries;
- the luma injection variables *muA* and *muB*;
- the HDR picture mastering display max luminance **hdrMasterDisplayMaxLuminance**.

The process generates as output:

- the decoded HDR 4:4:4 picture made of two-dimensional arrays HDR_R, HDR_G, HDR_B of width *picWidth* and height *picHeight*.

NOTE: The final conversion of the output HDR 4:4:4 RGB picture to the output format adapted to the rendering device is not described in the present document.

The HDR reconstruction process performs the following successive steps for each pixel x = 0..(*picWidth - 1*), y = 0..(*picHeight - 1*):

- the variables U_{post1} and V_{post1} are derived as follows:

$$\begin{cases} U_{post1} = SDR_{cb}[x][y] - midSampleVal \\ V_{post1} = SDR_{cr}[x][y] - midSampleVal \end{cases} \quad (24)$$

- the variable Y_{post1} is derived as follows:

$$Y_{post1} = SDR_y[x][y] + Max(\,0\,; muA \times U_{post1} + muB \times V_{post1}\,) \quad (25)$$

$$Y_{post1} = Clip3(0, maxSampleVal - 1, Y_{post1}) \quad (26)$$

- U_{post1} and V_{post1} are modified as follows:

$$\begin{cases} U_{post1} = lutCC[Y_{post1}] \times U_{post1} \\ V_{post1} = lutCC[Y_{post1}] \times V_{post1} \end{cases} \quad (27)$$

- the variables *R1*, *G1*, *B1* are derived as follows:
  - the variable *T* is computed as follows:

$$T = k0 \times U_{post1} \times V_{post1} + k1 \times U_{post1} \times U_{post1} + k2 \times V_{post1} \times V_{post1} \quad (28)$$

    where *k0 , k1, k2* are derived as specified in Table 14, with *colourSpace* equal to **sdrPicColourSpace**.

  - the variable *S0* is initialized to 0, and the following applies:
    - If ($T \leq 1$), *S0* is set to *Sqrt(1 - T)*
    - Otherwise ($T > 1$), U_{post1} and V_{post1} are modified as follows:

$$\begin{cases} U_{post1} = \frac{U_{post1}}{\sqrt{T}} \\ V_{post1} = \frac{V_{post1}}{\sqrt{T}} \end{cases} \quad (29)$$

- *R1*, *G1*, *B1* are derived as follows:

$$\begin{bmatrix} R1 \\ G1 \\ B1 \end{bmatrix} = M_{Y'CbCr-to-R'G'B'} \times \begin{bmatrix} S0 \\ U_{post1} \\ V_{post1} \end{bmatrix} \quad (30)$$

  where $M_{Y'CbCr-to-R'G'B'}$ is the conversion matrix from Y'CbCr to R'G'B', as specified in Table 17 with variable $matrixIdx$ equal to **sdrPicColourSpace**.

- the variables *R2*, *G2*, *B2* are derived from *R1*, *G1*, *B1* as follows:

$$\begin{cases} R2 = lutMapY[Y_{post1}] \times R1 \\ G2 = lutMapY[Y_{post1}] \times G1 \\ B2 = lutMapY[Y_{post1}] \times B1 \end{cases} \quad (31)$$

- the output samples $HDR_R[x][y]$, $HDR_G[x][y]$, $HDR_B[x][y]$ are derived from *R2*, *G2*, *B2* as follows:

$$\begin{cases} HDR_R[x][y] = Clip3(0, L_{HDR}, R2^2) \\ HDR_G[x][y] = Clip3(0, L_{HDR}, G2^2) \\ HDR_B[x][y] = Clip3(0, L_{HDR}, B2^2) \end{cases} \quad (32)$$

where L_{HDR} is equal to **hdrDisplayMasterMaxLuminance.**

Table 14: Variables k0, k1, k2.

colourSpace	**k0**	**k1**	**k2**
0	0,12562	0,27372	0,68402
1	0,12749	0,22826	0,79256

Table 15: Conversion matrices from XYZ to RGB.

matrixIdx	$M_{XYZ\text{-}to\text{-}RGB}$		
0	3,240970	-1,537383	-0,498611
	-0,969244	1,875968	0,041555
	0,055630	-0,203977	1,056972
1	1,716651	-0,355671	-0,253366
	-0,666684	1,616481	0,015768
	0,017640	-0,042771	0,942103

Table 16: Conversion matrices from RGB to XYZ

matrixIdx	$M_{RGB\text{-}to\text{-}XYZ}$		
0	0,412391	0,357584	0,180481
	0,212639	0,715169	0,072192
	0,019331	0,119195	0,950532
1	0,636958	0,144617	0,168881
	0,2627	0,677998	0,059302
	0	0,028073	1,060985
2	0,486571	0,265668	0,198217
	0,228975	0,691739	0,079287
	0	0,045113	1,043944

Table 17: Conversion matrices from Y'CbCr to R'G'B'

matrixIdx	$M_{Y'CbCr\text{-}to\text{-}R'G'B'}$		
0	1	0	1,574839015
	1	-0,187375853	-0,468146545
	1	1,855609707	0
1	1	0	1,4746
	1	-0,16455	-0,57135
	1	1,8814	0

7.3 Piecewise linear function computation

A piecewise linear function $f(x)$ that is specified by a list of{ x_i, y_i } pairs shall be computed as specified in this clause.

The $y = f(x)$ values that are not in the list of { x_i, y_i } pairs shall be obtained through interpolation. The default interpolation shall be linear interpolation.

The default processing of $y = f(x)$ shall be according to equation (33).

$$y = y[i] + (y[i+1] - y[i]) \times \frac{x - x[i]}{(x[i+1] - x[i])} \quad (33)$$

where:

- y = output value
- x = input value
- $y[i]$ = element y_i in the list
- $x[i]$ = element x_i in the list
- i = index into the list, such that x is in the interval ($x[i], x[i+1]$)

Annex A (normative): SDR-to-HDR reconstruction metadata mapping from HEVC specification

A.1 Introduction

This annex specifies the mapping between the syntax elements present in sections of the HEVC v3 specification 4 and the dynamic metadata variables provided in clause 6. Clause A.2 allows retrieval of the specification version and payload mode to which the associated HEVC bitstream conforms to. Clause A.3 specifies the mapping of input and output signal characteristics of the HDR reconstruction process. Clause A.4 specifies the mapping specific to **payloadMode** 0. Clause A.5 specifies the mapping specific to **payloadMode** 1.

An HEVC bitstream conforming to the present document shall at least contain for at least the first access unit of the CLVS the following SEI messages additionally to the coded video SDR stream:

- 1 User Data Registered SEI message identifying specification version and payload mode;
- 1 Mastering Display Colour Volume SEI message, common to both payload modes;
- 1 Colour Volume Reconstruction Information SEI message as defined in clause A.4.2 (wrapped in a User Data Registered SEI message), when **payloadMode** is equal to 0;
- 1 Colour Remapping Information SEI message, when **payloadMode** is equal to 1.

A.2 Specification version and payload mode mapping

A.2.1 Introduction

This clause specifies the format of the HEVC SEI message that enables identification of the specification version and the payload mode used for carrying dynamic metadata specified in the present document in an HEVC bitstream.

A.2.2 ETSI TS 103 433 information SEI message syntax

Specification version and payload mode are carried in an HEVC "User data registered by Recommendation ITU-T T.35" SEI message as specified in Table A.1.

Table A.1: ts_103_433_info SEI message syntax

Syntax	Descriptor
ts_103_433_info(payloadSize) {	
terminal_provider_code	u(16)
terminal_provider_oriented_code_message_idc	u(8)
ts_103_433_spec_version	u(4)
ts_103_433_payload_mode	u(4)
}	

A.2.3 ETSI TS 103 433 information SEI message semantics

This SEI message provides information to identify the payload mode and version number of the present document to which the associated bitstream conforms to.

- **terminal_provider_code** contains an identifying number that is provided by an Administration or a national body to register the SEI message. **terminal_provider_code** most significant byte shall be equal to 0b00000000 (0x00). **terminal_provider_code** least significant byte shall be equal to 0b00111010 (0x3A).

- **terminal_provider_oriented_code_message_idc** contains an identifying number to indicate the SEI message type. **terminal_provider_oriented_code_message_idc** shall be equal to 0b00000010 (0x02).

- **ts_103_433_spec_version** contains an identifying number that is used to identify the version number to which the associated bitstream conforms to. The value of **ts_103_433_spec_version** shall be in the range of 0 to 15, inclusive. In bitstreams conforming to this version of this document, **ts_103_433_spec_version** shall be equal to 0.

- **ts_103_433_payload_mode** contains an identifying number that is used to identify the payload mode to which the associated bitstream conforms to. In bitstreams conforming to this version of this document, the value of **ts_103_433_payload_mode** shall be in the range of 0 to 1, inclusive.

A.2.4 ETSI TS 103 433 information SEI message wrapping in HEVC user data registered SEI message

The present document information SEI message shall be wrapped in an HEVC "User data registered by Recommendation ITU-T T.35" SEI message as specified in clauses D.2.6 and D.3.6 of HEVC specification [4].

- **itu_t_t35_country_code** shall be equal to 0b10110101 (0xB5). This code designates United States of America.

- **itu_t_t35_payload_byte** shall be populated by syntax elements described in clause A.2.2.

A.3 Characteristics mapping

A.3.1 Introduction

This clause specifies the mapping between syntax elements of HEVC specification [4] and dynamic metadata variables representative of the characteristics of the reconstructed HDR picture (output signal of the HDR reconstruction process) and the SDR picture (input signal of the HDR reconstruction process) specified in clause 6.

A.3.2 Reconstructed HDR picture characteristics mapping

A.3.2.1 Introduction

The characteristics of the display used to master the associated HDR content, hdr_characteristics( ) variables, shall be mapped from HEVC bitstreams according to clauses A.3.2.2 to A.3.2.5.

A.3.2.2 hdrPicColourSpace mapping

When **payloadMode** is equal to 0, **hdrPicColourSpace** is inferred from colour volume reconstruction information SEI message (specified in clause A.4.2) as specified in Table A.2.

Table A.2: Mapping from HEVC CVRI to hdrPicColourSpace variable (payloadMode 0)

CVRI syntax element	CVRI syntax element value	hdrPicColourSpace value
cv_rec_primaries **cv_rec_matrix_coefficients**	1 1	0
cv_rec_primaries **cv_rec_matrix_coefficients**	9 9	1

When **payloadMode** is equal to 1, **hdrPicColourSpace** is inferred from colour remapping information SEI message as specified in Table A.3.

Table A.3: Mapping from HEVC CRI to hdrPicColourSpace variable (payloadMode 1)

CRI syntax element	CRI syntax element value	hdrPicColourSpace value
colour_remap_primaries **colour_remap_matrix_coefficients**	1 1	0
colour_remap_primaries **colour_remap_matrix_coefficients**	9 9	1

A.3.2.3 hdrMasterDisplayColourSpace mapping

The mapping to **hdrMasterDisplayColourSpace** variable from HEVC MDCV SEI syntax elements is specified in Table A.4.

Table A.4: Mapping of hdrMasterDisplayColourSpace from HEVC MDCV SEI message

MDCV SEI syntax element	MDCV SEI syntax element value	hdrMasterDisplayColourSpace value	Note: Colour space
display_primaries_x[0] **display_primaries_y[0]** **display_primaries_x[1]** **display_primaries_y[1]** **display_primaries_x[2]** **display_primaries_y[2]** **white_point_x** **white_point_y**	15 000 30 000 7 500 3 000 32 000 16 500 15 635 16 450	0	Recommendation ITU-R BT.709-6 [9]
display_primaries_x[0] **display_primaries_y[0]** **display_primaries_x[1]** **display_primaries_y[1]** **display_primaries_x[2]** **display_primaries_y[2]** **white_point_x** **white_point_y**	8 500 39 850 6 550 2 300 35 400 14 600 15 635 16 450	1	Recommendation ITU-R BT.2020-2 [10]
display_primaries_x[0] **display_primaries_y[0]** **display_primaries_x[1]** **display_primaries_y[1]** **display_primaries_x[2]** **display_primaries_y[2]** **white_point_x** **white_point_y**	13 250 34 500 7 500 3 000 34 000 16 000 15 635 16 450	2	SMPTE RP 431-2 [8]
NOTE: In case the MDCV SEI syntax elements retrieved from the bitstream do not exactly match with proposed values of column 2, it is recommended to allocate a value to **hdrMasterDisplayColourSpace** that is the closest match to the column 2 values.			

A.3.2.4 hdrMasterDisplayMaxLuminance mapping

hdrMasterDisplayMaxLuminance is mapped from **max_display_mastering_luminance** of HEVC Mastering Display Colour Volume SEI message as follows:

$$\textbf{hdrMasterDisplayMaxLuminance} = Min(50 \times (\textbf{max\_display\_mastering\_luminance} \times 0{,}0001 + 25) / 50 \,;\, 10000) \qquad \text{(A.1)}$$

A.3.2.5 hdrMasterDisplayMinLuminance mapping

hdrMasterDisplayMinLuminance is mapped from **min_display_mastering_luminance** of HEVC Mastering Display Colour Volume SEI message as follows:

$$\textbf{hdrMasterDisplayMinLuminance} = Min(\textbf{min\_display\_mastering\_luminance} \times 0{,}0001 \,;\, 10000) \qquad \text{(A.2)}$$

A.3.3 SDR picture characteristics mapping

The characteristics of the coded SDR picture shall be mapped from HEVC VUI syntax elements to sdr_characteristics( ) variables as specified in this clause and in Table A.5.

Table A.5: Mapping from HEVC VUI to sdrPicColourSpace variables

VUI syntax element	VUI syntax element value	sdrPicColourSpace value
colour_primaries **matrix_coeffs**	1 1	0
colour_primaries **matrix_coeffs**	9 9	1

Bitstreams that conform to this version of the present document shall obey to the following constraint:

- **colour_description_present_flag** shall be equal to 1.
- **sdrMasterDisplayColourSpace** value shall be derived as documented in clause 6.3.3.3.

Independently of the value of **video_full_range_flag** in HEVC VUI, it is recommended that the full 8-bit or 10-bit signal range is preserved at the encoding stage.

A.4 Mapping to parameter-based mode variables (payloadMode 0)

A.4.1 Introduction

This clause specifies the mapping from HEVC Colour Volume Reconstruction Information SEI message syntax elements to the parameter-based payload mode variables (**payloadMode** 0) specified in clause 6. In clause A.4.2, the Colour Volume Reconstruction Information SEI message is specified and wrapped in an HEVC User Data Registered SEI message. Clause A.4.3 specifies the mapping from CVRI SEI message syntax elements to the parameter-based payload mode variables.

A.4.2 HEVC colour volume reconstruction information SEI message

A.4.2.1 Colour volume reconstruction information SEI message syntax

This SEI message is a user data registered SEI message. It is derived from a subset of SMPTE ST 2094-20 [6] dynamic metadata. The SEI message is specified in Table A.6.

Table A.6: colour_volume_reconstruction_info SEI message syntax (based on subpart of ST 2094-20 [6])

Syntax	Descriptor
colour_volume_reconstruction_info(payloadSize) {	
terminal_provider_code	u(16)
terminal_provider_oriented_code_message_idc	u(8)
cv_rec_id	ue(v)
cv_rec_cancel_flag	u(1)
if(!cv_rec_cancel_flag) {	
cv_rec_persistence_flag	u(1)
cv_rec_target_info_flag	u(1)
if(cv_rec_target_info_flag) {	
cv_rec_primaries	u(8)
cv_rec_matrix_coefficients	u(8)
}	
tone_mapping_input_signal_black_level_offset	u(8)
tone_mapping_input_signal_white_level_offset	u(8)
shadow_gain_control	u(8)
highlight_gain_control	u(8)
mid_tone_width_adjustment_factor	u(8)
tone_mapping_output_fine_tuning_num_val	u(4)
for(i = 0; i < tone_mapping_output_fine_tuning_num_val; i++) {	
tone_mapping_output_fine_tuning_x[i]	u(8)
tone_mapping_output_fine_tuning_y[i]	u(8)
}	
saturation_gain_num_val	u(4)
for(i = 0; i < saturation_gain_num_val; i++) {	
saturation_gain_x[i]	u(8)
saturation_gain_y[i]	u(8)
}	
}	
}	

A.4.2.2 Colour volume reconstruction information SEI message semantics

This SEI message provides information to enable reconstruction of an original colour volume of a signal when combined with the decoded colour samples of the output pictures. The input to the indicated colour volume transform process is in luminance-chrominance colour space. The set of decoded sample values shall apply an (unspecified) upsampling conversion process to the 4:4:4 colour sampling format as necessary when chroma_format_idc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format).

The information conveyed in this SEI message is intended to be adequate for a subset of purposes corresponding to the use of SMPTE ST 2094-20 [6].

terminal_provider_code contains an identifying number that is provided by an Administration or a national body to register the SEI message. **terminal_provider_code** most significant byte shall be equal to 0b00000000 (0x00). **terminal_provider_code** least significant byte shall be equal to 0b00111010 (0x3A).

terminal_provider_oriented_code_message_idc contains an identifying number that is provided by an Administration or a national body to register the SEI message. **terminal_provider_oriented_code_message_idc** shall be equal to 0b00000100 (0x04).

cv_rec_id contains an identifying number that may be used to identify the purpose of the HDR reconstruction information. The value of **cv_rec_id** shall be in the range of 0 to 2^{32} - 2, inclusive.

Values of **cv_rec_id** from 0 to 255 and from 512 to 2^{31} - 1 may be used as determined by the application. Values of **cv_rec_id** from 256 to 511, inclusive, and from 2^{31} to 2^{32} - 2, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all colour volume reconstruction information SEI messages containing a value of **cv_rec_id** in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32} - 2, inclusive, and bitstreams shall not contain such values.

NOTE: The **cv_rec_id** can be used to support different colour volume reconstruction processes that are suitable for different reconstruction scenarios. For example, a value of **cv_rec_id** may indicate that no gamut mapping or colour space change is allowed during the colour volume reconstruction process. A non-null value of **cv_rec_id** may indicate the gamut mapping and colour space change that are allowed or that are to be performed during the colour volume reconstruction process.

cv_rec_cancel_flag equal to 1 indicates that the colour volume reconstruction information SEI message cancels the persistence of any previous colour volume reconstruction information SEI message in output order that applies to the current layer. **cv_rec_cancel_flag** equal to 0 indicates that colour volume reconstruction information follows.

cv_rec_persistence_flag specifies the persistence of the colour volume reconstruction information SEI message for the current layer.

cv_rec_persistence_flag equal to 0 specifies that the colour volume reconstruction information applies to the current picture only.

Let picA be the current picture. **cv_rec_persistence_flag** equal to 1 specifies that the colour volume reconstruction information persists for the current layer in output order until either of the following conditions is true.

- A new CLVS of the current layer begins.
- The bitstream ends.
- A picture picB in the current layer in an access unit containing a colour volume reconstruction information SEI message with the same value of cv_rec_id and applicable to the current layer is output for which PicOrderCnt(picB) is greater than PicOrderCnt(picA), where PicOrderCnt(picB) and PicOrderCnt(picA) are the PicOrderCntVal values of picB and picA, respectively, immediately after the invocation of the decoding process for picture order count for picB.

cv_rec_target_info_flag equal to 1 specifies that syntax elements **cv_rec_primaries** and **cv_rec_matrix_coefficients** are present, **cv_rec_target_info_flag** equal to 0 specifies that **cv_rec_primaries** and **cv_rec_matrix_coefficients** are not present.

cv_rec_primaries has the same semantics as specified in clause E.3.1 of HEVC specification [4] for the **colour_primaries** syntax element, except that **cv_rec_primaries** identifies the colour space of the colour volume reconstructed picture, rather than the colour space used for the CLVS. When not present, the value of **cv_rec_primaries** is inferred to be equal to the value of **colour_primaries**.

cv_rec_matrix_coefficients has the same semantics as specified in clause E.3.1 of HEVC specification [4] for the **matrix_coeffs** syntax element, except that **cv_rec_matrix_coefficients** identifies the colour space of the colour volume reconstructed picture, rather than the colour space used for the CLVS. When not present, the value of **cv_rec_matrix_coefficients** is inferred to be equal to the value of matrix_coeffs.

tone_mapping_input_signal_black_level_offset indicates the black level offset to be subtracted during the colour volume reconstruction process. The value of **tone_mapping_input_signal_black_level_offset** shall be in the range of 0 to 255, inclusive.

tone_mapping_input_signal_white_level_offset indicates the white level offset to be subtracted from during the colour volume reconstruction process. The value of **tone_mapping_input_signal_white_level_offset** shall be in the range of 0 to 255, inclusive.

shadow_gain_control indicates the adjustment to the shadow (darker) region of the luminance mapping curve. The value of **shadow_gain_control** shall be in the range of 0 to 255, inclusive.

highlight_gain_control indicates the adjustment to the highlight (brighter) region of the luminance mapping curve. The value of **highlight_gain_control** shall be in the range of 0 to 255, inclusive.

mid_tone_width_adjustment_factor indicates the adjustment to the mid-tone region of the tone mapping. The value of **mid_tone_width_adjustment_factor** shall be in the range of 0 to 255, inclusive.

tone_mapping_output_fine_tuning_num_val specifies the number of pivot points to be adjusted in the piece-wise linear luminance mapping curve. When **tone_mapping_output_fine_tuning_num_val** is equal to 0, no adjustment points are defined. In bitstreams conforming to this version of the present document, the value of **tone_mapping_output_fine_tuning_num_val** shall be in the range of 0 to 10, inclusive.

tone_mapping_output_fine_tuning_x[i] specifies the input value of the i-th adjusted pivot point for the luminance mapping curve. The value of **tone_mapping_output_fine_tuning_x**[i] shall be in the range of 0 to 255, inclusive.

tone_mapping_output_fine_tuning_y[i] specifies the output value of the i-th adjusted pivot point for the luminance mapping curve. The value of **tone_mapping_output_fine_tuning_y**[i] shall be in the range of 0 to 255, inclusive.

When **tone_mapping_output_fine_tuning_x**[0] is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **tone_mapping_output_fine_tuning_x**[0], inclusive, to target values ranging from 0 to **tone_mapping_output_fine_tuning_y**[0], inclusive.

When **tone_mapping_output_fine_tuning_x**[**tone_mapping_output_fine_tuning_num_val** - 1] is not equal to 255, a final linear segment shall be inferred that maps input values ranging from **tone_mapping_output_fine_tuning_x**[**tone_mapping_output_fine_tuning_num_val** - 1] to 255, inclusive, to target values ranging from **tone_mapping_output_fine_tuning_y**[**tone_mapping_output_fine_tuning_num_val** - 1] to 255, inclusive.

saturation_gain_num_val specifies the number of pivot points to be adjusted in the piece-wise colour correction curve. When **saturation_gain_num_val** is equal to 0, no adjustment points are defined. In bitstreams conforming to this version of the present document, the value of **saturation_gain_num_val** shall be in the range of 0 to 6, inclusive.

saturation_gain_x[i] specifies the input value of the i-th adjusted pivot point for the colour correction curve. The value of **saturation_gain_x**[i] shall be in the range of 0 to 255, inclusive.

saturation_gain_y[i] specifies the output value of the i-th adjusted pivot point for the colour correction curve. The value of **saturation_gain_y**[i] shall be in the range of 0 to 255, inclusive.

When **saturation_gain_x**[0] is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **saturation_gain_x**[0], inclusive, to target values ranging from 0 to **saturation_gain_y**[0], inclusive.

When **saturation_gain_x**[**saturation_gain_num_val** - 1] is not equal to 255, a final linear segment shall be inferred that maps input values ranging from **saturation_gain_x**[**saturation_gain_num_val** - 1] to 255, inclusive, to target values ranging from **saturation_gain_y**[**saturation_gain_num_val** - 1] to 255, inclusive.

A.4.2.3 Colour volume reconstruction information SEI message wrapping in HEVC user data registered SEI message

The colour volume reconstruction information SEI message shall be wrapped in an HEVC "User data registered by Recommendation ITU-T T.35" SEI message as specified in clauses D.2.6 and D.3.6 of HEVC specification [4].

- **itu_t_t35_country_code** shall be equal to 0b10110101 (0xB5). This code designates United States of America.
- **itu_t_t35_payload_byte** shall be populated by syntax elements described in clause A.4.2.1.

A.4.3 Luminance mapping and colour correction adjustment variables mapping

Syntax elements values of the colour volume reconstruction information shall be mapped to luminance_mapping_variables( ) and colour_correction_adjustment( ) variables as specified in Table A.7.

Table A.7: Mapping from CVRI SEI syntax elements to luminance_mapping_variables( ) and colour_correction_adjustment( ) variables

Variables	CVRI SEI syntax element value
tmInputSignalBlackLevelOffset	**tone_mapping_input_signal_black_level_offset** ÷ 255
tmInputSignalWhiteLevelOffset	**tone_mapping_input_signal_white_level_offset** ÷ 255
shadowGain	**shadow_gain_control** × 2 ÷ 255
highlightGain	**highlight_gain_control** × 2 ÷ 255
midToneWidthAdjFactor	**mid_tone_width_adjustment_factor** × 2 ÷ 255
tmOutputFineTuningNumVal	**tone_mapping_output_fine_tuning_num_val**
tmOutputFineTuningX[i]	**tone_mapping_output_fine_tuning_x**[i] ÷ 255
tmOutputFineTuningY[i]	**tone_mapping_output_fine_tuning_y**[i] ÷ 255
saturationGainNumVal	**saturation_gain_num_val**
saturationGainX[i]	**saturation_gain_x**[i] ÷ 255
saturationGainY[i]	**saturation_gain_y**[i] ÷ 255

In the colour_volume_reconstruction_info (CVRI SEI message), the following restrictions shall apply on the bitstream:

- When present, **cv_rec_primaries** shall be equal to 1 or 9.
- When present, **cv_rec_matrix_coefficients** shall be equal to 1 or 9.

A.5 Mapping to table-based mode variables (payloadMode 1)

A.5.1 Introduction

This clause specifies the mapping from HEVC Colour Remapping Information SEI message syntax elements to the table-based payload mode variables (**payloadMode** 1) specified in clause 6.

A.5.2 Luminance mapping and colour correction tables variables mapping

Syntax elements values of the colour volume reconstruction information shall be mapped to hdr_reconstruction_info( ), luminance_mapping_table( ) and colour_correction_table( ) variables as specified in Table A.8.

Table A.8: Mapping from CRI SEI syntax elements to luminance_mapping_table( ) and colour_correction_table( ) variables

Variables	CRI SEI syntax element value
luminanceMappingNumVal	**post_lut_num_val_minus1**[0] + 1
luminanceMappingX[i]	**post_lut_coded_value**[0][i] ÷ 16 383
luminanceMappingY[i]	**post_lut_target_value[** 0][i] ÷ 16 383
colourCorrectionNumVal	**post_lut_num_val_minus1**[1] + 1
colourCorrectionX[i]	**post_lut_coded_value**[1][i] ÷ 2 047
colourCorrectionY[i]	**post_lut_target_value**[1][i] ÷ 2 047
chromaToLumaInjectionMuA	**colour_remap_coeffs**[0][1] ÷ 16 383
chromaToLumaInjectionMuB	**colour_remap_coeffs**[0][2] ÷ 16 383

In the colour_remapping_info (CRI SEI message), the following restrictions shall apply on the bitstream:

- **colour_remap_id** shall be greater or equal to 512.
- When present, **colour_remap_primaries** shall be equal to 1 or 9.
- When present, **colour_remap_matrix_coefficients** shall be equal to 1 or 9.

- **colour_remap_input_bit_depth** shall be equal to 14.
- **colour_remap_bit_depth** shall be equal to 14.
- **pre_lut_num_val_minus1**[2] shall be equal to 0.
- **colour_remap_matrix_present_flag** shall be equal to 1.
- **log2_matrix_denom** shall be equal to 0.
- **post_lut_coded_value**[0][i] shall be in the bounded range [0 to 16 383].
- **post_lut_target_value**[0][i] shall be in the bounded range [0 to 8 192].
- **post_lut_coded_value**[1][i] shall be in the bounded range [0 to 2 047].
- **post_lut_target_value**[1][i] shall be in the bounded range [0 to 2 047].
- **colour_remap_coeffs**[0][1] shall be in the bounded range [0 to 8 191].
- **colour_remap_coeffs**[0][2] shall be in the bounded range [0 to 8 191].
- **colour_remap_coeffs**[0][0];
colour_remap_coeffs[1][1];
colour_remap_coeffs[2][2] shall be equal to 1.
- **colour_remap_coeffs**[1][0];
colour_remap_coeffs[2][0];
colour_remap_coeffs[2][1];
colour_remap_coeffs[1][2] shall be equal to 0.
- **pre_lut_num_val_minus1**[c] shall be equal to 0 for c = 0, 1, 2.

Annex B (normative): SDR-to-HDR reconstruction metadata mapping from AVC specification

B.1 Introduction

This normative annex specifies the mapping between the syntax elements present in sections of the AVC specification [3] and the dynamic metadata variables provided in clause 6. Clause B.2 allows retrieval of the specification version and payload mode to which the associated AVC bitstream conforms to. Clause B.3 specifies the mapping of input and output signal characteristics of the HDR reconstruction process. Clause B.4 specifies the mapping specific to **payloadMode** 0. Clause B.5 specifies the mapping specific to **payloadMode** 1.
An AVC bitstream conforming to the present document shall at least contain for at least the first access unit of the coded video sequence the following SEI messages additionally to the coded video SDR stream:

- 1 User Data Registered SEI message identifying specification version and payload mode;
- 1 Mastering Display Colour Volume SEI message, common to both payload modes;
- 1 Colour Volume Reconstruction Information SEI message as defined in clause B.4.2 (wrapped in a User Data Registered SEI message), when **payloadMode** is equal to 0;
- 1 Colour Remapping Information SEI message (wrapped in a User Data Unregistered SEI message), when **payloadMode** is equal to 1.

NOTE: CRI SEI is not specified in the February 2016 published version of AVC. CRI SEI message is embedded in a User Data Unregistered SEI message pending standardization of those SEI messages in AVC specification [3].

B.2 Specification version and payload mode mapping

B.2.1 Introduction

This clause specifies the format of the AVC SEI that enables identification of the specification version and the payload mode used for carrying dynamic metadata specified in the present document in an AVC bitstream.

B.2.2 ETSI TS 103 433 information SEI message syntax

Specification version and payload mode are carried in an AVC "User data registered by Recommendation ITU-T T.35" SEI message as specified in Table B.1.

Table B.1: ts_103_433_info SEI message syntax

Syntax	C	Descriptor
ts_103_433_info(payloadSize) {		
terminal_provider_code	5	u(16)
terminal_provider_oriented_code_message_idc	5	u(8)
ts_103_433_spec_version	5	u(4)
ts_103_433_payload_mode	5	u(4)
}		

B.2.3 ETSI TS 103 433 information SEI message semantics

This SEI message provides information to identify the payload mode and version number of the present document to which the associated bitstream conforms to:

- **terminal_provider_code** contains an identifying number that is provided by an Administration or a national body to register the SEI message. **terminal_provider_code** most significant byte shall be equal to 0b00000000 (0x00). **terminal_provider_code** least significant byte shall be equal to 0b00111010 (0x3A).
- **terminal_provider_oriented_code_message_idc** contains an identifying number that is provided by an Administration or a national body to register the SEI message. **terminal_provider_oriented_code_message_idc** shall be equal to 0b00000011 (0x03).
- **ts_103_433_spec_version** contains an identifying number that is used to identify the version number to which the associated bitstream conforms to. The value of **ts_103_433_spec_version** shall be in the range of 0 to 15, inclusive. In bitstreams conforming to this version of the present document, **ts_103_433_spec_version** shall be equal to 0.
- **ts_103_433_payload_mode** contains an identifying number that is used to identify the payload mode to which the associated bitstream conforms to. In bitstreams conforming to this version of the present document, the value of **ts_103_433_payload_mode** shall be in the range of 0 to 1, inclusive.

B.2.4 ETSI TS 103 433 information SEI message wrapping in AVC user data registered SEI message

The present document information SEI message shall be wrapped in an AVC "User data registered by Recommendation ITU-T T.35" SEI message as specified in clauses D.1.5 and D.2.5 of AVC specification [3].

- **itu_t_t35_country_code** shall be equal to 0b10110101 (0xB5). This code designates United States of America.
- **itu_t_t35_payload_byte** shall be populated by syntax elements described in clause B.2.2.

B.3 Characteristics mapping

B.3.1 Introduction

This clause specifies the mapping between syntax elements of AVC specification [3] and dynamic metadata variables representative of the characteristics of the reconstructed HDR picture (output signal of the HDR reconstruction process) and the SDR picture (input signal of the HDR reconstruction process) specified in clause 6.

B.3.2 Reconstructed HDR picture characteristics mapping

B.3.2.1 Introduction

The characteristics of the display used to master the associated HDR content, hdr_characteristics( ) variables, shall be mapped from AVC bitstreams according to clauses B.3.2.2 to B.3.2.5.

B.3.2.2 hdrPicColourSpace mapping

When **payloadMode** is equal to 0, **hdrPicColourSpace** is inferred from colour volume reconstruction information SEI message (specified in clause B.4.2) as specified in Table B.2.

Table B.2: Mapping from AVC CVRI to hdrPicColourSpace variable (payloadMode 0)

CVRI syntax element	CVRI syntax element value	hdrPicColourSpace value
cv_rec_primaries **cv_rec_matrix_coefficients**	1 1	0
cv_rec_primaries **cv_rec_matrix_coefficients**	9 9	1

When **payloadMode** is equal to 1, **hdrPicColourSpace** is inferred from colour remapping information SEI message (specified in clause B.5.2) as specified in Table B.3.

Table B.3: Mapping from AVC CRI to hdrPicColourSpace variable (payloadMode 1)

CRI syntax element	CRI syntax element value	hdrPicColourSpace value
colour_remap_primaries **colour_remap_matrix_coefficients**	1 1	0
colour_remap_primaries **colour_remap_matrix_coefficients**	9 9	1

B.3.2.3 hdrMasterDisplayColourSpace mapping

The mapping to **hdrMasterDisplayColourSpace** variable from AVC MDCV SEI syntax elements is specified in Table B.4.

Table B.4: Mapping of hdrMasterDisplayColourSpace from AVC MDCV SEI message

MDCV SEI syntax element	MDCV SEI syntax element value	hdrMasterDisplayColourSpace value	Note: Colour space
display_primaries_x[0] **display_primaries_y[0]** **display_primaries_x[1]** **display_primaries_y[1]** **display_primaries_x[2]** **display_primaries_y[2]** **white_point_x** **white_point_y**	15 000 30 000 7 500 3 000 32 000 16 500 15 635 16 450	0	Recommendation ITU-R BT.709-6 [9]
display_primaries_x[0] **display_primaries_y[0]** **display_primaries_x[1]** **display_primaries_y[1]** **display_primaries_x[2]** **display_primaries_y[2]** **white_point_x** **white_point_y**	8 500 39 850 6 550 2 300 35 400 14 600 15 635 16 450	1	Recommendation ITU-R BT.2020-2 [10]
display_primaries_x[0] **display_primaries_y[0]** **display_primaries_x[1]** **display_primaries_y[1]** **display_primaries_x[2]** **display_primaries_y[2]** **white_point_x** **white_point_y**	13 250 34 500 7 500 3 000 34 000 16 000 15 635 16 450	2	SMPTE RP 431-2 [8]
NOTE: In case the MDCV SEI syntax elements retrieved from the bitstream do not exactly match with proposed values of column 2, it is recommended to allocate a value to **hdrMasterDisplayColourSpace** that is the closest match to the column 2 values.			

B.3.2.4 hdrMasterDisplayMaxLuminance mapping

hdrMasterDisplayMaxLuminance is mapped from **max_display_mastering_luminance** of AVC Mastering Display Colour Volume SEI message as follows:

$$\textbf{hdrMasterDisplayMaxLuminance} = Min(\,50 \times (\,\textbf{max\_display\_mastering\_luminance} \times 0{,}0001 + 25\,)\,/\,50\,;\,10000\,) \qquad \text{(B.1)}$$

B.3.2.5 hdrMasterDisplayMinLuminance mapping

hdrMasterDisplayMinLuminance is mapped from **min_display_mastering_luminance** of AVC Mastering Display Colour Volume SEI message as follows:

$$\textbf{hdrMasterDisplayMinLuminance} = Min(\,\textbf{min\_display\_mastering\_luminance} \times 0{,}0001\,;\,10000\,) \qquad \text{(B.2)}$$

B.3.3 SDR picture characteristics mapping

The characteristics of the coded SDR picture shall be mapped from AVC VUI syntax elements to sdr_characteristics( ) variables as specified in this clause and in Table B.5.

Table B.5: Mapping from AVC VUI to sdrPicColourSpace( ) variables

VUI syntax element	VUI syntax element value	sdrPicColourSpace value
colour_primaries **matrix_coefficients**	1 1	0
colour_primaries **matrix_coefficients**	9 9	1

Bitstreams that conform to this version of the present document shall obey to the following constraint:

- **colour_description_present_flag** shall be equal to 1.

sdrMasterDisplayColourSpace value shall be derived as documented in clause 6.3.3.3.

Independently of the value of **video_full_range_flag** in AVC VUI, it is strongly recommended that the full 8-bit or 10-bit signal range is preserved at the encoding stage.

B.4 Mapping to parameter-based mode variables (payloadMode 0)

B.4.1 Introduction

This clause specifies the mapping from AVC Colour Volume Reconstruction Information SEI message syntax elements to the parameter-based payload mode variables (**payloadMode** 0) specified in clause 6. In clause B.4.2, the Colour Volume Reconstruction Information SEI message is specified and wrapped in an AVC User Data Registered SEI message is specified. Clause B.4.3. specifies the mapping from CVRI SEI message syntax elements to the parameter-based payload mode variables.

B.4.2 AVC colour volume reconstruction information SEI message

B.4.2.1 Colour volume reconstruction information SEI message syntax

This SEI message is a user registered SEI message. It is derived from a subset of SMPTE ST 2094-20 [6] dynamic metadata. The SEI message is specified in Table B.6.

Table B.6: colour_volume_reconstruction_info SEI message syntax (based on subpart of SMPTE ST 2094-20 [6])

Syntax	C	Descriptor
colour_volume_reconstruction_info(payloadSize) {		
terminal_provider_code	5	u(16)
terminal_provider_oriented_code_message_idc	5	u(8)
cv_rec_id	5	ue(v)
cv_rec_cancel_flag	5	u(1)
if(!cv_rec_cancel_flag) {		
cv_rec_repetition_period	5	ue(v)
cv_rec_target_info_flag	5	u(1)
if(cv_rec_target_info_flag) {		
cv_rec_primaries	5	u(8)
cv_rec_matrix_coefficients	5	u(8)
}		
tone_mapping_input_signal_black_level_offset	5	u(8)
tone_mapping_input_signal_white_level_offset	5	u(8)
shadow_gain_control	5	u(8)
highlight_gain_control	5	u(8)
mid_tone_width_adjustment_factor	5	u(8)
tone_mapping_output_fine_tuning_num_val	5	u(4)
for(i = 0; i < tone_mapping_output_fine_tuning_num_val; i++) {		
tone_mapping_output_fine_tuning_x[i]	5	u(8)
tone_mapping_output_fine_tuning_y[i]	5	u(8)
}		
saturation_gain_num_val	5	u(4)
for(i = 0; i < saturation_gain_num_val; i++) {		
saturation_gain_x[i]	5	u(8)
saturation_gain_y[i]	5	u(8)
}		
}		
}		

B.4.2.2 Colour volume reconstruction information SEI message semantics

This SEI message provides information to enable reconstruction of an original colour volume of a signal when combined with the decoded colour samples of the output pictures. The input to the indicated colour volume transform process is in luminance-chrominance colour space. The set of decoded sample values shall apply an (unspecified) upsampling conversion process to the 4:4:4 colour sampling format as necessary when chroma_format_idc is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format).

The information conveyed in this SEI message is intended to be adequate for a subset of purposes corresponding to the use of SMPTE ST 2094-20 [6].

terminal_provider_code contains an identifying number that is provided by an Administration or a national body to register the SEI message. **terminal_provider_code** most significant byte shall be equal to 0b00000000 (0x00). **terminal_provider_code** least significant byte shall be equal to 0b00111010 (0x3A).

terminal_provider_oriented_code_message_idc contains an identifying number that is provided by an Administration or a national body to register the SEI message**. terminal_provider_oriented_code_message_idc** shall be equal to 0b00000101 (0x05).

cv_rec_id contains an identifying number that may be used to identify the purpose of the HDR reconstruction information. The value of **cv_rec_id** shall be in the range of 0 to 2^{32} - 2, inclusive.

Values of **cv_rec_id** from 0 to 255 and from 512 to 2^{31} - 1 may be used as determined by the application. Values of **cv_rec_id** from 256 to 511, inclusive, and from 2^{31} to 2^{32} - 2, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all colour volume reconstruction information SEI messages containing a value of **cv_rec_id** in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32} - 2, inclusive, and bitstreams shall not contain such values.

NOTE: The **cv_rec_id** can be used to support different colour volume reconstruction processes that are suitable for different reconstruction scenarios. For example, a value of **cv_rec_id** may indicate that no gamut mapping or colour space change is allowed during the colour volume reconstruction process. A non-null value of **cv_rec_id** may indicate the gamut mapping and colour space change that are allowed or that are to be performed during the colour volume reconstruction process.

cv_rec_cancel_flag equal to 1 indicates that the colour volume reconstruction information SEI message cancels the persistence of any previous colour volume reconstruction information SEI message in output order that applies to the current layer. **cv_rec_cancel_flag** equal to 0 indicates that colour volume reconstruction information follows.

cv_rec_repetition_period specifies the persistence of the colour volume reconstruction information SEI message and may specify a picture order count interval within which another colour volume reconstruction information SEI message with the same value of **cv_rec_id** or the end of the coded video sequence shall be present in the bitstream. The value of **cv_rec_repetition_period** shall be in the range of 0 to 16 384, inclusive.

cv_rec_repetition_period equal to 0 specifies that the colour volume reconstruction information applies to the current decoded picture only.

cv_rec_repetition_period equal to 1 specifies that the colour volume reconstruction information persists in output order until any of the following conditions are true.

- A new coded video sequence begins.
- A picture in an access unit containing a colour volume reconstruction information SEI message with the same value of **cv_rec_id** is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic).

cv_rec_repetition_period equal to 0 or equal to 1 indicates that another colour volume reconstruction remapping information SEI message with the same value of **cv_rec_id** may or may not be present.

cv_rec_repetition_period greater than 1 specifies that the colour volume reconstruction information persists until any of the following conditions are true.

- A new coded video sequence begins.
- A picture in an access unit containing a colour volume reconstruction information SEI message with the same value of cv_rec_id is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + cv_rec_repetition_period.

cv_rec_repetition_period greater than 1 indicates that another colour volume reconstruction information SEI message with the same value of **cv_rec_id** shall be present for a picture in an access unit that is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + **cv_rec_repetition_period**; unless the bitstream ends or a new coded video sequence begins without output of such a picture.

cv_rec_target_info_flag equal to 1 specifies that syntax elements **cv_rec_primaries** and **cv_rec_matrix_coefficients** are present, **cv_rec_target_info_flag** equal to 0 specifies that **cv_rec_primaries** and **cv_rec_matrix_coefficients** are not present.

cv_rec_primaries has the same semantics as specified in clause E.3.1 of AVC specification [3] for the **colour_primaries** syntax element, except that **cv_rec_primaries** identifies the colour space of the colour volume reconstructed picture, rather than the colour space used for the coded video sequence. When not present, the value of **cv_rec_primaries** is inferred to be equal to the value of **colour_primaries**.

cv_rec_matrix_coefficients has the same semantics as specified in clause E.3.1 of AVC specification [3] for the matrix_coefficients syntax element, except that **cv_rec_matrix_coefficients** identifies the colour space of the colour volume reconstructed picture, rather than the colour space used for the coded video sequence. When not present, the value of **cv_rec_matrix_coefficients** is inferred to be equal to the value of matrix_coefficients.

tone_mapping_input_signal_black_level_offset indicates the black level offset to be subtracted during the colour volume reconstruction process. The value of **tone_mapping_input_signal_black_level_offset** shall be in the range of 0 to 255, inclusive.

tone_mapping_input_signal_white_level_offset indicates the white level offset to be subtracted from during the colour volume reconstruction process. The value of **tone_mapping_input_signal_white_level_offset** shall be in the range of 0 to 255, inclusive.

shadow_gain_control indicates the adjustment to the shadow (darker) region of the luminance mapping curve. The value of **shadow_gain_control** shall be in the range of 0 to 255, inclusive.

highlight_gain_control indicates the adjustment to the highlight (brighter) region of the luminance mapping curve. The value of **highlight_gain_contro**l shall be in the range of 0 to 255, inclusive.

mid_tone_width_adjustment_factor indicates the adjustment to the mid-tone region of the tone mapping. The value of **mid_tone_width_adjustment_factor** shall be in the range of 0 to 255, inclusive.

tone_mapping_output_fine_tuning_num_val specifies the number of pivot points to be adjusted in the piece-wise linear luminance mapping curve. When **tone_mapping_output_fine_tuning_num_val** is equal to 0, no adjustment points are defined. In bitstreams conforming to this version of the present document, the value of **tone_mapping_output_fine_tuning_num_val** shall be in the range of 0 to 10, inclusive.

tone_mapping_output_fine_tuning_x[i] specifies the input value of the i-th adjusted pivot point for the luminance mapping curve. The value of **tone_mapping_output_fine_tuning_x**[i] shall be in the range of 0 to 255, inclusive.

tone_mapping_output_fine_tuning_y[i] specifies the output value of the i-th adjusted pivot point for the luminance mapping curve. The value of **tone_mapping_output_fine_tuning_y**[i] shall be in the range of 0 to 255, inclusive.

When **tone_mapping_output_fine_tuning_x**[0] is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **tone_mapping_output_fine_tuning_x**[0], inclusive, to target values ranging from 0 to **tone_mapping_output_fine_tuning_y**[0], inclusive.

When **tone_mapping_output_fine_tuning_x**[**tone_mapping_output_fine_tuning_num_val** - 1] is not equal to 255, a final linear segment shall be inferred that maps input values ranging from **tone_mapping_output_fine_tuning_x**[**tone_mapping_output_fine_tuning_num_val** - 1] to 255, inclusive, to target values ranging from **tone_mapping_output_fine_tuning_y**[**tone_mapping_output_fine_tuning_num_val** - 1] to 255, inclusive.

saturation_gain_num_val specifies the number of pivot points to be adjusted in the piece-wise colour correction curve. When **saturation_gain_num_val** is equal to 0, no adjustment points are defined. In bitstreams conforming to this version of the present document, the value of **saturation_gain_num_val** shall be in the range of 0 to 6, inclusive.

saturation_gain_x[i] specifies the input value of the i-th adjusted pivot point for the colour correction curve. The value of **saturation_gain_x**[i] shall be in the range of 0 to 255, inclusive.

saturation_gain_y[i] specifies the output value of the i-th adjusted pivot point for the colour correction curve. The value of **saturation_gain_y**[i] shall be in the range of 0 to 255, inclusive.

When **saturation_gain_x**[0 **]** is greater than 0, an initial linear segment shall be inferred that maps input values ranging from 0 to **saturation_gain_x**[0], inclusive, to target values ranging from 0 to **saturation_gain_y**[0], inclusive.

When **saturation_gain_x**[**saturation_gain_num_val** - 1] is not equal to 255, a final linear segment shall be inferred that maps input values ranging from **saturation_gain_x**[**saturation_gain_num_val** - 1] to 255, inclusive, to target values ranging from **saturation_gain_y**[**saturation_gain_num_val** - 1] to 255, inclusive.

B.4.2.3 Colour volume reconstruction information SEI message wrapping in AVC user data registered SEI message

The colour volume reconstruction information SEI message shall be wrapped in an AVC "User data registered by Recommendation ITU-T T.35" SEI message as specified in clauses D.1.5 and D.2.5 of AVC specification [3].

- **itu_t_t35_country_code** shall be equal to 0b10110101 (0xB5). This code designates United States of America.
- **itu_t_t35_payload_byte** shall be populated by syntax elements described in clause B.4.2.1.

B.4.3 HDR picture characteristics and model parameters mapping

Syntax elements values of the colour volume reconstruction information shall be mapped to luminance_mapping_variables( ) and colour_correction_adjustment( ) variables as specified in Table B.7.

Table B.7: Mapping from CVRI SEI syntax elements to luminance_mapping_variables( ) and colour_correction_adjustment( ) variables

Variables	CVRI SEI syntax element value
tmInputSignalBlackLevelOffset	**tone_mapping_input_signal_black_level_offset** ÷ 255
tmInputSignalWhiteLevelOffset	**tone_mapping_input_signal_white_level_offset** ÷ 255
shadowGain	**shadow_gain_control** × 2 ÷ 255
highlightGain	**highlight_gain_control** × 2 ÷ 255
midToneWidthAdjFactor	**mid_tone_width_adjustment_factor** × 2 ÷ 255
tmOutputFineTuningNumVal	**tone_mapping_output_fine_tuning_num_val**
tmOutputFineTuningX[i]	**tone_mapping_output_fine_tuning_x**[i] ÷ 255
tmOutputFineTuningY[i]	**tone_mapping_output_fine_tuning_y**[i] ÷ 255
saturationGainNumVal	**saturation_gain_num_val**
saturationGainX[i]	**saturation_gain_x**[i] ÷ 255
saturationGainY[i]	**saturation_gain_y**[i] ÷ 255

In the colour_volume_reconstruction_info (CVRI SEI message), the following restrictions shall apply on the bitstream:

- When present, **cv_rec_primaries** shall be equal to 1 or 9.
- When present, **cv_rec_matrix_coefficients** shall be equal to 1 or 9.

B.5 Mapping to table-based mode variables (payloadMode 1)

B.5.1 Introduction

This clause specifies the mapping from AVC Colour Remapping Information SEI message syntax elements to the table-based payload mode variables (**payloadMode** 1) specified in clause 6. In clause B.5.2, the Colour Remapping Information SEI message is specified and wrapped in an AVC User Data Unregistered SEI message. Clause B.5.3 specifies the mapping from CRI SEI message syntax elements to the table-based payload mode variables.

B.5.2 AVC colour remapping information SEI message

B.5.2.1 Colour remapping information SEI syntax

This SEI message is a user unregistered SEI message. It is derived from a subset of SMPTE ST 2094-30 [5] dynamic metadata. The SEI message is specified in Table B.8.

Table B.8: colour_remapping_info SEI message syntax (based on subpart of SMPTE ST 2094-30 [5])

Syntax	C	Descriptor
colour_remapping_info(payloadSize) {		
colour_remap_id	5	ue(v)
colour_remap_cancel_flag	5	u(1)
if(!colour_remap_cancel_flag) {		
colour_remap_repetition_period	5	ue(v)
colour_remap_video_signal_info_present_flag	5	u(1)
if(colour_remap_video_signal_info_present_flag) {		
colour_remap_full_range_flag	5	u(1)
colour_remap_primaries	5	u(8)

Syntax	C	Descriptor
colour_remap_transfer_function	5	u(8)
colour_remap_matrix_coefficients	5	u(8)
}		
colour_remap_input_bit_depth	5	u(8)
colour_remap_output_bit_depth	5	u(8)
for(c = 0; c < 3; c++) {		
pre_lut_num_val_minus1[c]	5	u(8)
if(pre_lut_num_val_minus1[c] > 0)		
for(i = 0; i <= pre_lut_num_val_minus1[c]; i++) {		
pre_lut_coded_value[c][i]	5	u(v)
pre_lut_target_value[c][i]	5	u(v)
}		
}		
colour_remap_matrix_present_flag	5	u(1)
if(colour_remap_matrix_present_flag) {		
log2_matrix_denom	5	u(4)
for(c = 0; c < 3; c++)		
for(i = 0; i < 3; i++)		
colour_remap_coeffs[c][i]	5	se(v)
}		
for(c = 0; c < 3; c++) {		
post_lut_num_val_minus1[c]	5	u(8)
if(post_lut_num_val_minus1[c] > 0)		
for(i = 0; i <= post_lut_num_val_minus1[c]; i++) {		
post_lut_coded_value[c][i]	5	u(v)
post_lut_target_value[c][i]	5	u(v)
}		
}		
}		
}		

B.5.2.2 Colour remapping information SEI semantics

The colour remapping information SEI message provides information to enable remapping of the reconstructed colour samples of the output pictures for purposes such as converting the output pictures to a representation that is more suitable for an alternative display. The colour remapping model used in the colour remapping information SEI message is composed of a first piece-wise linear function applied to each colour component (specified by the "pre" set of syntax elements herein), followed by a three-by-three matrix applied to the three resulting colour components, followed by a second piece-wise linear function applied to each resulting colour component (specified by the "post" set of syntax elements herein).

NOTE 1: Colour remapping of the output pictures for the display process (which is outside the scope of the present document) is optional and does not affect the decoding process specified in the present document.

Unless indicated otherwise by some means not specified in the present document, the input to the indicated remapping process is the set of decoded sample values after applying an (unspecified) upsampling conversion process to the 4:4:4 colour sampling format as necessary when the colour remapping three-by-three matrix coefficients are present in the SEI message and **chroma_format_idc** is equal to 1 (4:2:0 chroma format) or 2 (4:2:2 chroma format). When **chroma_format_idc** is equal to 0 (monochrome), the colour remapping information SEI message shall not be present, although decoders shall allow such messages to be present and shall ignore any such colour remapping information SEI messages that may be present.

colour_remap_id contains an identifying number that may be used to identify the purpose of the colour remapping information. The value of **colour_remap_id** may be used (in a manner not specified in the present document) to indicate that the input to the remapping process is the output of some conversion process that is not specified in the present document, such as a conversion of the picture to some alternative colour representation (e.g., conversion from a YCbCr colour representation to a GBR colour representation). When more than one colour remapping information SEI message is present with the same value of **colour_remap_id**, the content of these colour remapping information SEI messages shall be the same. When colour remapping information SEI messages are present that have more than one value of **colour_remap_id**, this may indicate that the remapping processes indicated by the different values of **colour_remap_id** are alternatives that are provided for different purposes or that a cascading of remapping processes is to be applied in a sequential order (an order that is not specified in the present document). The value of **colour_remap_id** shall be in the range of 0 to 2^{32} - 2, inclusive.

Values of **colour_remap_id** from 0 to 255 and from 512 to 2^{31} - 1 may be used as determined by the application. Values of **colour_remap_id** from 256 to 511, inclusive, and from 2^{31} to 2^{32} - 2, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all colour remapping information SEI messages containing a value of **colour_remap_id** in the range of 256 to 511, inclusive, or in the range of 2^{31} to 2^{32} - 2, inclusive, and bitstreams shall not contain such values.

NOTE 2: The **colour_remap_id** can be used to support different colour remapping processes that are suitable for different display scenarios. For example, different values of **colour_remap_id** may correspond to different remapped colour spaces supported by displays.

colour_remap_cancel_flag equal to 1 indicates that the colour remapping information SEI message cancels the persistence of any previous colour remapping information SEI message in output order that applies to the current layer. **colour_remap_cancel_flag** equal to 0 indicates that colour remapping information follows.

colour_remap_repetition_period specifies the persistence of the colour remapping information SEI message and may specify a picture order count interval within which another colour remapping information SEI message with the same value of **colour_remap_id** or the end of the coded video sequence shall be present in the bitstream. The value of **colour_remap_repetition_period** shall be in the range of 0 to 16 384, inclusive.

colour_remap_repetition_period equal to 0 specifies that the colour remapping information applies to the current decoded picture only.

colour_remap_repetition_period equal to 1 specifies that the colour remapping information persists in output order until any of the following conditions are true.

- A new coded video sequence begins.
- A picture in an access unit containing a colour remapping information SEI message with the same value of colour_remap_id is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic).

colour_remap_repetition_period equal to 0 or equal to 1 indicates that another colour remapping information SEI message with the same value of **colour_remap_id** may or may not be present.

colour_remap_repetition_period greater than 1 specifies that the colour remapping information persists until any of the following conditions are true.

- A new coded video sequence begins.
- A picture in an access unit containing a colour remapping information SEI message with the same value of colour_remap_id is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + colour_remap_repetition_period.

colour_remap_repetition_period greater than 1 indicates that another colour remapping information SEI message with the same value of **colour_remap_id** shall be present for a picture in an access unit that is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + **colour_remap_repetition_period**; unless the bitstream ends or a new coded video sequence begins without output of such a picture.

colour_remap_video_signal_info_present_flag equal to 1 specifies that syntax elements **colour_remap_full_range_flag**, **colour_remap_primaries**, **colour_remap_transfer_function** and **colour_remap_matrix_coefficients** are present, **colour_remap_video_signal_info_present_flag** equal to 0 specifies that syntax elements **colour_remap_full_range_flag**, **colour_remap_primaries**, **colour_remap_transfer_function** and **colour_remap_matrix_coefficients** are not present.

colour_remap_full_range_flag has the same semantics as specified in clause E.2.1 of AVC specification [3] for the **video_full_range_flag** syntax element, except that **colour_remap_full_range_flag** identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the coded video sequence. When not present, the value of **colour_remap_full_range_flag** is inferred to be equal to the **value of video_full_range_flag**.

colour_remap_primaries has the same semantics as specified in clause E.2.1 of AVC specification [3] for the **colour_primaries** syntax element, except that **colour_remap_primaries** identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the coded video sequence. When not present, the value of **colour_remap_primaries** is inferred to be equal to the value of colour_primaries.

colour_remap_transfer_function has the same semantics as specified in clause E.2.1 of AVC specification [3] for the **transfer_characteristics** syntax element, except that **colour_remap_transfer_function** identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the coded video sequence. When not present, the value of **colour_remap_transfer_function** is inferred to be equal to the value of **transfer_characteristics**.

colour_remap_matrix_coefficients has the same semantics as specified in clause E.2.1 of AVC specification [3] for the **matrix_coeffs** syntax element, except that **colour_remap_matrix_coefficients** identifies the colour space of the remapped reconstructed picture, rather than the colour space used for the coded video sequence. When not present, the value of **colour_remap_matrix_coefficients** is inferred to be equal to the value of **matrix_coeffs**.

colour_remap_input_bit_depth specifies the bit depth of the colour components of the associated pictures for purposes of interpretation of the colour remapping information SEI message. When any colour remapping information SEI message is present with the value of **colour_remap_input_bit_depth** not equal to the bit depth of the decoded colour components, the SEI message refers to the hypothetical result of a conversion operation performed to convert the decoded colour components samples to the bit depth equal to **colour_remap_input_bit_depth**.

The value of **colour_remap_input_bit_depth** shall be in the range of 8 to 16, inclusive. Values of **colour_remap_input_bit_depth** from 0 to 7, inclusive, and from 17 to 255, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all colour remapping SEI messages that contain a **colour_remap_input_bit_depth** in the range of 0 to 7, inclusive, or in the range of 17 to 255, inclusive, and bitstreams shall not contain such values.

colour_remap_output_bit_depth specifies the bit depth of the output of the colour remapping function described by the colour remapping information SEI message.

The value of **colour_remap_output_bit_depth** shall be in the range of 8 to 16, inclusive. Values of **colour_remap_output_bit_depth** from 0 to 7, inclusive, and in the range of 17 to 255, inclusive, are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all colour remapping SEI messages that contain a value of **colour_remap_output_bit_depth** from 0 to 7, inclusive, or in the range of 17 to 255, inclusive, and bitstreams shall not contain such values.

pre_lut_num_val_minus1[c] plus 1 specifies the number of pivot points in the piece-wise linear remapping function for the c-th component, where c equal to 0 refers to the luma or G component, c equal to 1 refers to the Cb or B component, and c equal to 2 refers to the Cr or R component. When **pre_lut_num_val_minus1**[c] is equal to 0, the default end points of the input values are 0 and $2^{\text{colour\_remap\_input\_bit\_depth}}$ - 1, and the corresponding default end points of the output values are 0 and $2^{\text{colour\_output\_remap\_bit\_depth}}$ - 1, for the c-th component. In bitstreams conforming to this version of the present document, the value of **pre_lut_num_val_minus1**[c] shall be in the range of 0 to 32, inclusive.

pre_lut_coded_value[c][i] specifies the input value of the i-th pivot point for the c-th component. The number of bits used to represent **pre_lut_coded_value**[c][i] is ((colour_remap_input_bit_depth + 7) >> 3) << 3.

pre_lut_target_value[c][i] specifies the output value of the i-th pivot point for the c-th component. The number of bits used to represent **pre_lut_target_value**[c][i] is ((colour_remap_output_bit_depth + 7) >> 3) << 3.

When **pre_lut_coded_value**[c][0] is greater than 0, an initial linear segment should be inferred that maps input values ranging from 0 to **pre_lut_coded_value**[c][0], inclusive, to target values ranging from 0 to **pre_lut_target_value**[c][0], inclusive.

When **pre_lut_coded_value**[c][**pre_lut_num_val_minus1**[c]] is not equal to $2^{\text{colour\_remap\_input\_bit\_depth}}$ - 1, a final linear segment should be inferred that maps input values ranging from **pre_lut_coded_value**[c][**pre_lut_num_val_minus1**[c]] to $2^{\text{colour\_remap\_input\_bit\_depth}}$ - 1, inclusive, to target values ranging from **pre_lut_target_value**[c][**pre_lut_num_val_minus1**[c]] to $2^{\text{colour\_remap\_output\_bit\_depth}}$ - 1, inclusive.

colour_remap_matrix_present_flag equal to 1 indicates that the syntax elements **log2_matrix_denom** and **colour_remap_coeffs**[c][i], for c and i in the range of 0 to 2, inclusive, are present.

colour_remap_matrix_present_flag equal to 0 indicates that the syntax elements **log2_matrix_denom** and **colour_remap_coeffs**[c][i], for c and i in the range of 0 to 2, inclusive, are not present.

log2_matrix_denom specifies the base 2 logarithm of the denominator for all matrix coefficients. The value of **log2_matrix_denom** shall be in the range of 0 to 15, inclusive. When not present, the value of **log2_matrix_denom** is inferred to be equal to 0.

colour_remap_coeffs[c][i] specifies the value of the three-by-three colour remapping matrix coefficients. The value of **colour_remap_coeffs**[c][i] shall be in the range of -2^{15} to $2^{15} - 1$, inclusive. When **colour_remap_coeffs**[c][i] is not present, it is inferred to be equal to 1 if c is equal to i, and inferred to be equal to 0 otherwise.

NOTE 3: When colour_remap_matrix_present_flag is equal to 0, the colour remapping matrix is inferred to be equal to the identity matrix of size 3x3.

The variable *matrixOutput[c]* for c = 0, 1 and 2 is derived as follows:

roundingOffset = **log2_matrix_denom** = = 0 ? 0: 1 << (**log2_matrix_denom** - 1)
matrixOutput[c] = *Clip3*(0, (1 << **colour_remap_output_bit_depth**) - 1,
(**colour_remap_coeffs**[c][0] × *matrixInput[0]* + **colour_remap_coeffs**[c][1] × *matrixInput[1]*
+ **colour_remap_coeffs**[c][2] × *matrixInput[2]* + *roundingOffset*) >> **log2_matrix_denom**)

where *matrixInput[c]* is the input sample value of the c-th colour component, and *matrixOutput[c]* is the output sample value of the c-th colour component.

post_lut_num_val_minus1[c] has the same semantics as **pre_lut_num_val_minus1**[c], with "pre" replaced by "post", except that the default end points of the input values are 0 and $2^{\text{colour\_remap\_output\_bit\_depth}} - 1$ for the c-th colour component. The value of **post_lut_num_val_minus1**[c] shall be in the range of 0 to 32, inclusive.

post_lut_coded_value[c][i] has the same semantics as **pre_lut_coded_value**[c][i], with "pre" replaced by "post", except that the number of bits used to represent **post_lut_coded_value**[c][i] is
((**colour_remap_output_bit_depth** + 7) >> 3) << 3.

post_lut_target_value[c][i] has the same semantics as **pre_lut_target_value**[c][i], with "pre" replaced by "post", except that **colour_remap_input_bit_depth** is replaced by **colour_remap_output_bit_depth** in the semantics.

B.5.2.3 Colour remapping information SEI message wrapping in AVC user data unregistered SEI message

This clause describes how the colour remapping information SEI message shall be wrapped in an AVC user data unregistered SEI message.

The colour remapping information SEI message shall be wrapped in an AVC user data unregistered SEI message as specified in clauses D.1.6 and D.2.6 of AVC specification [3].

- **uuid_iso_iec_11578** value shall be equal to 391949a4-1142-11e6-b932-0050c2490048.
- **user_data_payload_byte** shall be populated by data syntax such as specified in clause B.5.2.1.

B.5.3 Luminance mapping and colour correction tables variables mapping

Syntax elements values of the colour volume reconstruction information shall be mapped to hdr_reconstruction_info( ), luminance_mapping_table( ) and colour_correction_table( ) variables as specified in Table B.9.

Table B.9: Mapping from CRI SEI syntax elements to luminance_mapping_table( ) and colour_correction_table( ) variables

Variables	CRI SEI syntax element value
luminanceMappingNumVal	**post_lut_num_val_minus1**[0] + 1
luminanceMappingX[i]	**post_lut_coded_value**[0][i] ÷ 16 383
luminanceMappingY[i]	**post_lut_target_value[** 0][i] ÷ 16 383
colourCorrectionNumVal	**post_lut_num_val_minus1**[1] + 1
colourCorrectionX[i]	**post_lut_coded_value**[1][i] ÷ 2 047
colourCorrectionY[i]	**post_lut_target_value**[1][i] ÷ 2 047
chromaToLumaInjectionMuA	**colour_remap_coeffs**[0][1] ÷ 16 383
chromaToLumaInjectionMuB	**colour_remap_coeffs**[0][2] ÷ 16 383

In the colour_remapping_info (CRI SEI message), the following restrictions shall apply on the bitstream:

- **colour_remap_id** shall be greater or equal to 512.
- When present, **colour_remap_primaries** shall be equal to 1 or 9.
- When present, **colour_remap_matrix_coefficients** shall be equal to 1 or 9.
- **colour_remap_input_bit_depth** shall be equal to 14.
- **colour_remap_output_bit_depth** shall be equal to 14.
- **pre_lut_num_val_minus1**[2] shall be equal to 0.
- **colour_remap_matrix_present_flag** shall be equal to 1.
- **log2_matrix_denom** shall be equal to 0.
- **post_lut_coded_value**[0][i] shall be in the bounded range [0 to 16 383].
- **post_lut_target_value**[0][i] shall be in the bounded range [0 to 8 192].
- **post_lut_coded_value**[1][i] shall be in the bounded range [0 to 2 047].
- **post_lut_target_value**[1][i] shall be in the bounded range [0 to 2 047].
- **colour_remap_coeffs**[0][1] shall be in the bounded range [0 to 8 191].
- **colour_remap_coeffs**[0][2] shall be in the bounded range [0 to 8 191].
- **colour_remap_coeffs**[0][0];
  colour_remap_coeffs[1][1];
  colour_remap_coeffs[2][2] shall be equal to 1.
- **colour_remap_coeffs**[1][0];
  colour_remap_coeffs[2][0];
  colour_remap_coeffs[2][1];
  colour_remap_coeffs[1][2] shall be equal to 0.
- **pre_lut_num_val_minus1**[c] shall be equal to 0 for c = 0, 1, 2.

Annex C (informative): HDR-to-SDR decomposition principle

C.1 Introduction

C.1.1 Process overview

The HDR-to-SDR decomposition process aims at converting the input linear-light 4:4:4 HDR, to an SDR compatible version (also in 4:4:4 format). The process also uses side information such as the mastering display peak luminance, colour primaries, and the colour gamut of the container of the HDR and SDR pictures. In this version of the specification, the HDR-to-SDR conversion operates without changes of the colour gamut or space. The HDR and SDR pictures are defined in the same colour gamut or space. When this is not true, a preliminary gamut mapping process may be applied to convert the HDR picture from its native colour gamut or space to the target SDR colour gamut or space.

The HDR-to-SDR decomposition process generates an SDR backward compatible version from the input HDR signal, using an invertible process that guarantees a high quality reconstructed HDR signal.

The process is summarized in Figure C.1. First, from the input HDR picture and its characteristics, mapping variables are derived. Then the luminance signal is mapped to an SDR luma signal using the luminance mapping variables. Then a mapping of the colour to derive the chroma components of the SDR signal is applied. This step results in a gamut shifting, which is corrected by a final step of colour gamut correction.

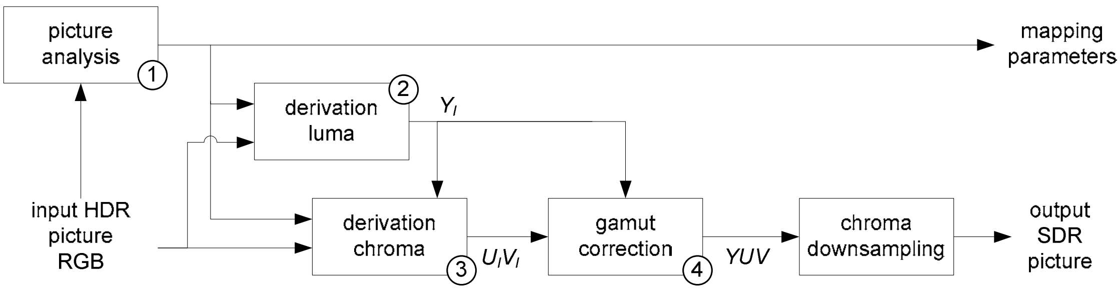

Figure C.1: synoptic of HDR-to-SDR decomposition process.

Clause C.1.2 provides more details on the steps 2 to 4. Clause C.1.3 describes the actual implementation of the process. The first step (picture analysis) that derives the mapping variables is described in clause C.3.

C.1.2 Theoretical decomposition process

Once the mapping parameters have been derived, as described in clause C.3, a luminance mapping function, noted $LUT_{TM,}$ is obtained. The next steps can be summarized as follows.

First, the luma signal is derived from the HDR linear-light RGB signal and from the luminance mapping function:

- derivation of linear-light luminance L from linear-light RGB signal:

$$L = A_1 \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{(C.1)}$$

with $A = [A_1\ A_2\ A_3]^T$ being the conventional 3x3 R'G'B'-to-Y'CbCr conversion matrix (e.g. Recommendation ITU-R BT.2020-2 [10] or Recommendation ITU-R BT.709-6 [9] depending on the colour space), A_1, A_2, A_3 being 1x3 matrices

- Then the linear-light luminance L is mapped to an SDR-like luma Y_{tmp}, using the luminance mapping function:

$$Y_{tmp} = (LUT_{TM}(L))^{\frac{1}{2,4}} \tag{C.2}$$

In the next step, the chroma components are built as follows:

- A mapping of R,G,B is applied with a ratio (Y_{tmp}/L)

$$\begin{bmatrix} R_N \\ G_N \\ B_N \end{bmatrix} = \frac{Y_{tmp}}{L} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{C.3}$$

  This generates the linear-light SDR version of RGB that will be used to derive the chroma components.

- A pseudo-gammatization using square-root (close to Recommendation ITU-R BT.709-6 OETF [9]) is applied:

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \begin{bmatrix} \sqrt{R_N} \\ \sqrt{G_N} \\ \sqrt{B_N} \end{bmatrix} = \sqrt{\frac{Y_{tmp}}{L}} \times \begin{bmatrix} \sqrt{R} \\ \sqrt{G} \\ \sqrt{B} \end{bmatrix} \tag{C.4}$$

  This generates the gammatized SDR version of RGB.

- The resulting R,G,B signal is converted to CbCr as follows:

$$\begin{bmatrix} U_{tmp} \\ V_{tmp} \end{bmatrix} = \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \sqrt{\frac{Y_{tmp}}{L}} \times \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} \sqrt{R} \\ \sqrt{G} \\ \sqrt{B} \end{bmatrix} \tag{C.5}$$

  where $A2, A3$ are made of the second and third lines of coefficients of the conversion matric from R'G'B'-to-Y'CbCr.

In the final step, a colour correction is applied as follows:

- First the chroma components are rescaled to correct the hue shift that results from the previous mapping step:

$$\begin{bmatrix} U_{sdr} \\ V_{sdr} \end{bmatrix} = \frac{1}{\beta(Y_{tmp})} \times \begin{bmatrix} U_{tmp} \\ V_{tmp} \end{bmatrix} = \frac{1}{\beta(Y_{tmp})} \times \sqrt{\frac{Y_{tmp}}{L}} \times \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} \sqrt{R} \\ \sqrt{G} \\ \sqrt{B} \end{bmatrix} \tag{C.6}$$

- Then a final adjustment of the luma component applies:

$$Y_{sdr} = Y_{tmp} - Max\,(0\,;\ muA \times U_{sdr} + muB \times V_{sdr}) \tag{C.7}$$

  with $\beta(Y_{tmp})$ a 1D function (or look-up table) depending on Y_{tmp}.

C.1.3 Reference implementation

The previous clause described the theoretical basics of the invertible HDR-to-SDR decomposition process that generates an SDR version from the input HDR content. In this clause, a reference implementation of the process is provided. Its inverse process corresponds to what is described in clause 7.2.4.

In the implementation of the HDR-to-SDR decomposition, some operations are concatenated into the look-up table used in the colour correction. The optimization is actually done directly for this look-up table. Then this "optimized" look-up table is used to derive the corresponding look-up table for the HDR reconstruction process (presented in clause 7.2.4).

The successive steps of the reference implementation are supplied by the following equations:

$$L = A_1 \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{C.8}$$

$$Y_{pre0} = (maxSampleVal - 1) \times\ (LUT_{TM}(L))^{\frac{1}{2,4}} \tag{C.9}$$

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \begin{bmatrix} \sqrt{R} \\ \sqrt{G} \\ \sqrt{B} \end{bmatrix} \quad (C.10)$$

$$\begin{bmatrix} U_{pre0} \\ V_{pre0} \end{bmatrix} = \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} \times maxSampleVal \quad (C.11)$$

$$\begin{bmatrix} U_{pre1} \\ V_{pre1} \end{bmatrix} = \frac{1}{\beta_0(Y_{pre0})} \times \begin{bmatrix} U_{pre0} \\ V_{pre0} \end{bmatrix} = \frac{maxSampleVal}{\beta_0(Y_{pre0})} \times \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} \sqrt{R} \\ \sqrt{G} \\ \sqrt{B} \end{bmatrix} \quad (C.12)$$

$$\begin{bmatrix} U_{pre1} \\ V_{pre1} \end{bmatrix} = \begin{bmatrix} Clip3(-midSampleVal, midSampleVal - 1, U_{pre1}) \\ Clip3(-midSampleVal, midSampleVal - 1, V_{pre1}) \end{bmatrix} \quad (C.13)$$

$$Y_{pre1} = Y_{pre0} - Max\ (0\ ;\ muA \times U_{pre1} + muB \times V_{pre1}) \quad (C.14)$$

$$\begin{bmatrix} U_{pre1} \\ V_{pre1} \end{bmatrix} = \begin{bmatrix} U_{pre1} + midSampleVal \\ V_{pre1} + midSampleVal \end{bmatrix} \quad (C.15)$$

In the last step, the signal is down-sampled to 4:2:0 format then converted from Full-to-Narrow Range, to generate the output SDR picture.

C.2 Mapping and colour functions derivation

C.2.1 Introduction

The mapping variables that are used to perform the HDR-to-SDR decomposition are derived in a first step that can be automatically, or manually driven. The variables are signalled in order, at the decoder side, to properly reconstruct, when required, the HDR signal from the SDR signal. Two modes of conveying those metadata are supported. In payload mode 0, a limited set of mapping variables are used to model the luminance mapping function (or look-up-table). These variables also enable to derive the colour correction function (or look-up-table). In payload mode 1, both functions are explicitly signalled using piece-wise linear functions. This gives more degrees of control of the mapping and colour correction functions, and therefore more flexibility to control the HDR-to-SDR decomposition. On the other hand, this requires an additional payload cost for those metadata.

The next clauses describe specifically the way the mapping function and colour correction functions are built for payload mode 0. Clause C.2.2 focuses on the luminance mapping function derivation. Clause C.2.3 relates to the colour correction derivation.

C.2.2 Computation of the function $LUT_{TM}(L)$ (payloadMode 0)

C.2.2.1 Overview of the computation of $\mathrm{LUT_{TM}}(L)$

The function $LUT_{TM}(L)$ performs the tone mapping. The tone mapping process is shown in Figure C.2.

The input signal L is first converted to the perceptually-uniform domain based on the mastering display maximum luminance, represented by **hdrMasterDisplayMaxLuminance**. In this domain, after black and white stretching, it is processed by the tone mapping curve, which in itself is controlled by the **shadowGain**, **highlightGain** and **midToneWidthAdjFactor**. Next the Tone Mapping Output Signal Offset Function is applied, which output is then gain limited. The gain limited signal is converted back to the linear-light domain based on the maximum luminance of the targeted system display maximum luminance, which is SDR, so 100 cd/m^2, yielding the output $LUT_{TM}(L)$.

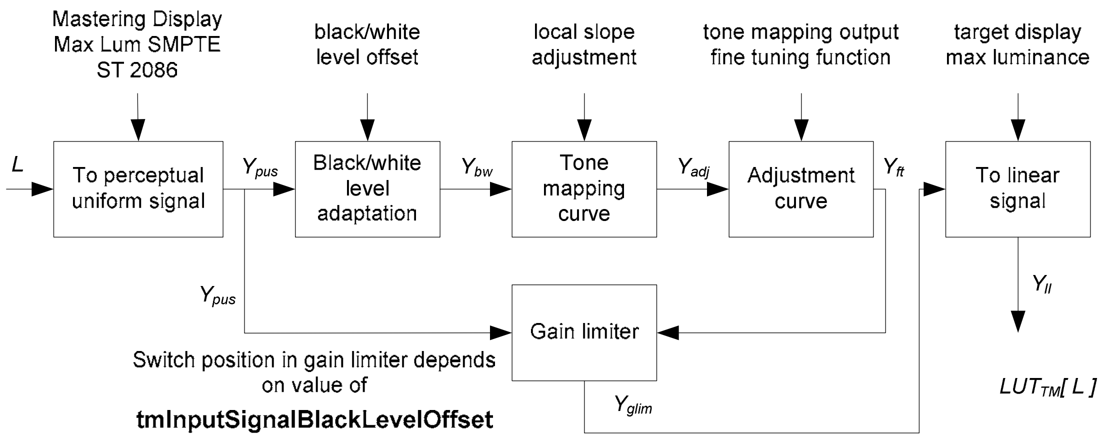

Figure C.2: Tone mapping process

The blocks shown in Figure C.2 are described in detail in clauses C.2.2.2 to C.2.2.8.

C.2.2.2 Block "To perceptual uniform signal"

The purpose of this block is to transform the linear-light input signal L, which is normalized to 0..1, where 1 corresponds to the peak luminance, to output signal Y_{pus} in the perceptually-uniform domain. It has the Mastering Display Maximum Luminance, L_{HDR}, as variable, in this version of the specification.

Y_{pus} is calculated as specified by equations (C.16) and (C.17).

$$Y_{pus} = v(L, L_{HDR}) \tag{C.16}$$

$$v(x, L) = \frac{\log_{10}\left(1 + (\rho(L) - 1) \times x^{\frac{1}{2,4}}\right)}{\log_{10}(\rho(L))} \tag{C.17}$$

The value for $\rho(L_{HDR})$ can be calculated using equation (C.18). L_{HDR} represents the peak luminance given by the Mastering Display Maximum Luminance, and is stored in the variable **hdrMasterDisplayMaxLuminance** in the structure hdr_characteristics( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.3).

$$\rho(L) = 1 + (33 - 1) \times \left(\frac{L}{10000}\right)^{\frac{1}{2,4}} \tag{C.18}$$

Figure C.3 depicts an example for L_{HDR}, = 5 000 cd/m^2.

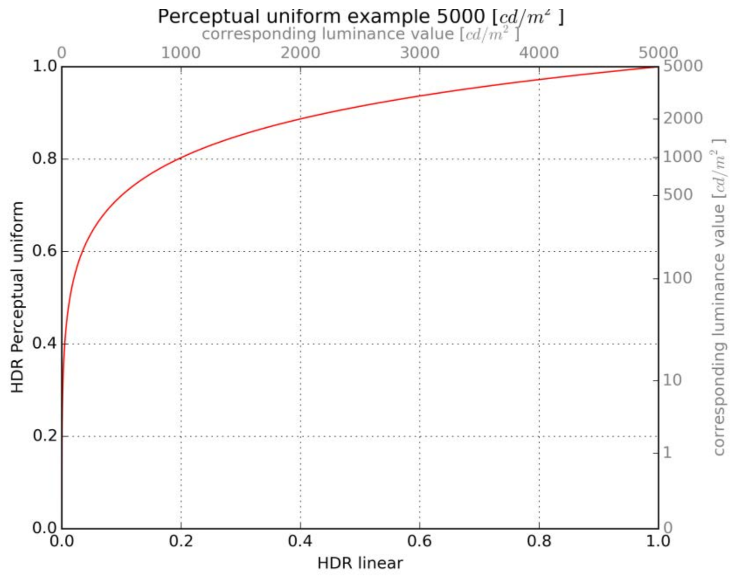

Figure C.3: Example curve $v(x, L_{HDR})$ for L_{HDR} = 5 000 cd/m^2

C.2.2.3 Block "Black/white level adaptation"

The purpose of this block is to adapt the input signal Y_{pus} by the black and white level offset to compute the output signal Y_{bw}.

The computations in this block are specified in equations (C.19), (C.20) and (C.21):

$$Y_{bw} = \frac{Y_{pus} - blo}{1 - wlo - blo} \quad \text{(C.19)}$$

$$wlo = \frac{255 \times \mathbf{tmInputSignalWhiteLevelOffset}}{510} \quad \text{(C.20)}$$

$$blo = \frac{255 \times \mathbf{tmInputSignalBlackLevelOffset}}{2040} \quad \text{(C.21)}$$

NOTE: Equation (C.19) is the inverse of equation (15).

The parameters **tmInputSignalBlackLevelOffset** and **tmInputSignalWhiteLevelOffset** are stored in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5.

Figure C.4 shows an example black white correction curve.

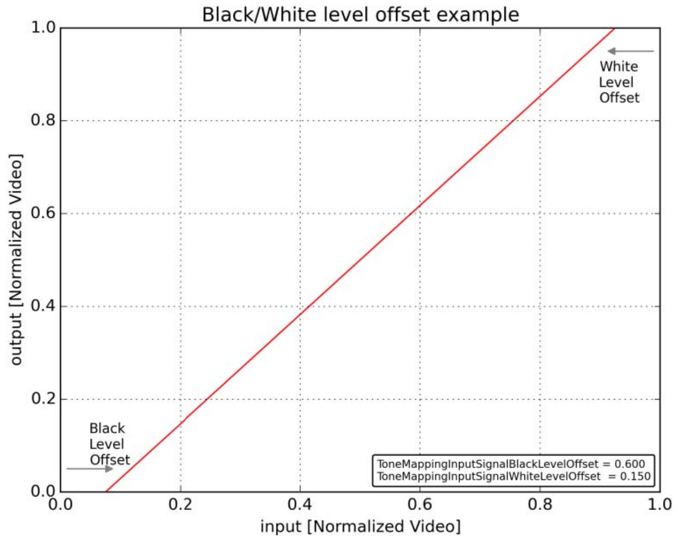

Figure C.4: Example curve for black and white level offset

C.2.2.4 Block "Tone mapping curve"

In this block, the Tone Mapping curve is applied on the input signal Y_{bw} to compute the output signal Y_{adj}, according to equation (C.22):

$$Y_{adj} = TMO(Y_{bw}) \qquad \text{(C.22)}$$

The basics of the curve are explained below and an example is shown in Figure C.5.

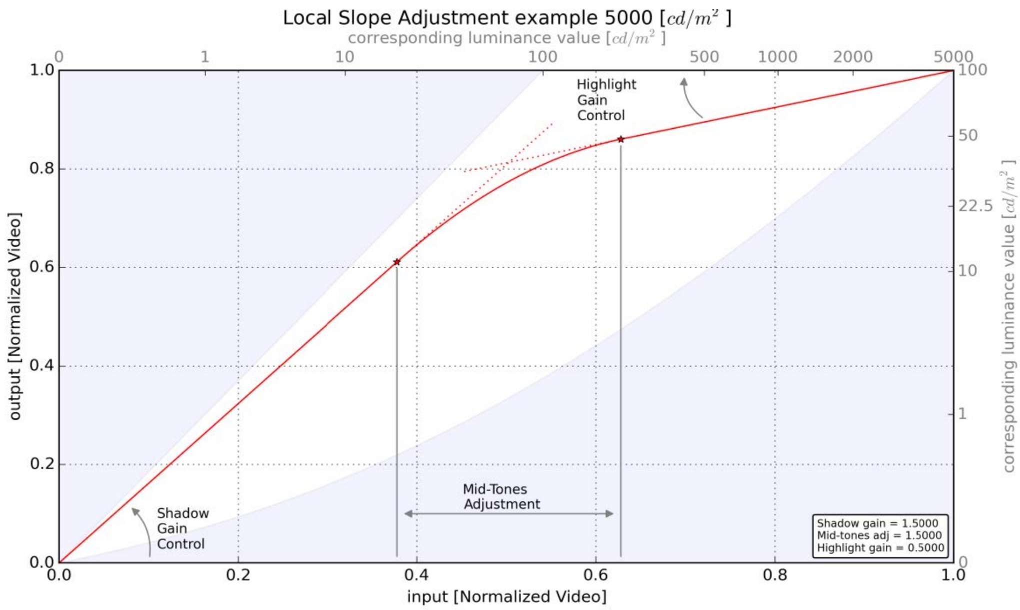

Figure C.5: Tone mapping curve shape example

The tone mapping curve is applied in a perceptually-uniform domain and is a piece-wise curve constructed out of three parts.

The bottom section is linear, and its steepness is determined by the **shadowGain**.
The upper section is also linear, and its steepness is determined by the **highlightGain**.

The mid-section is a parabola providing a smooth bridge between the two linear sections. The width of the cross-over is determined by the **midToneWidthAdjFactor**.

Equation (C.23) up to and including equation (C.32) are an overview of the calculations in order to arrive at the piece-wise constructed curve.

NOTE 1: These calculations are valid under the condition that $L_{HDR} > L_{target}$.

$$TMO(x) = \begin{cases} SGC \times x, & 0 \le x \le x_{SGC} \\ ax^2 + bx + c, & x_{SGC} < x < x_{HGC} \\ HGC \times x + 1 - HGC, & x_{HGC} \le x \le 1 \end{cases} \quad \text{(C.23)}$$

$$a = \begin{cases} 0, & para = 0 \\ -0{,}5 \times \frac{SGC-HGC}{para}, & otherwise \end{cases}$$
$$b = \begin{cases} 0, & para = 0 \\ \frac{1-HGC}{para} + \frac{SGC+HGC}{2}, & otherwise \end{cases} \quad \text{(C.24)}$$
$$c = \begin{cases} 0, & para = 0 \\ -\frac{\left((SGC-HGC)\times para - 2(1-HGC)\right)^2}{8\times(SGC-HGC)\times para}, & otherwise \end{cases}$$

$$x_{SGC} = \frac{1-HGC}{SGC-HGC} - \frac{para}{2}$$
$$x_{HGC} = \frac{1-HGC}{SGC-HGC} + \frac{para}{2} \quad \text{(C.25)}$$

$$exposure = \frac{\mathbf{shadowGain}}{4} + 0{,}5 \quad \text{(C.26)}$$

$$expgain = v\left(\frac{L_{HDR}}{L_{target}}, L_{target}\right) \quad \text{(C.27)}$$

$$L_{HDR} = \mathbf{hdrMasterDisplayMaxLuminance} \quad \text{(C.28)}$$

$$L_{target} = 100 \text{ cd}/m^2 \quad \text{(C.29)}$$

$$SGC = expgain \times exposure \quad \text{(C.30)}$$

$$HGC = \frac{\mathbf{highlightGain}}{4} \quad \text{(C.31)}$$

$$para = \frac{\mathbf{midToneWidthAdjFactor}}{2} \quad \text{(C.32)}$$

The value of **shadowGain**, **highlightGain** and **midToneWidthAdjFactor**, as used in equations (C.26), (C.31) and (C.32), are stored in the metadata structure luminance_mapping_variables( ) as specified in clause 6.2.5.

The value of **hdrMasterDisplayMaxLuminance**, as used in equation (C.28), is stored in the metadata structure hdr_characteristics( ) as specified in clause 6.2.3.

NOTE 2: It is the objective to create an SDR picture and therefore a value of $100\ cd/m^2$ is used for L_{target}.

C.2.2.5 Block "Adjustment curve"

In this block, the fine tuning curve is applied on the input signal Y_{adj} to compute the output signal Y_{ft}, according to equation (C.33).

$$Y_{ft} = \begin{cases} f_{ftlum}(Y_{adj}), & 0 \le Y_{adj} \le 1 \\ Y_{adj}, & otherwise \end{cases} \quad \text{(C.33)}$$

The *ToneMappingOutputFineTuningFunction* function f_{ftlum}( ), is a piecewise linear function; see clause 7.3 for the computation of f_{ftlum}( ) from the list of points.

The samples explicitly defining the *ToneMappingOutputFineTuningFunction* function are the pairs **tmOutputFineTuningX**[i], **tmOutputFineTuningY**[i], in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5, possibly extended with a point at the start and/or at the end, as specified in clause 6.3.4.9.

An example fine tuning curve is shown in Figure C.6

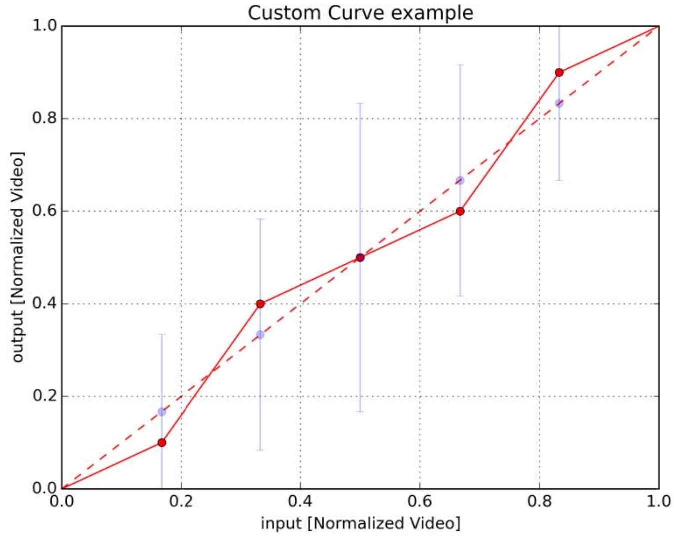

Figure C.6: Example fine-tuning curve

C.2.2.6 Block "Gain limiter"

The purpose of this block is to correct the input signal Y_{ft} for a gain that is too low, but only if the value of the variable **tmInputSignalBlackLevelOffset** is not equal to 0.

If the value of the variable **tmInputSignalBlackLevelOffset** is not equal to 0, the input signal Y_{ft} and the input signal Y_{pus} from clause C.2.2.2 are used to compute the output signal Y_{glim}, according to equations (C.34) and (C.35), based on the maximum display mastering luminance L_{HDR}, which is equal to the variable **hdrMasterDisplayMaxLuminance** that is stored in the structure hdr_characteristics( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.3).

$$Y_{glim} = Max\left(Y_{ft}\ ;\ Y_{pus} \times g\right) \tag{C.34}$$

$$g = v(0{,}1 \div L_{SDR}, L_{SDR}) \div v(1 \div L_{HDR}, L_{HDR}) \tag{C.35}$$

with the inverse EOTF, $v(x, y)$, taken from equation (C.17).

If the value of the variable **tmInputSignalBlackLevelOffset** is equal to 0, the output Y_{glim} of this block is equal to the input signal Y_{ft}.

C.2.2.7 Block "To linear signal"

The purpose of this block is to convert the input signal Y_{glim} from the perceptually uniform domain to the linear-light domain output signal Y_{ll}, using the EOTF $v_{inv}(x,y)$. It is based on the SDR max luminance of 100 cd/m^2, see equations (C.36), up to and including (C.38).

$$Y_{ll} = v_{inv}(Y_{glim}, 100) \tag{C.36}$$

$$v_{inv}(x, y) = \left(\frac{\rho(y)^{x} - 1}{\rho(y) - 1}\right)^{2{,}4} \tag{C.37}$$

NOTE: Equation (C.37) is the mathematical inverse function for equation (C.17).

$$\rho(y) = 1 + (33 - 1) \times \left(\frac{y}{10000}\right)^{\frac{1}{2{,}4}} \tag{C.38}$$

The curve $v_{inv}(Y, 100)$ is shown in Figure C.7.

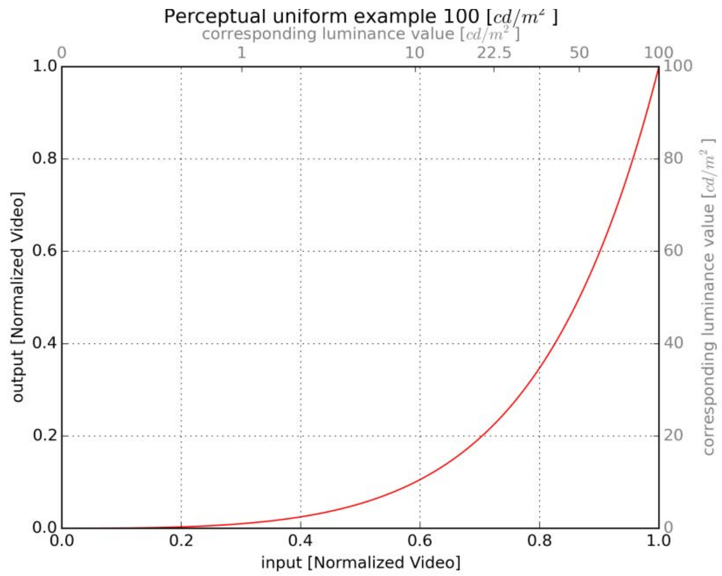

Figure C.7: Perceptual to linear curve based on 100 cd/m^2

C.2.2.8 Final output

The output $LUT_{TM}(L)$ is the linear-light value Y_{ll}.

C.2.3 Computation of the colour correction function

C.2.3.1 Introduction

Clause C.2.3.2 provides an exemplary process to derive the colour correction function (or look-up table) $\beta_0(\ \ )$, used in the decomposition process of clause C.1.3, is described. The process aims at generating for each luma value a beta value that gives an overall minimum colour shift error between HDR and SDR content in an CIELAB-like colour space, over the container gamut. Clause C.2.3.3 covers the colour shift error computation. The process proposed in clause C.2.3.2 can apply to both payload mode 0 and 1. A simplified process for colour correction function generation is provided in clause C.3.4.

C.2.3.2 Examplary process for the computation of the colour correction function

The process is independent from the content. It applies in the container colour gamut, takes into account the content colour gamut.

The synoptic of this process is summarized in Figure C.8.

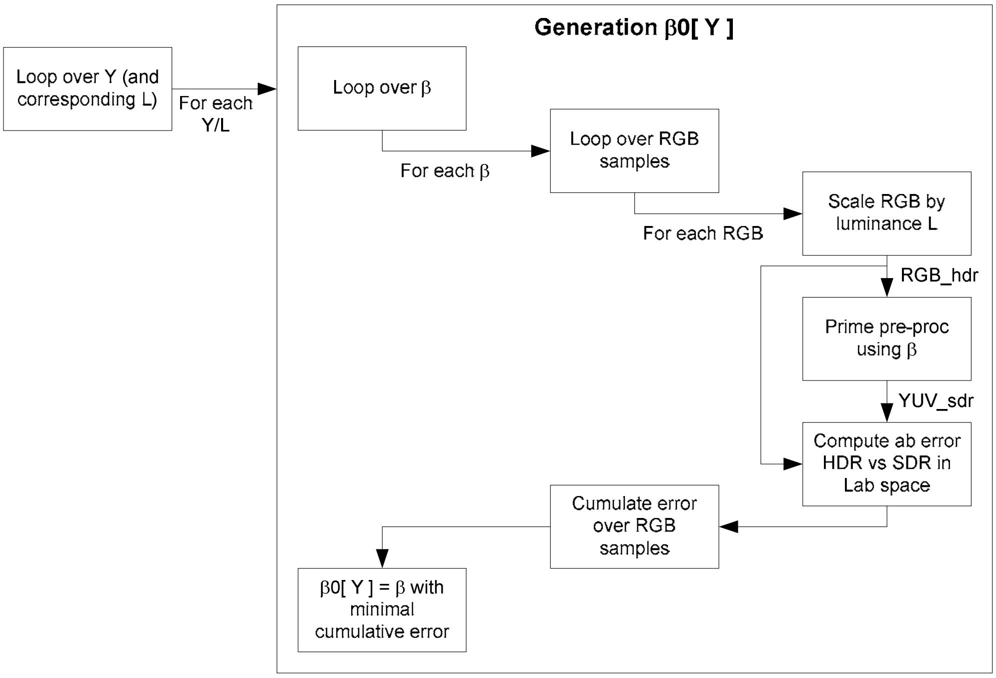

Figure C.8: Synoptic of LUT β_0 generation.

The process can be summarized as follows. For each luma value Y, the following steps are applied.

The luminance is generated using the inverse function of $(LUT_{TM}(L))^{\frac{1}{2,4}}$: $L = invLUT_{TM}[\ Y]$.

NOTE: $invLUT_{TM}[\ Y]$ corresponds to the square value of $lutMapY[\ Y]$.

Then the best $\beta_0[Y]$ for luminance L (and therefore for luma Y) is identified as follows.

- Sampling of beta values - For each beta value β_{test}
  - Set $error_{ab} = 0$: cumulative error for value β_{test} at luminance L computed on the content gamut (as specified by **hdrMasterDisplayColourSpace**), but considered in the container gamut (as specified by **hdrPicColourSpace**)
  - A scanning of the RGB cube is performed, and each RGB sample is modified to reach a luminance of 1 cd/m². Then the following applies.

$$\begin{cases} R = L \times R_{SDR} \\ G = L \times G_{SDR} \\ B = L \times B_{SDR} \end{cases} \quad \text{(C.39)}$$

    - If the sample RGB_{HDR} is in the container gamut (as specified by **hdrPicColourSpace**) (this test is achieved by checking if once converted to CIE xyY, the sample is inside the container triangle primaries), compute the output sample YUV_{SDR} as specified in clause C.1.3.
    - Compute $error$ in CIELAB-like space between RGB_{sdr} and RGB_{hdr} as specified in clause C.2.3.3.
    - Set $error_{ab} = error_{ab} + error$.
- $\beta_0[Y] = \beta_{test}$ giving the lowest $error_{ab}$ value

C.2.3.3 Derivation of the HDR-versus-SDR perceived colour error

This process takes as inputs:

- R, G, B values corresponding to a linear-light HDR signal;
- $Y_{sdr}, U_{sdr}, V_{sdr}$ values corresponding to an SDR converted version of the linear-light HDR signal;

- the SDR picture characteristics variable **sdrPicColourSpace**;
- the HDR picture characteristics variable **hdrPicColourSpace**.

The process generates as output:

- a value $errColour$ measuring a perceived colour difference between the HDR and SDR signal.

The process works as follows:

- Y_{sdr}, U_{sdr}, V_{sdr} are converted to R_{sdr}, G_{sdr}, B_{sdr} values corresponding to a linear-light HDR signal:

$$\begin{bmatrix} R_{sdr} \\ G_{sdr} \\ B_{sdr} \end{bmatrix} = M_{Y'CbCr-to-R'G'B'} \times \begin{bmatrix} Y_{sdr} \\ U_{sdr} \\ V_{sdr} \end{bmatrix} \quad (C.40)$$

  where $M_{Y'CbCr-to-R'G'B'}$ is the conversion matrix from Y'CbCr to R'G'B', as specified in Table 17, with variable $matrixIdx$ equal to **sdrPicColourSpace**.

- R_{sdr}, G_{sdr}, B_{sdr} are modified as follows:

$$\begin{cases} R_{sdr} = 100 \times R_{sdr}^2 \\ G_{sdr} = 100 \times G_{sdr}^2 \\ B_{sdr} = 100 \times B_{sdr}^2 \end{cases} \quad (C.41)$$

- For $idx = sdr$, hdr, the values a_{idx}, b_{idx} are derived as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M_{RGB-to-XYZ} \times \begin{bmatrix} R_{idx} \\ G_{idx} \\ B_{idx} \end{bmatrix} \quad (C.42)$$

  where $M_{RGB-to-XYZ}$ is the conversion matrix from RGB to XYZ, as specified in Table 16, with variable $matrixIdx$ equal to **sdrPicColourSpace** for $idx = sdr$ and $matrixIdx$ equal to **hdrPicColourSpace** for $idx = hdr$.

$$\begin{cases} a_{idx} = 500 \times \left(Flab\left(\frac{X}{Y}\right) - Flab(1) \right) \\ b_{idx} = 200 \times \left(Flab(1) - Flab\left(\frac{Z}{Y}\right) \right) \end{cases} \quad (C.43)$$

- Finally, $errColour$ is derived as follows:

$$errColour = (a_{hdr} - a_{sdr})^2 + (b_{hdr} - b_{sdr})^2 \quad (C.44)$$

C.3 Automatic parameter generation during encoding

C.3.1 Introduction

Clause C.3 describes one of the possible ways to calculate the tone mapping parameters used in clause C.2.2 "Computation of the function $LUT_{TM}(L)$ (**payloadMode** 0)". This can be used during e.g. live HDR transmissions.

Clause C.3.2 describes a possible calculation of these parameters from a single HDR picture.

Clause C.3.3 describes a possible temporal filtering of these parameters in a sequence of pictures.

Clause C.3.4 describes a possible simplified process for deriving the colour correction function.

C.3.2 Automatic tone mapping parameter generation from only an HDR picture

C.3.2.1 Introduction

Clause C.3.2 describes one of the possible ways to calculate the tone mapping parameters used in clause C.2.2 "Computation of the function $LUT_{TM}(L)$ (**payloadMode** 0)". This can be used during e.g. live HDR distribution.

The tone mapping curve in clause C.2.2 is variable, depending at least on the parameters **tmInputSignalBlackLevelOffset**, **tmInputSignalWhiteLevelOffset**, **shadowGain**, **highlightGain** and **midToneWidthAdjFactor**.

A way to calculate these parameters from an HDR picture is described in clauses C.3.2.2 to C.3.2.4. The constants used in these clauses are not necessarily the optimal ones and they can be subject to expert tuning.

C.3.2.2 Calculation of tmInputSignalBlackLevelOffset, tmInputSignalWhiteLevelOffset

This clause describes a way to calculate the parameters **tmInputSignalBlackLevelOffset**, **tmInputSignalWhiteLevelOffset** of the tone mapping curve of clause C.2.2.

First, let $luminancePeakSDR$ be the peak luminance of the SDR picture in the normalized linear-light domain, and let $luminancePeakHDR$ be the peak luminance in the HDR picture in the normalized linear-light domain. The variables $vMaxOut$ and $vMaxIn$ are the equivalent values of $luminancePeakSDR$ and $luminancePeakHDR$ converted to the perceptually uniform domain, see equations (C.45) and (C.46).

NOTE 1: The value of 1 is usually taken for $luminancePeakHDR$, which leads to a value of 1 for $vMaxIn$.

$$vMaxOut = v(luminancePeakSDR, L_{HDR}) \quad \text{(C.45)}$$

$$vMaxIn = v(luminancePeakHDR, L_{HDR}) \quad \text{(C.46)}$$

where the OETF v(;) is defined in equations (C.47) to (C.48), and where L_{HDR}, represents the Mastering Display Maximum Luminance used to master the HDR content in cd/m^2.

$$v(x, L) = \frac{\log_{10}\left(1 + (\rho(L) - 1) \times x^{\frac{1}{2,4}}\right)}{\log_{10}(\rho(L))} \quad \text{(C.47)}$$

$$\rho(L) = 1 + (33 - 1) \times \left(\frac{L}{10\,000}\right)^{\frac{1}{2,4}} \quad \text{(C.48)}$$

The value for the unclipped black stretch bsu can be taken as the 0,01 % percentile of the value Y of all pixels of the HDR picture, as defined in equation (C.49).

$$Y = v(L, L_{HDR}) \quad \text{(C.49)}$$

where L represents the pixel luminance value in the normalized linear domain, and where the OETF v(;) is defined in equations (C.47) and (C.48), and where L_{HDR}, represents the Mastering Display Maximum Luminance in cd/m2.

The value of the clipped black stretch bs can be calculated as defined in equation (C.50).

$$bs = Max(0\,;\, Min(bsu\,;\, 0{,}1)) \quad \text{(C.50)}$$

The value for the unclipped white stretch wsu can be taken as the 99,999 % percentile of the value V of all pixels of the HDR picture, as defined in equation (C.51).

$$V = v(Max(Max(R\,;G)\,;B), L_{HDR}) \quad \text{(C.51)}$$

where R, G and B are the normalized linear-light values R, G, B per pixel and where the OETF v(;) is defined in equations (C.47) and (C.48), and where L_{HDR}, represents the Mastering Display Maximum Luminance in cd/m^2.

The value of the clipped white stretch ws can be calculated as defined in equation (C.52).

$$ws = Max(vMaxOut\ ;\ Min(wsu\ ;\ vMaxIn)) \quad \text{(C.52)}$$

The black level is stretched for only 60 % and the white level is stretched for only for 80 %, yielding the variables bl and wh, see equations (C.53) and (C.54).

$$bl = 0{,}6 \times bs \quad \text{(C.53)}$$

$$wh = 0{,}8 \times ws + 0{,}2 \times vMaxIn \quad \text{(C.54)}$$

The parameters **tmInputSignalBlackLevelOffset**, and **tmInputSignalWhiteLevelOffset**, can be derived according to equations (C.55) and (C.56).

$$\mathbf{tmInputSignalBlackLevelOffset} = bl \div vMaxIn \quad \text{(C.55)}$$

$$\mathbf{tmInputSignalWhiteLevelOffset} = 1 - wh \div vMaxIn \quad \text{(C.56)}$$

NOTE 2: The value of 1 is usually taken for $luminancePeakHDR$, which leads to a value of 1 for $vMaxIn$.

C.3.2.3 Calculation of shadowGain

This clause describes a way to calculate the parameter **shadowGain** of the tone mapping curve of clause C.2.2.

Let $LightnessHDR$ be the average value of V from equation (C.51) over all pixels of a picture.

NOTE 1: $LightnessHDR$ can be indirectly computed from a Max(Max(R ; G) ; B) histogram.

The variable bg can be computed using equations (C.57) to (C.58).

$$bg = Min\left(nomGain \times \mathrm{Max}\left(1\ ;\left(2 - \frac{LightnessHDR}{LightnessHDRHigh}\right) \div bwGain\right)\ ;\ 1\right) \quad \text{(C.57)}$$

$$nomGain = vMaxOut \div vMaxIn \quad \text{(C.58)}$$

where $LightnessHDRHigh$ is the highest accepted value (e.g. the value above which mainly highlights occur) of the luminance of the HDR picture converted to the perceptually uniform domain, $bwGain$ is defined in equation (C.59) and where $vMaxOut$ and $vMaxIn$ are defined in equations (C.45) and (C.46).

$$bwGain = vMaxIn \div wh \quad \text{(C.59)}$$

where bl and wh are defined in equations (C.53) and (C.54).

NOTE 2: If the value of 1 is taken for $luminancePeakHDR$, $nomGain$ will become equal to $vMaxOut$.

NOTE 3: The variable bg is limited between $nomGain$ and 1. This means that variable bg reaches the minimum value for e.g. $LightnessHDR = 0{,}7$, which corresponds to an average luminance of 700 cd/m² when $luminancePeakHDR$ = 10 000 cd/m². For scenes with an average luminance of 700 cd/m² or more, the darkest tone mapping is applied. This is necessary for protecting white areas during MPEG compression of the SDR.

Finally, the parameter **shadowGain** can be computed according to equation (C.60).

$$\mathbf{shadowGain} = 4 \times (bg - 0{,}5) \quad \text{(C.60)}$$

NOTE 4: A value of 1 for bg, so 2 for **shadowGain** will lead to cd/m² out equals cd/m² in and may be used for dark scenes.

C.3.2.4 The parameters highlightGain and midToneWidthAdjFactor

This clause describes a way to calculate the parameters **highlightGain** and **midToneWidthAdjFactor** of the tone mapping curve of clause C.2.2.

The parameter **highlightGain** can be automatically computed according to equations (C.61) to (C.62).

$$dg = Max\big(0{,}25 \times nomGain\,; Min(0{,}375 - 0{,}25 \times bg\,; 0{,}5 \times nomGain)\big) \quad (C.61)$$

$$\mathbf{highlightGain} = \frac{4 \times dg}{nomGain} \quad (C.62)$$

where dg is the differential gain, $nomGain$ is from equation (C.58) and is from equation (C.57).

The parameter **midToneWidthAdjFactor** can be computed according to equations (C.63) to (C.67).

$$xp1 = Max(0{,}2\,; Min(1{,}12 - bg\,; 0{,}5)) \quad (C.63)$$

$$xm = (vMaxOut - dg \times vMaxIn) \div Max(10^{-8}; bg - dg) \quad (C.64)$$

$$xp2 = Min\big(2 \times \mathrm{xm}\,; 2 \times (vMaxIn - \mathrm{xm})\big) \quad (C.65)$$

$$xp = Min(xp1\,; xp2) \quad (C.66)$$

$$\mathbf{midToneWidthAdjFactor} = 2 \times xp \div vMaxIn \quad (C.67)$$

where bg is taken from equation (C.57) and dg is taken from equation (C.61).

NOTE: Equations (C.65) and (C.66) automatically reduce xp so that half of the mid-tones parabola can never be wider than each half of the tone mapping curve: the left half with shadow gain running from 0 to xm, and the right half with the highlights differential gain (which is 0,5 running from xm to $vMaxIn$).

C.3.3 Temporal filtering of tone mapping parameters

The parameters automatically generated in real-time from a live video stream should be temporally low-pass filtered to reduce unrest and to avoid unnecessary responses to short-term changes.

An example temporal filter is shown in equations (C.68) to (C.70).

$$e = Y[i] - YF[i-1] \quad (C.68)$$

$$d = F(e) \quad (C.69)$$

$$Y[i] = d + YF[i-1] \quad (C.70)$$

where $Y[i]$ is the raw parameter computed from picture i, $YF[i]$ is the temporally filtered parameter for picture i, i is the index of the current picture, i - 1 is the index of the previous picture and $F(e)$ is a function that is explained below.

The speed of the temporal filter is determined by the transfer of the function $F(e)$. A deliberate asymmetry in the response speed is introduced through a transfer that is different for negative and positive values of the input e, see Figure C.9.

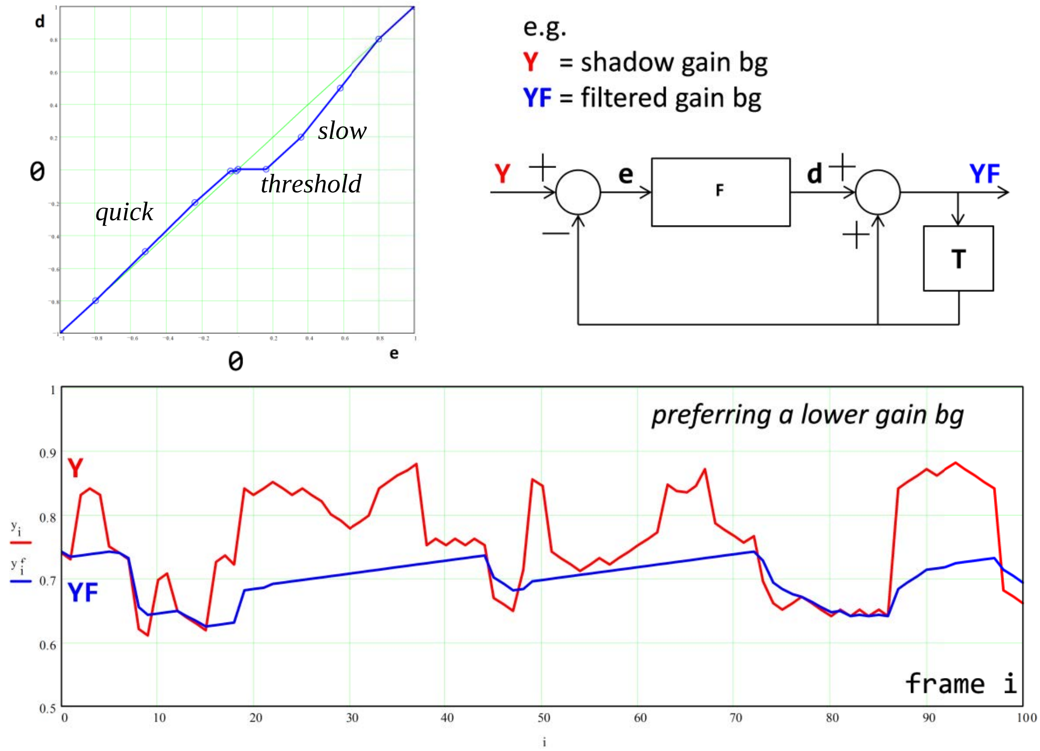

Figure C.9: Example temporal filter for the shadow gain parameter

In case the transfer of the function $F(e)$ in Figure C.9 is taken as $d = e$, the green line, then the output YF follows the input Y immediately and there is no temporal filtering.

The more the transfer of the function $F(e)$ moves to the right of d = e, the green line, and the closer to zero the value d>0 is, for positive input e, the slower YF follows Y for positive changes (increases of the raw parameter). Similarly, the more the transfer of the function $F(e)$ moves to the left of $d = e$, and the closer to zero the value d<0 is, the green line, for negative input e, the slower YF follows Y for negative changes (decreases of the raw parameter).

The knee marked "threshold" in the transfer of the function $F(e)$ in Figure C.9 marks the value of the positive difference e between the raw parameter values of two pictures above which the temporal filter starts responding quickly. The value of $F(e)$ for the horizontal part of the transfer of the function $F(e)$, the part between $e = 0$ and the knee marked "threshold" determines the slope of the temporally filtered output when the temporal filter responds slowly. This horizontal part in Figure C.9 has a very small but non-zero value for d.

In the case of Figure C.9, the filter responds quickly to a decreasing shadow gain parameter **shadowGain** (increasing lightness), and more slowly to an increasing shadow gain (decreasing lightness). It is a low-peak detector. This asymmetric variant of the temporal filter should be used when a lower value of the tone mapping parameter is deemed safer. A parameter change in the "unsafe" direction (increasing lightness, decreasing gain) is deemed critical, and should mostly be followed quickly. A large change in the "safe" direction (decreasing lightness, increasing gain) is treated as a scene change, and should be followed more quickly.

For a practical temporal filter for the parameter **shadowGain**, the threshold can be taken as e = 0,1 for the positive part and e = - 0,1 for the negative part of the transfer of the function $F(e)$ in the temporal filter. This means that the threshold for responding quickly instead of slowly is the same for increases and decreases in the input. The value for the transfer of the function $F(e)$ for 0 < e ≤ 0,1 can be taken as 0,002 and for - 0,1 ≤ e < 0, the value can be taken as - 0,05, which means that the temporal filter reacts more slowly for input changes between the thresholds of $F(e)$ for increasing input than for decreasing input.

The same filter can be applied to the calculated black level **tmInputSignalBlackLevelOffset**, responding quickly to a decreasing black level, and vice versa. For the calculated white level **tmInputSignalWhiteLevelOffset**, the filter should respond in the opposite way: quickly to an increasing white level, and slowly to a decreasing white level.

These two responses are illustrated together in Figure C.10.

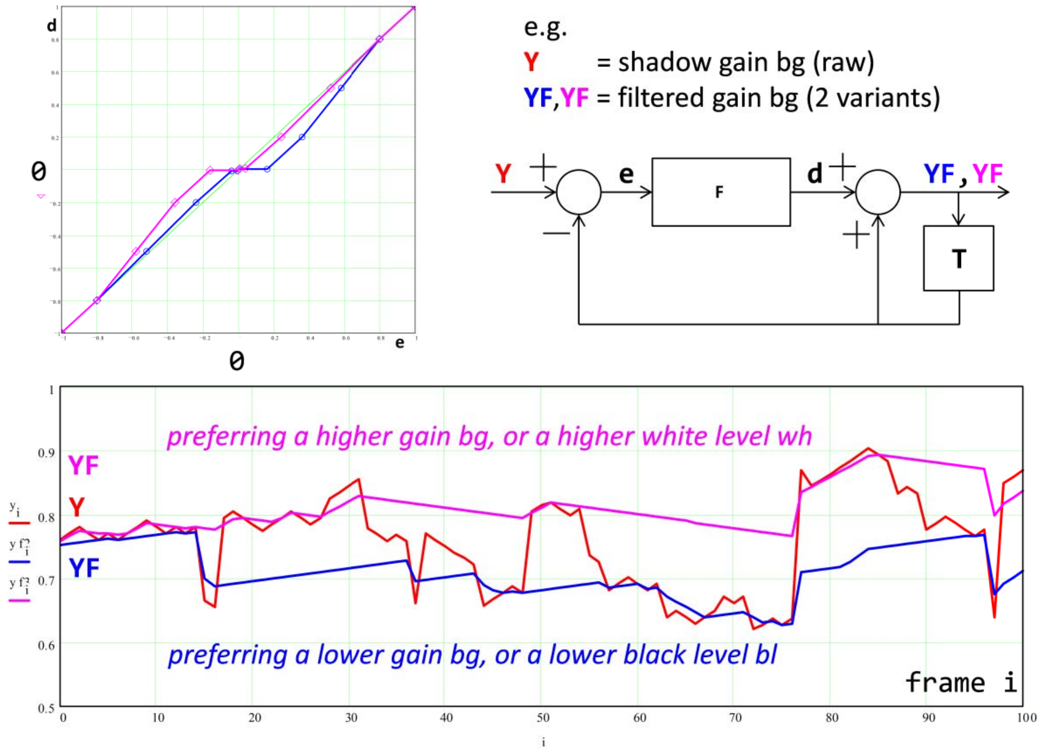

Figure C.10: Example temporal filters for the shadow gain and white level parameters

Three separate temporal filters can be used for the parameters **tmInputSignalBlackLevelOffset**, **tmInputSignalWhiteLevelOffset**, and **shadowGain**.

From the calculation of clause C.3.2.3, equations (C.57) and (C.59), one can see that the value of the (unfiltered) parameter **shadowGain** is dependent on the value of *wh*, from which the (unfiltered) parameter **tmInputSignalWhiteLevelOffset** is derived. If the parameters **tmInputSignalWhiteLevelOffset**, and **shadowGain** are filtered independently, the dependency of **shadowGain** on **tmInputSignalWhiteLevelOffset** may cause flicker. Therefore, it is better to perform a temporal filter on the variable $bgUf$ instead of on **shadowGain**, and perform a temporal filter on the variable *wh* instead of on **tmInputSignalWhiteLevelOffset**, see equations (C.71) to (C.78).

$$bgUf = nomGain \times \left(2 - \frac{LightnessHDR}{LightnessHDRHigh}\right) \quad (C.71)$$

$$bgUfCl = Min(Max(nomGain\,; bgUf)\,;\, 1) \quad (C.72)$$

$$whTf = \text{temporal\_filter}(wh) \quad (C.73)$$

$$bwGainTf = vMaxIn \div whTf \quad (C.74)$$

$$bgTf = \text{temporal\_filter}(bgUfCl) \quad (C.75)$$

$$bgTfM = \frac{bgTf}{bwGainTf} \quad (C.76)$$

$$bgTfCl = Min(Max(nomGain\,; bgTfM)\,;\, 1) \quad (C.77)$$

$$whTfCl = \frac{bgTfCl}{bgTf} \times vMaxIn \quad (C.78)$$

where $bgTf$ is the temporally filtered $bgUf$, $bgTfCl$ is the clipped and temporally filtered $bgUf$, $bwGainTf$ is the $bwGain$ derived from filtered parameters, $whTf$ is the temporally filtered parameter *wh* from clause C.3.2.2 and $whTfCl$ is the clipped and temporally filtered parameter *wh* from clause C.3.2.2.

The temporally filtered parameter **shadowGain** , **shadowGainTf**, can be computed according to equation (C.79).

$$\mathbf{shadowGainTf} = 4 \times (bgTfCl - 0{,}5) \quad (C.79)$$

The temporally filtered parameter **tmInputSignalWhiteLevelOffset** , **tmInputSignalWhiteLevelOffsetTf**, can be computed according to equation (C.80).

$$\mathbf{tmInputSignalWhiteLevelOffsetTf} = 1 - whTfCl \div vMaxIn \qquad \text{(C.80)}$$

The other parameters **highlightGain** and **midToneWidthAdjFactor** do not have to be filtered separately. Instead, the temporally filtered parameter **highlightGain** can be computed according to equations (C.61) to (C.62) and the temporally filtered parameter **midToneWidthAdjFactor** can be derived from equations (C.63) to (C.67), where $bgTfCl$ is taken for the value of bg in equations (C.61) and (C.67).

C.3.4 Simplified process for colour correction function generation

C.3.4.1 Introduction

This clause provides a simplified process for generating the colour correction function. It can be used in replacement of the process described in section C.2.3.2 typically for live environment implementation purposes. The process is based on the usage of a limited set of pre-defined colour correction LUTs that are used to derive the colour correction LUT used in the pre-processing.

Clause C.3.4.2 describes the pre-processing stage based on the simplified derivation of the colour correction LUT.

Clause C.3.4.3 describes the selection process of the colour correction LUT among different pre-defined LUTs.

C.3.4.2 Simplified colour correction derivation process

In practice, the colour correction function applied in the pre-processing, β_0 (cf section C.2.3.2), is implemented as a 1D LUT that is actually both linked to the colour correction LUT $lutCC$ applied during the post-processing and to the luminance mapping LUT $lutMapY$ by the following equation:

$$\beta_O(Y) = (maxSampleVal + 1) \times lutMapY[Y] \times lutCC[Y] \qquad \text{(C.81)}$$

$lutCC$ is mostly independent from the luminance mapping LUT $lutMapY$ and is mainly dependent on colour properties of the content, such as the saturation and hue. It is therefore possible to define a limited set of n pre-defined LUTs $lutCC_{ref\_i}$, for i=1..n, corresponding for instance to different categories of content, discriminated by their saturation and hue characteristics.

As illustrated in Figure C.11, the derivation of the colour correction LUT in the pre-processing stage consists of:

- identifying from the content properties the relevant LUT $lutCC_{ref\_k}$,
- computing the saturation gain function $f_{sgf}(\ )$ that enables the generation of $lutCC_{ref\_k}$ from the default colour correction LUT specified in Table 13,
- building the colour correction LUT $lutCC$ used in the post-processing stage,
- building the colour correction LUT β_O used in the pre-processing stage,

The pixel loop can then run using the luminance mapping LUT $lutMapY$ and the colour correction LUT β_O.

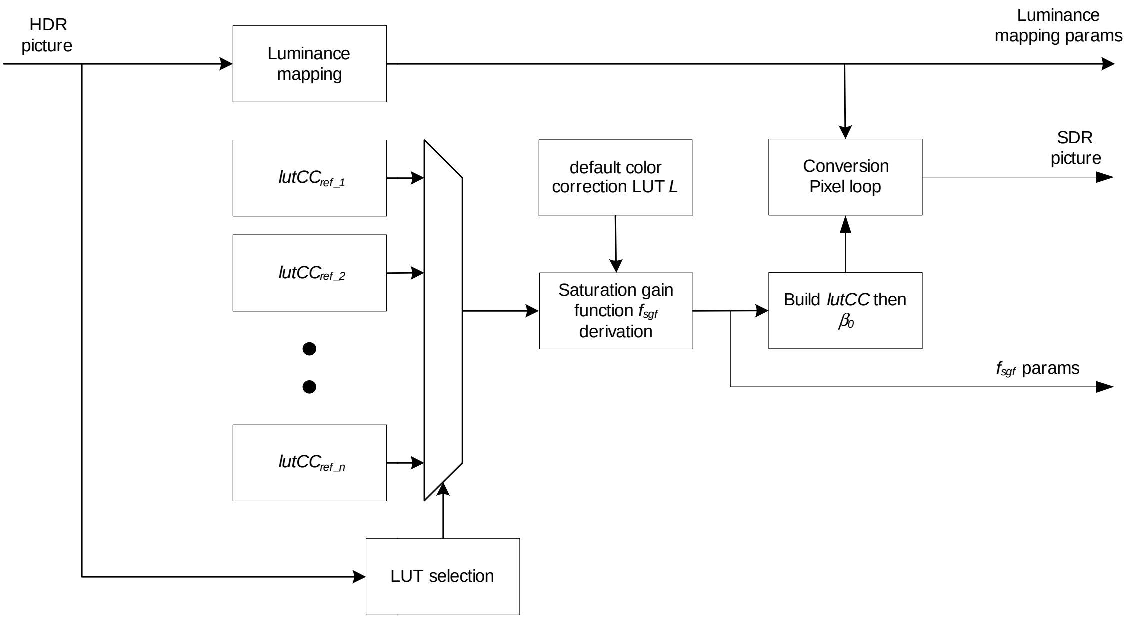

Figure C.11: Illustration of the simplified colour correction LUT usage in the pre-processing stage

Typically, three different pre-defined LUTs $lutCC_{ref\_i}$ are defined. Examples of three LUTs are depicted in Figure C.12. A possible process for the LUT selection is explained in clause C.3.4.3.

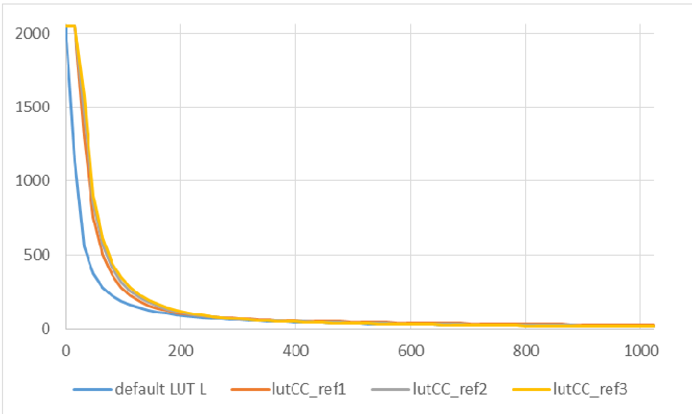

Figure C.12: Examples of pre-defined colour correction LUTs

C.3.4.3 Selection of the colour correction LUT

The LUT selection is based on the analysis of the input HDR image characteristics, in particular related to its saturation, hue, lightness properties. From this analysis, the relevant LUT is selected. The algorithm is summarized as follows.

- Computation of the saturation histogram and mean luminance on the HDR linear-light picture.
- Computation of the hue values, on the 5 %, 10 %, and 20 % most saturated pixels of the picture.
- Choice of the LUTs $lutCC_{ref\_k}$ based on those metrics.

In order to compute the saturation and hue of the HDR signal, R', G', B' non linear-light values are computed from R, G, B values corresponding to the linear-light HDR signal as follows:

$$\begin{cases} R' = f_{nll}(R) \\ G' = f_{nll}(G) \\ B' = f_{nll}(B) \end{cases} \quad \text{(C.82)}$$

where the $f_{nll}(\,)$ function is defined as follows:

$$f_{nll}\begin{cases} if\,(v < 0{,}0031308), f_{nll}(v) = Max\,(0\,; v \times 12{,}92) \\ else\ f_{nll}(v) = 1{,}055 \times v^{\frac{1}{2,4}} - 0{,}055 \end{cases} \tag{C.83}$$

The saturation histograms are derived as follows.

From the minimum value $minRGB$ and maximum value $maxRGB$ among R', G' and B', the saturation S is computed as follows:

$$\begin{cases} if\,(maxRGB > 0{,}01), S = \frac{maxRGB - minRGB}{maxRGB} \\ else\ S\ is\ not\ defined, and\ maxRGB\ is\ set\ to\ 0 \end{cases} \tag{C.84}$$

The histogram of S is computed over the HDR picture. From the histogram, three sets of pixels S_5, S_{10}, S_{20} are determined, corresponding to the 5 %, 10 % and 20 % most saturated pixels. The corresponding average saturation values $\overline{S_5}, \overline{S_{10}}, \overline{S_{20}}$ and average luminance values $meanL_5, meanL_{10}, meanL_{20}$ are computed. Similarly, the average luminance of the HDR picture is computed.

Hue histograms are computed as follows.

The hue value is only defined when $(maxRGB - minRGB < 0{,}0001)$ as follows:

$$\begin{cases} if\,(maxRGB = R'), hue_R = \left(\left[\frac{G' - B'}{maxRGB - minRGB} + 0\right] \times 60\right) \%\ 360 \\ if\,(maxRGB = G'), hue_G = \left(\left[\frac{B' - R'}{maxRGB - minRGB} + 2\right] \times 60\right) \%\ 360 \\ if\,(maxRGB = B'), hue_B = \left(\left[\frac{R' - G'}{maxRGB - minRGB} + 4\right] \times 60\right) \%\ 360 \end{cases} \tag{C.85}$$

Three histograms (one for each component), for each one of the three saturation sets S_5, S_{10}, S_{20}, are computed. This defines nine histograms H_{XY}, for X = R, G or B and Y = 5, 10 or 20. The mean hue values are computed for each of the 9 sets $\overline{H}_{XY}$, for X = R, G or B, and Y = 5, 10 or 20. Nine colour ratios R_{XY} are also computed as follows:

- for X = R, G or B, and Y = 5, 10 or 20:

$$R_{XY} = \frac{size(H_{XY})}{size(H_{RY}) + size(H_{GY}) + size(H_{BY})} \tag{C.86}$$

where $size()$ defines the cardinality of the associated histogram.

The selection of the LUT is based on the previously computed parameters representative of the HDR picture characteristics (e.g. saturation, hue, etc.). An example algorithm for 1 000 cd/m² peak luminance content is provided below.

- let k be the index of the pre-defined LUT $lutCC_{ref\_k}$.

- if ($Abs(\overline{H}_{R20}) < T_1$ and $\overline{S_5} > T_2$ and $\frac{size(S_{10})}{size(S_{20})} > T_3$ and $R_{R20} > R_{B20}$) $\quad k$ = 3

  else k = 2

- if (($\overline{L}$ < 5 and $\overline{S_{20}} < 0{,}6$) or ($\overline{L}$< 10 and $R_{B20} < 0{,}7$) or ($\overline{L}$< 20 and $R_{B20} < 0{,}8$) or ($\overline{L}$< 30 and $R_{B20} < 0{,}9$))

  $k = k - 1$

where T_1 = 8, T_2 = 0,95, T_3 = 0,8, and $\overline{L}$ is the average luminance of the HDR picture.

Annex D (informative): CE-friendly inverse gamut mapping

This annex provides the description of a lightweight inverse gamut mapping process that could apply when the input SDR picture of the SDR-to-HDR reconstruction process is provided in a BT.709 colour gamut (as specified by the variable **sdrPicColourSpace**), and is different from the target BT.2020 colour gamut of the HDR picture (as specified by the variable **hdrPicColourSpace**).

Figure D.1 illustrates a typical scenario where gamut mapping and inverse gamut mapping are required. In this example, the master HDR content coming from the production process is graded on a P3D65 HDR monitor (signalled by **hdrMasterDisplayColourSpace**), and represented in a BT.2020 colour gamut container (signalled by **hdrPicColourSpace**). In this scenario, the SDR backward compatible version derived from the HDR content are provided in a Recommendation ITU-R BT.709-6 [9] container, in order to be directly viewable on legacy BT.709 SDR displays. Therefore, for distribution of the SDR signal, a gamut mapping from Recommendation ITU-R BT.2020-2 [10] to BT.709 is required in addition to the dynamic range mapping from HDR to SDR before the distribution step. The distributed SDR content, represented in a Recommendation ITU-R BT.709-6 [9] container (signalled by **sdrPicColourSpace**), can then to be reconverted to an HDR version in a Recommendation ITU-R BT.2020-2 [10] container.

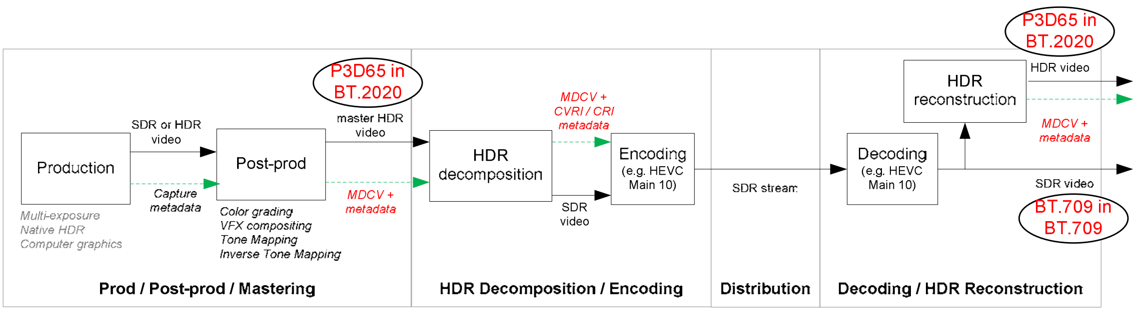

Figure D.1: example of use case requiring a gamut mapping

The inverse gamut mapping described in this annex typically addresses such use cases (corresponding to **hdrPicColourSpace** equal to 1, **hdrMasterDisplayColourSpace** equal to 2, **sdrMasterDisplayColourSpace** and **sdrPicColourSpace** equal to 0).

As illustrated in Figure 3, the process applies once the HDR picture has been reconstructed from the SDR picture by the SDR-to-HDR reconstruction process described in clause 7.2.4. The process applies when **hdrPicColourSpace** is not equal to **sdrPicColourSpace.**

The process takes as inputs:

- the reconstructed linear-light HDR 4:4:4 picture HDR_R, HDR_G, HDR_B ;
- the variables **hdrPicColourSpace**, **hdrMasterDisplayColourSpace** and **sdrPicColourSpace**.

The process provides as output:

- the gamut-mapped linear-light HDR 4:4:4 picture $mapHDR_R$, $mapHDR_G$, $mapHDR_B$.

The process applies the following steps for each pixel (x, y) of the picture:

- conversion from RGB to XYZ colour space, considering the RGB signal is represented in the colour gamut inferred from the variable **hdrMasterDisplayColourSpace**.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M_{RGB-to-XYZ} \times \begin{bmatrix} HDR_R[x][y] \\ HDR_G[x][y] \\ HDR_B[x][y] \end{bmatrix} \quad \text{(D.1)}$$

where $M_{RGB-to-XYZ}$ is the conversion matrix from RGB to XYZ, as specified in Table 16, with variable $matrixIdx$ equal to **hdrMasterDisplayColourSpace**.

- conversion from XYZ to RGB colour space, the RGB signal being represented in the colour gamut inferred from the variable **hdrPicColourSpace**.

$$\begin{bmatrix} mapHDR_R[x][y] \\ mapHDR_G[x][y] \\ mapHDR_B[x][y] \end{bmatrix} = M_{XYZ-to-RGB} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{D.2}$$

where $M_{XYZ-to-RGB}$ is the conversion matrix from XYZ to RGB, as specified in Table 15, with variable $matrixIdx$ equal to **hdrPicColourSpace**.

Annex E (informative): Display adaptation

E.1 Introduction

A video stream accompanied with metadata is generated with the system according to the present document and targets consumer HDR displays. Using the metadata standardized in the present document, an HDR output can be reconstructed with the original mastering display colour volume. The values of the minimum and maximum display luminance of the original mastering display are present in the metadata standardized in the present document (**hdrMasterDisplayMaxLuminance** and **hdrMasterDisplayMinLuminance**, see clause 6.2.3).

Display adaptation is the process that adapts the HDR output to a colour volume that has a different, typically lower, dynamic range than the dynamic range of the HDR mastering display.

Clause E.2 describes a display adaptation method that tries to maintain the artistic intent of the HDR and SDR images as much as possible in the image for the presentation display.

Clause E.3 describes how to set the MaxCLL and MaxFALL parameters defined in CTA-831.3 [i.1] when display adaptation is used and the presentation display is connected by HDMI.

E.2 Display adaptation maintaining creative intent

In this clause, a display adaptation method is described that uses recalculated metadata values based on the ratios between the original HDR peak luminance L_{HDR}. targeted SDR peak luminance (100 cd/m^2) and the presentation display maximum luminance, while maintaining the creative intent captured in the mapping between SDR and HDR as best as possible.

In the decoder, the linear-light value Y_{ll} coming out of the block "To linear signal", see clause 7.2.3.1.8, but before entering the block "Sqrt" in the decoder, see clause 7.2.3.1.9, is first processed by a tone mapping process as described in clause C.2.2 and see also Figure E.1.

NOTE: It is possible to create one LUT to implement the full process depicted in Figure E.1.

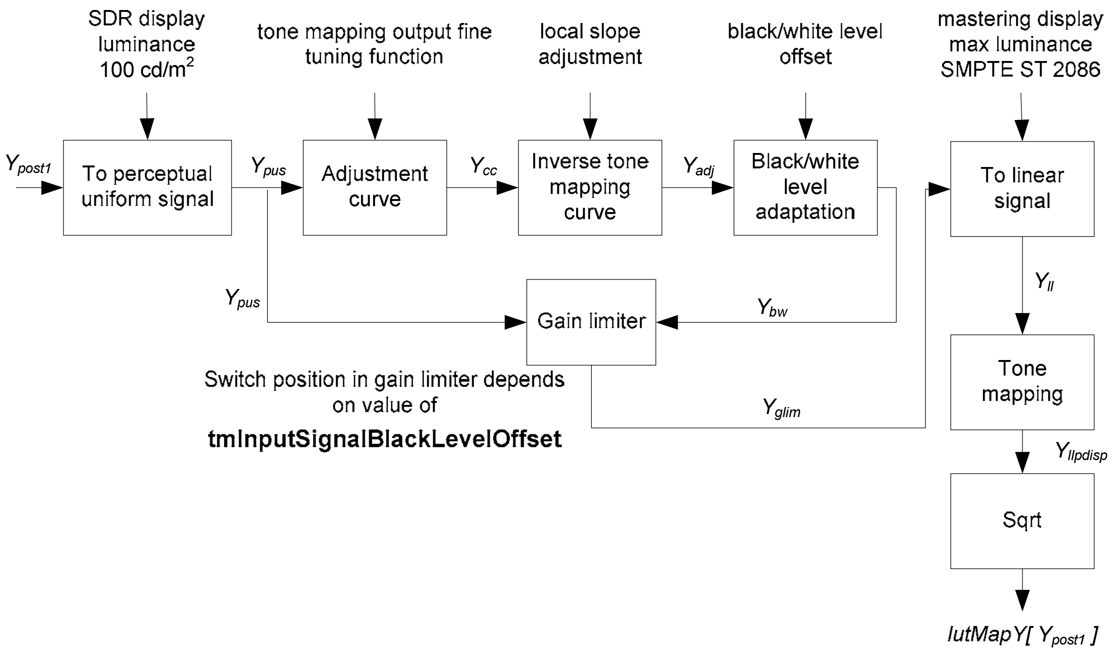

Figure E.1: Display adaptation process

The block "Tone mapping" in Figure E.1 consists of all of the blocks shown in Figure C.2 and is shown in Figure E.2.

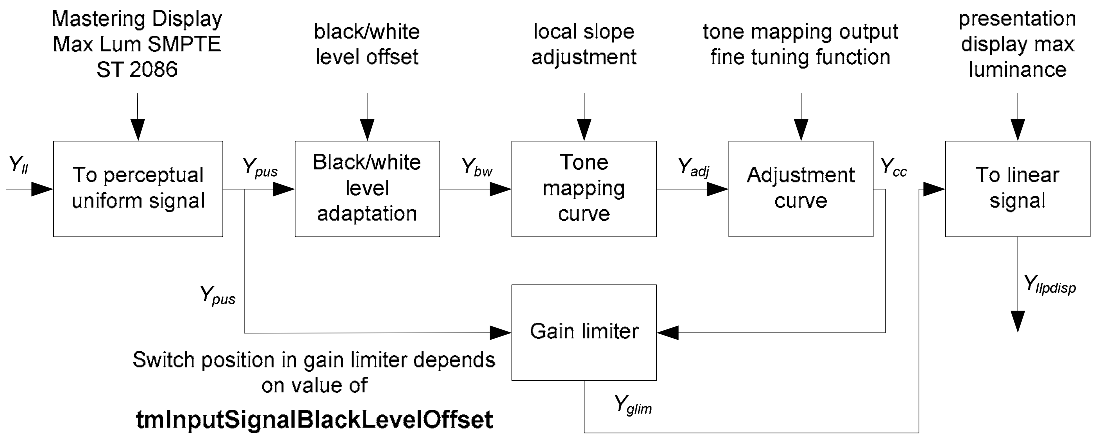

Figure E.2: Tone mapping process for display adaptation

The parameters to be used in the process "Tone mapping" in Figure E.2 can be recomputed based on the maximum luminance of the presentation display L_{pdisp}, as described below.

First, the scaling factors $scale$, $scaleHor$ and $scaleVer$ are computed, see equations (E.1) to (E.5).

$$\kappa = v\left(\frac{L_{HDR}}{L_{target}}, L_{target}\right) \tag{E.1}$$

$$\lambda = v\left(\frac{L_{HDR}}{L_{pdisp}}, L_{pdisp}\right) \tag{E.2}$$

$$scale = \frac{(\lambda-1)\times(\kappa+1)}{(\lambda+1)\times(\kappa-1)} \tag{E.3}$$

$$scaleHor = \frac{1-(1\div\lambda)}{1-(1\div\kappa)} \tag{E.4}$$

$$scaleVer = Max\left(\frac{1-\lambda}{1-\kappa}; 0\right) \tag{E.5}$$

where:

- L_{HDR} is the maximum display mastering luminance from the variable **hdrMasterDisplayMaxLuminance** in the structure hdr_characteristics( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.3);
- L_{target} is the maximum SDR luminance (100 cd/m^2);
- L_{pdisp} is the maximum luminance of the presentation display;
- and $v(x, y)$ is taken from equations (2) and (3).

The block "To perceptual uniform signal" in Figure E.2 can be used as described in clause C.2.2.2 with the parameters specified there.

NOTE: The operation in the block "To perceptual uniform signal" in Figure E.2 is the inverse of the block "To linear signal" in Figure E.1. Therefore, these two operations can be omitted in an implementation.

The block "Black/white level adaptation" in Figure E.2 can be used as described in clause C.2.2.3 with the parameters as recomputed according to equations (E.6) to (E.7).

$$TMWLO_{DA} = TMWLO \times Max(scaleHor; 0) \tag{E.6}$$

where:

- $TMWLO$ is the **tmInputSignalWhiteLevelOffset** as stored in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5; and

- $TMWLO_{DA}$ is the recomputed **tmInputSignalWhiteLevelOffset** to be used in the block "Black/white level adaptation" in Figure E.2.

$$TMBLO_{DA} = TMBLO \times Max(scaleHor; 0) \quad \text{(E.7)}$$

where:

- $TMBLO$ is the **tmInputSignalBlackLevelOffset** as stored in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5; and
- $TMBLO_{DA}$ is the recomputed **tmInputSignalBlackLevelOffset** to be used in the block "Black/white level adaptation" in Figure E.2.

The block "Tone mapping curve" in Figure E.2 can be used as described in clause C.2.2.4 with the parameters as recomputed according to equations (E.8) to (E.16).

$$MIDX = \frac{1-HGC}{SGC-HGC} \quad \text{(E.8)}$$

$$MIDX_{DA} = \frac{MIDX \times (SGC-1)}{2} \times (1-scale) + MIDX \quad \text{(E.9)}$$

$$MIDY_{DA} = -1 \times MIDX_{DA} + MIDX \times (SGC+1) \quad \text{(E.10)}$$

$$SGC_{DA} = \frac{MIDY_{DA}}{MIDX_{DA}} \quad \text{(E.11)}$$

where:

- SGC and HGC are computed according to equations (C.30) and (C.31).

$$para_{DA} = v(Abs(scale), L_{HDR}) \times para \quad \text{(E.12)}$$

where:

- $para$ is computed according to equation (C.32);
- and $v(x, y)$ is taken from equations (2) and (3).

$$HGC_{DA} = \begin{cases} 0, & \text{if } MIDX_{DA} - 1 = 0 \\ Max\left(\frac{MIDY_{DA}-1}{MIDX_{DA}-1}; 0\right), & otherwise \end{cases} \quad \text{(E.13)}$$

$$\mathbf{shadowGain_{DA}} = \left(\frac{SGC_{DA}}{\lambda} - 0.5\right) \times 4 \quad \text{(E.14)}$$

where:

- **shadowGain$_{DA}$** is the recomputed **shadowGain** to be used in the block "Tone mapping curve" in Figure E.2.

$$\mathbf{highlightGain_{DA}} = HGC_{DA} \times 4 \quad \text{(E.15)}$$

where:

- **highlightGain$_{DA}$** is the recomputed **highlightGain** to be used in the block "Tone mapping curve" in Figure E.2.

$$\mathbf{midToneWidthAdjFactor_{DA}} = para_{DA} \times 2 \quad \text{(E.16)}$$

where:

- **midToneWidthAdjFactor$_{DA}$** is the recomputed **midToneWidthAdjFactor** to be used in the block "Tone mapping curve" in Figure E.2.

The block "Adjustment curve" in Figure E.2 can be used as described in clause C.2.2.5 with the pairs **tmOutputFineTuningX**[i], **tmOutputFineTuningY**[i] as recomputed according to equations (E.17) to (E.19).

First, the points **tmOutputFineTuningX**[i], which are values in the perceptual uniform domain of the SDR image, are scaled to the corresponding values for the HDR image at the mastering display by 'going backwards' through the block "Tone mapping curve" and the block "Black/white level adaptation" in the encoder, Figure C.2. Going backwards means that first the inverse tone mapping has to be applied and then the inverse black/white adaptation, see equation (E.17).

$$x_{i_{HDR}} = BWAD_{inv}\left(TMO_{inv}(x_i)\right) \quad \text{(E.17)}$$

where:

- x_i is the **tmOutputFineTuningX**[i] as stored in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5;
- $x_{i_{HDR}}$ is the scaled **tmOutputFineTuningX**[i] corresponding to the HDR image at the mastering display;
- $TMO_{inv}(x)$ is taken from equation (7), using the values of the variables **shadowGain**, **highlightGain** and **midToneWidthAdjFactor** in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata (clause 6.2.5);
- $BWAD_{inv}\left(Y_{adj}\right) = Y_{bw}$ as computed by equation (15).

Next, the corresponding values for the HDR image at the mastering display, $x_{i_{HDR}}$, are scaled to correspond to the image at the presentation display, using the block "Black/white level adaptation" and the block "Tone mapping curve" in the encoder, see equation (E.18).

$$x_{i_{DA}} = TMO_{DA}\left(BWAD_{DA}\left(x_{i_{HDR}}\right)\right) \quad \text{(E.18)}$$

where:

- $x_{i_{DA}}$ is the recomputed **tmOutputFineTuningX**[i] to be used in the block "Adjustment curve" in Figure E.2;
- $BWAD_{DA}\left(Y_{pus}\right) = Y_{bw}$, as computed by equations (C.19), (C.20) and (C.21) and using the recomputed **tmInputSignalBlackLevelOffset** and **tmInputSignalWhiteLevelOffset** from equations (E.6) to (E.7);
- and $TMO_{DA}(X)$ is $TMO(X)$ from (C.23) up to and including equation (C.32), with the parameters as recomputed according to equations (E.8) to (E.16).

Last, the points **tmOutputFineTuningY**[i], are scaled to what they should be for the image at the presentation display with equation (E.19) using the scaling factor $scaleVer$ derived with equation (E.5).

$$y_{i_{DA}} = Min\left((y_i - x_i) \times scaleVer + x_{i_{DA}}\,;\ 1\right) \quad \text{(E.19)}$$

where:

- y_i is the **tmOutputFineTuningY**[i] as stored in the structure luminance_mapping_variables( ) of the SDR-to-HDR reconstruction metadata as specified in clause 6.2.5; and
- $y_{i_{DA}}$ is the recomputed **tmOutputFineTuningY**[i] to be used in the block "Adjustment curve" in Figure E.2.

The block "To linear signal" in Figure E.2 is used as described in clause C.2.2.7 with the value of L_{pdisp} for the parameter L_{HDR}.

The output of the tone mapping process, the signal $Y_{llpdisp}$ from the block "To linear signal" in Figure E.2, is then used as input signal Y_{ll} for the block "Sqrt" in the decoder, see clause 7.2.3.1.9.

E.3 Display adaptation and HDMI

In case the display adaptation process that is described in clause E.2 is performed and the presentation monitor is connected by HDMI, the values of the maximum content light level MaxCLL and the maximum picture average light level MaxFALL in the Dynamic Range and Mastering InfoFrame, as specified in the CTA-861.3 specification [i.1] should be set to 0.

Annex F (informative): Change History

Date	Version	Information about changes
February 2016	0.0.1	Early draft: Plan of the specification Introduction and scope
March 2016	0.0.2	Stable draft 1
April 2016	0.0.3	Stable draft 2
May 2016	0.0.4	Final draft for approval

History

Document history		
V1.1.1	August 2016	Publication