ETSI TS 103 433-1 V1.2.1 (2017-08)

TECHNICAL SPECIFICATION

High-Performance Single Layer High Dynamic Range (HDR) System for use in Consumer Electronics devices; Part 1: Directly Standard Dynamic Range (SDR) Compatible HDR System (SL-HDR1)

EBU

OPERATING EUROVISION

Reference
RTS/JTC-040-1

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

Contents

Intellectual Property Rights

Essential patents

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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

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).

The present document is part 1 of a multi-part document covering the High-Performance Single Layer High Dynamic Range (HDR) System for use in Consumer Electronics devices, as identified below:

Part 1: "Directly Standard Dynamic Range (SDR) Compatible HDR System (SL-HDR1)";

Part 2: "Enhancements for Perceptual Quantization (PQ) transfer function based High Dynamic Range (HDR) Systems (SL-HDR2)";

Part 3: "Enhancements for Hybrid Log Gamma (HLG) transfer function based High Dynamic Range (HDR) Systems (SL-HDR3)".

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 equipment 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 specified in Annex A and Annex B.

Post-processing

In the present document, the post-processing stage that occurs in the IRD 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 7, 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 present 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 abstraction layer (agnostic to the distribution format) implementing the content-based dynamic metadata common to systems based on ETSI TS 103 433 multi-part document. Specifically to the present document, the metadata are produced during the HDR-to-SDR decomposition stage and they 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 SEI messages representative of SL-HDR system that are specified for HEVC and AVC respectively in Annex A and Annex B. Informative Annex C, Annex D and Annex E provide information on an HDR-to-SDR decomposition process, a gamut mapping process as well as its inverse process and HDR-to-HDR display adaptation. Informative Annex F proposes a recovery procedure when dynamic metadata are detected as missing by the post-processor during the HDR signal reconstruction. Eventually, informative Annex G gives reference to a standard mechanism to carry SL-HDR reconstruction metadata through interfaces.

The structure of the present document is summarized in Table 1.

Table 1: Structure of the present document

Clause/Annex #	Description	Normative / Informative (in the present document)	Part(s) for which the clause/annex is valid
Clause 1	Scope of the document	Informative	1
Clause 2	References used in the document	Normative/Informative	1
Clause 3	Definitions, symbols, abbreviations	Normative	1
Clause 4	End-to-end system	Informative	1
Clause 5	Architecture of the HDR system	Informative	1
Clause 6	Metadata format abstraction layer (agnostic to the distribution format)	Normative	1, 2, 3
Clause 7	SDR-to-HDR reconstruction process	Normative	1
Annex A	SL-HDR reconstruction metadata using HEVC	Normative	1, 2, 3
Annex B	SL-HDR reconstruction metadata using AVC	Normative	1, 2, 3
Annex C	HDR-to-SDR decomposition process	Informative	1
Annex D	Invertible gamut mapping process	Informative	1
Annex E	HDR-to-HDR display adaptation process	Informative	1
Annex F	Error-concealment and recovery procedure	Informative	1
Annex G	ETSI TS 103 433 signalling in CTA-861-G	Informative	1, 2, 3

1 Scope

The present document specifies the content-based dynamic metadata common to systems based on ETSI TS 103 433 multi-part deliverable 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 RP 431-2:2011: "D-Cinema Quality - Reference Projector and Environment".

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

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

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

[9] 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: "A reference environment for evaluation of HDTV program material or completed programmes".

[i.3] SMPTE ST 2094-20:2016: "Dynamic Metadata for Color Volume Transform - Application #2".

[i.4] SMPTE ST 2094-30:2016: "Dynamic Metadata for Color Volume Transform - Application #3".

[i.5] SMPTE RP 2077:2013: "Full Range Image Mapping".

[i.6] Recommendation ITU-R BT.2100: "Image parameter values for high dynamic range television for use in production and international programme exchange".

[i.7] SMPTE Engineering Guideline EG 28-1993: "Annotated Glossary of Essential Terms for Electronic Production".

[i.8] CTA Standard CTA-861-G (November 2016): "A DTV Profile for Uncompressed High Speed Digital Interfaces".

[i.9] SMPTE RP 177:1993: "Derivation of Basic Television Color Equations".

[i.10] Recommendation ITU-T T.35 (02-2000): "Procedure for the allocation of ITU-T defined codes for non-standard facilities".

[i.11] JCTVC-Z1017: "Conversion and Coding Practices for HDR/WCG Y'CbCr 4:2:0 Video with PQ Transfer Characteristics (Draft 4)".

[i.12] ETSI TS 103 433 (V1.1.1): "High-Performance Single Layer Directly Standard Dynamic Range (SDR) Compatible High Dynamic Range (HDR) System for use in Consumer Electronics devices (SL-HDR1)".

3 Definitions, symbols, abbreviations and conventions

3.1 Definitions

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

anchor: invariant point that can be retrieved during the inverse hue mapping process

NOTE 1: Term specific to the invertible gamut mapping process.

NOTE 2: The anchor guarantees the invertibility of the hue mapping process.

chrominance: chrominance components are denoted U and V in the linear-light YUV colour space

NOTE 1: Term specific to the gamut mapping (or inverse) process.

NOTE 2: Typically, in the linear-light YUV colour space, it corresponds to the radial coordinate of the cylindrical representation of a colour i.e. chrominance(Y,U,V) = $\sqrt{U^2 + V^2}$.

cold colour: Colours for which the lightness value is less in the cusp of the wider gamut than in the cusp of the smaller gamut (e.g. green, blue and cyan)

NOTE: Term specific to the gamut mapping (or inverse) process.

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

decomposed picture: SDR picture derived from the HDR-to-SDR pre-processing stage

NOTE: Type of pre-processed picture.

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

gamut: complete subset of colours which can be represented within a given colour space or by a certain output device

NOTE: Also known as colour gamut.

gamut mapping: mapping of the colour space coordinates of the elements of a source image to colour space coordinates of the elements of a reproduction

NOTE: Gamut mapping intent is not to change the dynamic range of the source but to compensate for differences in the source and output medium colour gamut capability.

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 artefacts, including sufficient separation of diffuse white and specular highlights

hue: angular coordinate of the cylindrical representation of a colour in the linear-light YUV colour space

NOTE 1: Term specific to the gamut mapping (or inverse) process.

NOTE 2: Typically, hue may be computed as $\mathrm{hue}(Y,U,V) = \mathrm{atan2}(V, U)$, where $\mathrm{atan2}$ is the two-argument inverse tangent.

lightness: Y component of the linear-light YUV colour space

NOTE: Term specific to the gamut mapping (or inverse) process.

luma: linear combination of non-linear-light (gamma-corrected) primary colour signals

luminance: objective measure of the visible radiant flux weighted for colour by the CIE Photopic Spectral Luminous Efficiency Function ([i.7])

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

post-production: part of the process of filmmaking and video production gathering many different processes such as video editing, adding visual special effects, transfer of colour motion picture film to video

NOTE: The pre-processed picture is generated during the post-production stage at the encoding site.

pre-processed picture: output picture of SL-HDR pre-processing stage

reconstructed picture: output picture of SL-HDR post-processing stage

saturation: chrominance value normalized by the lightness value

NOTE: Term specific to the gamut mapping (or inverse) process.

Single Layer High Dynamic Range (SL-HDR) system: system implementing at least one of the parts of the ETSI TS 103 433 multi-part deliverable

source picture: input picture of SL-HDR pre-processing stage

NOTE: Typically an HDR picture coming from post-production facilities.

Standard Colour Gamut (SCG): chromaticity gamut equal to the chromaticity gamut defined by Recommendation ITU-R BT.709-6 [6]

Standard Dynamic Range (SDR) system: system having a reference reproduction using a luminance range constrained by Recommendation ITU-R BT.2035 [i.2], section 3.2

NOTE: Typically no more than 10 stops.

Supplemental Enhancement Information (SEI) message: carriage mechanism defined in Recommendation ITU-T H.264 [3] and Recommendation ITU-T H.265 [4] that is intended to assist in processes related to decoding, display or other purposes

target picture: picture graded on an SDR mastering display

warm colour: colours for which the lightness value is greater in the cusp of the wider gamut than in the cusp of the smaller gamut (e.g. yellow, red and magenta)

NOTE: Term specific to the gamut mapping (or inverse) process.

Wide Colour Gamut (WCG): chromaticity gamut larger than the chromaticity gamut defined by Recommendation ITU-R BT.709-6 [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 $x \geq 0$ and $y > 0$

3.2.2 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)	$\begin{cases} x & , \quad z < x \\ y & , \quad z > y \\ z & , \quad \text{otherwise} \end{cases}$
Floor(x)	the largest integer less than or equal to x

ln(x)	natural logarithm of x
log_{10}(x)	logarithm with base 10 of x
Min(x ; y)	$\begin{cases} x & , & x \le y \\ y & , & x > y \end{cases}$
Max(x ; y)	$\begin{cases} x & , & x \ge y \\ y & , & x < y \end{cases}$
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

3.2.3 Logical operators

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

x && *y*	Boolean logical "and" of x and y
x ? y : z	If x is TRUE or not equal to 0, evaluates to the value of y; otherwise, evaluates to the value of z

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
EIT	Event Information Table
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
SCG	Standard Colour Gamut
SDR	Standard Dynamic Range
SDT	Service Descriptor Table
SEI	Supplemental Enhancement Information (as in AVC and HEVC)
SL-HDR	Single Layer High Dynamic Range
SL-HDRI	Single Layer High Dynamic Range Information
SMPTE	Society of Motion Picture and Television Engineers
STB	Set Top Box
VUI	Video Usability Information
WCG	Wide Colour Gamut

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 lower 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. $variable$.

- Structures of syntactic elements or structures of variables 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. The primary goal of this HDR workflow 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 static and dynamic metadata:

- Mastering Display Colour Volume (MDCV) standardized in AVC [3], HEVC [4] and SMPTE ST 2086 [2] specifications;
- SL-HDR Information (SL-HDRI) based on both SMPTE ST 2094-20 [i.2] and SMPTE ST 2094-30 [i.3] specifications.

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 of broadcast applications.

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

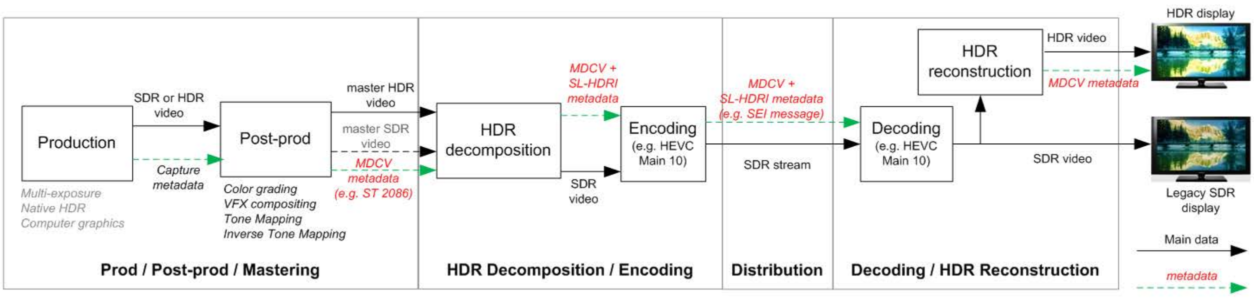

Figure 1: Example of HDR end-to-end system

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 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. Optionally, a block of HDR-to-HDR signal reconstruction may be used as a display adaptation process. The dynamic range output of the display adaptation process is less than the dynamic range of the HDR signal output of the SDR-to-HDR signal reconstruction process.

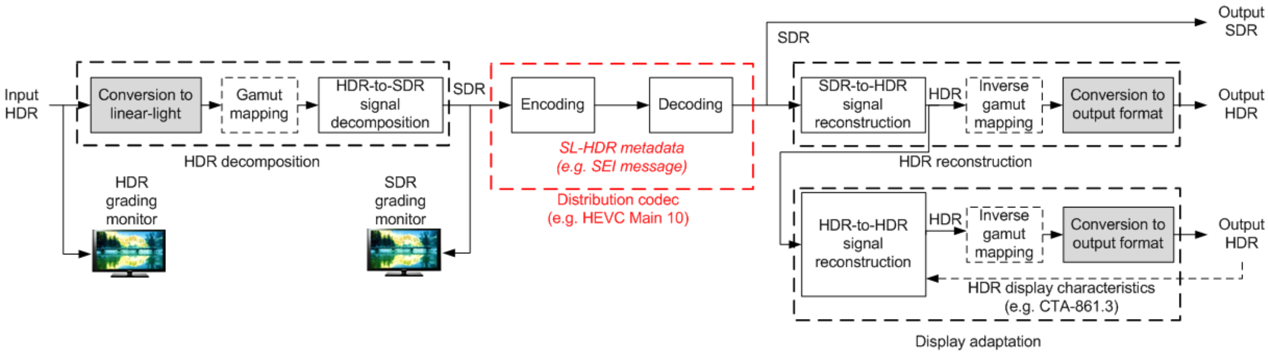

Figure 2: HDR system architecture overview

6 Dynamic metadata format for signal reconstruction

6.1 Introduction

Clause 6 specifies the dynamic metadata format for signal reconstruction. In the present document, the dynamic metadata allow SDR-to-HDR reconstruction of the HDR signal.

NOTE: Dynamic metadata format specified in clause 6 applies to each part of the ETSI TS 103 433 multi-part document. Specificities may be specified for each part of the multi-part deliverable.

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. The number of bits representative of the bit depth of the variables is provided in order to assist hardware buses sizing. When the number of bits used to encode a variable is not fixed and not bound by constants, it is indicated as VAR.

Clause 6.3 specifies the semantics and the precision of the reconstruction metadata.

6.2 Reconstruction metadata syntax

6.2.1 Introduction

This clause specifies a format abstraction layer implementing the static and dynamic metadata used for signal reconstruction (SDR-to-HDR reconstruction in the present document) agnostically to the distribution format (i.e. independent of the SEI message syntax specified in Annex A and Annex B). This format supports two mutually exclusive carriage modes: parameter-based mode and table-based mode. 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 Signal reconstruction information

The syntax of signal_reconstruction_info( ) is specified in Table 2. In the present document, the reconstructed signal is an HDR signal.

Table 2: Syntax of signal_reconstruction_info( )

Syntax	No. of bits
signal_reconstruction_info( )	
{	
partID	4
majorSpecVersionID	4
minorSpecVersionID	7
payloadMode	3
hdr_characteristics( )	36
sdr_characteristics( )	36
for(i=0; i<4 ; i++)	
matrixCoefficient[i]	10
for(i=0; i<2 ; i++)	
chromaToLumaInjection[i]	13
for(i=0; i<3 ; i++)	
kCoefficient[i]	6/7/8
switch(payloadMode) {	
case 0:	
luminance_mapping_variables( )	VAR
colour_correction_adjustment( )	VAR
break	
case 1:	
luminance_mapping_table( )	VAR
colour_correction_table( )	VAR
break	
}	
if(sdrPicColourSpace < hdrPicColourSpace) {	
gamutMappingMode	8
if(gamutMappingMode == 1)	
gamut_mapping_variables( )	VAR
}	
}	
NOTE: The number of bits used to represent **kCoefficient**[0], **kCoefficient**[1], **kCoefficient**[2] is respectively 6, 7 and 8 bits.	

6.2.3 HDR picture characteristics

HDR picture characteristics (i.e. source picture and reconstructed picture formats in the present document) are specified by syntax elements present in Table 3. From the IRD viewpoint, those variables can be mapped from HEVC and AVC Mastering Display Colour Volume (SMPTE ST 2086 [2]) SEI message or SL-HDR Information SEI message syntax elements as respectively specified in normative Annex A.

NOTE: It is noted that the dynamic range of the picture presented to the output HDR display may differ from the dynamic range of the source picture e.g. when HDR-to-HDR display adaptation (documented in clause E.2) is activated.

Table 3: Syntax of hdr_characteristics( )

Syntax	No. of bits
hdr_characteristics( )	
{	
hdrPicColourSpace	2
hdrDisplayColourSpace	2
hdrDisplayMaxLuminance	16
hdrDisplayMinLuminance	16
}	

6.2.4 SDR picture characteristics

SDR picture characteristics (i.e. pre-processed picture format in the present document) are specified in Table 4. From the IRD viewpoint, those variables can be mapped from SL-HDR Information SEI message syntax elements as respectively specified in normative Annex A and Annex B.

Table 4: Syntax of sdr_characteristics( )

Syntax	No. of bits
sdr_characteristics( )	
{	
sdrPicColourSpace	2
sdrDisplayMaxLuminance	16
sdrDisplayMinLuminance	16
}	

6.2.5 Luminance mapping variables

The luminance mapping variables are specified by syntax elements present in Table 5. Luminance mapping variables are invoked when **payloadMode** is equal to 0.

Table 5: 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 6. Colour correction adjustment variables are invoked when **payloadMode** is equal to 0.

Table 6: 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 7. Luminance mapping table is invoked when **payloadMode** is equal to 1.

Table 7: Syntax of luminance_mapping_table( )

Syntax	No. of bits
luminance_mapping_table( )	
{	
luminanceMappingNumVal	7
for(i = 0; i < luminanceMappingNumVal; i++) {	
luminanceMappingX[i]	14
luminanceMappingY[i]	13
}	
}	

6.2.8 Colour correction table

The colour correction table is specified by syntax elements present in Table 8. Colour correction table is invoked when **payloadMode** is equal to 1.

Table 8: Syntax of colour_correction_table( )

Syntax	No. of bits
colour_correction_table( )	
{	
colourCorrectionNumVal	7
for(i = 0; i < colourCorrectionNumVal; i++) {	
colourCorrectionX[i]	12
colourCorrectionY[i]	11
}	
}	

6.2.9 Gamut mapping variables

The gamut mapping variables are specified by syntax elements present in Table 9. Those variables are invoked when **sdrPicColourSpace** value is less than **hdrPicColourSpace** value and when **gamutMappingMode** is equal to 1.

Table 9: Syntax of gamut_mapping_variables( )

Syntax	No. of bits
gamut_mapping_variables( )	
{	
satMappingMode	2
switch(satMappingMode) {	
case 1:	
satGlobal1SegRatio	3
satGlobal2SegRatioWCG	3
satGlobal2SegRatioSCG	3
break	
case 2:	
for(c=0; c<6 ; c++) {	
sat1SegRatio[c]	3
sat2SegRatioWCG[c]	3
sat2SegRatioSCG[c]	3
}	
break	
}	
lightnessMappingMode	2
switch(lightnessMappingMode) {	
case 3:	
for(c=0; c<6 ; c++)	
lmWeightFactor[c]	3
break	
}	
croppingModeSCG	2
switch(croppingModeSCG) {	
case 3:	
for(c=0; c<6 ; c++)	
cmWeightFactor[c]	3
break	
}	
if(croppingModeSCG)	
cmCroppedLightnessMappingEnabledFlag	1
hueAdjMode	2
switch(hueAdjMode) {	
case 2:	
hueGlobalPreservationRatio	3
break	
case 3:	
for(c=0; c<6 ; c++)	
huePreservationRatio[c]	3
break	
}	
if(hueAdjMode) {	
hueAlignCorrectionPresentFlag	1
if(hueAlignCorrectionPresentFlag)	
for(c=0; c<6 ; c++)	
hueAlignCorrection[c]	3
chromAdjPresentFlag	1
if(chromAdjPresentFlag)	
for(c=0; c<6 ; c++)	
chromAdjParam[c]	2
}	
}	

6.3 Reconstruction metadata semantics

6.3.1 Introduction

This clause specifies the semantics of the static and dynamic metadata used for the signal reconstruction.

6.3.2 Signal reconstruction information

6.3.2.1 Introduction

In the present document, signal_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.

6.3.2.2 partID - part indicator of the multi-part document

This 4-bit code indicates the part of the ETSI TS 103 433 multi-part deliverable to which the bitstream conforms to.

6.3.2.3 majorSpecVersionID - Major specification version indicator

This 4-bit code indicates the specification version to which the bitstream conforms to.

6.3.2.4 minorSpecVersionID - Minor specification version indicator

This 7-bit code indicates the specification version to which the bitstream conforms to.

6.3.2.5 payloadMode - Payload carriage mode

This variable indicates the carriage mode used to implement the dynamic metadata representative of the colour volume transform. The value of **payloadMode** shall be equal to 0b000 or 0b001 only, see Table 10.

Table 10: Payload carriage mode

Value of payloadMode	Carriage mode
0b000	Parameter-based
0b001	Table-based
0b010 - 0b111	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 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: In the present document, 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 source and pre-processed pictures.

6.3.2.6 matrixCoefficient - Y'CC-to-R'G'B' conversion matrix coefficients

These variables specify matrix coefficients that are to be used to compute the Y'CC to R'G'B' conversion matrix that is employed in the SL-HDR reconstruction process. The value of **matrixCoefficient**[i] shall be in the bounded range $[-2 \text{ to } 2-\frac{1}{256}]$ and in multiples of (1÷256).

NOTE: Typically, the Y'CC to R'G'B' coefficients carried in **matrixCoefficient**[i] are representative of the coefficients of the appropriate canonical Y'CC to R'G'B' matrix. An example computation is provided in clause F.2. These matrix coefficients are not intended to be used in the gamut mapping process described in Annex D.

6.3.2.7 chromaToLumaInjection - Chroma to luma injection

This array of two variables respectively indicates the ratio of the blue and red colour-difference component injection into the luma component. The value of **chromaToLumaInjection**[i] shall be in the bounded range [0 to 0,5 $-\frac{1}{16384}$] and in multiples of (1÷16 384).

6.3.2.8 kCoefficient

This array is composed of three variables which specifies coefficients used in the computation of the reconstruction process. The value of **kCoefficient**[0] shall be in the bounded range [0 to (63 ÷ 256)] and in multiples of (1 ÷ 256). The value of **kCoefficient**[1] shall be in the bounded range [0 to (127 ÷ 256)] and in multiples of (1 ÷ 256). The value of **kCoefficient**[2] shall be in the bounded range [0 to (255 ÷ 256)] and in multiples of (1÷256).

6.3.2.9 gamutMappingMode

This variable is invoked when **sdrPicColourSpace** is less than **hdrPicColourSpace**. This variable indicates the gamut mapping mode that is representative of the gamut mapping parameters. In the present document, the value of **gamutMappingMode** shall be in the range of 0 to 3, inclusive and 64 to 127, inclusive, see Table 11.

Table 11: Gamut mapping mode

Value of gamutMappingMode	Gamut mapping mode
0	Implementation dependent method
1	Explicit parameters (see clause 6.3.9)
2	Preset #1: BT.709 to P3D65 gamut (see Table 12)
3	Preset #2: BT.709 to BT.2020 gamut (see Table 13)
4 - 63	Reserved for future use
64 - 127	Unspecified
128 - 255	Reserved for future use

When **gamutMappingMode** is equal to 0, no parameter information of gamut mapping (gamut compression or expansion) is carried in the bitstream and the implementer may use its own method to recover the initial HDR picture gamut. When **gamutMappingMode** is not equal to 0, parameters of the gamut mapping process are carried either explicitly (see clause 6.3.9) or implicitly (predetermined values). The unspecified values of **gamutMappingMode** in the range of 64 to 127, inclusive, are values unspecified by the present document and that may be used by some means unspecified in the present document to identify particular presets (e.g. determined by another standardization body).

Preset #1 and preset#2 shall only apply to the present document. Table 12 and Table 13 respectively provide the predetermined values of the gamut mapping variables that respectively correspond to an inverse gamut mapping (gamut expansion) from BT.709 gamut represented with BT.709 primaries to P3D65 gamut represented with BT.2020 primaries (preset #1) or from BT.709 gamut represented with BT.709 primaries to BT.2020 gamut represented with BT.2020 primaries (preset #2).

Table 12: Preset #1: BT.709 gamut to P3D65 gamut with BT.2020 primaries

Gamut mapping variable	Variable value
satMappingMode	2
sat1SegRatio[c]	$\{\frac{6}{8};\frac{7}{8};\frac{7}{8};\frac{6}{8};\frac{6}{8};\frac{7}{8}\}$
sat2SegRatioWCG[c]	$\{\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64}\}$
sat2SegRatioSCG[c]	$\{\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64}\}$
lightnessMappingMode	2
croppingModeSCG	2
cmCroppedLuminanceMappingEnabledFlag	1
hueAdjMode	2
hueGlobalPreservationRatio	1
hueAlignCorrectionPresentFlag	1
hueAlignCorrection[c]	{4; 5; 4; 4; 5; 4}
chromAdjPresentFlag	0

Table 13: Preset #2: BT.709 gamut to BT.2020 gamut with BT.2020 primaries

Gamut mapping variable	Variable value
satMappingMode	2
sat1SegRatio[c]	$\{\frac{5}{8};\frac{5}{8};\frac{6}{8};\frac{4}{8};\frac{5}{8};\frac{7}{8}\}$
sat2SegRatioWCG[c]	$\{\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64};\frac{35}{64}\}$
sat2SegRatioSCG[c]	$\{\frac{49}{64};\frac{49}{64};\frac{35}{64};\frac{49}{64};\frac{49}{64};\frac{35}{64}\}$
lightnessMappingMode	2
croppingModeSCG	2
cmCroppedLuminanceMappingEnabledFlag	1
hueAdjMode	2
hueGlobalPreservationRatio	1
hueAlignCorrectionPresentFlag	1
hueAlignCorrection[c]	{4; 5; 4; 4; 5; 4}
chromAdjPresentFlag	0

6.3.3 HDR picture characteristics

6.3.3.1 Introduction

The hdr_characteristics( ) contains 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 display used to master the HDR picture.

NOTE: The colour space and the display primaries which qualify the HDR reconstructed picture format are identical to the colour space and the mastering display primaries which qualify the HDR source picture format.

6.3.3.2 hdrPicColourSpace - HDR picture colour space

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 [9] and defined in Table 14.

Table 14: HDR picture colour primaries and white point

Value of hdrPicColourSpace	Primaries			NOTE: Colour space primaries
0	primary	x	y	as defined in Recommendation ITU-R BT.709-6 [6]
	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 [7]
	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			

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

NOTE: In the present document, an inverse gamut mapping conversion process is recommended to be invoked when the gamut in which the HDR picture is represented is greater than the gamut in which the SDR picture is represented.

6.3.3.3 hdrDisplayColourSpace - Colour space of the mastering display used to master the HDR picture

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 [9] and defined in Table 15. By extension, these values also apply to the reconstructed picture.

Table 15: HDR picture mastering display colour primaries and white point

Value of hdrDisplayColourSpace	Primaries			NOTE: Colour space primaries
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 [6]
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 [7]
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 RP 431-2 [5] (DCI-P3) white point as defined in Recommendation ITU-R BT.2020-2 [7]
3	Reserved for future use			

6.3.3.4 hdrDisplayMaxLuminance - HDR mastering display maximum luminance

This variable specifies the nominal maximum display luminance of the mastering display used to grade the HDR picture in units of 1 candela per square metre with a rounding to the nearest multiple of 50 candelas per square metre. The proper value of this variable shall be present in associated bitstreams that conform to the present document.

NOTE: In the present document, **hdrDisplayMaxLuminance** value is also used for the value of the display maximum luminance used for reconstructing the reconstructed picture when display adaptation is not activated.

6.3.3.5 hdrDisplayMinLuminance - HDR mastering display minimum luminance

This variable specifies the nominal minimum display luminance of the mastering display used to grade the HDR picture in units of 0,000 1 candelas per square metre. **hdrDisplayMinLuminance** shall be less than **hdrDisplayMaxLuminance**. If the proper value of **hdrDisplayMinLuminance** is unknown, it is recommended that it is set to 0.

6.3.4 SDR picture characteristics

6.3.4.1 Introduction

The sdr_characteristics( ) contains SDR picture signal characteristics namely: an indication on the colour space in which the SDR picture is represented, the nominal maximum and minimum luminance values of the mastering display that was used to master the SDR picture.

NOTE: In the present document, the SDR picture corresponds to the pre-processed picture that is intended to be encoded and transmitted on distribution networks. If no mastering display has been employed due to an automatic derivation process of the SDR picture, a value is inferred by the present document.

6.3.4.2 sdrPicColourSpace - SDR picture colour space

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 [9] and defined in Table 16.

Table 16: SDR picture colour primaries and white point

Value of sdrPicColourSpace	Primaries	Colour space primaries
0	primary x y green 0,300 0,600 blue 0,150 0,060 red 0,640 0,330 white D65 0,3127 0,3290	as defined in Recommendation ITU-R BT.709-6 [6]
1	primary x y green 0,170 0,797 blue 0,131 0,046 red 0,708 0,292 white D65 0,3127 0,3290	as defined in Recommendation ITU-R BT.2020-2 [7]
2 - 3	Reserved for future use	

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

NOTE: This constraint means that the colour space in which the SDR picture is represented is smaller or equal to the colour space in which the HDR picture is represented.

In the present document, when **sdrPicColourSpace** is less than **hdrPicColourSpace**, an inverse gamut mapping procedure should be invoked after the reconstruction process. An informative inverse gamut mapping process is provided in Annex D.

6.3.4.3 sdrDisplayMaxLuminance - SDR mastering display maximum luminance

This variable specifies the nominal maximum display luminance of the mastering display used to grade the SDR picture in units of 1 candela per square metre with a rounding to the nearest multiple of 50 candelas per square metre.

6.3.4.4 sdrDisplayMinLuminance - SDR mastering display minimum luminance

This variable specifies the nominal minimum display luminance of the mastering display used to grade the SDR picture in units of 0,0001 candelas per square metre. **sdrDisplayMinLuminance** shall be less than **sdrDisplayMaxLuminance**. If the proper value of **sdrDisplayMinLuminance** is unknown, it is recommended that it is set to 0.

6.3.5 Luminance mapping variables

6.3.5.1 Introduction

The luminance mapping variables defined in this clause are representative of the inverse EOTF 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 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 [i.2]. In the present document, those variables are used in the HDR reconstruction process specified in clause 7.

6.3.5.2 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).

6.3.5.3 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).

6.3.5.4 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).

6.3.5.5 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).

6.3.5.6 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).

6.3.5.7 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].

6.3.5.8 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.

6.3.5.9 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 equal to 0, **tmOutputFineTuningY**[0] shall be equal to 0. Otherwise, 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 equal to 1, **tmOutputFineTuningY**[**tmOutputFineTuningNumVal** - 1] shall be equal to 1. Otherwise, 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.

6.3.6 Colour correction adjustment variables

6.3.6.1 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*. The colour correction curve is a required input of the 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 [i.2]. In the present document, those variables are used in the HDR reconstruction process specified in clause 7.

6.3.6.2 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.

6.3.6.3 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.

6.3.6.4 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 constant segment shall be inferred that maps input values ranging from 0 to **saturationGainX**[0], inclusive, to a target value of **saturationGainY**[0].

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

6.3.7 Luminance mapping table

6.3.7.1 Introduction

The variables defined in this clause are piece-wise linear pivot points representative of the inverse EOTF of the inverse luminance mapping curve, implemented by the look-up table *lutMapY*. This look-up table is taken as one input of the 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 present clause is based on metadata definition as specified in SMPTE ST 2094-30 [i.3]. In the present document, those variables are used in the HDR reconstruction process specified in clause 7.

6.3.7.2 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 65].

6.3.7.3 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 ÷ 8 192). 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.

6.3.7.4 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 $1 - \frac{1}{8\,192}$] and in multiples of (1 ÷ 8 192).

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 $1 - \frac{1}{8\,192}$, inclusive.

6.3.8 Colour correction table

6.3.8.1 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 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 present clause is based on metadata definition as specified in SMPTE ST 2094-30 [i.4]. In the present document, those variables are used in the HDR reconstruction process specified in clause 7.

6.3.8.2 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 65].

6.3.8.3 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 ÷ 2 048). 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.

6.3.8.4 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 $0{,}125 - \frac{1}{16\,384}$] and in multiples of (1 ÷ 16 384).

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 $0{,}125 - \frac{1}{16\,384}$ 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 0, inclusive.

6.3.9 Gamut mapping variables

6.3.9.1 Introduction

This clause specifies the semantics of the metadata used for the gamut mapping process (and its inverse process) documented informatively in Annex D.

The index *c* of an array of variables that are defined in this clause represents an indicator specified as follows: the index value *c* equal to 0 should correspond to the red primary, *c* equal to 1 should correspond to the magenta secondary, *c* equal to 2 should correspond to the blue primary, *c* equal to 3 should correspond to the cyan secondary, *c* equal to 4 should correspond to the green primary, *c* equal to 5 should correspond to the yellow secondary.

6.3.9.2 satMappingMode - Saturation mapping mode

This variable indicates the mode of chrominance (re)mapping (a.k.a. saturation expansion or compression) used by the gamut mapping process. The value of **satMappingMode** shall be in the bounded range of [0 to 2] and as defined in Table 17.

Table 17: Chroma (re)mapping (saturation mapping) mode

Value of satMappingMode	Definition
0	Saturation mapping disabled
1	Saturation mapping with global parameters
2	Saturation mapping defined for each primary and secondary colour
3	Reserved for future use

6.3.9.3 satGlobal1SegRatio

This variable specifies the WCG and SCG chrominance coordinates of the first inflection point of the three piece-wise linear expansion or compression curve. This variable shall be invoked only when **satMappingMode** is equal to 1. The value of **satGlobal1SegRatio** shall be in the bounded range [0 to (7 ÷ 8)] and in multiples of (1 ÷ 8).

6.3.9.4 satGlobal2SegRatioWCG

This variable specifies the WCG chrominance coordinates of the second inflection point of the three piece-wise linear expansion or compression curve. This variable shall be invoked only when **satMappingMode** is equal to 1. The value of **satGlobal2SegRatioWCG** shall be in the bounded range [(7 ÷ 64) to (56 ÷ 64)] and in multiples of (7 ÷ 64).

6.3.9.5 satGlobal2SegRatioSCG

This variable specifies the SCG chrominance coordinates of the second inflection point of the three piece-wise linear expansion or compression curve. This variable shall be invoked only when **satMappingMode** is equal to 1. The value of **satGlobal2SegRatioSCG** shall be in the bounded range [(7 ÷ 64) to (56 ÷ 64)] and in multiples of (7 ÷ 64).

6.3.9.6 sat1SegRatio

This array of six variables specifies the WCG and SCG chrominance coordinates of the first inflection point of the three piece-wise linear expansion or compression curve. This array shall be invoked only when **satMappingMode** is equal to 2. The value of **sat1SegRatio**[c] shall be in the bounded range [0 to (7 ÷ 8)] and in multiples of (1 ÷ 8).

6.3.9.7 sat2SegRatioWCG

This array of six variables specifies the WCG chrominance coordinates of the second inflection point of the three piece-wise linear expansion or compression curve. This array shall be invoked only when **satMappingMode** is equal to 2. The value of **sat2SegRatioWCG**[c] shall be in the bounded range [(7 ÷ 64) to (56 ÷ 64)] and in multiples of (7 ÷ 64).

6.3.9.8 sat2SegRatioSCG

This array of six variables specifies the SCG chrominance coordinates of the second inflection point of the three piece-wise linear expansion or compression curve. This array shall be invoked only when **satMappingMode** is equal to 2. The value of **sat2SegRatioSCG**[c] shall be in the bounded range [(7 ÷ 64) to (56 ÷ 64)] and in multiples of (7 ÷ 64).

6.3.9.9 lightnessMappingMode

This variable indicates the mode of lightness (re)mapping used by the gamut mapping process. The value of **lightnessMappingMode** shall be in the bounded range of [0 to 3] and as defined in Table 18.

Table 18: Lightness (re)mapping mode

Value of lightnessMappingMode	Definition
0	Lightness (re)mapping disabled
1	Lightness (re)mapping applied to each primary and secondary colour
2	Lightness (re)mapping applied to the warm colours
3	Lightness (re)mapping with weighting factor applied to each primary and secondary colour

6.3.9.10 lmWeightFactor

This array of six variables specifies the weight to be applied to each primary and secondary colour during the lightness (re)mappping process. This array shall be invoked only when **lightnessMappingMode** is equal to 3. The value of **lmWeightFactor**[c] shall be in the bounded range $[(-1 \div 4)$ to $(6 \div 4)]$ and in multiples of $(1 \div 4)$.

6.3.9.11 croppingModeSCG

This variable indicates the cropping mode of the SCG used by the gamut mapping process (or its inverse process). The value of **croppingModeSCG** shall be in the bounded range of [0 to 3] and as defined in Table 19.

Table 19: SCG cropping mode

Value of croppingModeSCG	Definition
0	SCG cropping disabled
1	SCG cropping applied to each primary and secondary colour
2	SCG cropping applied to the cold colours
3	SCG cropping with weighting factor applied to each primary and secondary colour

6.3.9.12 cmWeightFactor

This array of six variables specifies the weight to be applied to each primary and secondary colour during the SCG cropping process. This array shall be invoked only when **croppingModeSCG** is equal to 3. The value of **cmWeightFactor**[c] shall be in the bounded range $[0$ to $(63 \div 64)]$ and in multiples of $(9 \div 64)$.

6.3.9.13 cmCroppedLightnessMappingEnabledFlag

This flag indicates whether the lightness (re)mapping is applied on the cropped SCG or not. When **cmCroppedLightnessMappingEnabledFlag** is equal to 0, it indicates that the lightness (re)mapping is not applied on the cropped SCG. When **cmCroppedLightnessMappingEnabledFlag** is equal to 1, it indicates that the lightness (re)mapping is applied on the cropped SCG. This flag shall be invoked only when **croppingModeSCG** is greater than 0.

6.3.9.14 hueAdjMode

This variable indicates the mode of hue (re)mapping (a.k.a. hue adjustment) used by the gamut mapping process (or its inverse process). The value of **hueAdjMode** shall be in the bounded range of [0 to 3] and as defined in Table 20.

Table 20: Hue (re)mapping (hue adjustment) mode

Value of hueAdjMode	Definition
0	Hue adjustment disabled
1	Global linear hue adjustment method
2	Piece-wise hue adjustment with globally preserved area
3	Piece-wise hue adjustment with preservation of areas per primary and secondary colours

6.3.9.15 hueGlobalPreservationRatio

This variable indicates the global preservation percentage for the hue (re)mapping. This variable shall be invoked only when **hueAdjMode** is equal to 2. The value of **hueGlobalPreservationRatio** shall be in the bounded range [0 to (7 ÷ 8)] and in multiples of (1 ÷ 8).

6.3.9.16 huePreservationRatio

This array of six variables indicates the preservation ratio to be applied to each primary or secondary colour during the hue (re)mapping process. This array shall be invoked only when **hueAdjMode** is equal to 3. The value of **huePreservationRatio**[c] shall be in the bounded range [0 to (7÷ 8)] and in multiples of (1 ÷ 8).

6.3.9.17 hueAlignCorrectionPresentFlag

This flag indicates the presence of hue correction parameters for each primary and secondary colour. A value of **hueAlignCorrectionPresentFlag** equal to 0 indicates that the parameters are not present. A value of **hueAlignCorrectionPresentFlag** equal to 1 indicates that the parameters are present. This flag shall be invoked only when **hueAdjMode** is greater than 0.

6.3.9.18 hueAlignCorrection

This array of six indices indicates the hue correction angle associated to each primary or secondary colour during the hue (re)mapping process. This array shall be invoked only when **hueAlignCorrectionPresentFlag** is equal to 1 and **hueAdjMode** is greater than 0. The value of **hueAlignCorrection**[c] shall be in the integer bounded range [0 to 7].

6.3.9.19 chromAdjPresentFlag

This variable indicates the presence of chrominance adjustment mapping parameters for each primary and secondary colour during the hue (re)mapping process. This flag shall be invoked only when **hueAdjMode** is greater than 0. A value of **chromAdjPresentFlag** equal to 0 indicates that the parameters are not present. A value of **chromAdjPresentFlag** equal to 1 indicates that the parameters are present.

6.3.9.20 chromAdjParam

This array of six indices indicates values for tuning the chrominance adjustment parameters associated to each primary or secondary colour during the hue (re)mapping process. This array shall be invoked only when **chromAdjPresentFlag** is equal to 1 and **hueAdjMode** is greater than 0. The value of **chromAdjParam**[c] shall be in the bounded range [(15 ÷ 16) to (18 ÷ 16)] and in multiples of (1 ÷ 16).

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, both payload carriage modes are carried by the SL-HDR Information SEI message specified in the present document, that is a User Data Registered 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 [i.5]). 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 [i.5] or Recommendation ITU-R BT.2100 [i.6]). This process assumes that the SDR picture signal is represented with a bit depth of 10-bit per component. For SDR picture represented with a different bit depth, 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. An invertible gamut mapping process is documented in Annex D, which parameters can be carried when **gamutMappingMode** is equal to 1 and when **sdrPicColourSpace** is less than **hdrPicColourSpace**.

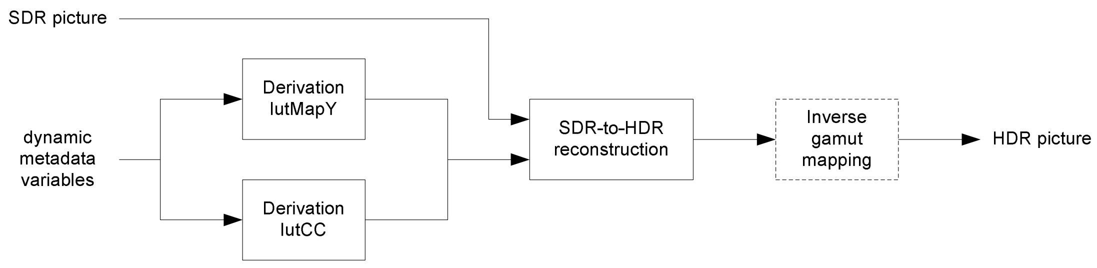

Figure 3: Overview of the HDR reconstruction process

In the next clauses, the variables *picWidth*, *picHeight* and *maxSampleVal* 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.

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 **hdrDisplayMaxLuminance** (clause 6.3.3.4);
- the luminance mapping variables **tmInputSignalBlackLevelOffset**, **tmInputSignalWhiteLevelOffset**, **shadowGain**, **highlightGain**, **midToneWidthAdjFactor**, **tmOutputFineTuningNumVal**, **tmOutputFineTuningX**[i] and **tmOutputFineTuningY**[i] (clause 6.3.5).

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 luminance), 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 **hdrDisplayMaxLuminance** 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_{glim} and the variable **hdrDisplayMaxLuminance** 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.

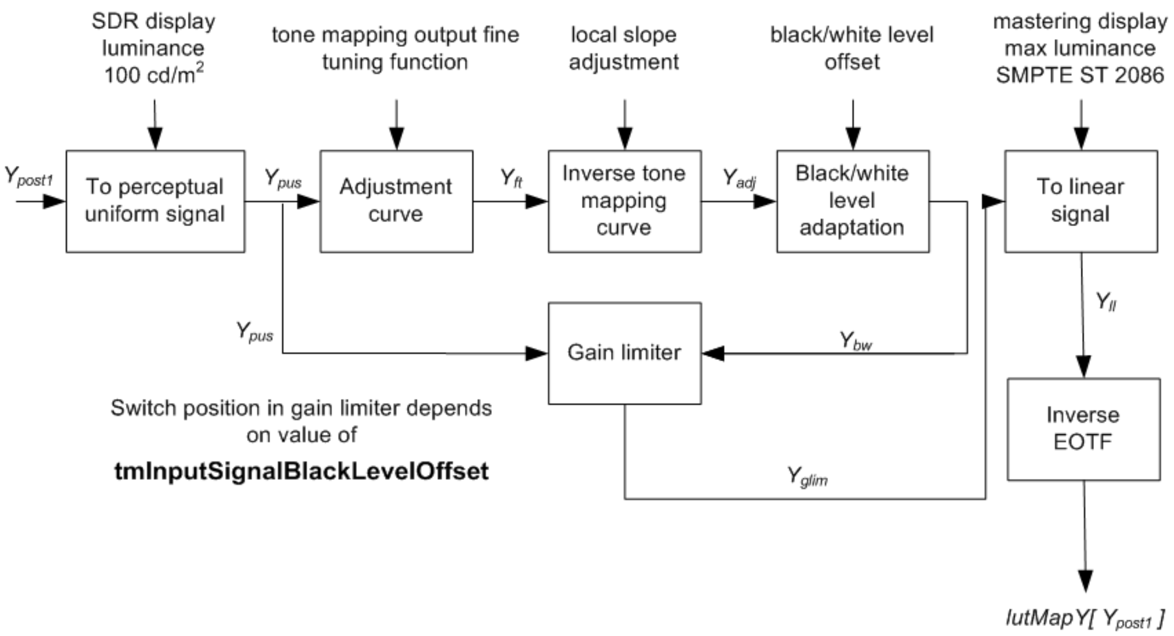

Figure 4: Inverse tone mapping process

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 input:

- the luma value Y_{post1}.

The process generates as output:

- the perceptual uniform value Y_{pus}.

In the first step, Y_{post1}, which is a Recommendation ITU-R BT.1886 [8] compatible luma signal, shall be taken to the power γ= 2,4 to yield the linear-light signal Y_2, see equation (1).

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

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 γ = 2,4, in order to get the perceptually uniform signal Y_{pus}, see equations (2) up to and including (4).

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

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

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

7.2.3.1.4 Block "Adjustment curve"

This process takes as inputs:

- the perceptual uniform value Y_{pus};
- the variables **tmOutputFineTuningNumVal**, **tmOutputFineTuningX**[i] and **tmOutputFineTuningY**[i], for i=0..(**tmOutputFineTuningNumVal** - 1) (clause 6.2.5).

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}, see equation (5).

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 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 6.3.5.9.

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

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 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** (clause 6.2.5);
- the variable **hdrDisplayMaxLuminance** (clause 6.3.3.4).

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 equations (6) up to and including (14).

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

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 in figure 5.

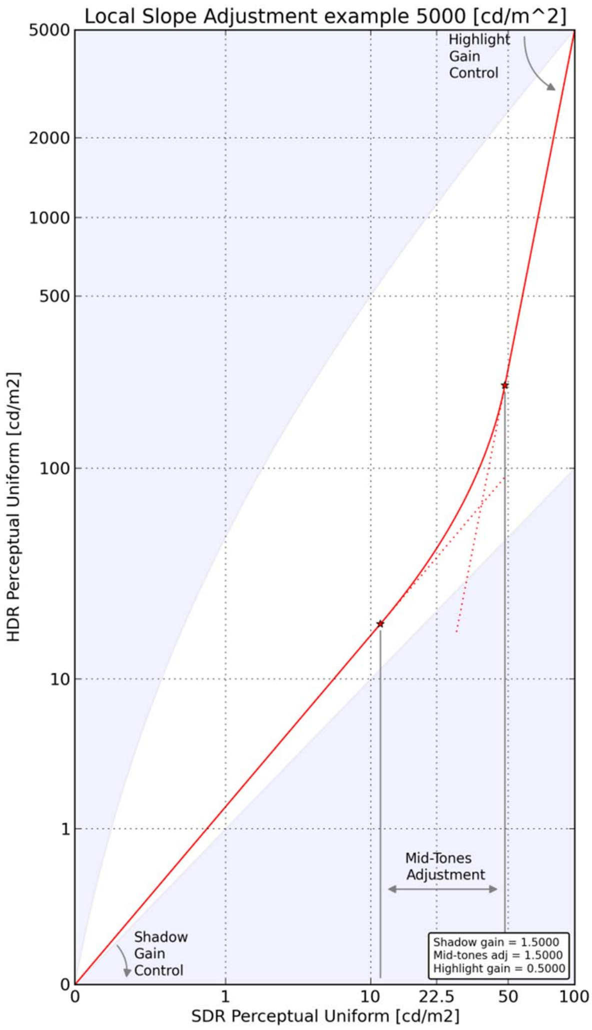

Figure 5: Inverse tone mapping curve shape

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 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 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 reconstruction metadata (clause 6.2.5).

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

NOTE 1: Due to the limitation on **shadowGain**, see clause 6.3.5, $SGC > 0,5$ for $L_{HDR} \geq 100$ 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 6.3.5, $HGC \leq 0{,}5$ and $SGC > 0{,}5$ for $L_{HDR} \geq 100$ 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 **hdrDisplayMaxLuminance** in the structure hdr_characteristics( ) of the reconstruction metadata (clause 6.3.3.4).

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

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

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

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}, see equation (15).

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

where:

- blo shall be equal to the variable **tmInputSignalBlackLevelOffset** in the structure luminance_mapping_variables( ) of the reconstruction metadata (clause 6.2.5); and
- wlo shall be equal to the variable **tmInputSignalWhiteLevelOffset** in the structure luminance_mapping_variables( ) of the 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** in the structure luminance_mapping_variables( ) of the reconstruction metadata (clause 6.2.5);
- the variable **hdrDisplayMaxLuminance** in the structure hdr_characteristics( ) of the reconstruction metadata (clause 6.3.3.4).

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 **hdrDisplayMaxLuminance** in the structure hdr_characteristics( ) of the reconstruction metadata (clause 6.3.3.4), 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, see equations (16) and (17).

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

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

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 **hdrDisplayMaxLuminance** (clause 6.3.3.4).

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 **hdrDisplayMaxLuminance** in the structure hdr_characteristics( ) of the reconstruction metadata (clause 6.3.3.4), see equations (18) and (19).

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

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

7.2.3.1.9 Block "Inverse EOTF"

This process takes as inputs:

- the linear-light value Y_{ll} ;

the variables **kCoefficient**[i] (clause 6.2.2).

The process generates as output:

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

In this block, an inverse EOTF is applied to the value Y_{ll} in order to compute the value of $lutMapY[\, Y_{post1} \,]$, see equation (20).

$$lutMapY\left[\, Y_{post1} \,\right] = {Y_{ll}}^{\frac{1}{\gamma}} \tag{20}$$

where:

- If **kCoefficient**[i] = 0 for all $i \in [\![0, 2]\!]$, $\gamma = 2{,}4$
- Otherwise, $\gamma = 2{,}0 + 0{,}4 \times (1 - modFactor)$

  where:

  - $modFactor = 1$, if the (optional) display adaptation of clause E.2 is not used;
  - $modFactor$ is determined by equation (E.20), if the display adaptation of clause E.2 is used.

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] (clause 6.2.6).

The process generates as output:

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

$$lutCC[0] = 0{,}125 \tag{21}$$

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

$$lutCC[Y] = Min\left(lutCC[0]\,;\, \frac{1}{Max\left(R_{sgf} \div 255\,;\, R_{sgf} \times g(Y_n)\right)} \times L(Y_n) \right) \tag{22}$$

where:

- $Y_n = \frac{Y}{maxSampleVal - 1}$
- $L(x) = \frac{1}{(maxSampleVal - 1) \times x}$;
- $g(Y_n) = f_{sgf}(Y_n) \times modFactor + \ (1 - modFactor) \div R_{sgf}$;

  where:

  - $modFactor = 1$, if the (optional) display adaptation of clause E.2 is not used;
  - $modFactor$ is determined by equation (E.20), if the display adaptation of clause E.2 is used.

- 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. When **saturationGainNumVal** is equal to 0, $f_{sgf}(\,) = \ 1 \div R_{sgf}$.
- R_{sgf} = 2 in the present document.

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 6.2.7.

This process takes as inputs:

- the luminance mapping table variables **luminanceMappingNumVal**, **luminanceMappingX**[i] and **luminanceMappingY**[i].

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:

$$lutMapY[\,Y\,] = f_{luma}\left(\frac{Y}{maxSampleVal-1}\right) \tag{23}$$

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 6.2.8.

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:

$$lutCC[\,Y\,] = f_{chroma}\left(\frac{Y}{maxSampleVal-1}\right) \tag{24}$$

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) bit depth conversion to 10 bits per component;
- the luminance mapping table $lutMapY$ of $maxSampleVal$ entries;
- the colour correction table $lutCC$ of $maxSampleVal$ entries;
- the four matrix coefficients variables **matrixCoefficient**[i] (clause 6.3.2.6);
- the two luma injection variables **chromaToLumaInjection**[i] (clause 6.3.2.7);

- the three "k" coefficients variables **kCoefficient**[i] (clause 6.3.2.8);
- the HDR picture mastering display max luminance **hdrDisplayMaxLuminance** (clause 6.3.3.4).

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:

$$\begin{cases} U_{post1} = SDR_{cb}[x][y] - midSampleVal \\ V_{post1} = SDR_{cr}[x][y] - midSampleVal \end{cases} \quad (25)$$

  where $midSampleVal$ is equal to $2^9 = 512$.

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

$$Y_{post1} = SDR_y[x][y] + Max(\ 0\ ; mu_0 \times U_{post1} + mu_1 \times V_{post1}\ ) \quad (26)$$

where:

  - mu_0 = **chromaToLumaInjection**[0] × $modFactor$;
  - mu_1 = **chromaToLumaInjection**[1] × $modFactor$;

  with:

    - $modFactor = 1$, if the (optional) display adaptation of clause E.2 is not used;
    - $modFactor$ is determined by equation (E.20), if the display adaptation of clause E.2 is used;

- the variable Y_{post2} is derived from Y_{post1} as follows:

$$Y_{post2} = Clip3(0\ ; maxSampleVal - 1\ ; Y_{post1}) \quad (27)$$

- U_{post2} and V_{post2} are derived from U_{post1} and V_{post1} as follows:

$$\begin{cases} U_{post2} = lutCC[Y_{post2}] \times U_{post1} \\ V_{post2} = lutCC[Y_{post2}] \times V_{post1} \end{cases} \quad (28)$$

- the variables R_1, G_1, $B1$ are derived as follows:

  - the variable T is computed as follows:

$$T = k_0 \times U_{post2} \times V_{post2} + k_1 \times U_{post2} \times U_{post2} + k_2 \times V_{post2} \times V_{post2} \quad (29)$$

    where k_i = **kCoefficient**[i] × $modFactor$ and $i \in [\![0, 2]\!]$

    with:

    - $modFactor = 1$, if the (optional) display adaptation of clause E.2 is not used;
    - $modFactor$ is determined by equation (E.20), if the display adaptation of clause E.2 is used;

  - the variable S_0 is initialized to 0, and the following applies:

    - If ($T \leq 1$), $S_0 = \sqrt{1 - T}$, $U_{post3} = U_{post2}$ and $V_{post3} = V_{post2}$

- Otherwise ($T > 1$), U_{post3} and V_{post3} are derived from U_{post2} and V_{post2} as follows:

$$\begin{cases} U_{post3} = \frac{U_{post2}}{\sqrt{T}} \\ V_{post3} = \frac{V_{post2}}{\sqrt{T}} \end{cases} \quad (30)$$

- R_1, G_1, B_1 are derived as follows:

$$\begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & m_0 \\ 1 & m_1 & m_2 \\ 1 & m_3 & 0 \end{bmatrix} \times \begin{bmatrix} S_0 \\ U_{post3} \\ V_{post3} \end{bmatrix} \quad (31)$$

where $m_i =$ **matrixCoefficient**[i] and $i \in [\![0, 3]\!]$

- the variables R_2, G_2, B_2 are derived from R_1, G_1, B_1 as follows:

$$\begin{cases} R_2 = lutMapY[Y_{post2}] \times R_1) \\ G_2 = lutMapY[Y_{post2}] \times G_1) \\ B_2 = lutMapY[Y_{post2}] \times B_1) \end{cases} \quad (32)$$

- the output samples $HDR_R[x][y]$, $HDR_G[x][y]$, $HDR_B[x][y]$ are derived from R_2, G_2, B_2 as follows:

$$\begin{cases} HDR_R[x][y] = L_{HDR} \times R_2^{\gamma} \\ HDR_G[x][y] = L_{HDR} \times G_2^{\gamma} \\ HDR_B[x][y] = L_{HDR} \times B_2^{\gamma} \end{cases} \quad (33)$$

where:

- $\gamma = 2{,}4$ if **kCoefficient**[i] = 0 for all $i \in [\![0, 2]\!]$;
- $\gamma = 2{,}0 + 0{,}4 \times (1 - modFactor)$ otherwise, with:
  - $modFactor = 1$, if the (optional) display adaptation of clause E.2 is not used;
  - $modFactor$ is determined by equation (E.20), if the display adaptation of clause E.2 is used;
- L_{HDR} is equal to **hdrDisplayMaxLuminance**.

7.3 Piecewise linear function computation

A piecewise linear function $f(x)$ is specified by a list of { x_i, y_i } pairs. $f(x)$ shall be defined on the domain [0 to 1]. $f(x)$ 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 processing of $y = f(x)$ shall be according to equation (34):

$$y = y[i] + (y[i+1] - y[i]) \times \frac{x - x[i]}{(x[i+1] - x[i])} \quad (34)$$

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): SL-HDR reconstruction metadata using HEVC

A.1 Introduction

This annex specifies the format of the SEI message that carries the SL-HDR reconstruction metadata for HEVC specification [4] as well as the mapping between the syntax elements of this SEI message and the dynamic metadata variables provided in clause 6. Clause A.2 specifies the SL-HDR Information SEI message that is a user data registered SEI message. It also specifies the mapping of the syntax elements values of the SL-HDRI SEI message into the dynamic metadata variables values specified in clause 6. Clause A.3 specifies the mapping between SL-HDRI SEI message syntax elements values and the Mastering Display Colour Volume SEI message (specified in HEVC) syntax elements values.

An HEVC bitstream conforming to this annex shall contain an SL-HDR Information SEI message (specified in clause A.2) in the first access unit of the CLVS. It is recommended that an HEVC bitstream conforming to Annex A contains an SL-HDR Information SEI message at least at each random access point of the bitstreams.

An HEVC bitstream conforming to Annex A may contain a Mastering Display Colour Volume SEI message (specified in HEVC specification [4]) for at least the first access unit of the CLVS.

A.2 SL-HDR SEI message definition and mapping

A.2.1 Introduction

Clause A.2.2 specifies the format of the HEVC SEI message that carries the specification version, the characteristics of the SDR and HDR signals, the payload mode and the related dynamic metadata. Clause A.2.3 specifies the mapping between the SL-HDR metadata carried through a user data registered SEI message and the SL-HDR metadata specified in clause 6 and invoked in the reconstruction process (clause 7).

A.2.2 SL-HDR information SEI message

A.2.2.1 Introduction

The SL-HDR information SEI message specified in clause A.2.2 applies to ETSI TS 103 433-1 (the present document), ETSI TS 103 433-2 [i.13] and ETSI TS 103 433-3 [i.12] documents. SL-HDR 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]. Clause A.2.2.2 specifies the SL-HDR information SEI message syntax. Clause A.2.2.3 specifies the syntax of the portion of the SL-HDR information SEI message that relates to the gamut mapping process. Clauses A.2.2.4 and A.2.2.5 respectively specifies the semantics of the SL-HDR information SEI message and the portion of this SEI message dedicated to the gamut mapping process.

A.2.2.2 SL-HDR information SEI message syntax

The SL-HDR information SEI message is an HEVC "User data registered by Recommendation ITU-T T.35 [i.14]" SEI message as specified in Table A.1.

Table A.1: sl_hdr_info SEI message syntax

Syntax	Descriptor
sl_hdr_info(payloadSize) {	
itu_t_t35_country_code	b(8)
terminal_provider_code	u(16)
terminal_provider_oriented_code_message_idc	u(8)
sl_hdr_mode_value_minus1	u(4)
sl_hdr_spec_major_version_idc	u(4)
sl_hdr_spec_minor_version_idc	u(7)
sl_hdr_cancel_flag	u(1)
if(!sl_hdr_cancel_flag) {	
sl_hdr_persistence_flag	u(1)
coded_picture_info_present_flag	u(1)
target_picture_info_present_flag	u(1)
src_mdcv_info_present_flag	u(1)
sl_hdr_extension_present_flag	u(1)
sl_hdr_payload_mode	u(3)
if(coded_picture_info_present_flag) {	
coded_picture_primaries	u(8)
coded_picture_max_luminance	u(16)
coded_picture_min_luminance	u(16)
}	
if(target_picture_info_present_flag) {	
target_picture_primaries	u(8)
target_picture_max_luminance	u(16)
target_picture_min_luminance	u(16)
}	
if(src_mdcv_info_present_flag) {	
for(c = 0; c < 3; c++) {	
src_mdcv_primaries_x[c]	u(16)
src_mdcv_primaries_y[c]	u(16)
}	
src_mdcv_ref_white_x	u(16)
src_mdcv_ref_white_y	u(16)
src_mdcv_max_mastering_luminance	u(16)
src_mdcv_min_mastering_luminance	u(16)
}	
for(i = 0; i < 4; i++)	
matrix_coefficient_value[i]	u(16)
for(i = 0; i < 2; i++)	
chroma_to_luma_injection[i]	u(16)
for(i = 0; i < 3; i++)	
k_coefficient_value[i]	u(8)
if(sl_hdr_payload_mode = = 0) {	
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)
saturation_gain_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)
}	
for(i = 0; i < saturation_gain_num_val; i++) {	
saturation_gain_x[i]	u(8)
saturation_gain_y[i]	u(8)
}	
}	
else if(sl_hdr_payload_mode = = 1) {	
lm_uniform_sampling_flag	u(1)
luminance_mapping_num_val	u(7)
for(i = 0; i < luminance_mapping_num_val; i++) {	

Syntax	Descriptor
if(!lm_uniform_sampling_flag)	
luminance_mapping_x[i]	u(16)
luminance_mapping_y[i]	u(16)
}	
cc_uniform_sampling_flag	u(1)
colour_correction_num_val	u(7)
for(i = 0; i < colour_correction_num_val; i++) {	
if(!cc_uniform_sampling_flag)	
colour_correction_x[i]	u(16)
colour_correction_y[i]	u(16)
}	
}	
if(GamutMappingEnabledFlag) {	
gamut_mapping_mode	u(8)
if (gamut_mapping_mode = = 1)	
gamut_mapping_params( )	
}	
if(sl_hdr_extension_present_flag) {	
sl_hdr_extension_6bits	u(6)
sl_hdr_extension_length	u(10)
for(i = 0; i< sl_hdr_extension_length; i++)	
sl_hdr_extension_data_byte[i]	u(8)
}	
}	
}	

A.2.2.3 Gamut mapping syntax

The portion of the HEVC SL-HDR information SEI message related to the invertible gamut mapping documented in Annex D is specified in Table A.2.

Table A.2: gamut_mapping_params( ) syntax

Syntax	Descriptor
gamut_mapping_params( ) {	
sat_mapping_mode	u(2)
if(sat_mapping_mode = = 1) {	
sat_global_1seg_ratio	u(3)
sat_global_2seg_ratio_wcg	u(3)
sat_global_2seg_ratio_scg	u(3)
}	
else if(sat_mapping_mode = = 2) {	
for(c = 0; c < 6; c++) {	
sat_1seg_ratio[c]	u(3)
sat_2seg_ratio_wcg[c]	u(3)
sat_2seg_ratio_scg[c]	u(3)
}	
}	
lightness_mapping_mode	u(2)
if(lightness_mapping_mode = = 3)	
for(c = 0; c < 6; c++)	
lm_weight_factor[c]	u(3)
cropping_mode_scg	u(2)
if(cropping_mode_scg = = 3)	
for(c =0; c < 6; c++)	
cm_weight_factor[c]	u(3)
if(cropping_mode_scg != 0)	
cm_cropped_lm_enabled_flag	u(1)
hue_adjustment_mode	u(2)
if(hue_adjustment_mode = = 2)	
hue_global_preservation_ratio	u(3)
if(hue_adjustment_mode = = 3)	

Syntax	Descriptor
for(c = 0; c < 6; c++)	
hue_preservation_ratio[c]	u(3)
if(hue_adjustment_mode != 0) {	
hue_adjustment_correction_info_present_flag	u(1)
if(hue_adjustment_correction_info_present_flag)	
for(c = 0; c < 6; c++)	
hue_alignment_correction[c]	u(3)
chrom_adjustment_info_present_flag	u(1)
if(chrom_adjustment_info_present_flag)	
for(c = 0; c < 6; c++)	
chrom_adjustment_param[c]	u(2)
}	
}	

A.2.2.4 SL-HDR SEI message semantics

This SEI message provides information to identify the specification version number, the characteristics of the source, decomposed and reconstructed signals, the payload mode and the dynamic metadata used by the SL-HDR systems.

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

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 0b00000000 (0x00).

sl_hdr_mode_value_minus1 plus 1 specifies the ETSI TS 103 433 part the SL-HDR information SEI message payload applies to. The value of sl_hdr_mode_value_minus1 shall be in the range of 0 to 2, inclusive. In bitstreams conforming to the present document, sl_hdr_mode_value_minus1 shall be equal to 0.

sl_hdr_spec_major_version_idc contains an identifying number that is used to identify the specification version number to which the associated bitstream conforms to. The value of sl_hdr_spec_major_version_idc shall be in the range of 0 to 15, inclusive. In bitstreams conforming to the present document, sl_hdr_spec_major_version_idc shall be equal to 1.

sl_hdr_spec_minor_version_idc contains an identifying number that is used to identify the specification version number to which the associated bitstream conforms to. The value of sl_hdr_spec_minor_version_idc shall be in the range of 0 to 127, inclusive. In bitstreams conforming to the present document, sl_hdr_spec_minor_version_idc shall be equal to 0.

sl_hdr_cancel_flag equal to 1 indicates that the SL-HDR information SEI message cancels the persistence of any previous SL-HDR information SEI message in output order with the same value of sl_hdr_mode_value_minus1 that applies to the current layer. sl_hdr_cancel_flag equal to 0 indicates that SL-HDR information follows.

sl_hdr_persistence_flag specifies the persistence of the SL-HDR information SEI message for the current layer. sl_hdr_persistence_flag equal to 0 specifies that the SL-HDR information applies to the current picture only.

Let picA be the current picture. sl_hdr_persistence_flag equal to 1 specifies that the SL-HDR 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 SL-HDR information SEI message with the same value of sl_hdr_mode_value_minus1 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.

coded_picture_info_present_flag equal to 1 specifies that coded_picture_primaries, coded_picture_max_luminance and coded_picture_min_luminance are present. coded_picture_info_present_flag equal to 0 specifies that coded_picture_primaries, coded_picture_max_luminance and coded_picture_min_luminance are not present. coded_picture_info_present_flag value shall be equal to 0 in bitstreams conforming to the present document.

NOTE 1: coded_picture_info_present_flag and associated syntax elements provide information on the picture that is coded in the bitstream. It is expected that the picture that is coded corresponds to the picture that was graded on a mastering display during the post-production stage.

target_picture_info_present_flag equal to 1 specifies that target_picture_primaries, target_picture_max_luminance and target_picture_min_luminance are present. target_picture_info_present_flag equal to 0 specifies that target_picture_primaries, target_picture_max_luminance and target_picture_min_luminance are not present.

NOTE 2: target_picture_info_present_flag and associated syntax elements provide information on the SDR picture that was graded on the SDR mastering display during the post-production stage.

src_mdcv_info_present_flag equal to 1 specifies that src_mdcv_primaries_x[c], src_mdcv_primaries_y[c], src_mdcv_ref_white_x, src_mdcv_ref_white_y, src_mdcv_max_mastering_luminance and src_mdcv_min_mastering_luminance are present. src_mdcv_info_present_flag equal to 0 specifies that src_mdcv_primaries_x[c], src_mdcv_primaries_y[c], src_mdcv_ref_white_x, src_mdcv_ref_white_y, src_mdcv_max_mastering_luminance and src_mdcv_min_mastering_luminance are not present.

NOTE 3: The actual intent for providing src_mdcv_info_present_flag and the associated syntax elements in the bitstream is to assist SL-HDR decoding systems to properly interpret the colour volume of the mastering display used to grade the HDR source picture when a Mastering Display Colour Volume SEI message is already used by a bitstream that was not generated by SL-HDR encoding systems. It is recommended that SL-HDR encoding systems use a Mastering Display Colour Volume SEI message (rather than enabling src_mdcv_info_present_flag and the associated syntax elements) for representing the colour volume of the mastering display of the HDR source picture.

When a Mastering Display Colour Volume SEI message and a SL-HDR Information SEI message with src_mdcv_info_present_flag equal to 1 are both present in the same CLVS, SL-HDR decoding systems shall use the value of the syntax elements enabled by src_mdcv_info_present_flag during the reconstruction process.

When an MDCV SEI message is not present with an SL-HDR Information SEI message in the same CLVS, src_mdcv_info_present_flag shall be equal to 1.

When src_mdcv_flag is equal to 0, an MDCV SEI message shall be present in the CLVS.

sl_hdr_extension_present_flag equal to 1 specifies that sl_hdr_extension_6bits, sl_hdr_extension_length and sl_hdr_extension_data_byte[i] are present. sl_hdr_extension_present_flag equal to 0 specifies that sl_hdr_extension_6bits, sl_hdr_extension_length and sl_hdr_extension_data_byte[i] are not present. sl_hdr_extension_present_flag shall be equal to 0 in bitstreams conforming to the present document.

sl_hdr_payload_mode contains an identifying number that is used to identify the payload mode (as described in clause 6.3.2.5) to which the associated bitstream conforms to. The value of sl_hdr_payload_mode shall be in the range of 0 to 1, inclusive.

NOTE 4: Parameter-based mode is indicated by sl_hdr_payload_mode value equal to 0. Table-based mode is indicated by sl_hdr_payload_mode value equal to 1.

coded_picture_primaries has the same semantics as specified in clause E.3.1 of HEVC specification [4] for the colour_primaries syntax element. The value of coded_picture_primaries shall be equal to 1 (Rec. 709 primaries) or 9 (Rec. 2020 primaries). coded_picture_primaries identifies the colour space in which the coded picture is represented. Decoders that comply with the present document shall ignore the value of coded_picture_primaries.

coded_picture_max_luminance has the same semantics as specified in clause D.3.28 of HEVC specification [4] for the max_display_mastering_luminance syntax element, except that the value is coded in units of 1 candela per square metre. coded_picture_max_luminance applies to the coded picture. Decoders that comply with the present document shall ignore the value of coded_picture_max_luminance.

coded_picture_min_luminance has the same semantics as specified in clause D.3.28 of HEVC specification [4] for the min_display_mastering_luminance syntax element i.e. the value is coded in units of 0,000 1 candelas per square metre. coded_picture_min_luminance applies to the coded picture. Decoders that comply with this version of the present document shall ignore the value of coded_picture_min_luminance.

target_picture_primaries has the same semantics as specified in clause E.3.1 of HEVC specification [4] for the colour_primaries syntax element, except that target_picture_primaries identifies the colour space of the target picture rather than the colour space used for the CLVS. The value of target_picture_primaries shall be equal to 1 (Rec. 709 primaries) or 9 (Rec. 2020 primaries).

target_picture_max_luminance has the same semantics as specified in clause D.3.28 of HEVC specification [4] for the max_display_mastering_luminance syntax element, except that the value is coded in units of 1 candela per square metre. target_picture_max_luminance applies to the target picture. When target_picture_max_luminance is not present, it is inferred to be equal to 100. In bitstreams conforming to the present document, target_picture_max_luminance value shall be equal to 100.

target_picture_min_luminance has the same semantics as specified in clause D.3.28 of HEVC specification [4] for the min_display_mastering_luminance syntax element i.e. the value is coded in units of 0,000 1 candelas per square metre. target_picture_min_luminance applies to the target picture. When target_picture_min_luminance is not present, it is inferred to be equal to 0. In bitstreams conforming to the present document, target_picture_min_luminance value shall be equal to 0.

src_mdcv_primaries_x[i] and **src_mdcv_primaries_y**[i] have the same semantics as specified in clause D.3.28 of HEVC specification [4] for the display_primaries_x[c] and display_primaries_y[c] syntax elements i.e. the value is coded in normalized increments of 0,000 02, according to CIE 1931 definition of x and y. It applies to the associated source picture.

src_mdcv_ref_white_x and **src_mdcv_ref_white_y** have the same semantics as specified in clause D.3.28 of HEVC specification [4] for the white_point_x and white_point_y syntax elements i.e. the value is coded in normalized increments of 0,000 02, according to CIE 1931 definition of x and y. It applies to the associated source picture.

src_mdcv_max_mastering_luminance has the same semantics as specified in clause D.3.28 of HEVC specification [4] for the max_display_mastering_luminance syntax element, except that the value is coded in units of 1 candela per square metre. It applies to the associated source picture.

src_mdcv_min_mastering_luminance has the same semantics as specified in clause D.3.28 of HEVC specification [4] for the min_display_mastering_luminance syntax element, i.e. the value is coded in units of 0,000 1 candelas per square metre. It applies to the associated source picture.

NOTE 5: src_mdcv_max_mastering_luminance and src_mdcv_min_mastering_luminance semantics are aligned with CTA-861.3 [i.1] definitions of max_display_mastering_luminance and min_display_mastering_luminance, respectively.

matrix_coefficient_value[i] specifies the value of the i-th coefficient used to compute the Y'CC to R'G'B' conversion matrix in the SL-HDR reconstruction process. The value of matrix_coefficient_value[i] shall be in the range of 0 to 1023, inclusive.

chroma_to_luma_injection[i] indicates the ratio of the blue and red colour difference components injection into the luma component with i set to 0 or 1, respectively. The value of chroma_to_luma_injection[i] shall be in the range of 0 to 8 191, inclusive.

k_coefficient_value[i] specifies the value of the i-th coefficient used to compute the variable T (see clause 7.2.4) in the SL-HDR reconstruction process. The value of k_coefficient_value[0] shall be in the range of 0 to 63, inclusive. The value of k_coefficient_value[1] shall be in the range of 0 to 127, inclusive. The value of k_coefficient_value[2] shall be in the range of 0 to 255, inclusive.

tone_mapping_input_signal_black_level_offset indicates the black level offset to be subtracted during the SL-HDR 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 SL-HDR the 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 indicates 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. The value of tone_mapping_output_fine_tuning_num_val shall be in the range of 0 to 10, inclusive.

saturation_gain_num_val indicates 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. The value of saturation_gain_num_val shall be in the range of 0 to 6, inclusive.

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

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

lm_uniform_sampling_flag equal to 1 specifies that the x-coordinates of the pivot points representative of the luminance mapping piece-wise linear curve are distributed uniformly across the horizontal axis. lm_uniform_sampling_flag equal to 0 specifies that the x-coordinates of the pivot points representative of the luminance mapping piece-wise linear curve are specified by luminance_mapping_x[i].

luminance_mapping_num_val indicates the number of pivot points representative of the luminance mapping piece-wise linear curve. The value of luminance_mapping_num_val shall be in the range of 0 to 65, inclusive.

luminance_mapping_x[i] specifies the input value of the i-th pivot point for the luminance mapping piece-wise linear curve. The value of luminance_mapping_x[i] shall be in the range of 0 to 8 192, inclusive.

luminance_mapping_y[i] specifies the output value of the i-th pivot point for the luminance mapping piece-wise linear curve. The value of luminance_mapping_y[i] shall be in the range of 0 to 8 191, inclusive.

cc_uniform_sampling_flag equal to 1 specifies that the x-coordinates of the pivot points representative of the colour correction piece-wise linear curve are distributed uniformly across the horizontal axis. cc_uniform_sampling_flag equal to 0 specifies that the x-coordinates of the pivot points representative of the colour correction piece-wise linear curve are specified by colour_correction_x[i].

colour_correction_num_val indicates the number of pivot points representative of the colour correction piece-wise linear curve. The value of colour_correction_num_val shall be in the range of 0 to 65, inclusive.

colour_correction_x[i] specifies the input value of the i-th pivot point for the colour correction curve. The value of colour_correction_x[i] shall be in the range of 0 to 2 048, inclusive.

colour_correction_y[i] specifies the output value of the i-th pivot point for the colour correction curve. The value of colour_correction_y[i] shall be in the range of 0 to 2 047, inclusive.

GamutMappingEnabledFlag is specified in clause A.2.2.5.

gamut_mapping_mode equal to 0 specifies that there is no parameter representative of the gamut mapping process (or its inverse process) carried in the bitstream. gamut_mapping_mode equal to 1 specifies that parameters representative of the gamut mapping process (or its inverse process) are present in the bitstream (see clauses A.2.2.3 and A.2.2.5). In the present document, gamut_mapping_mode equal to 2 and 3 specifies predetermined values used by the inverse gamut mapping process (documented in Annex D) to respectively map the BT.709 gamut of the reconstructed HDR picture to P3D65 (preset #1) or BT.2020 gamut (preset #2). In the present document, the value of gamut_mapping_mode shall be in the range of 0 to 3, inclusive. Values of gamut_mapping_mode in the range of 64 to 127, inclusive are unspecified by the present document and may be used by some means unspecified in the present document to identify particular presets. Values of gamut_mapping_mode in the range of 8 to 63, inclusive and in the range of 128 to 255, inclusive are reserved for future use.

NOTE 6: In the present document, when gamut_mapping_mode is equal to 0, an inverse gamut mapping is recommended to be performed by the CE manufacturer after the HDR signal reconstruction process.

sl_hdr_extension_6bits, when present, shall be equal to 0 in bitstreams conforming to this version of the present document. Values of sl_hdr_extension_6bits not equal to 0 are reserved for future use. Decoders that comply with this version of the present document shall ignore the value of sl_hdr_extension_6bits.

sl_hdr_extension_length specifies the length of the SL-HDR extension data in bytes, not including the bits used for signalling sl_hdr_extension_length itself. The value of sl_hdr_extension_length shall be in the range of 0 to 1023, inclusive. When not present, the value of sl_hdr_extension_length is inferred to be equal to 0.

sl_hdr_extension_data_byte[i] may have any value. Decoders that comply with this version of the present document shall ignore the value of sl_hdr_extension_data_byte[i].

A.2.2.5 Gamut mapping semantics

The variable **GamutMappingEnabledFlag** is derived as follows:

GamutMappingEnabledFlag = 0
if(**sdrPicColourSpace** < **hdrPicColourSpace**)
    GamutMappingEnabledFlag = 1

The variables **sdrPicColourSpace** and **hdrPicColourSpace** are respectively derived in clauses A.2.3.4 and A.2.3.3.

NOTE 1: In the present document, an inverse gamut mapping should be enabled when the gamut in which the SDR picture is represented is smaller than the gamut in which the HDR picture is represented.

In this clause, the index value c equal to 0 should correspond to the red primary, c equal to 1 should correspond to the magenta secondary, c equal to 2 should correspond to the blue primary, c equal to 3 should correspond to the cyan secondary, c equal to 4 should correspond to the green primary, c equal to 5 should correspond to the yellow secondary.

sat_mapping_mode indicates the mode of the chroma (re)mapping (saturation expansion or compression) process used by the gamut mapping process (or its inverse process). The value of sat_mapping_mode shall be in the range of 0 to 2, inclusive.

NOTE 2: A value of 0 indicates that the chroma (re)mapping is not active. A value of 1 indicates that chroma (re)mapping is defined with global parameters. A value of 2 indicates that chroma (re)mapping is defined for each primary and secondary colour.

sat_global_1seg_ratio specifies the WCG and SCG chrominance coordinates of the first inflection point of the three piece-wise linear expansion or compression curve for the chroma (re)mapping process. The value of sat_global_1seg_ratio shall be in the range of 0 to 7, inclusive.

sat_global_2seg_ratio_wcg and **sat_global_2seg_ratio_scg** specify the WCG and SCG chrominance coordinates of the second inflection point of three piece-wise linear expansion or compression curve for the chroma (re)mapping process. The value of sat_global_2seg_ratio_wcg and sat_global_2seg_ratio_scg shall be in the range of 0 to 7, inclusive.

sat_1seg_ratio[c] specifies the c-th WCG and SCG chrominance coordinates of the first inflection point of the three piece-wise linear expansion or compression curve for the chroma (re)mapping process. The value of sat_1seg_ratio[c] shall be in the range of 0 to 7, inclusive.

sat_2seg_ratio_wcg[c] and **sat_2seg_ratio_scg**[c] specify the c-th WCG and SCG chrominance coordinates of the second inflection point of the three piece-wise linear expansion or compression curve for the chroma (re)mapping process. The value of sat_2seg_ratio_wcg[c] and sat_2seg_ratio_scg[c] shall be in the range of 0 to 7, inclusive.

lightness_mapping_mode indicates the mode of the lightness (re)mapping process used by the gamut mapping process (or its inverse process). The value of lightness_mapping_mode shall be in the range of 0 to 3, inclusive.

NOTE 3: A value of 0 indicates that the lightness (re)mapping is not active. A value of 1 indicates that lightness (re)mapping is applied to each primary and secondary colour. A value of 2 indicates that the lightness (re)mapping is applied to warm colours. A value of 3 indicates that the lightness (re)mapping is applied with weighting factors applied to each primary and secondary colour.

lm_weight_factor[c] specifies the c-th weight to be applied to each primary and secondary colour during the lightness (re)mapping process. The value of lm_weight_factor[c] shall be in the range of 0 to 7, inclusive.

cropping_mode_scg indicates the cropping mode of the SCG during the lightness cropped gamut derivation process. The value of cropping_mode_scg shall be in the range of 0 to 3, inclusive.

NOTE 4: A value of 0 indicates that the SCG cropping is not active. A value of 1 indicates that the SCG cropping is applied to each primary and secondary colour. A value of 2 indicates that the SCG cropping is applied to the cold colours. A value of 3 indicates that SCG cropping is applied with weighting factors applied to each primary and secondary colour.

cm_weight_factor[c] specifies the c-th weight to be applied to each primary and secondary colour during the lightness cropped gamut derivation process. The value of cm_weight_factor[c] shall be in the range of 0 to 7, inclusive.

cm_cropped_lm_enabled_flag indicates that the lightness (re)mapping shall be applied on the cropped SCG when its value is equal to 1. cm_cropped_lm_enabled_flag indicates that the lightness (re)mapping shall not be applied on the cropped SCG when its value is equal to 0.

hue_adjustment_mode indicates the mode of the hue (re)mapping (hue adjustment) process used by the gamut mapping process (or its inverse process). The value of hue_adjustment_mode shall be in the range of 0 to 3, inclusive.

NOTE 5: A value of 0 indicates that the hue (re)mapping is not active. A value of 1 indicates that the hue (re)mapping method is global linear. A value of 2 indicates that a piece-wise hue (re)mapping method is used with globally preserved area. A value of 3 indicates that a piece-wise hue (re)mapping method is used with preservation of areas per primary and secondary colours.

hue_global_preservation_ratio indicates the global preservation percentage for the hue (re)mapping process. The value of hue_global_preservation_ratio shall in the range of 0 to 7, inclusive.

hue_preservation_ratio[c] indicates the c-th preservation ratio to be applied to each primary or secondary colours during the hue (re)mapping process. The value of hue_preservation_ratio[c] shall be in the range of 0 to 7, inclusive.

hue_adjustment_correction_info_present_flag equal to 0 indicates that hue_alignment_correction[c] syntax elements are not present. hue_adjustment_correction_info_present_flag equal to 1 indicates that hue_alignment_correction[c] syntax elements are present.

hue_alignment_correction[c] indicates the c-th hue correction angle associated to each primary or secondary colour during the hue (re)mapping process. The value of hue_alignment_correction[c] shall be in the range of 0 to 7, inclusive.

chrom_adjustment_info_present_flag equal to 0 indicates that chrom_adjustment_param[c] syntax elements are not present. chrom_adjustment_info_present_flag equal to 1 indicates that chrom_adjustment_param[c] syntax elements are present.

chrom_adjustment_param[c] specifies the c-th value for tuning the chrominance adjustment parameters associated to each primary or secondary colour during the hue (re)mapping process. The value of chrom_adjustment_param[c] shall be in the range of 0 to 3, inclusive.

A.2.3 SL-HDR metadata mapping

A.2.3.1 Introduction

This clause A.2.3 provides information on mapping SL-HDR Information SEI message syntax elements values to the respective SL-HDR metadata values that are used in the reconstruction process (clause 7).

A.2.3.2 Signal reconstruction information

A.2.3.2.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the root of the signal reconstruction information structure.

A.2.3.2.2 partID mapping

$$\textbf{partID} = \textbf{sl\_hdr\_mode\_value\_minus1} + 1 \tag{A.1}$$

A.2.3.2.3 majorSpecVersionID mapping

$$\textbf{majorSpecVersionID} = \textbf{sl\_hdr\_spec\_major\_version\_idc} \tag{A.2}$$

A.2.3.2.4 minorSpecVersionID mapping

$$\textbf{minorSpecVersionID} = \textbf{sl\_hdr\_spec\_minor\_version\_idc} \tag{A.3}$$

A.2.3.2.5 payloadMode mapping

$$\textbf{payloadMode} = \textbf{sl\_hdr\_payload\_mode} \tag{A.4}$$

A.2.3.2.6 matrixCoefficient mapping

$$\textbf{matrixCoefficient}[\,i\,] = (\,\textbf{matrix\_coefficient\_value}[\,i\,] - 512\,) \div 256 \tag{A.5}$$

where $i \in [\![0, 3]\!]$

A.2.3.2.7 chromaToLumaInjection mapping

$$\textbf{chromaToLumaInjection}[\,i\,] = \textbf{chroma\_to\_luma\_injection}[\,i\,] \div 16\,384 \tag{A.6}$$

where $i \in [\![0, 1]\!]$

A.2.3.2.8 kCoefficient mapping

$$\textbf{kCoefficient}[\,i\,] = \textbf{k\_coefficient\_value}[\,i\,] \div 256 \tag{A.7}$$

where $i \in [\![0, 2]\!]$

A.2.3.2.9 gamutMappingMode mapping

$$\textbf{gamutMappingMode} = \textbf{gamut\_mapping\_mode} \tag{A.8}$$

A.2.3.3 HDR picture characteristics

A.2.3.3.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the HDR picture characteristics.

A.2.3.3.2 hdrPicColourSpace mapping

The value of **hdrPicColourSpace** shall be inferred from the value of **hdrDisplayColourSpace** and **sdrPicColourSpace** as specified in Table A.3.

Table A.3: Inferred value of hdrPicColourSpace

Value of hdrDisplayColourSpace	Value of sdrPicColourSpace	Inferred value of hdrPicColourSpace
0 (BT.709)	0 (BT.709)	0 (BT.709)
0 (BT.709)	1 (BT.2020)	1 (BT.2020)
1 (BT.2020)	0 (BT.709) or 1 (BT.2020)	1 (BT.2020)
2 (P3D65)	0 (BT.709) or 1 (BT.2020)	1 (BT.2020)

NOTE: When the HDR picture input in the SL-HDR1 pre-processor is represented with BT.2020 primaries and its mastering display colour volume information indicates BT.709 (i.e. **hdrDisplayColourSpace** is equal to 0 (BT.709)) and the SDR picture output of the SL-HDR1 pre-processor is represented with BT.709 primaries (i.e. **sdrPicColourSpace** is equal to 0 (BT.709)), an unspecified conversion of the HDR picture colour space is performed in the SL-HDR1 pre-processor from BT.2020 to BT.709 primaries (so that **hdrPicColourSpace** is equal to 0 (BT.709) after the colour space conversion in the SL-HDR1 pre-processor).

A.2.3.3.3 hdrDisplayColourSpace mapping

The mapping to **hdrDisplayColourSpace** variable depends on the value of **src_mdcv_info_present_flag**:

- When **src_mdcv_info_present_flag** is equal to 1, **hdrDisplayColourSpace** shall be mapped as specified in Table A.4.
- When **src_mdcv_info_present_flag** is equal to 0, **hdrDisplayColourSpace** shall be mapped as specified in clause A.3.2.

NOTE: MDCV SEI message should be the preferred carriage mechanism for carrying the colour primaries of the mastering display used to grade the HDR picture.

Table A.4: Mapping to hdrDisplayColourSpace

SEI syntax elements	SEI syntax elements value	hdrDisplayColourSpace value	NOTE: Colour space primaries
src_mdcv_primaries_x[0] **src_mdcv_primaries_y[0]** **src_mdcv_primaries_x[1]** **src_mdcv_primaries_y[1]** **src_mdcv_primaries_x[2]** **src_mdcv_primaries_y[2]** **src_mdcv_ref_white_x** **src_mdcv_ref_white_y**	15 000 30 000 7 500 3 000 32 000 16 500 15 635 16 450	0	Recommendation ITU-R BT.709-6 [6]
src_mdcv_primaries_x[0] **src_mdcv_primaries_y[0]** **src_mdcv_primaries_x[1]** **src_mdcv_primaries_y[1]** **src_mdcv_primaries_x[2]** **src_mdcv_primaries_y[2]** **src_mdcv_ref_white_x** **src_mdcv_ref_white_y**	8 500 39 850 6 550 2 300 35 400 14 600 15 635 16 450	1	Recommendation ITU-R BT.2020-2 [7]
src_mdcv_primaries_x[0] **src_mdcv_primaries_y[0]** **src_mdcv_primaries_x[1]** **src_mdcv_primaries_y[1]** **src_mdcv_primaries_x[2]** **src_mdcv_primaries_y[2]** **src_mdcv_ref_white_x** **src_mdcv_ref_white_y**	13 250 34 500 7 500 3 000 34 000 16 000 15 635 16 450	2	SMPTE RP 431-2 (DCI P3) [5]
NOTE: In case the 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 **hdrDisplayColourSpace** that is the closest match to the column 2 values.			

A.2.3.3.4 hdrDisplayMaxLuminance mapping

The mapping to **hdrDisplayMaxLuminance** variable depends on the value of **src_mdcv_info_present_flag**:

- When **src_mdcv_info_present_flag** is equal to 0, **hdrDisplayMaxLuminance** shall be mapped from the MDCV SEI message syntax elements as specified in clause A.3.2.
- When **src_mdcv_info_present_flag** is equal to 1, **hdrDisplayMaxLuminance** shall be mapped as specified below:

$$\textbf{hdrDisplayMaxLuminance} = Min(\,50 \times ((\,\textbf{src\_mdcv\_max\_mastering\_luminance} + 25\,)\,/\,50\,);\,10\,000\,) \qquad \text{(A.9)}$$

NOTE: Integer division with truncation of the result toward zero.

A.2.3.3.5 hdrDisplayMinLuminance mapping

The mapping to **hdrDisplayMinLuminance** variable depends on the value of **src_mdcv_info_present_flag**:

- When **src_mdcv_info_present_flag** is equal to 0, **hdrDisplayMinLuminance** shall be mapped from the MDCV SEI message syntax elements as specified in clause A.3.2.
- When **src_mdcv_info_present_flag** is equal to 1, **hdrDisplayMinLuminance** shall be mapped as specified below:

$$\textbf{hdrDisplayMinLuminance} = \textbf{src\_mdcv\_min\_mastering\_luminance} \times 0{,}000\,1 \qquad \text{(A.10)}$$

A.2.3.4 SDR picture characteristics

A.2.3.4.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the SDR picture characteristics.

A.2.3.4.2 sdrPicColourSpace mapping

When **target_picture_primaries** is not present, the value of **sdrPicColourSpace** shall be set to the same value as **hdrPicColourSpace**.

Otherwise, the following applies:

- When **target_picture_primaries** is equal to 1, **sdrPicColourSpace** shall be equal to 0.
- When **target_picture_primaries** is equal to 9, **sdrPicColourSpace** shall be equal to 1.

A.2.3.4.3 sdrDisplayMaxLuminance mapping

When **target_picture_max_luminance** is not present, the value of **sdrDisplayMaxLuminance** shall be set to 100.

Otherwise, the following applies:

$$\textbf{sdrDisplayMaxLuminance} = Min(\, 50 \times ((\, \textbf{target\_picture\_max\_luminance} + 25\,) \,/\, 50\,);\, 10\,000\,) \qquad \text{(A.11)}$$

NOTE: In the present document as **target_picture_max_luminance** is equal to 100, **sdrDisplayMaxLuminance** is equal to 100.

A.2.3.4.4 sdrDisplayMinLuminance mapping

When **target_picture_min_luminance** is not present, the value of **sdrDisplayMinLuminance** shall be set to 0.

Otherwise, the following applies:

$$\textbf{sdrDisplayMinLuminance} = \textbf{target\_picture\_min\_luminance} \times 0{,}000\,1 \qquad \text{(A.12)}$$

NOTE: In the present document as **target_picture_min_luminance** is equal to 0, **sdrDisplayMinLuminance** is equal to 0.

A.2.3.5 Luminance mapping variables

A.2.3.5.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the luminance mapping variables.

A.2.3.5.2 tmInputSignalBlackLevelOffset mapping

$$\textbf{tmInputSignalBlackLevelOffset} = \textbf{tone\_mapping\_input\_black\_level\_offset} \div 255 \qquad \text{(A.13)}$$

A.2.3.5.3 tmInputSignalWhiteLevelOffset mapping

$$\textbf{tmInputSignalWhiteLevelOffset} = \textbf{tone\_mapping\_input\_white\_level\_offset} \div 255 \qquad \text{(A.14)}$$

A.2.3.5.4 shadowGain mapping

$$\textbf{shadowGain} = \textbf{shadow\_gain\_control} \times 2 \div 255 \qquad \text{(A.15)}$$

A.2.3.5.5 highlightGain mapping

$$\mathbf{highlightGain} = \mathbf{highlight\_gain\_control} \times 2 \div 255 \quad \text{(A.16)}$$

A.2.3.5.6 midToneWidthAdjFactor mapping

$$\mathbf{midToneWidthAdjFactor} = \mathbf{mid\_tone\_width\_adjustment\_factor} \times 2 \div 255 \quad \text{(A.17)}$$

A.2.3.5.7 tmOutputFineTuningNumVal mapping

$$\mathbf{tmOutputFineTuningNumVal} = \mathbf{tone\_mapping\_output\_fine\_tuning\_num\_val} \quad \text{(A.18)}$$

A.2.3.5.8 tmOutputFineTuningX mapping

for $i \in [\![0, \mathbf{tmOutputFineTuningNumVal} - 1]\!]$

$$\mathbf{tmOutputFineTuningX}[\,i\,] = \mathbf{tone\_mapping\_output\_fine\_tuning\_x}[\,i\,] \div 255 \quad \text{(A.19)}$$

A.2.3.5.9 tmOutputFineTuningY mapping

for $i \in [\![0, \mathbf{tmOutputFineTuningNumVal} - 1]\!]$, the following applies:

$$\mathbf{tmOutputFineTuningY}[\,i\,] = \mathbf{tone\_mapping\_output\_fine\_tuning\_y}[\,i\,] \div 255 \quad \text{(A.20)}$$

A.2.3.6 Colour correction adjustment variables

A.2.3.6.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the colour correction adjustment variables.

A.2.3.6.2 saturationGainNumVal mapping

$$\mathbf{saturationGainNumVal} = \mathbf{saturation\_gain\_num\_val} \quad \text{(A.21)}$$

A.2.3.6.3 saturationGainX mapping

for $i \in [\![0, \mathbf{saturationGainNumVal} - 1]\!]$

$$\mathbf{saturationGainX}[\,i\,] = \mathbf{saturation\_gain\_x}[\,i\,] \div 255 \quad \text{(A.22)}$$

A.2.3.6.4 saturationGainY mapping

for $i \in [\![0, \mathbf{saturationGainNumVal} - 1]\!]$

$$\mathbf{saturationGainY}[\,i\,] = \mathbf{saturation\_gain\_y}[\,i\,] \div 255 \quad \text{(A.23)}$$

A.2.3.7 Luminance mapping table

A.2.3.7.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the luminance mapping table elements. Clauses A.2.3.7.3 and A.2.3.7.4 apply only when **luminance_mapping_num_val** is greater than 0.

A.2.3.7.2 luminanceMappingNumVal mapping

$$\mathbf{luminanceMappingNumVal} = \mathbf{luminance\_mapping\_num\_val} \quad \text{(A.24)}$$

A.2.3.7.3 luminanceMappingX mapping

When **lm_uniform_sampling_flag** is equal to 1, the following applies:

for $i \in [\![0, \mathbf{luminanceMappingNumVal} - 1]\!]$

$$\mathbf{luminanceMappingX}[\,i\,] = i \div (\mathbf{luminanceMappingNumVal} - 1) \quad \text{(A.25)}$$

Otherwise, (when **lm_uniform_sampling_flag** is equal to 0) the following applies:

for $i \in [\![0, \mathbf{luminanceMappingNumVal} - 1]\!]$

$$\mathbf{luminanceMappingX}[\,i\,] = \mathbf{luminance\_mapping\_x}[\,i\,] \div 8\,192 \quad \text{(A.26)}$$

A.2.3.7.4 luminanceMappingY mapping

for $i \in [\![0, \mathbf{luminanceMappingNumVal} - 1]\!]$

$$\mathbf{luminanceMappingY}[\,i\,] = \mathbf{luminance\_mapping\_y}[\,i\,] \div 8\,192 \quad \text{(A.27)}$$

A.2.3.8 Colour correction table

A.2.3.8.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the colour correction table elements. Clauses A.2.3.8.3 and A.2.3.8.4 apply only when **colour_correction_num_val** is greater than 0.

A.2.3.8.2 colourCorrectionNumVal mapping

$$\mathbf{colourCorrectionNumVal} = \mathbf{colour\_correction\_num\_val} \quad \text{(A.28)}$$

A.2.3.8.3 colourCorrectionX mapping

When **cc_uniform_sampling_flag** is equal to 1, the following applies:

for $i \in [\![0, \mathbf{colourCorrectionNumVal} - 1]\!]$

$$\mathbf{colourCorrectionX}\,[\,i\,] = i \div (\mathbf{colourCorrectionNumVal} - 1) \quad \text{(A.29)}$$

Otherwise, (when **cc_uniform_sampling_flag** is equal to 0) the following applies:

for $i \in [\![0, \mathbf{colourCorrectionNumVal} - 1]\!]$

$$\mathbf{colourCorrectionX}[\,i\,] = \mathbf{colour\_correction\_x}[\,i\,] \div 2\,048 \quad \text{(A.30)}$$

A.2.3.8.4 colourCorrectionY mapping

for $i \in [\![0, \mathbf{colourCorrectionNumVal} - 1]\!]$, the following applies:

$$\mathbf{colourCorrectionY}[\,i\,] = \mathbf{colour\_correction\_y}[\,i\,] \div 2\,048 \quad \text{(A.31)}$$

A.2.3.9 Gamut mapping variables

A.2.3.9.1 Introduction

This clause specifies the mapping of SL-HDR Information SEI message syntax elements values that relate to the forward and inverse gamut mapping elements.

A.2.3.9.2 satMappingMode mapping

$$\textbf{satMappingMode} = \textbf{sat\_mapping\_mode} \quad (A.32)$$

A.2.3.9.3 satGlobal1SegRatio mapping

$$\textbf{satGlobal1SegRatio} = \textbf{sat\_global\_1seg\_ ratio} \div 8 \quad (A.33)$$

A.2.3.9.4 satGlobal2SegRatioWCG mapping

$$\textbf{satGlobal2SegRatioWCG} = 7 \times (\textbf{sat\_global\_2seg\_ratio\_wcg} + 1) \div 64 \quad (A.34)$$

A.2.3.9.5 satGlobal2SegRatioSCG mapping

$$\textbf{satGlobal2SegRatioSCG} = 7 \times (\textbf{sat\_global\_2seg\_ratio\_scg} + 1) \div 64 \quad (A.35)$$

A.2.3.9.6 sat1SegRatio mapping

for c ∈ ⟦0,5⟧

$$\textbf{sat1SegRatio}[\,c\,] = \textbf{sat\_1seg\_ratio}[\,c\,] \div 8 \quad (A.36)$$

A.2.3.9.7 sat2SegRatioWCG mapping

for c ∈ ⟦0,5⟧

$$\textbf{sat2SegRatioWCG}[\,c\,] = 7 \times (\textbf{sat\_2seg\_ratio\_wcg}[\,c\,] + 1) \div 64 \quad (A.37)$$

A.2.3.9.8 sat2SegRatioSCG mapping

for c ∈ ⟦0,5⟧

$$\textbf{sat2SegRatioSCG}[\,c\,] = 7 \times (\textbf{sat\_2seg\_ratio\_scg}[\,c\,] + 1) \div 64 \quad (A.38)$$

A.2.3.9.9 lightnessMappingMode mapping

$$\textbf{lightnessMappingMode} = \textbf{lightness\_mapping\_mode} \quad (A.39)$$

A.2.3.9.10 lmWeightFactor mapping

for c ∈ ⟦0,5⟧

$$\textbf{lmWeightFactor}[\,c\,] = (\textbf{lm\_weight\_factor}[\,c\,] - 1) \div 4 \quad (A.40)$$

A.2.3.9.11 croppingModeSCG mapping

$$\textbf{croppingModeSCG} = \textbf{cropping\_mode\_scg} \quad (A.41)$$

A.2.3.9.12 cmWeightFactor mapping

for c ∈ ⟦0,5⟧

$$\textbf{cmWeightFactor}[\,c\,] = 9 \times \textbf{cm\_weight\_factor}[\,c\,] \div 64 \quad (A.42)$$

A.2.3.9.13 cmCroppedLightnessMappingEnabledFlag mapping

$$\textbf{cmCroppedLightnessMappingEnabledFlag} = \textbf{cm\_cropped\_lm\_enabled\_flag} \quad (A.43)$$

A.2.3.9.14 hueAdjMode mapping

hueAdjMode = **hue_adjustment_mode** (A.44)

A.2.3.9.15 hueGlobalPreservationRatio mapping

hueGlobalPreservationRatio = **hue_global_preservation_ratio** $\div 8$ (A.45)

A.2.3.9.16 huePreservationRatio mapping

for $c \in [\![0,5]\!]$

huePreservationRatio[c] = **hue_preservation_ratio**[c] $\div 8$ (A.46)

A.2.3.9.17 hueAlignCorrectionPresentFlag mapping

hueAlignCorrectionPresentFlag = **hue_adjustment_correction_info_present_flag** (A.47)

A.2.3.9.18 hueAlignCorrection mapping

for $c \in [\![0,5]\!]$

hueAlignCorrection[c] = **hue_alignment_correction**[c] (A.48)

A.2.3.9.19 chromAdjPresentFlag mapping

chromAdjPresentFlag = **chrom_adjustment_info_present_flag** (A.49)

A.2.3.9.20 chromAdjParam mapping

for $c \in [\![0,5]\!]$

chromAdjParam[c] = (**chrom_adjustment_param**[c] $+ 15) \div 16$ (A.50)

A.3 Mastering Display Colour Volume SEI message mapping

A.3.1 Introduction

This clause specifies the mapping of HDR picture characteristics when a Mastering Display Colour Volume SEI message is present in the same CVLS with a SL-HDR Information SEI message with a value of **src_mdcv_info_present_flag** equal to 0. When **src_mdcv_info_present_flag** is equal to 0, the Mastering Display Colour Volume SEI message shall be present and the mapping to **hdrDisplayColourSpace, hdrDisplayMaxLuminance** and **hdrDisplayMinLuminance** shall be as specified in clause A.3.2. The mapping to **hdrPicColourSpace** is specified in clause A.2.3.3.2.

A.3.2 HDR picture characteristics

When **src_mdcv_info_present_flag** is equal to 0, **hdrDisplayColourSpace** shall be mapped as specified in Table A.5.

Table A.5: Mapping to hdrDisplayColourSpace (MDCV)

MDCV SEI syntax elements	MDCV SEI syntax elements value	hdrDisplayColourSpace value	Colour space primaries
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 [6]
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 [7]
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 (DCI P3) [5]
NOTE: In case the 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 **hdrDisplayColourSpace** that is the closest match to the column 2 values.			

When **src_mdcv_info_present_flag** is equal to 0, **hdrDisplayMaxLuminance** and **hdrDisplayMinLuminance** shall be mapped as specified below:

$$\textbf{hdrDisplayMaxLuminance} = Min(\ 50 \times ((\ \textbf{max\_display\_mastering\_luminance} \times 0{,}000\ 1 + 25\ )\ /\ 50\ );\ 10\ 000\ ) \qquad \text{(A.51)}$$

$$\textbf{hdrDisplayMinLuminance} = Min(\ \textbf{min\_display\_mastering\_luminance} \times 0{,}000\ 1\ ;\ 10\ 000\ ) \qquad \text{(A.52)}$$

Annex B (normative): SL-HDR reconstruction metadata using AVC

This annex specifies the format of the SEI message that carries the SL-HDR reconstruction metadata for AVC specification [3] as well as the mapping between the syntax elements of this SEI message and the dynamic metadata variables provided in clause 6. The SL-HDR Information SEI message syntax, semantics and mapping for the AVC specification are identical to those specified in clauses A.2.2 and A.2.3 for the HEVC specification except for the SL-HDR Information SEI message persistence mechanism and the terminal provider oriented code message value that are specific to AVC.

The SL-HDR 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].

The value of **terminal_provider_oriented_code_message_idc** shall be equal to 0b00000001 (0x01).

The **sl_hdr_persistence_flag** syntax element and descriptor of Table A.1 and the associated semantics are replaced as follows:

sl_hdr_repetition_period specifies the persistence of the SL-HDR information SEI message and may specify a picture order count interval within which another SL-HDR information SEI message with the same value of sl_hdr_mode_value_minus1 or the end of the coded video sequence shall be present in the bitstream. The value of sl_hdr_repetition_period shall be in the range of 0 to 16 384, inclusive.

sl_hdr_repetition_period descriptor is u(17).

sl_hdr_repetition_period equal to 0 specifies that the SL-HDR information applies to the current decoded picture only.

sl_hdr_repetition_period equal to 1 specifies that the SL-HDR 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 SL-HDR information SEI message with the same value of sl_hdr_mode_value_minus1 is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic).

sl_hdr_repetition_period equal to 0 or equal to 1 indicates that another SL-HDR information SEI message with the same value of sl_hdr_mode_value_minus1 may or may not be present.

sl_hdr_repetition_period greater than 1 specifies that the SL-HDR information persists until any of the following conditions are true:

- A new coded video sequence begins.
- A picture in an access unit containing a SL-HDR information SEI message with the same value of sl_hdr_mode_value_minus1 is output having PicOrderCnt( ) greater than PicOrderCnt(CurrPic) and less than or equal to PicOrderCnt(CurrPic) + sl_hdr_repetition_period.

sl_hdr_repetition_period greater than 1 indicates that another SL-HDR information SEI message with the same value of sl_hdr_mode_value_minus1 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) + sl_hdr_repetition_period; unless the bitstream ends or a new coded video sequence begins without output of such a picture.

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 space in which the HDR and SDR pictures are represented. 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.

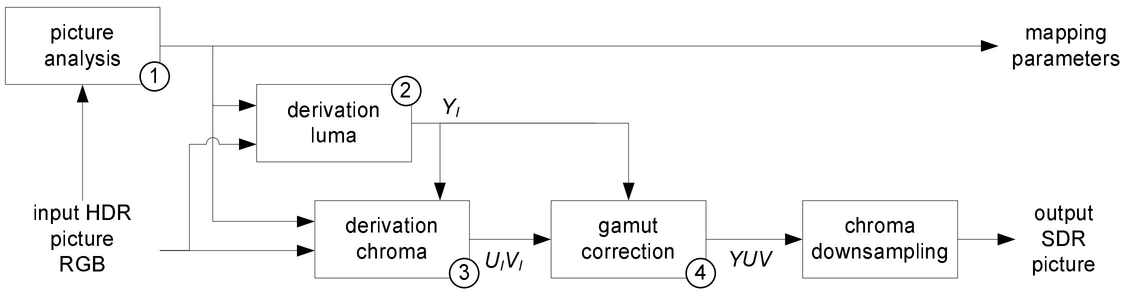

Figure C.1: Synopsis of HDR-to-SDR decomposition process

Clause C.1.2 provides more details on the steps 2 to 4. Clause C.1.3 describes the actual implementation of the process. Clause C.2 gives information on mapping and colour functions derivation. 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. Coefficients mu_0 and mu_1 are mapping parameters of the model as well. 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 canonical 3x3 R'G'B'-to-Y'CbCr conversion matrix (e.g. Recommendation ITU-R BT.2020-2 [7] or Recommendation ITU-R BT.709-6 [6] 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:

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

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

- A pseudo-gammatization is applied and the resulting signal is converted to CbCr as follows:

$$\begin{bmatrix} U_{tmp} \\ V_{tmp} \end{bmatrix} = \frac{1}{L^{\frac{1}{\gamma}}} \times \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} R^{1/\gamma} \\ G^{1/\gamma} \\ B^{1/\gamma} \end{bmatrix} \quad \text{(C.3)}$$

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

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

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

$$\begin{bmatrix} U_{sdr} \\ V_{sdr} \end{bmatrix} = \frac{1}{\beta(Y_{tmp})} \times \begin{bmatrix} U_{tmp} \\ V_{tmp} \end{bmatrix} = \frac{1}{\beta(Y_{tmp})} \times \frac{1}{L^{\frac{1}{\gamma}}} \times \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} R^{1/\gamma} \\ G^{1/\gamma} \\ B^{1/\gamma} \end{bmatrix} \quad \text{(C.4)}$$

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

- Then a final adjustment of the luma component applies:

$$Y_{sdr} = Y_{tmp} - Max(0\ ;\ mu_0 \times U_{sdr} + mu_1 \times V_{sdr}) \quad \text{(C.5)}$$

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:

$$L = A_1 \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{(C.6)}$$

$$Y_{pre0} = (maxSampleVal - 1) \times (LUT_{TM}(L))^{\frac{1}{2,4}} \quad \text{(C.7)}$$

The luminance is generated using the inverse function of $(LUT_{TM}(L))^{\frac{1}{2,4}}$: $\hat{L}^{1/\gamma} = invLUT_{TM}[\ Y_{pre0}]$.

NOTE 1: $invLUT_{TM}[\ Y]$ corresponds to the $lutMapY[\ Y]$.

$$\begin{bmatrix} U_{pre0} \\ V_{pre0} \end{bmatrix} = \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} R^{1/\gamma} \\ G^{1/\gamma} \\ B^{1/\gamma} \end{bmatrix} \quad \text{(C.8)}$$

$$\begin{bmatrix} U_{pre1} \\ V_{pre1} \end{bmatrix} = \frac{1}{\beta_0(Y_{preo})} \times \begin{bmatrix} U_{pre0} \\ V_{pre0} \end{bmatrix} = \frac{1}{\beta_0(Y_{preo})} \times \begin{bmatrix} A_2 \\ A_3 \end{bmatrix} \times \begin{bmatrix} R^{1/\gamma} \\ G^{1/\gamma} \\ B^{1/\gamma} \end{bmatrix} \quad \text{(C.9)}$$

where $\beta_0(Y_{tmp,SDR}) = \beta(Y_{tmp,SDR}) \times \hat{L}^{1/\gamma}$. $\beta_0(Y_{tmp,SDR})$ is defined in clause C.3.4.

NOTE 2: $\beta(Y_{tmp,SDR})$ corresponds to the $lutCC[\ Y]$.

$$\begin{bmatrix} U_{pre2} \\ V_{pre2} \end{bmatrix} = \begin{bmatrix} Clip3(-midSampleVal\ ;\ midSampleVal - 1\ ;\ U_{pre1}) \\ Clip3(-midSampleVal\ ;\ midSampleVal - 1\ ;\ V_{pre1}) \end{bmatrix} \quad \text{(C.10)}$$

$$Y_{sdr} = Y_{pre0} - Max(0\ ;\ mu_0 \times U_{pre2} + mu_1 \times V_{pre2}) \quad \text{(C.11)}$$

$$\begin{bmatrix} U_{sdr} \\ V_{sdr} \end{bmatrix} = \begin{bmatrix} U_{pre2} + midSampleVal \\ V_{pre2} + midSampleVal \end{bmatrix} \quad \text{(C.12)}$$

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 **hdrDisplayMaxLuminance**. 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 Fine Tuning 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 $LUT_{TM}(L)$.

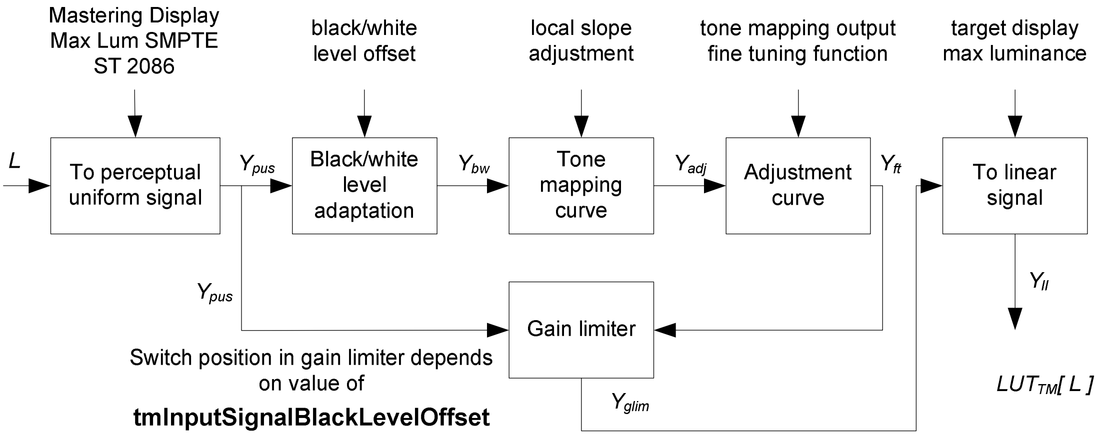

Figure C.2: Tone mapping process

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 present document.

Y_{pus} is calculated as specified by equations (C.13) and (C.14).

$$Y_{pus} = v(L, L_{HDR}) \quad \text{(C.13)}$$

$$v(x, L) = \frac{\log_{10}\left(1 + (\rho(L) - 1) \times x^{\frac{1}{2,4}}\right)}{\log_{10}(\rho(L))} \quad \text{(C.14)}$$

The value for $\rho(L_{HDR})$ can be calculated using equation (C.15). L_{HDR} represents the peak luminance given by the Mastering Display Maximum Luminance, and is stored in the variable **hdrDisplayMaxLuminance** in the structure hdr_characteristics( ) of the reconstruction metadata as specified in clause 6.3.3.4).

$$\rho(L) = 1 + (33 - 1) \times \left(\frac{L}{10000}\right)^{\frac{1}{2,4}} \quad \text{(C.15)}$$

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

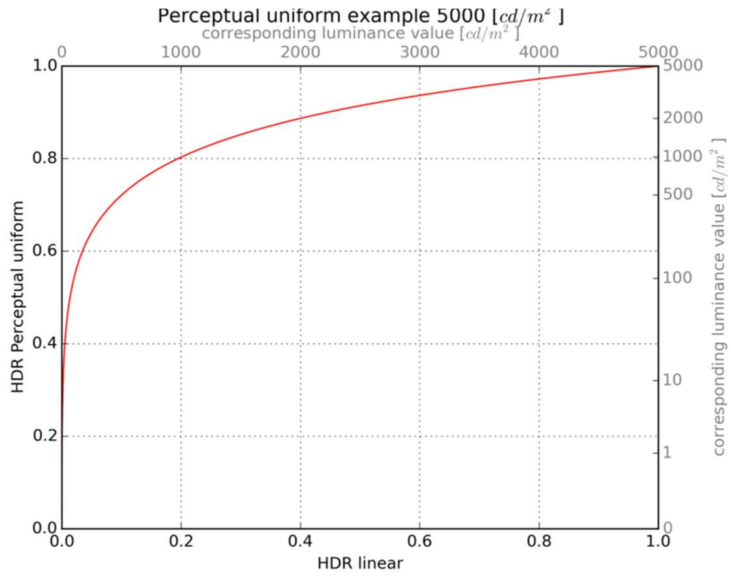

Figure C.3: Example curve $v(x, L_{HDR})$ for L_{HDR} = 5 000 cd/m²

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.16), (C.17) and (C.18):

$$Y_{bw} = \frac{Y_{pus} - blo}{1 - wlo - blo} \quad \text{(C.16)}$$

$$wlo = \frac{255 \times \mathbf{tmInputSignalWhiteLevelOffset}}{510} \quad \text{(C.17)}$$

$$blo = \frac{255 \times \mathbf{tmInputSignalBlackLevelOffset}}{2040} \quad \text{(C.18)}$$

NOTE: Equation (C.16) is the inverse of equation (15).

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

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

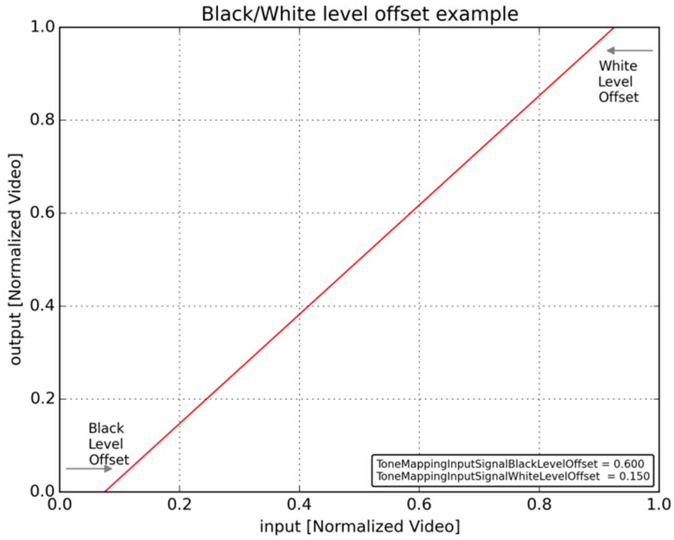

Figure C.4: Example curve for black and white level offset

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.19):

$$Y_{adj} = TMO(Y_{bw}) \quad \text{(C.19)}$$

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

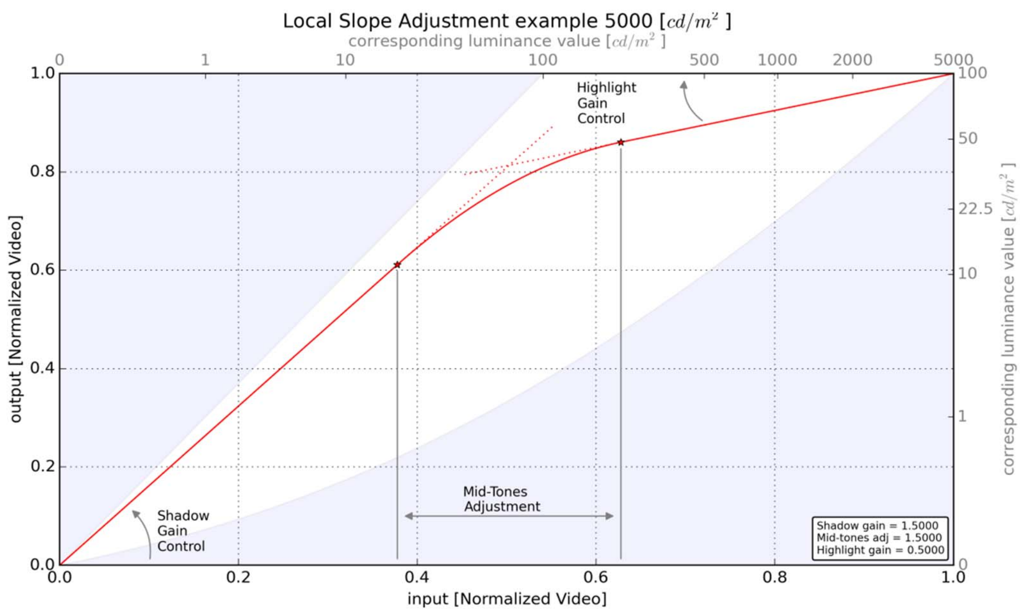

Figure C.5: Tone mapping curve shape example

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.20) up to and including equation (C.29) 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 $L_{HDR} > L_{target}$.

$$TMO(x) = \begin{cases} SGC \times x, & 0 \leq x \leq x_{SGC} \\ ax^2 + bx + c, & x_{SGC} < x < x_{HGC} \\ HGC \times x + 1 - HGC, & x_{HGC} \leq x \leq 1 \end{cases} \quad \text{(C.20)}$$

$$a = \begin{cases} 0, & para = 0 \\ -0{,}5 \times \frac{SGC-HGC}{para}, & otherwise \end{cases}$$
$$b = \begin{cases} 0, & para = 0 \\ \frac{1-HGC}{para} + \frac{SGC+HGC}{2}, & otherwise \end{cases} \quad \text{(C.21)}$$
$$c = \begin{cases} 0, & para = 0 \\ -\frac{\left((SGC-HGC)\times para - 2(1-HGC)\right)^2}{8\times(SGC-HGC)\times para}, & otherwise \end{cases}$$

$$x_{SGC} = \frac{1-HGC}{SGC-HGC} - \frac{para}{2}$$
$$x_{HGC} = \frac{1-HGC}{SGC-HGC} + \frac{para}{2} \quad \text{(C.22)}$$

$$exposure = \frac{\mathbf{shadowGain}}{4} + 0{,}5 \quad \text{(C.23)}$$

$$expgain = v\left(\frac{L_{HDR}}{L_{target}}, L_{target}\right) \quad \text{(C.24)}$$

$$L_{HDR} = \mathbf{hdrDisplayMaxLuminance} \quad \text{(C.25)}$$

$$L_{target} = 100\ \text{cd}/m^2 \quad \text{(C.26)}$$

$$SGC = expgain \times exposure \quad \text{(C.27)}$$

$$HGC = \frac{\mathbf{highlightGain}}{4} \quad \text{(C.28)}$$

$$para = \frac{\mathbf{midToneWidthAdjFactor}}{2} \quad \text{(C.29)}$$

The value of **shadowGain**, **highlightGain** and **midToneWidthAdjFactor**, as used in equations (C.23), (C.28) and (C.29), are stored in the metadata structure luminance_mapping_variables( ) as specified in clause 6.2.5.

The value of **hdrDisplayMaxLuminance**, as used in equation (C.25), is stored in the metadata structure hdr_characteristics( ) as specified in clause 6.3.3.4.

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.30).

$$Y_{ft} = \begin{cases} f_{ftlum}(Y_{adj}), & 0 \leq Y_{adj} \leq 1 \\ Y_{adj}, & otherwise \end{cases} \quad \text{(C.30)}$$

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 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 6.3.5.9.

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

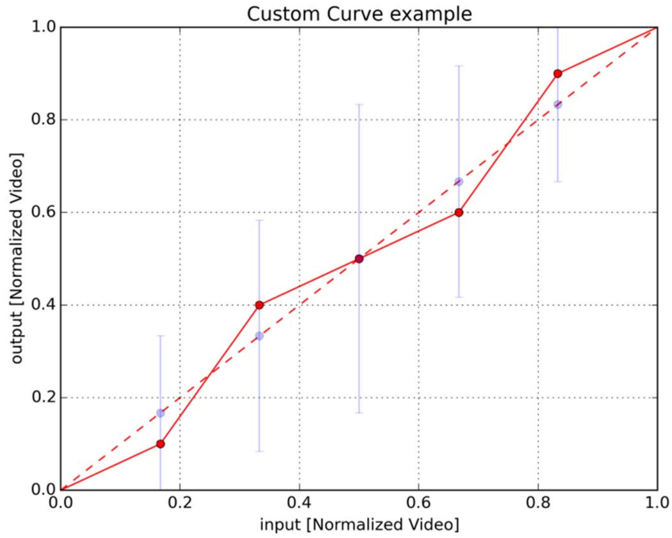

Figure C.6: Example fine-tuning curve

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.31) and (C.32), based on the maximum display mastering luminance L_{HDR}, which is equal to the variable **hdrDisplayMaxLuminance** that is stored in the structure hdr_characteristics( ) of the reconstruction metadata (clause 6.3.3.4).

$$Y_{glim} = Max\left(Y_{ft}\ ;\ Y_{pus} \times g\right) \quad \text{(C.31)}$$

$$g = v(0{,}1 \div L_{SDR}, L_{SDR}) \div v(1 \div L_{HDR}, L_{HDR}) \quad \text{(C.32)}$$

with the inverse EOTF, $v(x, y)$, taken from equation (C.14).

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.33), up to and including (C.35).

$$Y_{ll} = v_{inv}(Y_{glim}, 100) \quad \text{(C.33)}$$

$$v_{inv}(x, y) = \left(\frac{\rho(y)^{x} - 1}{\rho(y) - 1}\right)^{2{,}4} \quad \text{(C.34)}$$

NOTE: Equation (C.34) is the mathematical inverse function for equation (C.14).

$$\rho(y) = 1 + (33 - 1) \times \left(\frac{y}{10000}\right)^{\frac{1}{2{,}4}} \quad \text{(C.35)}$$

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

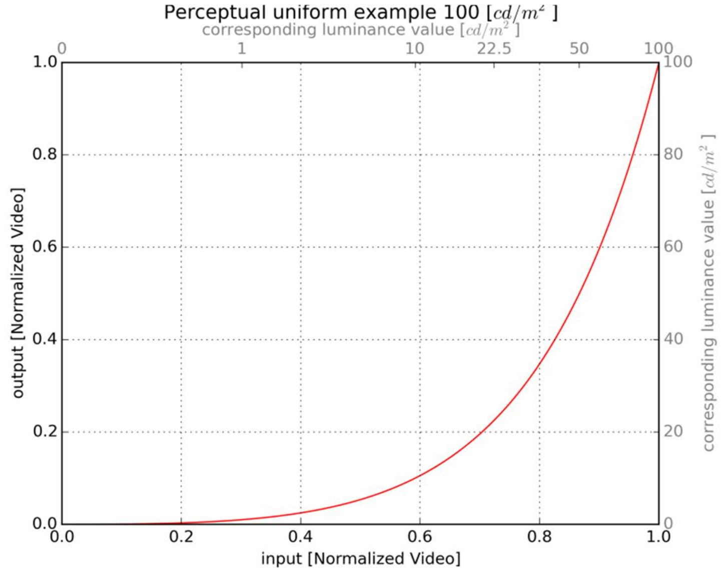

Figure C.7: Perceptual to linear curve based on 100 cd/m²

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

Clause 7.2.3.2 specifies the construction of the colour correction table. The default colour correction curve $L(x)$ is the inverse of the normalized SDR luma: $L(Y_{post1}) = \frac{1}{1023 \times Y_{post1}}$. This default curve is scaled by the saturation gain function $f_{sgf}(\,)$. This clause proposes a method to determine parameters representative of a simple function $f_{sgf}(\,)$.

The variables **kCoefficient**[i] and **chromaToLumaInjection**[0] are set to 0. **chromaToLumaInjection**[1] is set to 0,1.

Experiments show that in order to preserve the hue and saturation fidelity to the source picture, the colour correction is of type:

$$lutCC[Y_{post1}] = L(Y_{post1}) \times \Omega \quad \text{(C.36)}$$

where Ω is a constant that depends on the colour space of the source picture.

This value of Ω may be conveyed by the function $f_{sgf}(\,)$ which is defined by the variables **saturationGainX**[i], **saturationGainY**[i] and **saturationGainNumVal.** As Ω is a constant number, only one pair of **saturationGainX**[i], **saturationGainY**[i] is required.

From experiments, it can be determined that when the source picture colour space is Rec. 2020, $\Omega = 1{,}2$. This value is coded as follows:

$$\mathbf{saturationGainNumVal} = 1 \quad \text{(C.37)}$$

$$\mathbf{saturationGainX}[0] = 0 \quad \text{(C.38)}$$

$$\mathbf{saturationGainY}[0] = \frac{1}{R_{sgf} \times \Omega} = \frac{1}{2 \times 1{,}2} = 0{,}417 \quad \text{(C.39)}$$

By mean of the $f_{sgf}(\,)$, saturation of the pre-processed picture can be controlled.

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.40) and (C.41).

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

$$vMaxOut = v(luminancePeakSDR, L_{HDR}) \tag{C.40}$$

$$vMaxIn = v(luminancePeakHDR, L_{HDR}) \tag{C.41}$$

where the OETF $v(;)$ is defined in equations (C.42) to (C.43), and where L_{HDR}, represents the Mastering Display Maximum Luminance used to master the HDR content in cd/m^2.

$$v(x, L) = \frac{\log_{10}\left(1 + (\rho(L) - 1) \times x^{\frac{1}{2,4}}\right)}{\log_{10}(\rho(L))} \tag{C.42}$$

$$\rho(L) = 1 + (33 - 1) \times \left(\frac{L}{10\,000}\right)^{\frac{1}{2,4}} \tag{C.43}$$

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.44).

$$Y = v(L, L_{HDR}) \tag{C.44}$$

where L represents the pixel luminance value in the normalized linear domain, and where the OETF v(;) is defined in equations (C.42) and (C.43), 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.45).

$$bs = Clip3(0\ ;\ 0{,}1\ ;\ bsu\ ) \quad \text{(C.45)}$$

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.46).

$$V = v(Max(Max(R\ ;G)\ ;B), L_{HDR}) \quad \text{(C.46)}$$

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.42) and (C.43), 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.47).

$$ws = Clip3(vMaxOut\ ;\ vMaxIn\ ;wsu\ ) \quad \text{(C.47)}$$

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.48) and (C.49).

$$bl = 0{,}6 \times bs \quad \text{(C.48)}$$

$$wh = 0{,}8 \times ws + 0{,}2 \times vMaxIn \quad \text{(C.49)}$$

The parameters **tmInputSignalBlackLevelOffset**, and **tmInputSignalWhiteLevelOffset**, can be derived according to equations (C.50) and (C.51).

$$\mathbf{tmInputSignalBlackLevelOffset} = bl \div vMaxIn \quad \text{(C.50)}$$

$$\mathbf{tmInputSignalWhiteLevelOffset} = 1 - wh \div vMaxIn \quad \text{(C.51)}$$

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.46) 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.52) to (C.53).

$$bg = Min\left(nomGain \times \mathrm{Max}\left(1\ ;\left(2 - \frac{LightnessHDR}{LightnessHDRHigh}\right) \div bwGain\right)\ ;\ 1\right) \quad \text{(C.52)}$$

$$nomGain = vMaxOut \div vMaxIn \quad \text{(C.53)}$$

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.54) and where $vMaxOut$ and $vMaxIn$ are defined in equations (C.40) and (C.41).

$$bwGain = vMaxIn \div wh \quad \text{(C.54)}$$

where *bl* and *wh* are defined in equations (C.48) and (C.49).

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.55).

$$\mathbf{shadowGain} = 4\times(bg - 0{,}5) \tag{C.55}$$

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.56) to (C.57).

$$dg = Clip3(0{,}25\times nomGain\,;0{,}5\times nomGain\,;\ 0{,}375 - 0{,}25\times bg\ ) \tag{C.56}$$

$$\mathbf{highlightGain} = \frac{4\times dg}{nomGain} \tag{C.57}$$

where dg is the differential gain, $nomGain$ is from equation (C.53) and is from equation (C.52).

The parameter **midToneWidthAdjFactor** can be computed according to equations (C.58) to (C.62).

$$xp1\ =\ Clip3(0{,}2\,;0{,}5\,;\ 1{,}12 - bg\ ) \tag{C.58}$$

$$xm\ =\ (vMaxOut\ -\ dg\times vMaxIn) \div Max(10^{-8}; bg\ -\ dg) \tag{C.59}$$

$$xp2\ =\ Min\big(2\times \mathrm{xm}\,;2\times(vMaxIn\ -\ \mathrm{xm})\big) \tag{C.60}$$

$$xp\ =\ Min(xp1\,;xp2) \tag{C.61}$$

$$\mathbf{midToneWidthAdjFactor} = 2\times xp \div vMaxIn \tag{C.62}$$

where bg is taken from equation (C.52) and dg is taken from equation (C.56).

NOTE: Equations (C.60) and (C.61) 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.63) to (C.65).

$$e = Y[i] - YF[i-1] \tag{C.63}$$

$$d = F(e) \tag{C.64}$$

$$Y[i] = d + YF[i-1] \tag{C.65}$$

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.8.

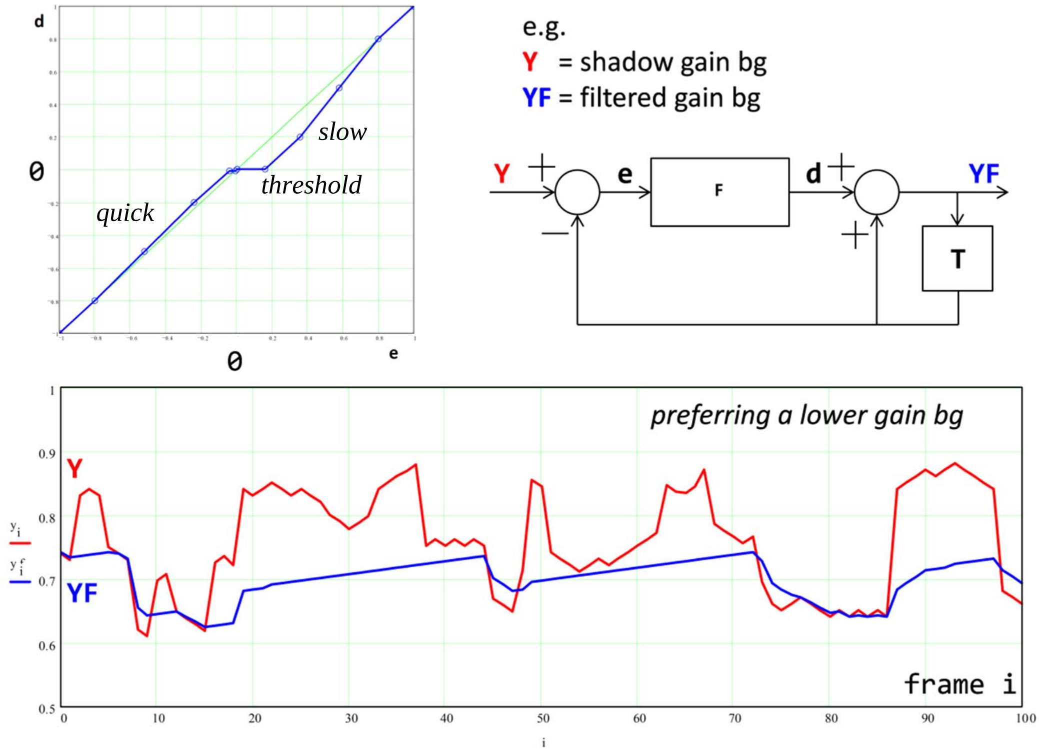

Figure C.8: Example temporal filter for the shadow gain parameter

In case the transfer of the function $F(e)$ in Figure C.8 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.8 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.8 has a very small but non-zero value for d.

In the case of Figure C.8, 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.9.

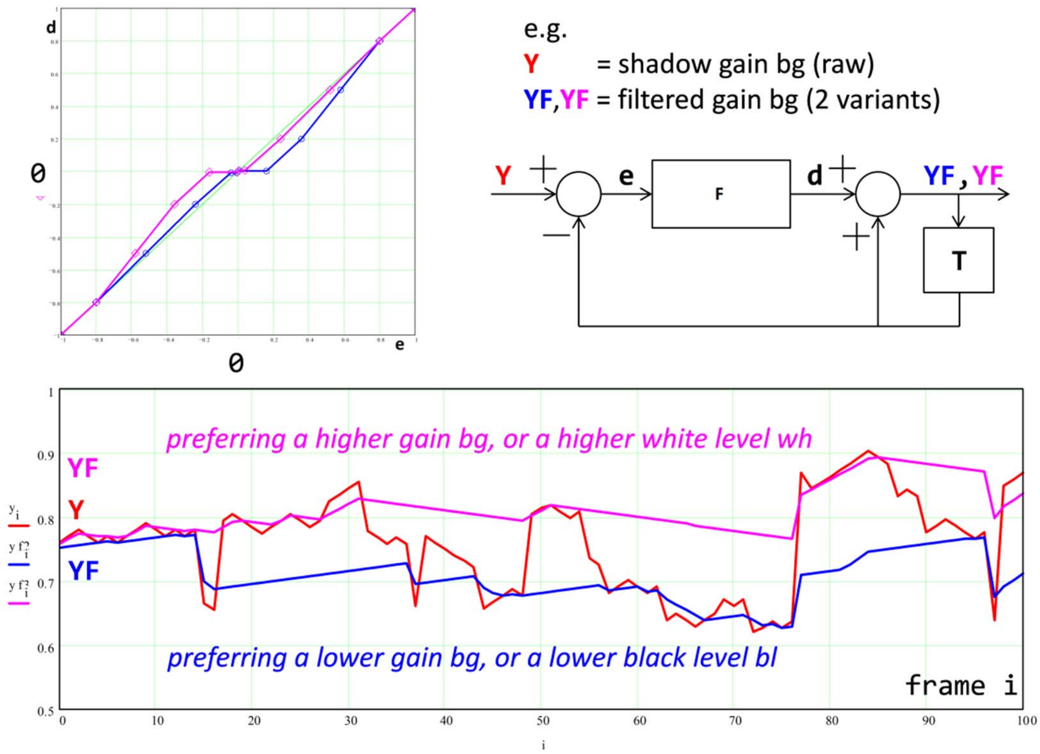

Figure C.9: Example temporal filters for the shadow gain and white level parameters

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.52) and (C.54), 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.66) to (C.73).

$$bgUf = nomGain \times \left(2 - \frac{LightnessHDR}{LightnessHDRHigh}\right) \quad (C.66)$$

$$bgUfCl = Min(Max(nomGain\,;bgUf)\,;\,1) \quad (C.67)$$

$$whTf = \text{temporal\_filter}(wh) \quad (C.68)$$

$$bwGainTf = vMaxIn \div whTf \quad (C.69)$$

$$bgTf = \text{temporal\_filter}(bgUfCl) \quad (C.70)$$

$$bgTfM = \frac{bgTf}{bwGainTf} \quad (C.71)$$

$$bgTfCl = Min(Max(nomGain\,;bgTfM)\,;\,1) \quad (C.72)$$

$$whTfCl = \frac{bgTfCl}{bgTf} \times vMaxIn \quad (C.73)$$

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.74).

$$\mathbf{shadowGainTf} = 4 \times (bgTfCl - 0{,}5) \quad (C.74)$$

The temporally filtered parameter **tmInputSignalWhiteLevelOffset**, **tmInputSignalWhiteLevelOffsetTf**, can be computed according to equation (C.75).

$$\mathbf{tmInputSignalWhiteLevelOffsetTf} = 1 - whTfCl \div vMaxIn \quad \text{(C.75)}$$

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.56) to (C.57) and the temporally filtered parameter **midToneWidthAdjFactor** can be derived from equations (C.58) to (C.62), where $bgTfCl$ is taken for the value of bg in equations (C.56) and (C.62).

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 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, 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:

$$\beta_0(Y) = lutMapY[Y] \times lutCC[Y] \quad \text{(C.76)}$$

$lutCC$ 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.10, the derivation of the colour correction LUT in the pre-processing stage consists of:

- identifying from the content properties the relevant LUT $lutCC_{ref\_k}$;
- optionally computing the saturation gain function $f_{sgf}(\,)$ that enables the generation of $lutCC_{ref\_k}$ from the default colour correction LUT as documented in clause C.2.3;
- building the colour correction LUT $lutCC$ used in the post-processing stage;
- building the colour correction LUT β_0 used in the pre-processing stage.

The pixel loop can then run using the luminance mapping LUT $lutMapY$ and the colour correction LUT β_0.

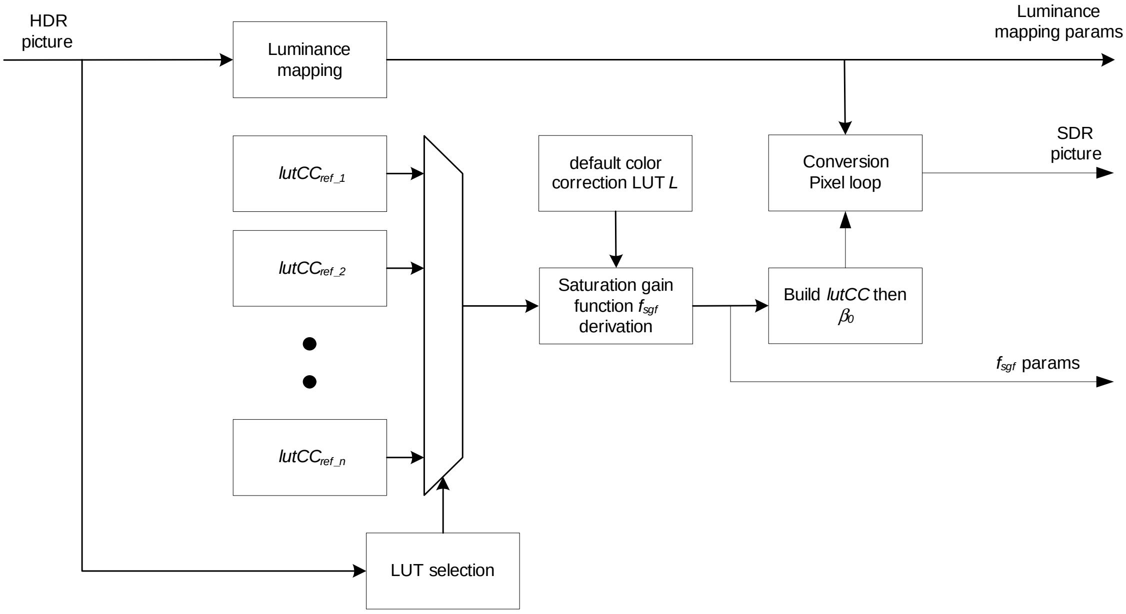

Figure C.10: Illustration of the simplified colour correction LUT usage in the pre-processing stage

Typically, three different pre-defined LUTs $lutCC_{ref\_i}$ are defined. Examples of three LUTs are depicted in Figure C.11. A possible process for the LUT selection is explained in clause C.3.4.3.

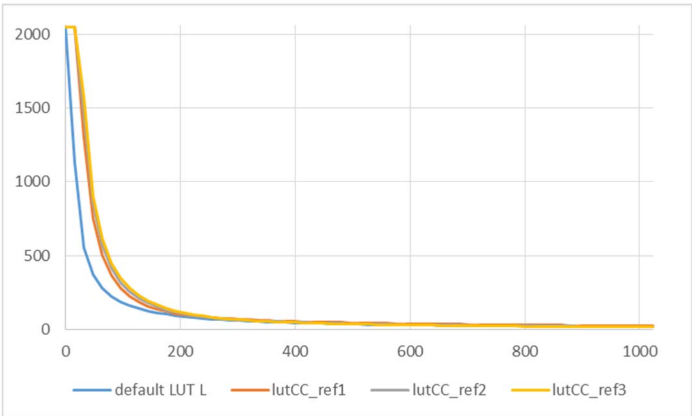

Figure C.11: Examples of pre-defined colour correction LUTs

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:

$$\begin{cases} R' = f_{nll}(R) \\ G' = f_{nll}(G) \\ B' = f_{nll}(B) \end{cases} \quad \text{(C.77)}$$

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

$$f_{nll}\begin{cases} if\,(v < 0{,}0031308), f_{nll}(v) = Max(0\,;v\times 12{,}92) \\ else\ f_{nll}(v) = 1{,}055\times v^{\frac{1}{2{,}4}} - 0{,}055 \end{cases} \quad \text{(C.78)}$$

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:

$$\begin{cases} if\,(maxRGB > 0{,}01), S = \frac{maxRGB - minRGB}{maxRGB} \\ else\ S\ is\ not\ defined, and\ maxRGB\ is\ set\ to\ 0 \end{cases} \quad \text{(C.79)}$$

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 $(maxRGB - minRGB < 0{,}000\,1)$ as follows:

$$\begin{cases} if\,(maxRGB = R'), hue_R = \left(\left[\frac{G'-B'}{maxRGB - minRGB} + 0\right]\times 60\right)\%\,360 \\ if\,(maxRGB = G'), hue_G = \left(\left[\frac{B'-R'}{maxRGB - minRGB} + 2\right]\times 60\right)\%\,360 \\ if\,(maxRGB = B'), hue_B = \left(\left[\frac{R'-G'}{maxRGB - minRGB} + 4\right]\times 60\right)\%\,360 \end{cases} \quad \text{(C.80)}$$

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:

$$R_{XY} = \frac{size(H_{XY})}{size(H_{RY}) + size(H_{GY}) + size(H_{BY})} \quad \text{(C.81)}$$

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 $R_{R20} > R_{B20}$) 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): Invertible gamut mapping

D.1 Introduction

This annex provides the description of an invertible gamut mapping process that could apply when the input SDR picture of the SDR-to-HDR reconstruction process is provided in a SDR legacy standard colour gamut (e.g. Recommendation ITU-R BT.709-6 [6]) as specified by the variable **sdrPicColourSpace**), and is different from the target wide colour gamut of the HDR picture (e.g. Recommendation ITU-R BT.2020-2 [7] as specified by the variable **hdrPicColourSpace**). In this annex, colour backward compatibility is defined such that the SDR CE receiver only supports Recommendation ITU-R BT.709-6 [6] colour space while the video to be distributed using SL-HDR1 can support Recommendation ITU-R BT.2020-2 colour space. When **hdrPicColourSpace** is not equal to **sdrPicColourSpace,** at the HDR-to-SDR decomposition stage the WCG HDR video should be converted to a standard colour gamut SDR video (plus metadata) while the inverse process at the HDR reconstruction side reverts this conversion by rendering the WCG HDR video from the standard colour gamut SDR video (plus metadata). The cascading of those two-colour processes should be visually lossless, while the standard colour gamut SDR video should entirely preserve the artistic intent of the original WCG HDR video with minimal impairments. Both colour reconstruction (inverse gamut mapping) and compression (forward gamut mapping) conversions are designed to be reciprocal.

Figure D.1 illustrates a typical scenario where (forward) 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 **hdrDisplayColourSpace**), and represented in Recommendation ITU-R BT 2020 colour space (signalled by **hdrPicColourSpace**). In this scenario, the SDR backward compatible version derived from the HDR content is provided in Recommendation ITU-R BT.709-6 [6] container, in order to be directly viewable on legacy BT 709 SDR displays. Therefore, for distribution of the SDR signal, a (forward) gamut mapping from Recommendation ITU-R BT.2020-2 [7] to Recommendation ITU-R BT.709-6 [6] is required in addition to the dynamic range mapping from HDR to SDR before the distribution step. The distributed SDR content, represented in Recommendation ITU-R BT.709-6 [6] colour space (signalled by **sdrPicColourSpace**), can then to be converted back to an HDR version in a Recommendation ITU-R BT.2020-2 [7] colour space or any other colour space with a wide colour gamut.

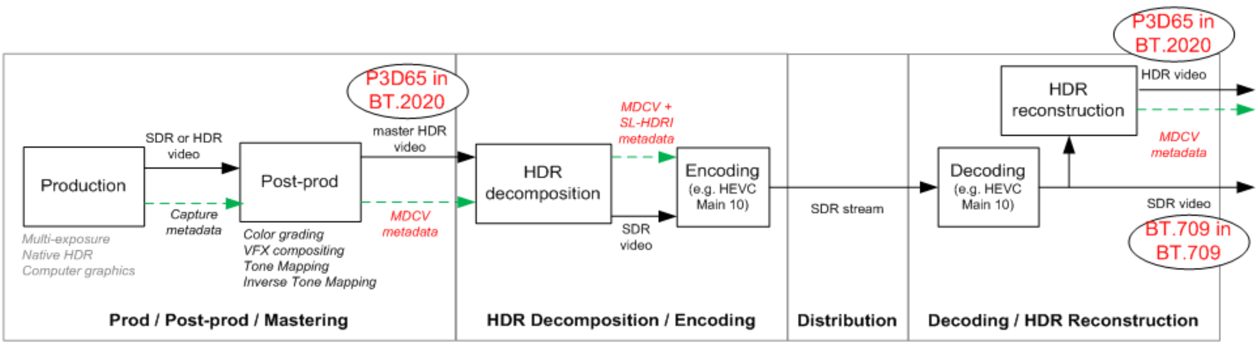

Figure D.1: Example of use case requiring a gamut mapping

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

As illustrated in Figure 2, the inverse gamut mapping process applies once the HDR picture has been reconstructed from the SDR picture by the SDR-to-HDR reconstruction process described in clause 7. The process should apply when **hdrPicColourSpace** is not equal to **sdrPicColourSpace**.

The processes specified in Annex D work on signals represented in a linear-light YUV colour space which is a linear-light colour space obtained from the conversion of a linear-light RGB colour space with the canonical R'G'B'-to-Y'CbCr matrix. Consequently, a tristimulus sample in the linear-light YUV colour space consists of a lightness sample with two chrominance samples. The combination of these three samples constitutes a colour that belongs to at least one gamut.

NOTE 1: The invertible gamut mapping process operates in a linear-light YUV colour space as this colour space makes computational processes simpler than perceptual colour spaces (e.g. CIELab, IPT, etc.) with a good invertibility / rendition quality trade-off.

NOTE 2: The U and V chrominance components are represented with signed values typically contained in the [- 1,0 to 1,0] range. The Y component is also signed although its typical range is [0,0 to 1,0].

Clause D.2 provides notations and definitions essential to the understanding of the processes specified in Annex D. Clauses D.3 and D.4 respectively document the forward gamut mapping and the inverse gamut mapping processes.

D.2 Notations and definitions

D.2.1 Introduction

This clause provides the notations (clause D.2.2).and definitions (clause D.2.3) that are used to specify the forward gamut mapping process (clause D.3) and the inverse gamut mapping process (clause D.4).

D.2.2 Notations

$\mathfrak{D}$ denotes a line. A line may be indexed by a subscript (e.g. $\mathfrak{D}_0$).

(AB) denotes a line passing by points A and B.

[AB[denotes a segment from point A (included) to point B (excluded).

$(\mathrm{A} + \vec{\mu})$ denotes a line passing by A with the $\vec{\mu}$ direction vector.

$\mathfrak{D}_0 \cap \mathfrak{D}_1$ denotes the intersection of line $\mathfrak{D}_0$ with line $\mathfrak{D}_1$. Similarly, the $\cap$ symbol denotes an intersection of two planes, or the intersection of a volume and a plane.

$Plane_{(y=\,P_y)}$ denotes the plane having a constant lightness of P_y in the linear-light YUV colour space.

$HalfPlane_{(\theta=\,P_\theta)}$ denotes the half plane containing all colours of the same hue where the hue is defined as the angle P_θ between P_u and P_v.

The cross-product of vector $\overrightarrow{oa}$ with vector $\overrightarrow{ob}$ is denoted $\overrightarrow{oa} \times \overrightarrow{ob}$ where:

$$\overrightarrow{oa} \times \overrightarrow{ob} \; = \; a_u \times b_v - \; a_v \times b_u \qquad \text{(D.1)}$$

Colour gamut (or gamut) is indicated by italics Cambria Math 10 points. A colour gamut of a colour space is denoted with an uppercase letter (e.g. $\mathcal{G}$). A gamut that is denoted with a lowercase letter (e.g. w) refers to a projection of a gamut on a plane (e.g. projection of a gamut on the chrominance plane). A tilde on a gamut notation denotes a so called "warped" gamut that is to say a gamut which faces are no planar (e.g. $\tilde{\mathcal{G}}$) unlike usual planar faces gamut (gamut notation without tilde).

W and K respectively denote the white point and the black point in a linear-light YUV colour space. W and K are similar in all gamuts used by the invertible gamut mapping process.

R, G and B respectively denote the red, green and blue primary colours in a linear-light YUV colour space.

NOTE 1: Video signal components are normalized with a black level at 0 and a peak white level at 1.

M, C and J respectively denote the magenta, cyan and yellow secondary colours in a linear-light YUV colour space.

NOTE 2: J designates yellow secondary colour instead of Y, to avoid any confusion with Y of the linear-light YUV colour space. Similarly, K represents the black point to avoid confusion with the blue primary colour notation B.

In a linear-light YUV colour space, the gamut can be represented by a hexahedron which vertices are defined by the octuplet { K, R, M, B, C, G, J, W }. An index i is used as a superscript letter onto a gamut symbol to indicate a gamut vertex in terms of primary or secondary colour. As an example, $\mathcal{W}^R$ designates the red primary vertex of the wide colour gamut $\mathcal{W}$.

Each coordinate of a primary or secondary colour represented in the linear-light YUV colour space is indicated by a subscript letter representative of a component. As an example, the Cartesian coordinates in the linear-light YUV colour space of the red primary colour vertex R of the wide colour gamut $\mathcal{W}$ are denoted by $\mathcal{W}_y^R, \mathcal{W}_u^R, \mathcal{W}_v^R$. Cylindrical coordinates are denoted by ρ and θ where the angle representative of the hue in a wide colour gamut is denoted $\mathcal{W}_\theta^R$ and the radius representative of the chrominance $\mathcal{W}_\rho^R$.

In computational loops, each of the six vertices corresponding to the primary and secondary colours is indicated by an index value c. The index value c equal to 0 should correspond to the red primary, c equal to 1 should correspond to the magenta secondary, c equal to 2 should correspond to the blue primary, c equal to 3 should correspond to the cyan secondary, c equal to 4 should correspond to the green primary, c equal to 5 should correspond to the yellow secondary.

D.2.3 Definitions

D.2.3.1 Introduction

Clause D.2.3.2 to clause D.2.3.7 provide definitions of concepts and terms that are extensively used in the specification of the forward and inverse gamut mapping processes.

D.2.3.2 Line defined by two points

In a three-dimensional space, the parametric definition of a line defined by two points is generally formulated as follows:

Let A and B of coordinates represented by a triplet (y,u,v) denotes two separate points. If the point P belongs to the (AB) line, the parameter t is specified as follows:

$$\overrightarrow{AP} = t \times \overrightarrow{AB} \quad \text{(D.2)}$$

As A and B are two separate points, at least one of their three coordinate are distinct. For example, if $A_y \neq B_y$, then t is computed using the coordinates of A, B and P as follows:

$$t = \frac{(P_y - A_y)}{(B_y - A_y)} \quad \text{(D.3)}$$

In a two-dimensional space, let A and B of coordinates represented by a duet (u,v) or (ρ, y); a line is defined by an equation generally formulated as:

$$a \times u + b \times v + c = 0 \quad \text{(D.4)}$$

where:

$$a = A_v - B_v \quad \text{(D.5)}$$

$$b = B_u - A_u \quad \text{(D.6)}$$

$$c = A_u \times B_v - A_v \times B_u \quad \text{(D.7)}$$

NOTE: This formulation enables the representation of vertical lines.

D.2.3.3 Intersections

The intersection of a segment (defined by a primary colour and a secondary colour) with a plane (defined by a given constant lightness) is computed with equations (D.2) and (D.3) of clause D.2.3.2 considering that:

- the points A and B are respectively the primary colour and secondary colour which define the segment;

- the point P is the intersection of the segment and the plane.

As the primary and secondary colours have different lightness, t is computed using the lightness as illustrated by equation (D.3).

The intersection of two lines in a two-dimensional space is specified as follows:

Considering:

- a first line of equation $a_1 \times u + b_1 \times v + c_1 = 0$,
- a second line of equation $a_2 \times u + b_2 \times v + c_2 = 0$,
- P, the intersection point of the first and second lines, represented by the homogeneous coordinate triplet (u, v, w) with $\frac{u}{w} = P_u$, $\frac{v}{w} = P_v$ and w = 1.

NOTE: Homogeneous coordinates allows to compute intersection of a vertical line with a non-vertical line.

Then, the homogeneous coordinates of P are as follows:

$$u = b_1 \times c_2 - b_2 \times c_1 \quad \text{(D.8)}$$

$$v = a_2 \times c_1 - a_1 \times c_2 \quad \text{(D.9)}$$

$$w = a_1 \times b_2 - a_2 \times b_1 \quad \text{(D.10)}$$

- If $w = 0$, the first and the second lines do not intersect (they are parallel).
- Otherwise, the first and the second lines intersect in P with:

$$P_u = \frac{b_1 \times c_2 - b_2 \times c_1}{a_1 \times b_2 - a_2 \times b_1} \quad \text{(D.11)}$$

$$P_v = \frac{a_2 \times c_1 - a_1 \times c_2}{a_1 \times b_2 - a_2 \times b_1} \quad \text{(D.12)}$$

D.2.3.4 Cusp

The cusp colour $Cusp(\mathrm{P}, \mathcal{G})$ of a colour sample P belonging to the gamut $\mathcal{G}$ is the most saturated colour (i.e. the colour with the largest chrominance value in the linear-light YUV colour space).

The cusp line $\mathcal{G}^{cusp}$ is the collection of all the cusp colours belonging to the gamut $\mathcal{G}$ (see Figure D.2).

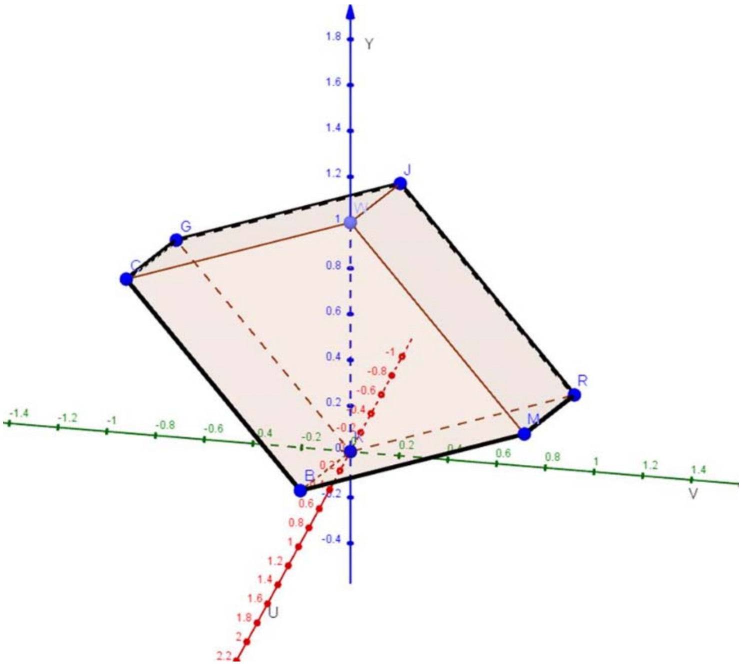

Figure D.2: Illustration of a cusp line of a gamut in linear-light YUV colour space (thick black line)

The projection of a cusp line $\mathcal{G}^{cusp}$ on the plane $Plane_{(y=0)}$ is denoted $(g^R g^M g^B g^C g^G g^J)$ and represented by a hexagon with the primary and secondary colours as vertices (see Figure D.3). This hexagon is made of the following segments ordered clockwise $[\mathcal{G}^R\mathcal{G}^M[$, $[\mathcal{G}^M\mathcal{G}^B[$, $[\mathcal{G}^B\mathcal{G}^C[$, $[\mathcal{G}^C\mathcal{G}^G[$, $[\mathcal{G}^G\mathcal{G}^J[$ and $[\mathcal{G}^J\mathcal{G}^R[$.

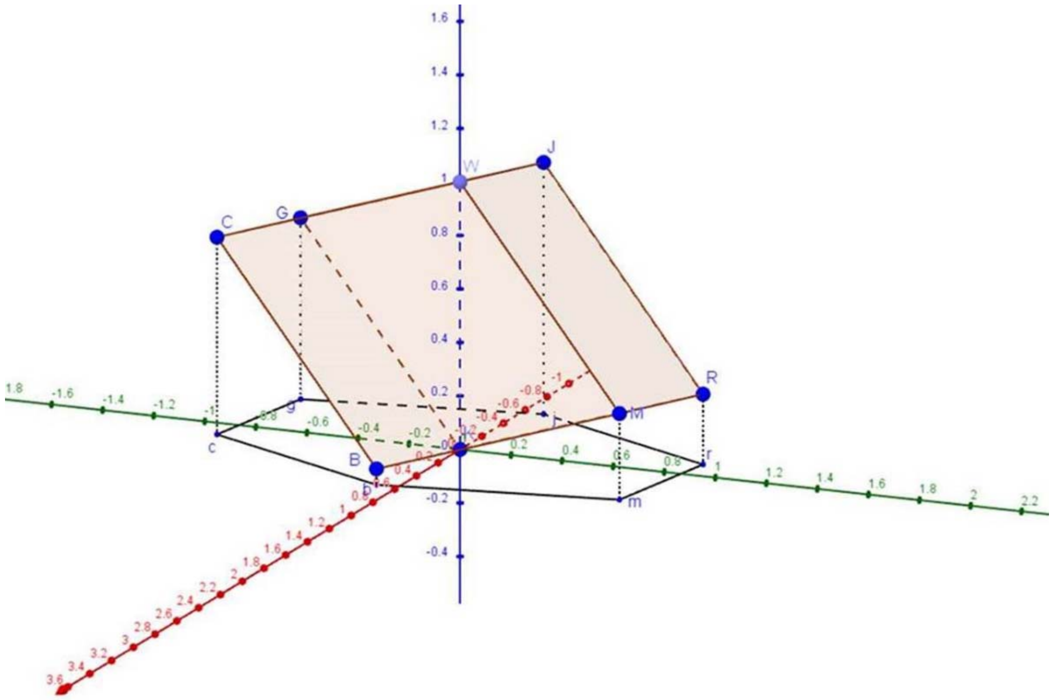

Figure D.3: Illustration of the projection of a cusp line on the UV plane (black hexagon)

D.2.3.5 Sector

The projection of the cusp line $\mathcal{G}^{cusp}$ on the plane $Plane_{(y=0)}$ is a hexagon. This hexagon can be partitioned in six triangles which vertices are:

- the origin of the plane (black point);
- one primary colour; and
- one adjacent secondary colour.

The sector $Sector(\mathrm{P}, \mathcal{G}^{cusp})$ of a colour sample P is the triangle containing the projection of P on the plane $Plane_{(y=0)}$, see Figure D.4.

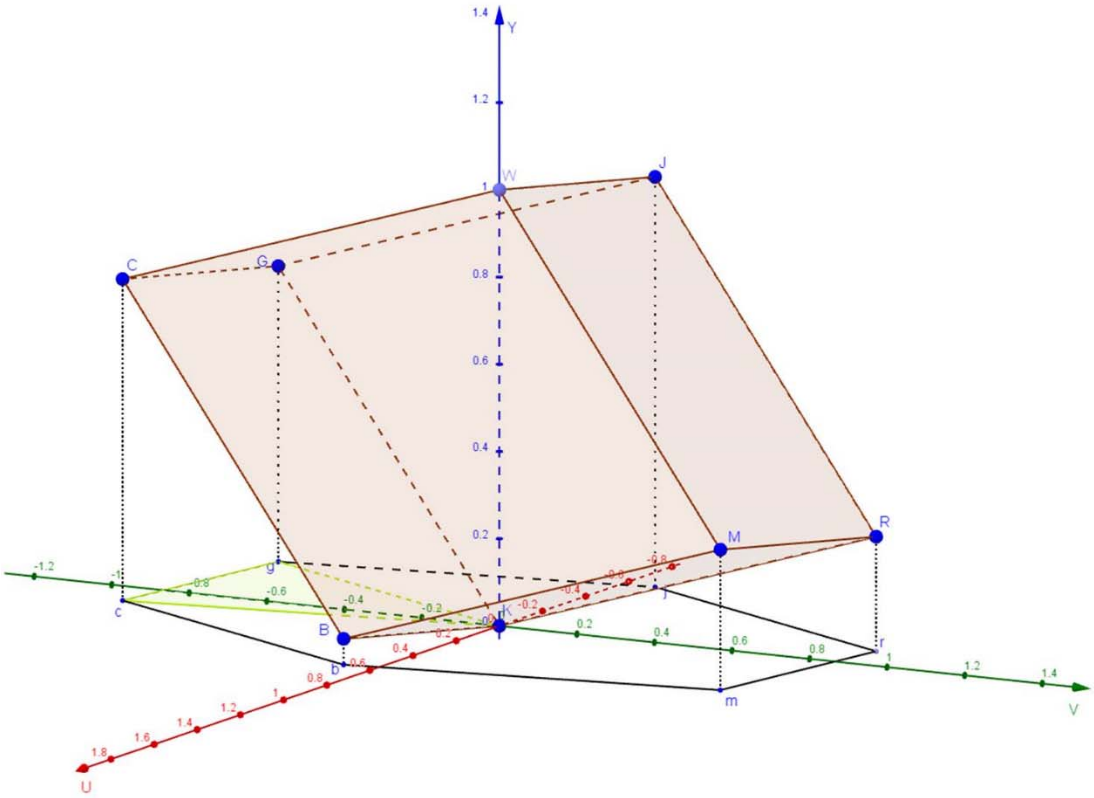

Figure D.4: Illustration of a sector of the UV plane (light green triangle with c, g, o as vertices)

The following variables and notation allow defining the sector $Sector(\mathrm{P}, \mathcal{G}^{cusp})$:

- P, a colour sample in the linear-light YUV colour space with:
  - p, the projection of P on the UV plane (e.g. $Plane_{(y=0)}$);
  - o, the origin of the UV plane;
- $(g^R g^M g^B g^C g^G g^J)$, a projection of the cusp line $\mathcal{G}^{cusp}$ on the UV plane;
- β^i, the result of a cross-product;

NOTE: The cross-product helps determining the position of p against the line (og^i).

- beg and end, are the two vertices defining a sector.

$Sector(\mathrm{P}, \mathcal{G}^{cusp})$ computation is specified in the following pseudo-code:

```
for(i = 0; i <= 5; i++)
    β^i = op × og^i;
end = 5;
for (i = 0; i <= 5; i++) {
    if( (β^i ≥ 0) && (β^end < 0) ) {
        beg = i;
        break;
    }
    else
```

```
        end = i;
    }
```

The resulting $Sector(\mathrm{P}, \mathcal{G}^{cusp})$ is defined or characterized by the $[g^{\mathrm{beg}} g^{\mathrm{end}}[$ segment.

D.2.3.6 Computing the cusp colour using a sector

Obtaining the cusp colour of P, with P being a tri-stimulus colour sample (P_y, P_u, P_v) belonging to gamut $\mathcal{G}$, can be achieved by finding first the sector to which P belongs to. This operation is performed in the UV plane (e.g. $Plane_{(y=0)}$) using the projection of the cusp line ($g^R g^M g^B g^C g^G g^J$) on this plane and the projection p of P on the UV plane with o the origin of the UV plane.

The cusp colour of P can be computed as follows:

- Compute the intersection γ of the line (op) with the segment $[g^{\mathrm{beg}} g^{\mathrm{end}}[$:

$$\gamma = (\mathrm{op}) \cap [g^{\mathrm{beg}} g^{\mathrm{end}}[\qquad \text{(D.13)}$$

NOTE: γ is the projection of $Cusp(\mathrm{P}, \mathcal{G})$ on the plane $Plane_{(y=0)}$.

- Compute the parametric coordinate t as follows:

$$\overrightarrow{g^{\mathrm{beg}}\gamma} = t \times \overrightarrow{g^{\mathrm{beg}} g^{\mathrm{end}}} \qquad \text{(D.14)}$$

$$t = \frac{\left\| \overrightarrow{g^{\mathrm{beg}}\gamma} \right\|}{\left\| \overrightarrow{g^{\mathrm{beg}} g^{\mathrm{end}}} \right\|}) \qquad \text{(D.15)}$$

- Finally, determine the cusp of $\mathrm{P}(\mathrm{y}, \mathrm{u}, \mathrm{v})$ for each component:

$$Cusp(\mathrm{P}, \mathcal{G})_u = \gamma_u \qquad \text{(D.16)}$$

$$Cusp(\mathrm{P}, \mathcal{G})_v = \gamma_v \qquad \text{(D.17)}$$

$$Cusp(\mathrm{P}, \mathcal{G})_y = \mathcal{G}_y^{\mathrm{beg}} \times (1 - t) + \mathcal{G}_y^{\mathrm{end}} \times t \qquad \text{(D.18)}$$

D.2.3.7 Boundary

The boundary line $\mathcal{G}_{(y=P_y)}^{bound}$ is defined as the collection of the most saturated colours for the gamut $\mathcal{G}$ intersected by the plane $Plane_{(y=P_y)}$. Leveraging the boundary line $\mathcal{G}_{(y=\mathrm{P_y})}^{bound}$, the sector $Sector(\mathrm{P}, \mathcal{G}_{(y=\mathrm{P_y})}^{bound})$ can be computed in a similar way as for $Sector(\mathrm{P}, \mathcal{G}^{cusp})$ (see clause D.2.3.5).

The boundary colour $Bound(\mathrm{P}, \mathcal{G})$ computation is based on the same process as computing $Cusp(\mathrm{P}, \mathcal{G})$ (clause D.2.3.6) by replacing $\mathcal{G}^{cusp}$ with $\mathcal{G}_{(y=\mathrm{P_y})}^{bound}$.

In the linear-light YUV colour space, the shape of a RGB gamut can be represented by a hexahedron. This shape is transformed during the conversions required by the gamut mapping process and its inverse process. During the gamut mapping process (and its inverse process), the hue is rotated with a hue dependent rotation angle. Similarly, the lightness is shifted. This results in transforming the hexahedron in a dodecahedron, as illustrated in the Figure D.5.

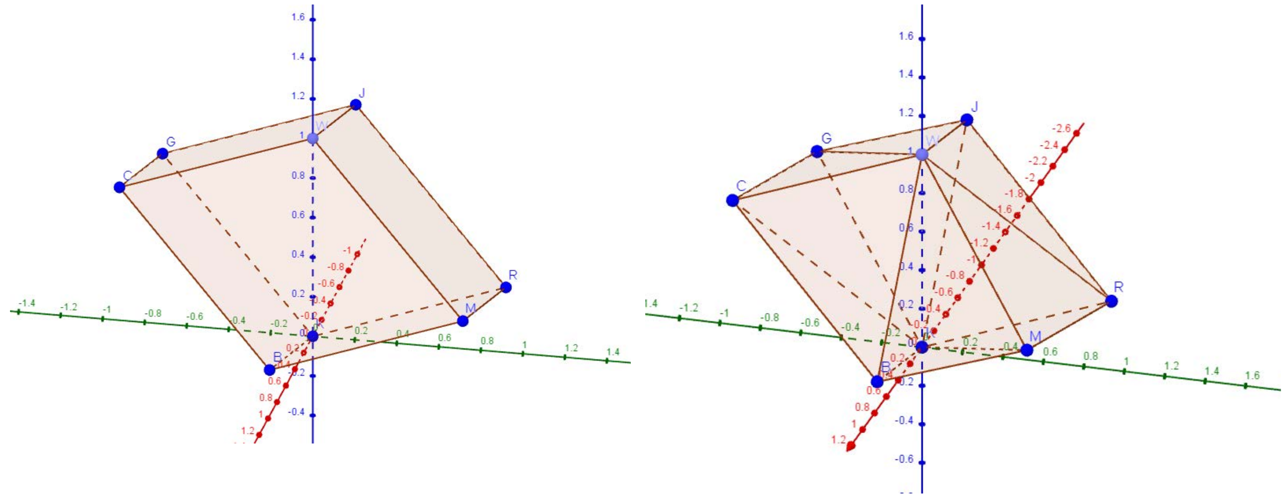

Figure D.5: Views of a hexahedron gamut (left) and a dodecahedron gamut (right)

Similarly to the cusp line, the boundary line is a polygon made of three to five segments when the gamut is a hexahedron or five to ten segments when the gamut is a dodecahedron.

The vertices of $\mathcal{G}^{bound}_{(y=\mathrm{P_y})}$ are determined by computing the intersection of the gamut polygon edges (twelve for a hexahedron and eighteen for a dodecahedron) with the $Plane_{(y=\mathrm{P_y})}$. The only edges to be considered are the edges simultaneously having one vertex above $\mathrm{P_y}$ and a second vertex below $\mathrm{P_y}$.

In practice, when the gamut is a hexahedron, the edges to be used to compute the edges intersection with the $Plane_{(y=\mathrm{P_y})}$ are documented in Table D.1. These intersections define the vertices of the boundary polygon in $Plane_{(y=\mathrm{P_y})}$.

Table D.1: Determination of boundary colour - hexahedron case

Value of y	Edge 1	Edge 2	Edge 3	Edge 4	Edge 5
y = K	-	-	-	-	-
K < y ≤ B	KG	KR	KB	-	-
B < y ≤ R	MB	BC	KG	KR	-
R < y < M	JR	RM	MB	BC	KG
M ≤ y ≤ G	JR	WM	BC	KG	-
G < y < C	CG	GJ	JR	WM	BC
C ≤ y < J	WC	GJ	JR	WM	-
J ≤ y < W	WC	WJ	WM	-	-
y = W	-	-	-	-	-
NOTE: The notation XY should be read $[\mathcal{G}^X\mathcal{G}^Y[$. Edges lists is ordered clockwise.					

In practice, when the gamut is a dodecahedron, the edges to be used to compute the edges intersection with the $Plane_{(y=\mathrm{P_y})}$ are documented in Table D.2. These intersections define the vertices of the boundary polygon in $Plane_{(y=\mathrm{P_y})}$.

Table D.2: Determination of boundary colour - dodecahedron case

Value of y	Edge 1	Edge 2	Edge 3	Edge 4	Edge 5	Edge 6	Edge 7	Edge 8	Edge 9	Edge 10
y = K	-	-	-	-	-	-	-	-	-	-
K < y ≤ B	KB	KC	KG	KJ	KR	KM	-	-	-	-
B < y ≤ R	WB	BC	KC	KG	KJ	KR	KM	MB	-	-
R < y < M	WB	BC	KC	KG	KJ	JR	WR	RM	KM	MB
M ≤ y ≤ G	WB	BC	KC	KG	KJ	JR	WR	WM	-	-
G < y < C	WB	BC	KC	CG	WG	GJ	KJ	JR	WR	WM
C ≤ y < J	WB	WC	WG	GJ	KJ	JR	WR	WM	-	-
J ≤ y < W	WB	WC	WG	WJ	WR	WM	-	-	-	-
y = W	-	-	-	-	-	-	-	-	-	-
NOTE: The notation XY should be read $[\mathcal{G}^X\mathcal{G}^Y[$. Edges lists is ordered clockwise.										

D.3 Forward gamut mapping process

D.3.1 Introduction

This clause specifies the forward gamut mapping process enabling the generation of a standard colour gamut picture from a wide colour gamut picture according to a set of associated dynamic metadata.

The forward gamut mapping documented in this annex aims at compressing a wider mastering display colour gamut (WCG) $\mathcal{W}$ into a standard colour gamut (SCG) $\mathcal{S}$. For this purpose, the following colour gamuts are introduced during the conversion stages: a wide gamut $\mathcal{K}$, a preserved gamut $\mathcal{P}$, a rotated gamut $\mathcal{R}$, a lightness mapped gamut $\mathcal{L}$, a warped lightness mapped gamut $\tilde{\mathcal{L}}$ and a truncated gamut $\mathcal{T}$. The gamut is transformed during the forward gamut mapping stages as described below:

$$\mathcal{K} \rightarrow \mathcal{W} \rightarrow \mathcal{R} \rightarrow \tilde{\mathcal{L}} \rightarrow \mathcal{T} \rightarrow \mathcal{S} \quad \text{(D.19)}$$

This process is defined for a 4:4:4 chroma sampling and full range YUV linear-light signal. In a typical implementation of a SL-HDR1 pre-processing stage with gamut mapping, the input of the system is a Y'CbCr 4:2:2 chroma sampling legal range with a transfer function such as SMPTE ST 2084 [1]. Several signal conversions are occurring to derive the linear-light 4:4:4 YUV, notably:

- 4:2:2 to 4:4:4 chroma upsampling conversion;
- conversion to R'G'B' colour space;
- removal of the transfer function to obtain a linear-light RGB signal;
- conversion to linear-light YUV 4:4:4.

In Figure D.6 that depicts the gamut mapping process and the associated gamuts these conversions are called "video content adaptation".

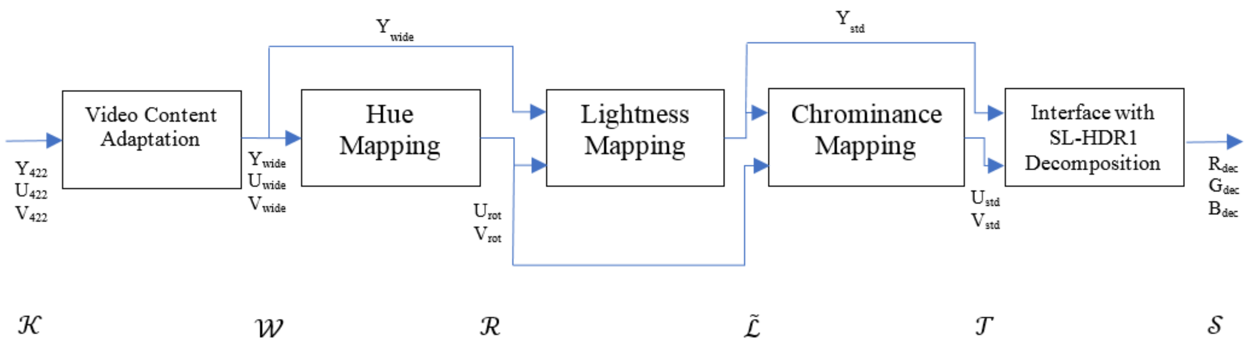

Figure D.6: Overview of the forward gamut mapping process and the associated gamuts

The forward gamut mapping process can be summarized by the following successive steps (for each pixel x = 0..(*picWidth - 1*), y = 0..(*picHeight - 1*)):

- The input 4:2:2 Y'CbCr signal is converted to a linear-light YUV signal (see clause D.3.2) and the colour gamut is converted from $\mathcal{K}$ to $\mathcal{W}$.
- The hue is compensated (see clause D.3.3) and the colour gamut is converted from $\mathcal{W}$ to $\mathcal{R}$.
- The lightness is mapped (see clause D.3.4) and the colour gamut is converted from $\mathcal{R}$ to $\tilde{\mathcal{L}}$.
- The chrominance is compressed (see clause D.3.5) and the colour gamut is converted from $\tilde{\mathcal{L}}$ to $\mathcal{T}$.
- The standard gamut mapped signal is input in SL-HDR1 decomposition stage (see clause D.3.6) and the colour gamut is converted from $\mathcal{T}$ to $\mathcal{S}$.

Due to the reversible design of the gamut mapping documented in Annex D, processes specified in the clauses D.3.3 to D.3.6 are highly similar to those used by the reverse processes.

During the forward gamut mapping phase, parameters are stored in the variables specified in clause 6.3.9 while in the inverse gamut mapping phase, parameters are derived from those variables.

D.3.2 Video content adaptation

This process takes as inputs:

- the luma and chroma values Y_{422}, U_{422} and V_{422} in the colour gamut $\mathcal{K}$,
- the variables related to colour spaces: **sdrPicColourSpace, hdrPicColourSpace** and **hdrDisplayColourSpace**.

The process generates as output:

- the adapted lightness and chrominance components values Y_{wide}, U_{wide} and V_{wide} of the colour gamut $\mathcal{W}$.

The block diagram depicted in Figure D.7 summarizes the different processing blocks that may be used to prepare the content to be ingested into the actual forward gamut mapping process. The video content adaptation process is composed of bricks either documented in existing recommendations or standards. As an example, the draft recommendation JCTVC-Z1017 [i.11] proposes in clause 10 details to adapt such a video content.

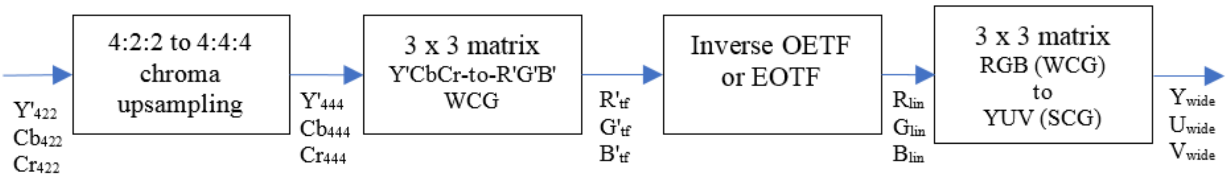

Figure D.7: Overview of the input video adaptation

- The first block entitled "4:2:2 to 4:4:4 chroma upsampling" is a 4:2:2 to 4:4:4 Y'CbCr chroma upsampling. The input Y'CbCr 4:2:2 signal is contained in the $\mathcal{K}$ colour gamut (specified by **hdrPicColourSpace**) with $\mathcal{W}$ (specified by **hdrDisplayColourSpace**) as a colour volume.
- The second block named "3x3 matrix Y'CbCr-to-R'G'B' WCG" is a canonical Y'CbCr-to-R'G'B' matrix operating in the colour space corresponding to the $\mathcal{K}$ colour gamut**.**
- The third block called 'Inverse OETF or EOTF" is removing the transfer function of the R'G'B signal to produce a RGB linear-light (display-referred or scene referred) value. For instance, the input 4:2:2 HDR signal transfer function is the SMPTE ST 2084 [1] transfer function environment. However, it is worth noting that there is not limitation in the HDR transfer function that can be used.

- The last block which name is "3x3 matrix RGB (WCG) to YUV (SCG)" converts a linear-light RGB signal represented in the colour space whose colour gamut is $\mathcal{K}$ to a linear-light YUV signal represented in a colour space whose colour gamut is $\mathcal{S}$ (specified by **sdrPicColourSpace**). The fourth block matrix is a compilation of matrix that is detailed in the following equation:

$$\begin{bmatrix} Y_{wide} \\ U_{wide} \\ V_{wide} \end{bmatrix} = M^{\mathcal{S}}_{RGB-to-YUV} \times M^{\mathcal{S}}_{XYZ-to-RGB} \times M^{\mathcal{K}}_{RGB-to-XYZ} \times \begin{bmatrix} R_{lin} \\ G_{lin} \\ B_{lin} \end{bmatrix} \quad \text{(D.20)}$$

where:

- $M^{\mathcal{K}}_{RGB-to-XYZ}$ is a canonical 3x3 RGB to XYZ matrix (computed with **hdrDisplayColourSpace** and SMPTE RP 177 [i.9]);
- $M^{\mathcal{S}}_{XYZ-to-RGB}$ is a canonical 3x3 XYZ to RGB matrix (computed with **sdrPicColourSpace** and SMPTE RP 177 [i.9]);
- $M^{\mathcal{S}}_{RGB-to-YUV}$ is a canonical 3x3 R'G'B' to Y'CbCr matrix (computed with **sdrPicColourSpace**. In this version of the present document, this matrix is specified in Recommendation ITT-R BT.709-6 [6]).

NOTE: Unlike the conventional usage where the canonical R'G'B' to Y'CbCr matrix ($M^{\mathcal{S}}_{RGB-to-YUV}$) is applied on a signal that is compressed by a transfer function, in the process described in this clause and in clause D.4.2 the matrix is applied on a on a RGB linear-light signal.

All the computations that are documented in the clauses D.3.3 to D.3.6 are performed in the standard colour space (frame of reference) even though the converted gamut and signals may exceed the gamut of the standard colour space.

D.3.3 Hue mapping

D.3.3.1 Introduction

This process takes as inputs:

- the lightness and chrominance components values Y_{wide}, U_{wide} and V_{wide} in the colour gamut $\mathcal{W}$ generated in clause D.3.2;
- the variables related to hue remapping: **hueAdjMode**, **hueGlobalPreservationRatio** and **huePreservationRatio**[c];
- the variables related to hue alignment: **hueAlignCorrectionPresentFlag** and **hueAlignCorrection**[i];
- the variables related to chrominance adjustment: **chromAdjPresentFlag** and **chromAdjParam**[i].

The process generates as output:

- the hue mapped chrominance components values U_{rot} and V_{rot} which with the lightness Y_{wide} reside in the colour gamut $\mathcal{R}$.

The hue mapping process is designed to optimize the rendition of highly saturated colours when compressing them from the wide gamut to the standard one. This process relies on a hue, chrominance and lightness dependent hue rotation.

The hue mapping process consists in transforming the wide gamut $\mathcal{W}$ (specified by **hdrDisplayColourSpace**) into a rotated gamut $\mathcal{R}$. Because of the input video adaptation process (see clause D.3.2), the wide colour gamut $\mathcal{W}$ is represented into the standard colour space which gamut is denoted by $\mathcal{S}$ (specified by **sdrPicColourSpace**).

In a first time, the primary and secondary colours of the gamut $\mathcal{W}$ are rotated as specified in clause D.3.3.2. During the forward gamut mapping phase, it is possible to define a preserved gamut $\mathcal{P}$ (specified in clause D.3.3.4) in which the colours remain unchanged during the hue mapping process. Then, clauses D.3.3.3 and D.3.3.5 respectively specify the hue mapping of colours inside $\mathcal{W}$ without or with the consideration of a preserved area.

D.3.3.2 Deriving the rotated gamut

The parameters of the conversion of the wide gamut $\mathcal{W}$ into the rotated gamut $\mathcal{R}$ expressed in the standard colour space are derived from the variables **hueAlignCorrectionPresentFlag**, **hueAlignCorrection**[i], **chromAdjPresentFlag** and **chromAdjParam**[i]. The hue rotation angle $\Delta\theta^i$ as well as the chrominance gain ε^i are computed as follows:

- Compute for each primary and secondary colour $\Delta\theta^i$, the hue rotation angle, as follows:
  - Compute the hue angles $\mathcal{W}_\theta^i$ and $\mathcal{S}_\theta^i$ (Cartesian coordinates to cylindrical coordinates conversion).
  - Derive the rotation angle $\Delta\theta^i$, that enables the derivation of the rotated gamut $\mathcal{R}$, from the hue angles $\mathcal{W}_\theta^i$ and $\mathcal{S}_\theta^i$ as specified by the following pseudo-code:

```
for (i = 0; i ≤ 5; i++){
    if (hueAlignCorrectionPresentFlag)
        Hue_index = hueAlignCorrection[ i ];
    else
        Hue_index = Hue_default[ i ];
    Δθ^i = (S_θ^i − W_θ^i) / Hue_ratio[ i ] × (Hue_index − Hue_offset[ i ]);
}
```

with $Hue_{default}[\ i\ ]$, $Hue_{ratio}[\ i\ ]$ and $Hue_{offset}[\ i\ ]$ specified in Table D.3.

Table D.3: Hue alignment correction factors

Primary/secondary colour (i)	$Hue_{ratio}[i]$	$Hue_{offset}[i]$	$Hue_{default}[i]$
0(red)	2	2	4
1 (magenta)	4	1	5
2 (blue)	2	2	4
3 (cyan)	2	2	4
4 (green)	4	1	5
5 (yellow)	2	2	4

- Compute for each primary and secondary colour ε^i, the chrominance gain, as specified by the following pseudo-code:

```
for (i = 0; i ≤ 5; i++){
    if (chromAdjPresentFlag)
        ε^i = chromAdjParam[ i ]
    else
        ε^i = 1;
}
```

- Then, the rotated gamut $\mathcal{R}$ is derived as specified by the following pseudo-code:

```
for (i = 0; i ≤ 5; i++){
    if (hueAdjMode){
        R_u^i = ε^i × (W_u^i × cos Δθ^i − W_v^i × sin Δθ^i)
        R_v^i = ε^i × (W_u^i × sin Δθ^i + W_v^i × cos Δθ^i)
    }
    else{
        R_u^i = W_u^i
        R_v^i = W_v^i
    }
}
```

with $\mathcal{R}^K = \mathcal{W}^K$ and $\mathcal{R}^W = \mathcal{W}^W$.

D.3.3.3 Hue mapping without preserved area

When this process is applied, it is stored as **hueAdjMode** equal to 1.

The hue mapping without preserved area process rotates the wide colour gamut $\mathcal{W}$ into the rotated colour gamut $\mathcal{R}$ (see Figure D.8). The primary and secondary colours of $\mathcal{W}$ are respectively projected to the primary and secondary colours of $\mathcal{R}$ as specified by the following pseudo-code:

```
for (i = 0 ; i ≤ 5 ; i++)
```

$\mathcal{W}^i \rightarrow \mathcal{R}^i$

The colours in-between the primary and secondary colours are projected linearly depending on the hue and chrominance as specified below.

Assuming an input YUV triplet *(P_y, P_u, P_v)* equal to *(Y_{wide}, U_{wide}, V_{wide})*, the computation is as follows:

- Compute the cusp line $\mathcal{W}^{cusp}$ (see clause .2.3.4 and D.2.3.6). This is a hexagon resulting of the projection on a $Plane_{(y\,=\,0)}$ of the six segments $[\mathcal{W}^R\mathcal{W}^M[$, $[\mathcal{W}^M\mathcal{W}^B[$, $[\mathcal{W}^B\mathcal{W}^C[$, $[\mathcal{W}^C\mathcal{W}^G[$, $[\mathcal{W}^G\mathcal{W}^J[$, $[\mathcal{W}^J\mathcal{W}^R[$. Lowercase notation denotes the projected segment: e.g. $[\mathcal{W}^B\mathcal{W}^C[$ is projected as $[w^B w^C[$ on the $Plane_{(y\,=\,0)}$.
- Determine to which sector $Sector(\mathrm{P}, \mathcal{W}^{cusp})$ of this w hexagon the hue of P belongs to.

NOTE: This means that if the hue of P is between blue and cyan, the sector will be $[w^B w^C[$. More generally this hue-selected sector is denoted by $[w^{\mathrm{beg}} w^{\mathrm{end}}[$.

In $Plane_{(y=\,0)}$, o denotes the origin of the plane of coordinates $o_u = 0$ and $o_v = 0$. Lowercase notation denotes the projection of P on this plane.

Compute the intersection q of the line (op) with the segment $[w^{\mathrm{beg}} w^{\mathrm{end}}[$ (see clause D.2.3.3) as follows:

$$\mathrm{q} = (\mathrm{op}) \cap \ [w^{\mathrm{beg}} w^{\mathrm{end}}[\tag{D.21}$$

- Compute the parametric coordinates of q, t and k such that:

$$\overrightarrow{op} = \mathrm{t} \times \overrightarrow{oq} \tag{D.22}$$

$$\overrightarrow{w^{\mathrm{beg}} q} = \ \mathrm{k} \times \overrightarrow{w^{\mathrm{beg}} w^{\mathrm{end}}} \tag{D.23}$$

- The cusp displacement vector $\vec{\delta}$ is determined as follows:

$$\vec{\delta} = \ \mathrm{k} \times \overrightarrow{w^{\mathrm{end}} r^{\mathrm{end}}} + (1 - \mathrm{k}) \times \overrightarrow{w^{\mathrm{beg}} r^{\mathrm{beg}}} \tag{D.24}$$

- Compute h the rotation of p such that:

$$\overrightarrow{oh} = \overrightarrow{op} + \mathrm{t} \times \vec{\delta} \tag{D.25}$$

- The output of this hue mapping process without preserved area is the triplet *(Y_{wide}, U_{rot}, V_{rot})* such that:

$$\mathrm{Y_{wide}} = \mathrm{P_y} \tag{D.26}$$

$$\mathrm{U_{rot}} = \mathrm{h_u} \tag{D.27}$$

$$\mathrm{V_{rot}} = \mathrm{h_v} \tag{D.28}$$

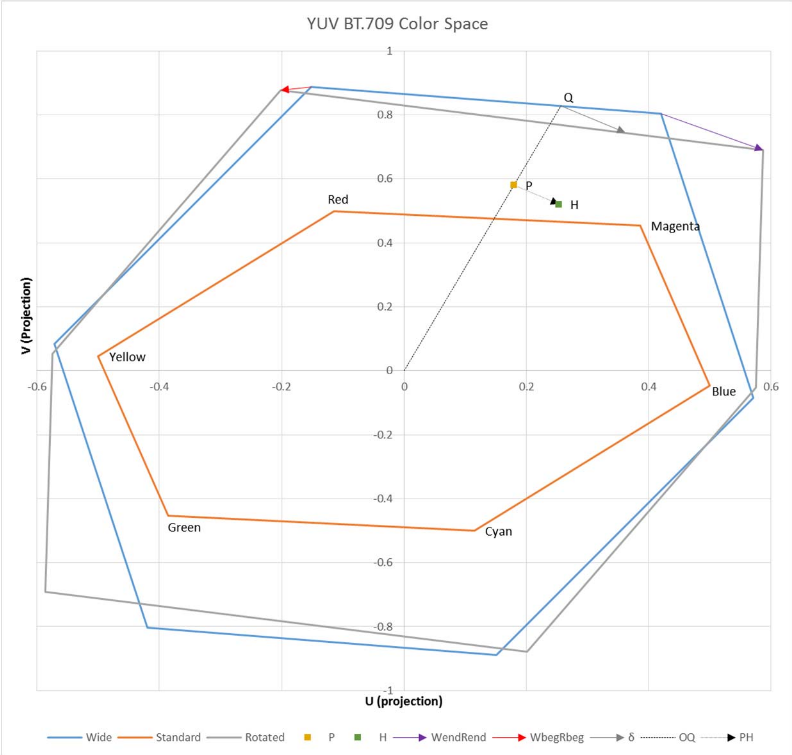

Figure D.8: Principle of the hue mapping without preserved area

D.3.3.4 Deriving the preserved gamut

The preserved gamut $\mathcal{P}$ is derived from the wide gamut $\mathcal{W}$ and from **hueAdjMode**, **hueGlobalPreservationRatio** and **huePreservationRatio**[i] variables, as follows:

```
switch (hueAdjMode){
    case 1: // no preserved area
        break;
    case 2:// global preservation ratio
        for(i = 0; i ≤ 5; i++){
            $\mathcal{P}_y^i = \mathcal{W}_y^i$;
            $\mathcal{P}_u^i$ = hueGlobalPreservationRatio × $\mathcal{W}_u^i$;
            $\mathcal{P}_v^i$ = hueGlobalPreservationRatio × $\mathcal{W}_v^i$;
        }
        break;
    case 3: // preservation ratio per primary/secondary colour
        for(i = 0; i ≤ 5; i++) {
            $\mathcal{P}_y^i = \mathcal{W}_y^i$;
            $\mathcal{P}_u^i$ = huePreservationRatio[ i ] × $\mathcal{W}_u^i$;
            $\mathcal{P}_v^i$ = huePreservationRatio[ i ] × $\mathcal{W}_v^i$;
        }
        break ;
}
```

D.3.3.5 Hue mapping with preserved area

When this process is applied, it is stored as **hueAdjMode** equal to 2 when the hue mapping process occurs with a globally preserved area or 3 when the hue mapping process occurs with preservation of areas per primary and secondary colours.

The hue mapping (also known as hue adjustment) with preserved area process rotates the wide colour gamut $\mathcal{W}$ into the rotated colour gamut $\mathcal{R}$ while leaving unchanged colours belonging to a preserved area that is named the preserved colour gamut $\mathcal{P}$ (see Figure D.9). Note that the rotated colour gamut $\mathcal{R}$ is represented by a dodecahedron in the linear-light YUV colour space due to the rotation with preservation of a colour area. This dodecahedron has the same height corners than the hexahedron representing the rotated colour gamut without preserved area i.e. $\{ \mathcal{R}^K, \mathcal{R}^R, \mathcal{R}^M, \mathcal{R}^B, \mathcal{R}^C, \mathcal{R}^G, \mathcal{R}^J, \mathcal{R}^W \}$ but its faces are no longer six coplanar quadrilaterals {(KRBM), (KBGC), (KRGJ), (WCMB), (WJMR), (WJCG)} but twelve triangles {(KRM), (KBM), (KBC), (KGC), (KRJ), (KGJ), (WCB), (WMB), (WJR), (WMR), (WJG), (WCG) }.

NOTE 1: The notation of the quadrilaterals and triangles have been simplified by removing the related gamut notation e.g. from $(\mathcal{R}^x\mathcal{R}^y\mathcal{R}^z)$ to (XYZ).

In this clause, both the wide colour gamut $\mathcal{W}$ and the preserved colour gamut $\mathcal{P}$ are considered as dodecahedrons, even if they actually are hexahedron.

Assuming an input YUV triplet (P_y, P_u, P_v) equal to $(Y_{wide}, U_{wide}, V_{wide})$, the computation is as follows:

- Compute the boundary line $\mathcal{W}_{(y=\mathrm{P_y})}^{bound}$ (see clause D.2.3.7) as follows:

$$\mathcal{W}_{(y=\mathrm{P_y})}^{bound} = \mathcal{W} \cap Plane_{(y\,=\,\mathrm{P_y})} \quad \text{(D.29)}$$

  where $\mathcal{W}_{(y=\mathrm{P_y})}^{bound}$ is a convex polygon with between three and ten edges depending on the value of $\mathrm{P_y}$. Some edges may be collinear because $\mathcal{W}$ is a hexahedron considered as a dodecahedron and all the vertices from the dodecahedron are necessary in the computation.

NOTE 2: In the linear-light YUV colour space, $\mathcal{W}$ is considered as a degenerated dodecahedron defined by its height corners $\{ \mathcal{W}^K, \mathcal{W}^R, \mathcal{W}^M, \mathcal{W}^B, \mathcal{W}^C, \mathcal{W}^G, \mathcal{W}^J, \mathcal{W}^W \}$ and its twelve faces {(KRM), (KBM), (KBC), (KGC), (KRJ), (KGJ), (WCB), (WMB), (WJR), (WMR), (WJG), (WCG) }.

- Determine to which sector of the polygon $\mathcal{W}_{(y=\mathrm{P_y})}^{bound}$ p belongs to (see clause D.2.3.5).

NOTE 3: A sector of the polygon $\mathcal{W}_{(y=\mathrm{P_y})}^{bound}$ is defined by the triangle $(\mathrm{o}, \mathcal{w}^{\mathrm{beg}}, \mathcal{w}^{\mathrm{end}})$, o being the origin of the $Plane_{(y\,=\,\mathrm{P_y})}$.

- Compute the boundary line of both $\mathcal{R}$ and $\mathcal{P}$ as follows:

$$\mathcal{R}_{(y=P_y)}^{bound} = \mathcal{R} \cap Plane_{(y\,=\,\mathrm{P_y})} \quad \text{(D.30)}$$

$$\mathcal{P}_{(y=P_y)}^{bound} = \mathcal{P} \cap Plane_{(y\,=\,\mathrm{P_y})} \quad \text{(D.31)}$$

- Determine the vertices $\mathcal{r}^{\mathrm{beg}}$, $\mathcal{r}^{\mathrm{end}}$, $\mathcal{p}^{\mathrm{beg}}$ and $\mathcal{p}^{\mathrm{end}}$ that define the sector of which p projection of P belongs to where beg and end are two vertices of the dodecahedron intersected by the plane $Plane_{(y\,=\,\mathrm{P_y})}$ (see clause D.2.3.5).

NOTE 4: The gamut boundaries $\mathcal{W}_{(y=\mathrm{P_y})}^{bound}$, $\mathcal{R}_{(y=\mathrm{P_y})}^{bound}$ and $\mathcal{P}_{(y=\mathrm{P_y})}^{bound}$ are polygons with the same number of vertices.

- Check if p is in the preserved area, by computing a cross-product as follows:

$$\mathrm{k} = \overrightarrow{\mathcal{p}^{\mathrm{beg}}p} \times \overrightarrow{\mathcal{p}^{\mathrm{beg}}\mathcal{p}^{\mathrm{end}}} \quad \text{(D.32)}$$

  - If ($k < 0$), the projection p is in the preserved area. Then, the rotated projection of the colour h, is the projection (h = p) and the computation stops for the current projection p.
  - Otherwise (p is not in the preserved area), the computation continues as follows:
    - Compute the anchor α in the $Plane_{(y=\,\mathrm{P_y})}$ as the intersection of line $(\mathcal{p}^{\mathrm{beg}}\mathcal{r}^{\mathrm{beg}})$ and $(\mathcal{p}^{\mathrm{end}}\,\mathcal{r}^{\mathrm{end}})$:

$$\alpha = \left(\mathcal{p}^{\mathrm{beg}}\mathcal{r}^{\mathrm{beg}}\right) \cap \left(\mathcal{p}^{\mathrm{end}}\,\mathcal{r}^{\mathrm{end}}\right) \quad \text{(D.33)}$$

- Compute the rotation centre q that is the intersection of line (op) with the preserved sector segment:

$$q = (op) \cap [\mathscr{p}^{beg}\mathscr{p}^{end}[\quad (D.34)$$

- Compute the projection direction $\vec{\delta}$ in thee steps as described by the following equations:

$$\beta = (oq) \cap [\mathscr{w}^{beg}\mathscr{w}^{end}[\quad (D.35)$$

$$\gamma = (\alpha q) \cap [\mathscr{r}^{beg}\mathscr{r}^{end}[\quad (D.36)$$

$$\vec{\delta} = \overrightarrow{\beta\gamma} \quad (D.37)$$

- Compute, the projection h as follows:

$$h = (\alpha q) \cap (p + \vec{\delta}) \quad (D.38)$$

- The output of the hue mapping with preserved area process is the triplet *(Y_{wide}, U_{rot}, V_{rot})* such that:

$$Y_{wide} = P_y \quad (D.39)$$

$$U_{rot} = h_u \quad (D.40)$$

$$V_{rot} = h_v \quad (D.41)$$

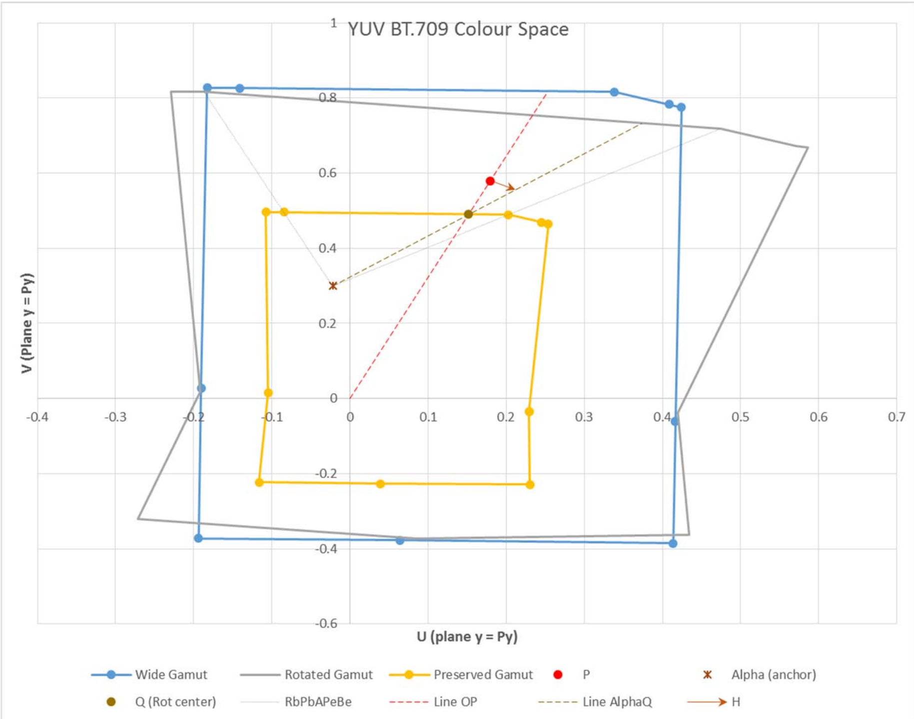

Figure D.9: Hue mapping with preserved area

D.3.4 Lightness mapping

D.3.4.1 Introduction

This process takes as inputs:

- the lightness Y_{wide} and the hue rotated values U_{rot} and V_{rot} in the colour gamut $\mathcal{R}$ generated in clauses D.3.2 and D.3.3,
- the variables related to lightness mapping and cropping: **lightnessMappingMode**, **lmWeightFactor**[i], **croppingModeSCG, cmCroppedLightnessMappingEnabledFlag** and **cmWeightFactor**[i].

The process generates as output:

- the lightness mapped value Y_{std} which combined with U_{rot} and V_{rot} reside in the gamut $\tilde{\mathcal{L}}$.

The lightness mapping stage aims at achieving a trade-off between saturation and lightness for the saturated colours of the wide colour gamut in order to get a better preservation of the intent of the wide colour gamut source. The higher the chrominance the stronger the lightness correction. In this process, the lightness is corrected while keeping the hue and chrominance unchanged.

The lightness mapping parameters offer the possibility to either do a global remapping, or to differentially remap warm and cold colours or to weight the remapping of each individual primary and secondary colour.

First, the lightness cropping process specified in clause D.3.4.2 derives, from the standard gamut $\mathcal{S}$, a lightness cropped gamut $\mathcal{T}$, which is used as an intermediate gamut for the lightness mapping and the chrominance mapping processes.

Then, the lightness mapping weighting factors are computed in clause D.3.4.3.

Finally, the pixel processing is specified in clause D.3.4.4.

D.3.4.2 Deriving the lightness cropped gamut

To avoid or minimize saturation inversion, a cropping of the gamut $\mathcal{S}$ is performed thus resulting in a new gamut $\mathcal{T}$, contained in $\mathcal{S}$. $\mathcal{T}$ is primarily used during the chrominance mapping process (see clause D.3.5) and depending on **cmCroppedLightnessMappingEnabledFlag**, $\mathcal{T}$ may also be used during the lightness mapping process, especially to derive the lightness mapped rotated gamut $\mathcal{L}$ (see clause D.3.4.3).

This is done as follows from $\mathcal{R}$, $\mathcal{S}$, **croppingModeSCG** and **cmWeightFactor**[i]**:**

```
for (i = 0; i ≤ 5; i++){
    switch (croppingModeSCG){
        case 0: // no cropping
            $\mathcal{T}_y^i = \mathcal{S}_y^i$;
            $\mathcal{T}_u^i = \mathcal{S}_u^i$;
            $\mathcal{T}_v^i = \mathcal{S}_v^i$;
            break;
        case 1: // global cropping
            $\mathcal{T}_y^i = \mathcal{R}^{cusp}_{(\theta=\mathcal{S}_\theta^i),y}$;
            if ($\mathcal{T}_y^i \geq \mathcal{S}_y^i$ ){
                $\mathcal{T}_u^i = \mathcal{S}_u^i \times \frac{(1-\mathcal{T}_y^i)}{(1-\mathcal{S}_y^i)}$;
                $\mathcal{T}_v^i = \mathcal{S}_v^i \times \frac{(1-\mathcal{T}_y^i)}{(1-\mathcal{S}_y^i)}$;
            }
            else{
                $\mathcal{T}_u^i = \mathcal{S}_u^i \times \frac{\mathcal{T}_y^i}{\mathcal{S}_y^i}$;
                $\mathcal{T}_v^i = \mathcal{S}_v^i \times \frac{\mathcal{T}_y^i}{\mathcal{S}_y^i}$;
            }
            break;
        case 2: // cropping for cold colours only
            if ($\mathcal{R}^{cusp}_{(\theta=\mathcal{S}_\theta^i),y} \geq \mathcal{S}_y^i$){
                $\mathcal{T}_y^i = \mathcal{S}_y^i$;
                $\mathcal{T}_u^i = \mathcal{S}_u^i$;
```

```
                $\mathcal{T}_v^i = \mathcal{S}_v^i$;
            else{
                $\mathcal{T}_y^i = \mathcal{R}^{cusp}_{(\theta=\mathcal{S}_\theta^i),y}$;
                $\mathcal{T}_u^i = \mathcal{S}_u^i \times \frac{\mathcal{T}_y^i}{\mathcal{S}_y^i}$;
                $\mathcal{T}_v^i = \mathcal{S}_v^i \times \frac{\mathcal{T}_y^i}{\mathcal{S}_y^i}$;
            }
            break;
        case 3: // weighted cropping
            $\delta = \mathcal{R}^{cusp}_{(\theta=\mathcal{S}_\theta^i),y} - \mathcal{S}_y^i$;
            $\mathcal{T}_y^i = \mathcal{S}_y^i + \delta \times$ cmWeightFactor[ i ];
            if ( $\mathcal{T}_y^i \geq \mathcal{S}_y^i$){
                $\mathcal{T}_u^i = \mathcal{S}_u^i \times \frac{(1-\mathcal{T}_y^i)}{(1-\mathcal{S}_y^i)}$;
                $\mathcal{T}_v^i = \mathcal{S}_v^i \times \frac{(1-\mathcal{T}_y^i)}{(1-\mathcal{S}_y^i)}$;
            }
            else{
                $\mathcal{T}_u^i = \mathcal{S}_u^i \times \frac{\mathcal{T}_y^i}{\mathcal{S}_y^i}$;
                $\mathcal{T}_v^i = \mathcal{S}_v^i \times \frac{\mathcal{T}_y^i}{\mathcal{S}_y^i}$;
            }
            break;
    }
}
```

D.3.4.3 Deriving the lightness mapping weighting factors

The gamut $\mathcal{W}$ has been transformed by the hue mapping process (clause D.3.3) and it has resulted in the rotated gamut $\mathcal{R}$. In this clause, the actual lightness mapping weighting factors ωG[i] are derived from the **lightnessMappingMode,** the **lmWeightFactor**[i] and the gamuts $\mathcal{R}, \mathcal{S}$ and $\mathcal{T}$.

The derivation process of ωG[i] is specified by the following pseudo-code:

```
for (i = 0; i ≤ 5; i++){
    switch (lightnessMappingMode){
        case 0: // no lightness mapping
            ωG[i] = 0 ;
            break;
        case 1: // global lightness mapping
            ωG[i] = 1 ;
            break;
        case 2: // lightness mapping for warm colours only
            $\sigma^i$ =cmCroppedLightnessMappingEnabledFlag ? $\mathcal{T}^{cusp}_{(\theta=\mathcal{R}_\theta^i)}$ : $\mathcal{S}^{cusp}_{(\theta=\mathcal{R}_\theta^i)}$
            if ($\mathcal{R}_y^i > \sigma_y^i$ )
                ωG[i] = 1 ;
            else
                ωG[i] = 0 ;
                break;
        case 3: // weighted lightness mapping
                ωG[i] = lmWeightFactor[ i ];
                break;
    }
}
```

D.3.4.4 Parabolic lightness mapping

The lightness mapping is done using a parabolic equation, transforming the gamut into a new warped gamut.

Assuming L the input colour with coordinates (L_y, L_u, L_v) equal to $(Y_{wide}, U_{rot}, V_{rot})$. The parabolic lightness mapping process is as follows:

- Determine to which sector $Sector(\mathrm{P}, \mathcal{R}^{cusp}) = [\mathcal{r}^{\mathrm{beg}}\mathcal{r}^{\mathrm{end}}[$ the hue of L belongs to. This is done similarly as in clause D.3.3.3, replacing the $\mathcal{w}$ hexagon by the $\mathcal{r}$ hexagon and the point "p" by the point "l" (projection of L on the cusp plane), see equations (D.21), (D.22) and (D.23), ending up with the computation of the "k" ratio, that will be used to interpolate the weighting factor.

- Using "k", compute the weighting factor "ωL" corresponding to L as follows:

$$\omega\mathrm{L} = \omega\mathrm{G}[\mathrm{beg}] \times (1 - k) + \omega\mathrm{G}[\mathrm{end}] \times k \quad \text{(D.42)}$$

- Compute the cusp colours of L relative to the relevant gamuts as follows:

$$\mathrm{N} = (\mathbf{cmCroppedLightnessMappingEnabledFlag})\ ?\ \mathcal{T}^{cusp}_{(\theta=\mathrm{L}_\theta)}\ :\ \mathcal{S}^{cusp}_{(\theta=\mathrm{L}_\theta)} \quad \text{(D.43)}$$

$$\mathrm{Z} = \mathcal{R}^{cusp}_{(\theta=\mathrm{L}_\theta)} \quad \text{(D.44)}$$

- Then derive δL the lightness mapped value of the cusp colour of L as follows:

$$\delta\mathrm{L} = \left(\mathrm{Z}_y - \mathrm{N}_y\right) \times \omega\mathrm{L} \quad \text{(D.45)}$$

- Finally, Q represents the lightness mapped value of L, calculated as follows:

$$\mathrm{Q_y} = \mathrm{L_y} - \delta\mathrm{L} \times \left(\frac{\mathrm{L}_\rho}{\mathrm{Z}_\rho}\right)^2 \quad \text{(D.46)}$$

This process output is as follows (see Figure D.10):

$$\mathrm{Y_{std}} = \mathrm{Q_y} \quad \text{(D.47)}$$

Figure D.10: Parabolic lightness mapping

D.3.5 Chrominance mapping

D.3.5.1 Introduction

This process takes as inputs:

- the lightness Y_{std} and the hue rotated values U_{rot} and V_{rot} in the colour gamut $\mathcal{R}$ generated in clause D.3.3 and lightness mapped in clause D.3.4.4. $\tilde{\mathcal{L}}$ represents the gamut resulting from the lightness mapping of $\mathcal{R}$;
- the variables related to chrominance mapping (also known as saturation compression): **satMappingMode**, **satGlobal1SegRatio**, **satGlobal2SegRatioWCG**, **satGlobal2SegRatioSCG**, **sat1SegRatio**[c]**,** **sat2SegRatioWCG**[c] and **sat2SegRatioSCG**[c].

The process generates as output:

- the mapped chrominance components values U_{std} and V_{std} which combined with Y_{std} reside in the colour gamut $\mathcal{S}$.

The chrominance mapping process consists in compressing the wide gamut that has been transformed by the preceding steps (hue mapping D.3.3 then lightness mapping D.3.4), denoted $\tilde{\mathcal{L}}$ into the gamut $\mathcal{T}$ without changing the lightness of the input signal. Chrominance mapping is also referred as saturation compression.

The boundary of the warped gamut is computed in clause D.3.5.2. Then the chrominance mapping process parameters are derived from metadata in clause D.3.5.3. The pixel-loop representative of the chrominance compression is documented in clause D.3.5.4.

D.3.5.2 Computing the boundary of a warped gamut

The lightness mapping process is a parabolic transformation (see clause D.3.4.4). The parabolic transformation reshapes the gamut by replacing the linear edges of the gamut by fragments of parabolas and the planar faces by warped faces. The parabolic transformation preserves the hexagonal shape of the cusp line. However, lightness mapped colours have their boundary lines passing by the white and black points replaced by a parabola (that is defined by a second-degree equation).

The lightness mapping transforms the rotated gamut $\mathcal{R}$ into $\tilde{\mathcal{L}}$ (see clauses D.3.4.3 and D.3.4.4) using parabolic transformations. During the chrominance mapping process, the warped gamut $\tilde{\mathcal{L}}$ is transformed into the truncated gamut $\mathcal{T}$ which is a dodecahedron or a hexahedron (depending on the value of **croppingModeSCG** and **cmWeightFactor**[i]) with planar faces. During chrominance mapping process, the boundary lines of both $\tilde{\mathcal{L}}$ and $\mathcal{T}$ should be computed. The computation of $\mathcal{T}$ boundary lines is specified in clause D.2.3.7 while the computation of $\tilde{\mathcal{L}}$ boundary lines is specified in this clause.

The boundary lines of $\tilde{\mathcal{L}}$ are fragments of parabolas following a second-degree equation:

$$y = \alpha \times \rho^2 + \beta \times \rho + \gamma \quad \text{(D.48)}$$

where α, β and γ are the coefficients of the equation, ρ the unknown and y the lightness.

The computation of the colour boundary for a given colour P in the parabolic gamut $\tilde{\mathcal{L}}$ is specified as follows:

- The second-degree equation (D.48) is solved as described by the following equations:

  The coefficients α, β and γ are computed as follows (where s designates the sign) reusing δL from (D.45):

$$\alpha = \frac{\delta L}{Cusp(P,\mathcal{R})_\rho^{\,2}} \quad \text{(D.49)}$$

If $\left(1 \geq y \geq Cusp(P,\mathcal{R})_y - \delta L\right)$

$$\beta = \frac{Cusp(P,\mathcal{R})_y - 1}{Cusp(P,\mathcal{R})_\rho} \quad \text{(D.50)}$$

$$\gamma = 1 \quad \text{(D.51)}$$

$$s = 1 \qquad (D.52)$$

Otherwise:

$$\beta = \frac{Cusp(P,\mathcal{R})_y}{Cusp(P,\mathcal{R})_\rho} \qquad (D.53)$$

$$\gamma = 0 \qquad (D.54)$$

$$s = -1 \qquad (D.55)$$

- Compute the boundary colour $Bound(P, \tilde{\mathcal{L}})$ i.e. the intersection between the boundary line $\tilde{\mathcal{L}}^{bound}_{(\theta = P_\theta)}$ and the line defined by $y = P_y$ as follows:

$$Bound(P, \tilde{\mathcal{L}})_y = P_y \qquad (D.56)$$

$$Bound(P, \tilde{\mathcal{L}})_\theta = P_\theta \qquad (D.57)$$

$$Bound(P, \tilde{\mathcal{L}})_\rho = \left(\beta + s \times \sqrt{\beta^2 - 4 \times \alpha \times (\gamma - P_y)}\right) \times \frac{1}{-2 \times \alpha} \qquad (D.58)$$

D.3.5.3 Deriving the compression parameters from the metadata

The chrominance mapping behaviour is derived from **satMappingMode**:

- When **satMappingMode** is equal to 0, the chrominance mapping process is by-passed (output sample equal to the input sample).
- When **satMappingMode** is greater than 0, the compression curve consists of up to three linear segments that are defined by three parameters: $ratioPresArea$, $ratioElWCG$ and $ratioElSCG$ (see Figure D.11). These three parameters are derived (inverse gamut mapping phase) from or stored (forward gamut mapping phase) in variables specified in clause 6.3.9.
  - When **satMappingMode** is equal to 1, the three parameters are global to the gamut and are derived from the variables **satGlobal1SegRatio**, **satGlobal2SegRatioWCG** and **satGlobal2SegRatioSCG** as specified in the following equations:

$$ratioPresArea = \textbf{satGlobal1SegRatio} \qquad (D.59)$$

$$ratioElWCG = \textbf{satGlobal2SegRatioWCG} \qquad (D.60)$$

$$ratioElSCG = \textbf{satGlobal2SegRatioSCG} \qquad (D.61)$$

  - When **satMappingMode** is equal to 2, the three parameters are interpolated as follows:

    Assuming Q the input colour, with (Q_y, Q_u, Q_v) equal to $(Y_{std}, U_{rot}, V_{rot})$:

    - Determine the sector $Sector(Q, \mathcal{L}^{cusp})$ which the hue of Q belongs to. $Sector(Q, \mathcal{L}^{cusp})$ results in the $[\ell^{beg}\ell^{end}[$ segment. Since ℓ^i is equivalent to r^i, in the following r^i will be used.
    - Compute the intersection e of line (oq) with $[r^{beg}r^{end}[$, o being the origin of the UV plane ($Plane_{(y=0)}$) and q the projection of Q on this UV plane:

$$e = (op) \cap [r^{beg}r^{end}[\qquad (D.62)$$

    - Compute the interpolation ratio t:

$$\overrightarrow{r^{beg}e} = t \times \overrightarrow{r^{beg}r^{end}} \qquad (D.63)$$

    - Interpolate the variables to obtain the parameters for each primary and secondary colour:

$$ratioPresArea = \textbf{sat1SegRatio}[\,beg\,] + t \times (\textbf{sat1SegRatio}[\,end\,] - \textbf{sat1SegRatio}[\,beg\,]) \qquad (D.64)$$

$$ratioElWCG = \textbf{sat2SegRatioWCG}[\,beg\,] + t \times (\textbf{sat2SegRatioWCG}[\,end\,] - \textbf{sat2SegRatioWCG}[\,beg\,]) \qquad (D.65)$$

$$ratioElSCG = \mathbf{sat2SegRatioSCG}[\,\mathrm{beg}\,] + t \times (\mathbf{sat2SegRatioSCG}[\,\mathrm{end}\,] - \mathbf{sat2SegRatioSCG}[\,\mathrm{beg}\,]) \quad (D.66)$$

- When **satMappingMode** is equal to 3 the three parameters are defined per primary/secondary colours of the wide gamut and are derived from the variables **sat1SegRatio**[c], **sat2SegRatioWCG**[c] and **sat2SegRatioSCG**[c].

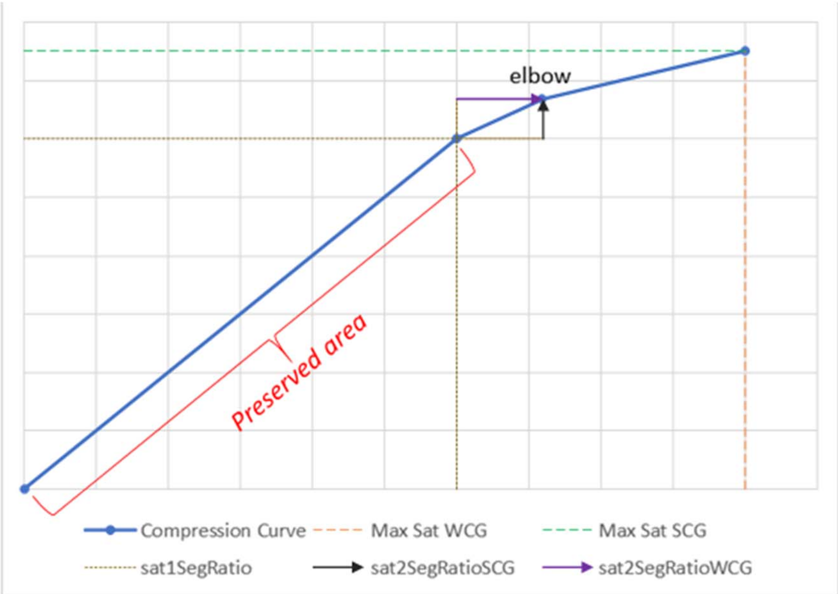

Figure D.11: Shape of the compression curve

D.3.5.4 Compressing the chrominance

This stage aims at fitting the input colour to the cropped (or truncated) gamut $\mathcal{T}$ (included in the standard gamut $\mathcal{S}$) as depicted in Figure D.12.

Assuming Q the input colour, with (Q_y, Q_u, Q_v) equal to $(Y_{std}, U_{rot}, V_{rot})$.

- Compute T, the boundary colour of Q in the gamut $\mathcal{T}$:

$$\mathrm{T} = Bound(\mathrm{Q}, \mathcal{T}) \quad (D.67)$$

- Compute L, the boundary colour of Q in the gamut $\tilde{\mathcal{L}}$ from clause D.3.5.2:

$$\mathrm{L} = Bound(\mathrm{Q}, \tilde{\mathcal{L}}) \quad (D.68)$$

- Compute $preservedArea$, the preserved area ratio:

$$preservedArea = ratioPresArea \times \mathrm{T}_\rho \quad (D.69)$$

- Determine the compressed chrominance F as follows:

```
if (Q_ρ ≤ preservedArea)
      F_ρ = Q_ρ;
else{
      elbowWCG = preservedArea + (L_ρ − preservedArea) × ratioElWCG;
      elbowSCG = preservedArea + (T_ρ − preservedArea) × ratioElSCG;
      if (Q_ρ ≤ elbowWCG)
            F_ρ = preservedArea + (elbowSCG − preservedArea)/(elbowWCG − preservedArea) × (Q_ρ − preservedArea);
      else
            F_ρ = elbowSCG + (T_ρ − elbowSCG)/(L_ρ − elbowWCG) × (Q_ρ − elbowWCG);
}
```

- The output of the chrominance compression is $(Y_{std}, U_{std}, V_{std})$ such that:

$$(\mathrm{Y}_{std}, \mathrm{U}_{std}, \mathrm{V}_{std}) = (\mathrm{F}_y, \mathrm{F}_u, \mathrm{F}_v) \quad (D.70)$$

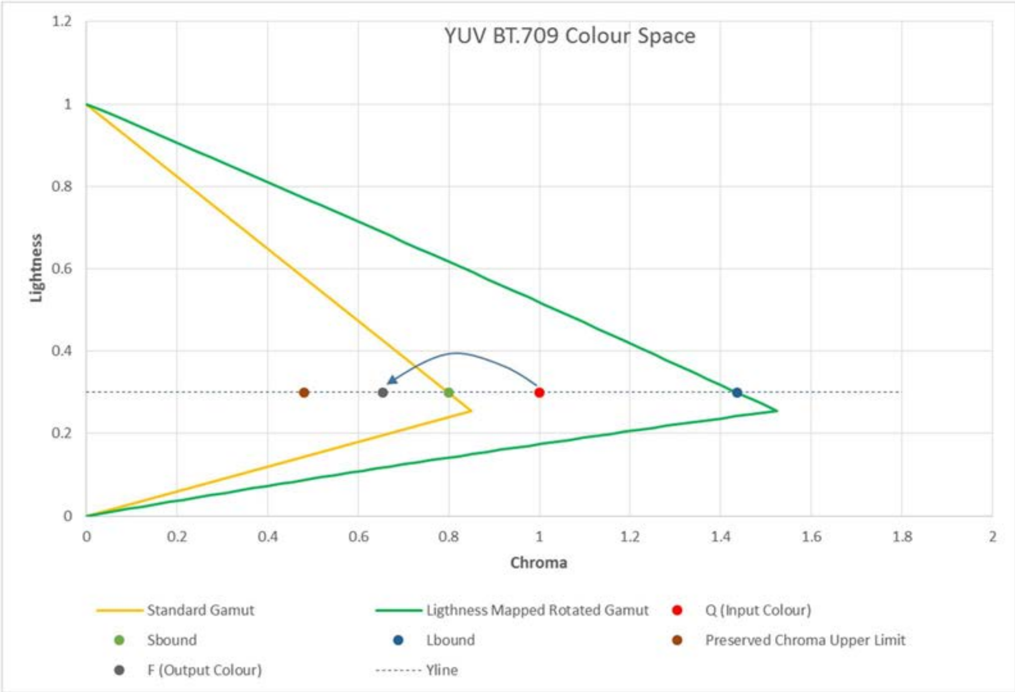

Figure D.12: Chrominance compression

D.3.6 Interfacing with SL-HDR1 decomposition

This process takes as inputs:

- the lightness mapped Y_{std} and the chrominance components mapped U_{std} and V_{std} in gamut generated in clauses D.3.4 and D.3.5;
- the variable related to the standard colour space: **sdrPicColourSpace**.

The process generates as output:

- the RGB 4:4:4 linear-light R_{dec}, G_{dec} and B_{dec} in gamut $\mathcal{S}$.

The SL-HDR1 HDR-to-SDR decomposition (see Annex C) starts from linear-light 4:4:4 RGB as documented in clause C.1.2. Since clause D.3.6 is producing linear-light 4:4:4 YUV, a canonical Y'CbCr-to-R'G'B' matrix operating in the $\mathcal{S}$ colour space is required to interface the forward gamut mapping with SL-HDR1 decomposition process, as follows:

$$\begin{bmatrix} R_{dec} \\ G_{dec} \\ B_{dec} \end{bmatrix} = M^{\mathcal{S}}_{YUV-to-RGB} \times \begin{bmatrix} Y_{std} \\ U_{std} \\ V_{std} \end{bmatrix} \quad \text{(D.71)}$$

where $M^{\mathcal{S}}_{YUV-to-RGB}$ is a canonical 3x3 Y'CbCr-to-R'G'B' matrix (e.g.as specified in Recommendation ITU-R BT.709-6 [6]).

D.4 Inverse gamut mapping process

D.4.1 Introduction

This clause specifies the colour reconstruction (or gamut expansion or inverse gamut mapping) process enabling the generation of a wide colour gamut picture from a standard colour gamut picture with associated dynamic metadata.

This process is defined for a 4:4:4 chroma sampling and full range YUV linear-light signal normalized by **hdrDisplayMaxLuminance**. The input YUV signal comes from the conversion of an input RGB linear-light signal (output of the SDR-to-HDR reconstruction process specified in clause 7) to a linear-light YUV colour space thanks to the canonical R'G'B'-to-Y'CbCr matrix (computed thanks to SMPTE RP 177 [i.9]).

The inverse gamut mapping process is reversing all the processes operated during the forward gamut mapping. the gamut is transformed during the inverse gamut mapping stages as described below:

$$\mathcal{S} \rightarrow \mathcal{T} \rightarrow \tilde{\mathcal{L}} \rightarrow \mathcal{R} \rightarrow \mathcal{W} \rightarrow \mathcal{K} \quad \text{(D.72)}$$

The inverse gamut mapping process and the associated gamuts are depicted in Figure D.13.

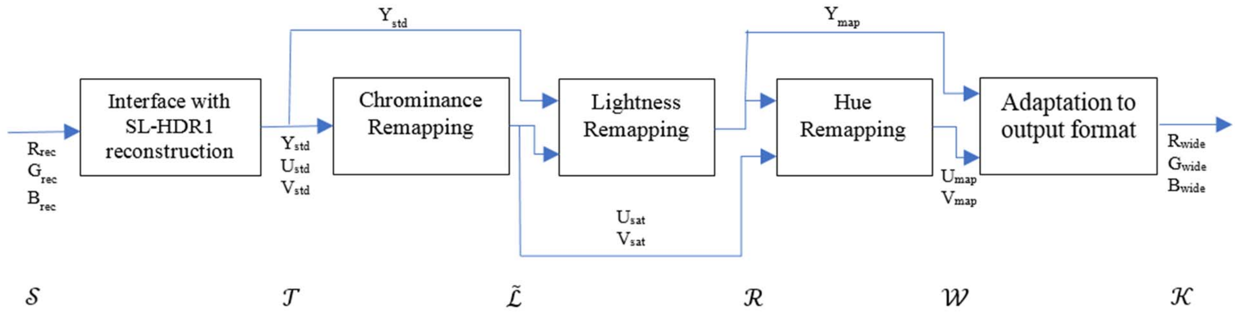

Figure D.13: Overview of the inverse gamut mapping process and the associated gamuts

The inverse gamut mapping process can be summarized as follows (for each pixel x = 0..($picWidth - 1$), y = 0..($picHeight - 1$)):

- The input of the inverse gamut mapping process is the output of the SL-HDR1 reconstruction process i.e. a RGB 4:4:4 linear-light signal represented in the standard colour space. This RGB signal is converted to a YUV 4:4:4 linear-light signal as documented in clause D.4.2. During this process, the gamut changes from the standard gamut $\mathcal{S}$ to the cropped (or truncated) gamut $\mathcal{T}$ specified in clause D.3.5.

- Then, the chrominance remapping stage is operated on the chrominance (see clause D.4.3). During this process, the gamut changes from the cropped (or truncated) gamut $\mathcal{T}$ to the lightness mapped gamut $\tilde{\mathcal{L}}$ specified in clause D.3.4.

- The next step consists in remapping the lightness (see clause D.4.4). During this process, the gamut changes from the lightness remapped gamut $\tilde{\mathcal{L}}$ to the chrominance hue mapped (or rotated) gamut $\mathcal{R}$ specified in clause D.3.3.

- The last inverse gamut mapping step is the hue remapping (see clause D.4.5). During this process, the gamut changes from the chrominance hue mapped (or rotated) gamut $\mathcal{R}$ to the wide gamut $\mathcal{W}$ specified in clause D.3.2.

- Eventually, the remapped YUV is converted to the wide colour space and a transfer function is applied (see clause D.4.6). During this process, the gamut changes from the mastering display gamut $\mathcal{W}$ to the wide gamut $\mathcal{K}$ specified in clause D.3.2.

D.4.2 Interfacing with SL-HDR1 reconstruction

This process takes as inputs:

- the linear-light standard $\mathcal{S}$ gamut 4:4:4 R_{rec}, G_{rec} and B_{rec} samples of the standard gamut $\mathcal{S}$ generated in clause 7.2.4 and normalized by **hdrDisplayMaxLuminance**;

- the variable related to the SDR colour space: **sdrPicColourSpace**.

The process generates as output:

- the linear-light standard 4:4:4 Y_{std}, U_{std} and V_{std} samples that reside in the gamut $\mathcal{T}$.

The process converts a linear-light RGB sample into a linear-light YUV sample as follows:

$$\begin{bmatrix} Y_{std} \\ U_{std} \\ V_{std} \end{bmatrix} = M^{\mathcal{S}}_{RGB-to-YUV} \times \begin{bmatrix} R_{rec} \\ G_{rec} \\ B_{rec} \end{bmatrix} \quad \text{(D.73)}$$

where $M^{\mathcal{S}}_{RGB-to-YUV}$ is a canonical 3x3 R'G'B' to Y'CbCr matrix that depends on **sdrPicColourSpace**. In this version of the present document, the R'G'B' to Y'CbCr matrix should be as specified in Recommendation ITU-R BT.709-6 709 [6].

D.4.3 Chrominance remapping

This process takes as inputs:

- the lightness and chrominance components values Y_{std}, U_{std} and V_{std} in the cropped gamut $\mathcal{T}$ generated in clause D.4.2;
- the variables related to chrominance remapping (also known as saturation expansion): **satMappingMode**, **satGlobal1SegRatio**, **satGlobal2SegRatioWCG**, **satGlobal2SegRatioSCG**, **sat1SegRatio**[i], **sat2SegRatioWCG**[i] and **sat2SegRatioSCG**[i];
- the variables related to the cropped (or truncated) gamut conversion: **croppingModeSCG** and **cmWeightFactor**[i].

The process generates as output:

- the chrominance components remapped values U_{sat} and V_{sat} which with the lightness sample Y_{std} are in the lightness mapped rotated gamut $\tilde{\mathcal{L}}$.

The chrominance remapping process aims at re-expanding the standard gamut $\mathcal{S}$ back to the lightness mapped rotated gamut $\tilde{\mathcal{L}}$, thus reversing the gamut compression made during the chrominance mapping process (see clause D.3.5) of the forward gamut mapping phase. Compression and expansion stages are performed without changing the lightness of the signal.

The expansion is using the same parameters as for the compression (i.e. **satMappingMode**, **satGlobal1SegRatio**, **satGlobal2SegRatioWCG**, **satGlobal2SegRatioSCG**, **sat1SegRatio**[i], **sat2SegRatioWCG**[i] and **sat2SegRatioSCG**[i]) to control the chrominance remapping either in a global manner or independently per primary and secondary colour. The cropped gamut is derived thanks to the variables **croppingModeSCG** and **cmWeightFactor**[i] as specified in clause D.3.4.2. The variables are used to derive the shape of the expansion curve that is expanding the input chrominance into an output chrominance as shown in Figure D.14.

The chrominance remapping process is quite similar to the chrominance mapping process specified in clause D.3.5.

The process is as follows:

Assuming F the input colour, with (F_y, F_u, F_v) equal to $(Y_{std}, U_{std}, V_{std})$.

- Derive the gamut $\tilde{\mathcal{L}}$ as specified in clauses D.3.4.2 or D.3.4.3.
- Derive the expansion curves $ratioPresArea$, $ratioElWCG$ and $ratioElSCG$ as specified in clause D.3.5.3.
- Compute T, the boundary colour of F in the gamut $\mathcal{T}$:

$$\mathrm{T} = Bound(\mathrm{F}, \mathcal{T}) \quad \text{(D.74)}$$

- Compute L, the boundary colour of F in the gamut $\tilde{\mathcal{L}}$, as in clause D.3.5.2:

$$\mathrm{L} = Bound(\mathrm{F}, \tilde{\mathcal{L}}) \quad \text{(D.75)}$$

- Compute $preservedArea$, the preserved area ratio:

$$preservedArea = ratioPresArea \times \mathrm{T}_{\rho} \quad \text{(D.76)}$$

- Determine the expanded chrominance Q as follows:

```
if (Fρ ≤ preservedArea)
    Qρ = Fρ;
else{
    elbowWCG = preservedArea + (Lρ − preservedArea) × ratioElWCG;
    elbowSCG = preservedArea + (Tρ − preservedArea) × ratioElSCG;
    if (Fρ ≤ elbowSCG)
        Qρ = preservedArea + (elbowWCG − preservedArea)/(elbowSCG − preservedArea) × (Fρ − preservedArea);
    else
        Qρ = elbowWCG + (Lρ − elbowWCG)/(Tρ − elbowSCG) × (Fρ − elbowSCG);
}
```

- The output of the chrominance compression is $(Y_{std}, U_{sat}, V_{sat})$ such that:

$$(Y_{std}, U_{sat}, V_{sat}) = (Q_y, Q_u, Q_v) \quad (D.77)$$

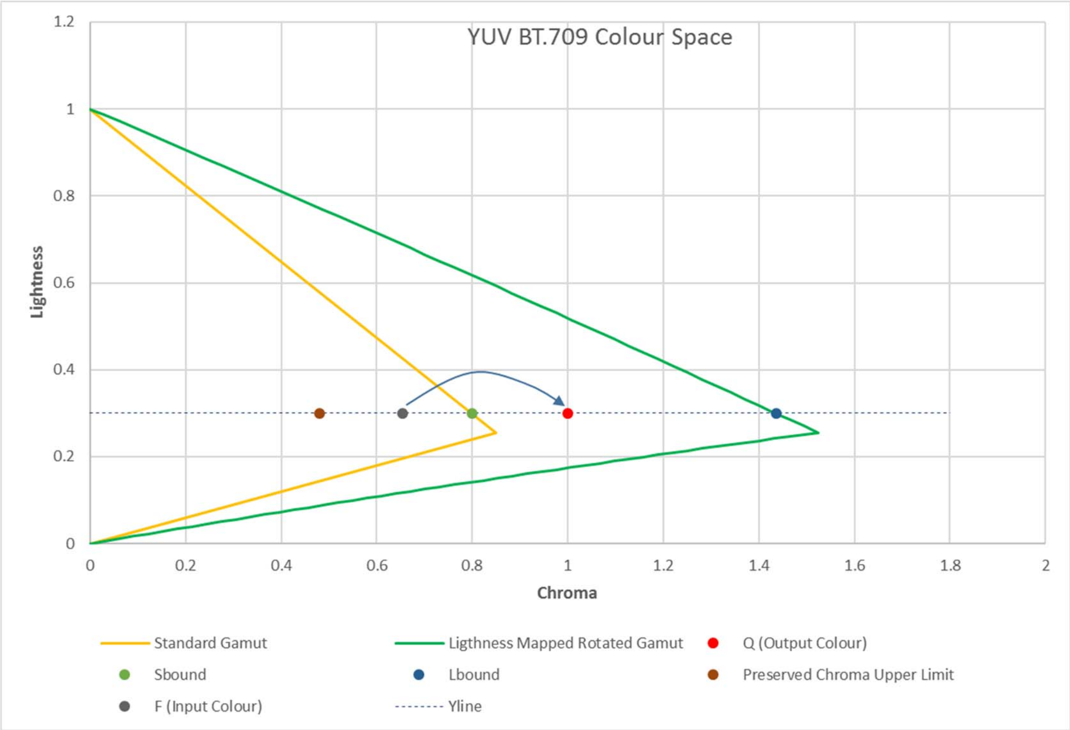

Figure D.14: Chrominance remapping

D.4.4 Lightness remapping

This process takes as inputs:

- the lightness value Y_{std} generated in clause D.4.2 which with the chrominance components remapped values U_{sat} and V_{sat} generated in clause D.4.3 are located in the lightness mapped rotated gamut $\tilde{\mathcal{L}}$;
- the variables related to lightness remapping: **lightnessMappingMode, cmCroppedLightnessMappingEnabledFlag** and **lmWeightFactor**[i];
- the variables related to hue rotation: **hueAlignCorrectionPresentFlag**, **hueAlignCorrection**[i], **chromAdjPresentFlag** and **chromAdjParam**[i].

The process generates as output:

- the lightness remapped value Y_{map} which with the chrominance components remapped values U_{sat} and V_{sat} are located in the hue mapped (rotated) gamut $\mathcal{R}$.

As depicted in Figure D.15, The inverse gamut mapping reverts the lightness mapping of the forward gamut mapping process specified in clause D.3.4. The lightness is corrected while leaving the hue and chrominance of the processed sample unchanged.

The lightness remapping converts the lightness mapped gamut $\tilde{\mathcal{L}}$ back to the hue mapped gamut $\mathcal{R}$. The gamut $\tilde{\mathcal{L}}$ is derived from clause D.3.4.3 and **lightnessMappingMode** and **lmWeightFactor**[i] variables. The hue mapped gamut $\mathcal{R}$ is derived from **hueAlignCorrectionPresentFlag**, **hueAlignCorrection**[i], **chromAdjPresentFlag** variables and **chromAdjParam**[i] as specified in clause D.3.3.2.

Assuming Q the input colour to this process with coordinates (Q_y, Q_u, Q_v) equal to $(Y_{std}, U_{map}, V_{map})$, the lightness remapping process operates similarly as the lightness mapping (see equations (D.42) to (D.46)) inclusive replacing L by Q), only the final mapping changes as follows:

$$L_y = Q_y + \delta Q \times \left(\frac{Q_\rho}{Z_\rho}\right)^2 \quad \text{(D.78)}$$

δQ is computed as δL in equation (D.45) with ωQ being calculated as ωL in equation.(D.42).

This process output is as follows:

$$Y_{map} = L_y \quad \text{(D.79)}$$

Figure D.15: Lightness remapping

D.4.5 Hue remapping

D.4.5.1 Introduction

This process takes as inputs:

- the lightness remapped value Y_{map} generated in clause D.4.4 and the chrominance components remapped values U_{sat} and V_{sat} generated in clause D.4.3 which are located in the hue mapped (rotated) gamut $\mathcal{R}$;
- the variables related to hue remapping: **hueAdjMode**, **hueGlobalPreservationRatio** and **huePreservationRatio**[c];

- the variables related to hue alignment: **hueAlignCorrectionPresentFlag** and **hueAlignCorrection**[c];
- the variables related to chrominance adjustment: **chromAdjPresentFlag** and **chromAdjParam**[c].

The process generates as output:

- the hue remapped chrominance components values U_{map} and V_{map} which with Y_{map} reside in the wide gamut $\mathcal{W}$.

The hue remapping process leaves the lightness unchanged. This process is equivalent to a gamut rotation based on a hue and chrominance dependent angles. This process enables the preservation of a portion of the input gamut when **hueAdjMode** ≥ 2.

The hue remapping process converts the hue mapped (rotated) gamut $\mathcal{R}$ back to the wide gamut $\mathcal{W}$ using the same variables than those of use in clause D.3.3. The gamut $\mathcal{R}$ is derived from clause D.3.3.2 while the preserved $\mathcal{P}$ gamut is derived from **hueAdjMode**, **hueGlobalPreservationRatio** and **huePreservationRatio**[i], as specified in clause D.3.3.4.

Clause D.4.5.2 specifies the hue remapping without preserved area process and applies when **hueAdjMode** is equal to 1. Clause D.4.5.3 specifies the hue remapping with preserved area process and applied when **hueAdjMode** is equal to 2 or 3.

D.4.5.2 Hue remapping without preserved area

This clause applies when **hueAdjMode** is equal to 1.

The same process as specified in clause D.3.3.3 applies with the following changes:

- the input of the process is the YUV triplet $(Y_{map}, U_{sat}, V_{sat})$;
- the output of the process is the YUV triplet $(Y_{map}, U_{map}, V_{map})$;
- the gamuts $\mathcal{R}$ and $\mathcal{W}$ as well as their derivations are swapped;
- the colour P and H as well as their derivations are swapped.

Figure D.16 represents the hue remapping without preserved area process.

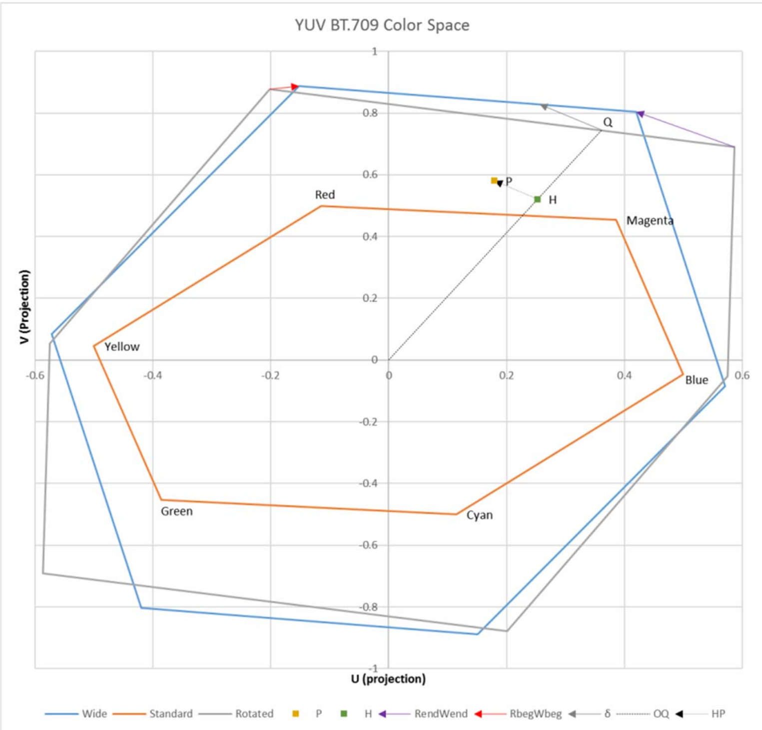

Figure D.16: Hue remapping without preserved area

D.4.5.3 Hue remapping with preserved area

This process applies when **hueAdjMode** is equal to 2 or 3.

The process specified in this clause differs by some aspects from the dual process (hue mapping) specified in clause D.3.3.5. Notations simplifications are the same as in clause D.3.3.5.

NOTE 1: The process specified in this clause is still the reversible part of clause D.3.3.5. The difference in processing between **hueAdjMode** equal to 2 or 3 is explicated in clause D.3.3.4.

The hue remapping with preserved area rotates the gamut $\mathcal{R}$ (see clause D.3.3.4) into the gamut $\mathcal{W}$, while preserving the gamut $\mathcal{P}$ that is a portion of the gamut $\mathcal{R}$, thus reverting the hue mapping (rotation) performed during the forward gamut mapping phase.

Assuming an input YUV triplet *(H_y, H_u, H_v)* equal to *(Y_{map}, U_{map}, V_{map})*, the first step is to determine whether H is in the preserved gamut $\mathcal{P}$ or in the modified portion of the gamut $\mathcal{R}$. The computation is as follows:

- Compute the boundary line $\mathcal{P}^{bound}_{(y=\mathrm{H_y})}$ as follows:

$$\mathcal{P}^{bound}_{(y=\mathrm{H_y})} = \mathcal{P} \cap Plane_{(y\,=\,\mathrm{H_y})} \quad \text{(D.80)}$$

where $\mathcal{P}^{bound}_{(y=\mathrm{H_y})}$ is a convex polygon with between three and ten edges depending on the value of $\mathrm{H_y}$.

NOTE 2: In the linear-light YUV colour space, the preserved gamut $\mathcal{P}$ is a dodecahedron defined by its eight vertices { $\mathcal{P}^K, \mathcal{P}^R, \mathcal{P}^M, \mathcal{P}^B, \mathcal{P}^C, \mathcal{P}^G, \mathcal{P}^J, \mathcal{P}^W$} and its twelve faces {(KRM), (KBM), (KBC), (KGC), (KRJ), (KGJ), (WCB), (WMB), (WJR), (WMR), (WJG), (WCG) }.

- Determine to which sector of the polygon $\mathcal{P}^{bound}_{(y=\mathrm{H_y})}$ or $\mathcal{R}^{bound}_{(y=P_y)}$ h belongs to as follows:

  Assuming η the number of vertices (or sectors) of $\mathcal{P}^{bound}_{(y=\mathrm{H_y})}$ and $[\wp^{\sigma}\wp^{(\sigma+1)\%\eta}[$ that characterizes the current sector of $\mathcal{P}^{bound}_{(y=\mathrm{H_y})}$, compute for each sector the following pseudo-code to determine the sector which h belongs to:

```
for(σ = 0; σ < η; σ++) {
    ϕ^σ = (oh→ × o℘^σ→ ≥ 0) && (oh→ × o℘^((σ+1)%η)→ < 0) && (℘^σh→ × ℘^σ℘^((σ+1)%η)→ ≥ 0)
    if(ϕ^σ == TRUE){
        beg = σ
        end = (beg + 1)%η
        break;
    }
}
```

  where the η vertices are ordered clockwise, ϕ^{σ} is a logical variable stating the presence or not of h in the sector characterized by $[\wp^{\sigma}\wp^{(\sigma+1)\%\eta}[$, h is the projection of H on $Plane_{(y\,=\,\mathrm{H_y})}$ and o is the origin of this plane.

  - If (ϕ^{σ} = = TRUE), the point h is in the preserved area. Then the rotated projection of the colour p, is the projection h (i.e. p = h) and the computation stops for the current projection h.
  - If (ϕ^{σ}= = FALSE) for every σ (i.e. h does not belong to the preserved area), determine the sector of $\mathcal{R}^{bound}_{(y=P_y)}$ which h belongs to as follows:
    - Compute $\mathcal{R}^{bound}_{(y=\mathrm{H_y})}$ as follows:

$$\mathcal{R}^{bound}_{(y=\mathrm{H_y})} = \mathcal{R} \cap Plane_{(y\,=\,\mathrm{H_y})} \quad \text{(D.81)}$$

    where $\mathcal{R}^{bound}_{(y=\mathrm{H_y})}$ is a convex polygon with between three to ten edges depending on the value of $\mathrm{H_y}$.

NOTE 3: In the linear-light YUV colour space, the rotated gamut $\mathcal{R}$ is a dodecahedron defined by its eight vertices { $\mathcal{R}^K, \mathcal{R}^R, \mathcal{R}^M, \mathcal{R}^B, \mathcal{R}^C, \mathcal{R}^G, \mathcal{R}^J, \mathcal{R}^W$} and its twelve faces {(KRM), (KBM), (KBC), (KGC), (KRJ), (KGJ), (WCB), (WMB), (WJR), (WMR), (WJG), (WCG) }.

- Assuming η the number of vertices (or sectors) of $\mathcal{R}^{bound}_{(y=\mathrm{H_y})}$ and $\mathcal{P}^{bound}_{(y=\mathrm{H_y})}$, and $[r^{\sigma}r^{(\sigma+1)\%\eta}[$ the current sector of $\mathcal{R}^{bound}_{(y=\mathrm{H_y})}$, compute for each sector the following pseudo-code to determine the sector h belongs to:

for(σ = 0; σ < η; σ ++)

$$\varphi^{\sigma} = \left(\overrightarrow{r^{\sigma}\mathrm{h}} \times \overrightarrow{r^{\sigma}p^{\sigma}} < 0\right) \&\& \left(\overrightarrow{r^{(\sigma+1)\%\eta}h} \times \overrightarrow{r^{(\sigma+1)\%\eta}p^{(\sigma+1)\%\eta}} \geq 0\right) \&\& \left(\overrightarrow{p^{\sigma}\mathrm{h}} \times \overrightarrow{p^{\sigma}p^{(\sigma+1)\%\eta}} < 0\right)$$

where the η vertices are ordered clockwise, φ^{σ} is a logical variable stating the presence or not of h in the sector characterized by $[r^{\sigma}r^{(\sigma+1)\%\eta}[$, h is the projection of H on $Plane_{(y\,=\,\mathrm{H_y})}$ and o is the origin of this plane.

The projection h is in the sector characterized by $[r^{\mathrm{beg}}r^{\mathrm{end}}[$ with beg defined by $\varphi^{\mathrm{beg}} == TRUE$ and end defined by $\mathrm{end} = (\mathrm{beg}+1)\%\eta$.

Figure D.17 shows the $Plane_{(y\,=\,\mathrm{H_y})}$ with the different sectors depicted by pentagons that are defined by $(\mathrm{o}, p^{\sigma}, r^{\sigma}, r^{(\sigma+1)\%\eta}, p^{(\sigma+1)\%\eta})$. In the example below the sector corresponding to h has its beg vertex set to WR and its end vertex set to RM.

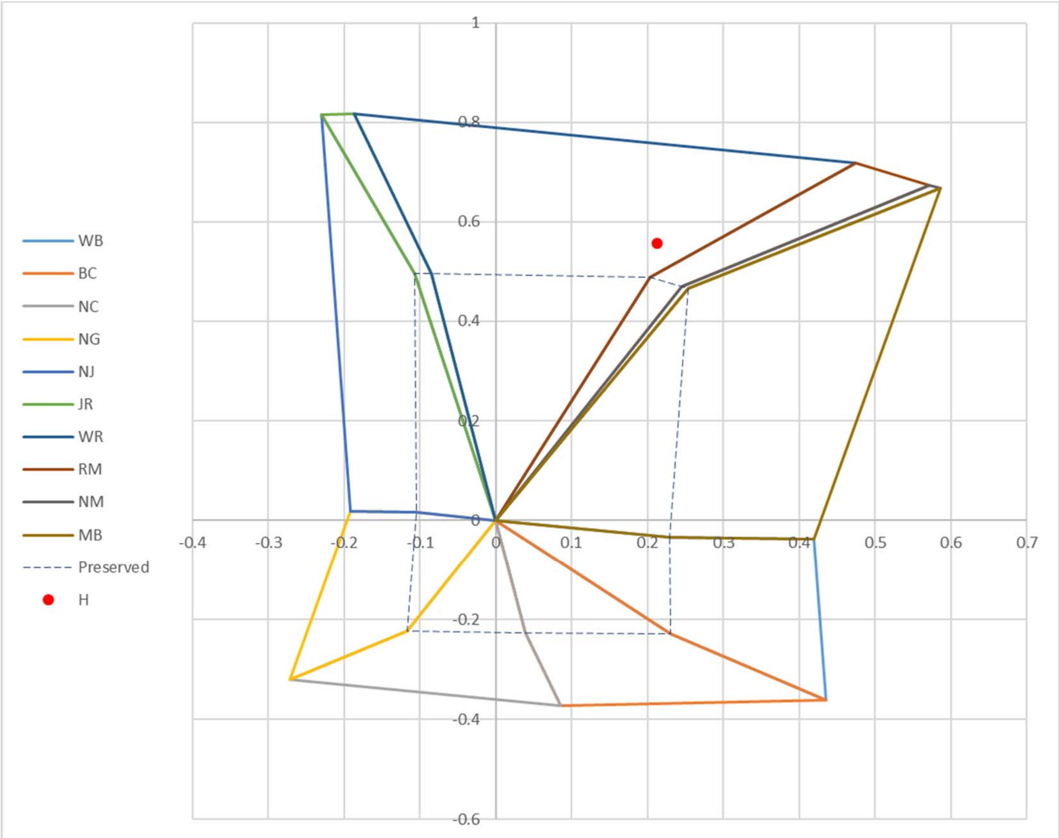

Figure D.17: Sectors decomposition in the gamut $\mathcal{R}$

- Then, the next steps are similar to those specified in clause D.3.3.5 of the forward gamut mapping process:
  - Compute the anchor α as follows:

$$\alpha = \left(p^{\mathrm{beg}}r^{\mathrm{beg}}\right) \cap \left(p^{\mathrm{end}}\,r^{\mathrm{end}}\right) \quad \text{(D.82)}$$

  - Compute the rotation centre q that is the intersection of line (αh) with the preserved sector segment as follows:

$$\mathrm{q} = (\alpha\mathrm{h}) \cap [p^{\mathrm{beg}}p^{\mathrm{end}}[\quad \text{(D.83)}$$

- Compute $\mathcal{W}^{bound}_{(y=H_y)}$ as follows:

$$\mathcal{W}^{bound}_{(y=\mathrm{H_y})} = \mathcal{W} \cap Plane_{(y\,=\,\mathrm{H_y})} \qquad \text{(D.84)}$$

Using lowercase, the sector of $\mathcal{W}$ corresponding to h is defined using beg and end coordinates computed from $\mathcal{P}^{bound}_{(y=\mathrm{H_y})}$: $[w^{\mathrm{beg}} w^{\mathrm{end}}[$.

NOTE 4: In the linear-light YUV colour space the colour gamut $\mathcal{W}$ is considered as a degenerated dodecahedron defined by its eight vertices { $\mathcal{W}^K, \mathcal{W}^R, \mathcal{W}^M, \mathcal{W}^B, \mathcal{W}^C, \mathcal{W}^G, \mathcal{W}^J, \mathcal{W}^W$ } and its twelve faces {(KRM), (KBM), (KBC), (KGC), (KRJ), (KGJ), (WCB), (WMB), (WJR), (WMR), (WJG), (WCG) }.

- Compute the projection direction $\vec{\delta}$ in three steps as described by the following equations:

$$\beta = (\mathrm{oq}) \cap [w^{\mathrm{beg}} w^{\mathrm{end}}[\qquad \text{(D.85)}$$

$$\gamma = (\alpha\mathrm{q}) \cap [r^{\mathrm{beg}} r^{\mathrm{end}}[\qquad \text{(D.86)}$$

$$\vec{\delta} = \overrightarrow{\gamma\beta} \qquad \text{(D.87)}$$

- Compute, the projection p as follows:

$$\mathrm{p} = (\alpha\mathrm{q}) \cap (\mathrm{h} + \vec{\delta}) \qquad \text{(D.88)}$$

- The output of this hue remapping with preserved area is the triplet $(Y_{map}, U_{map}, V_{map})$ such that:

$$\mathrm{Y_{map}} = \mathrm{H_y} \qquad \text{(D.89)}$$

$$\mathrm{U_{map}} = \mathrm{h_u} \qquad \text{(D.90)}$$

$$\mathrm{V_{map}} = \mathrm{h_v} \qquad \text{(D.91)}$$

The rotation and different parameters representative of the hue remapping with preserved area process are depicted in Figure D.18.

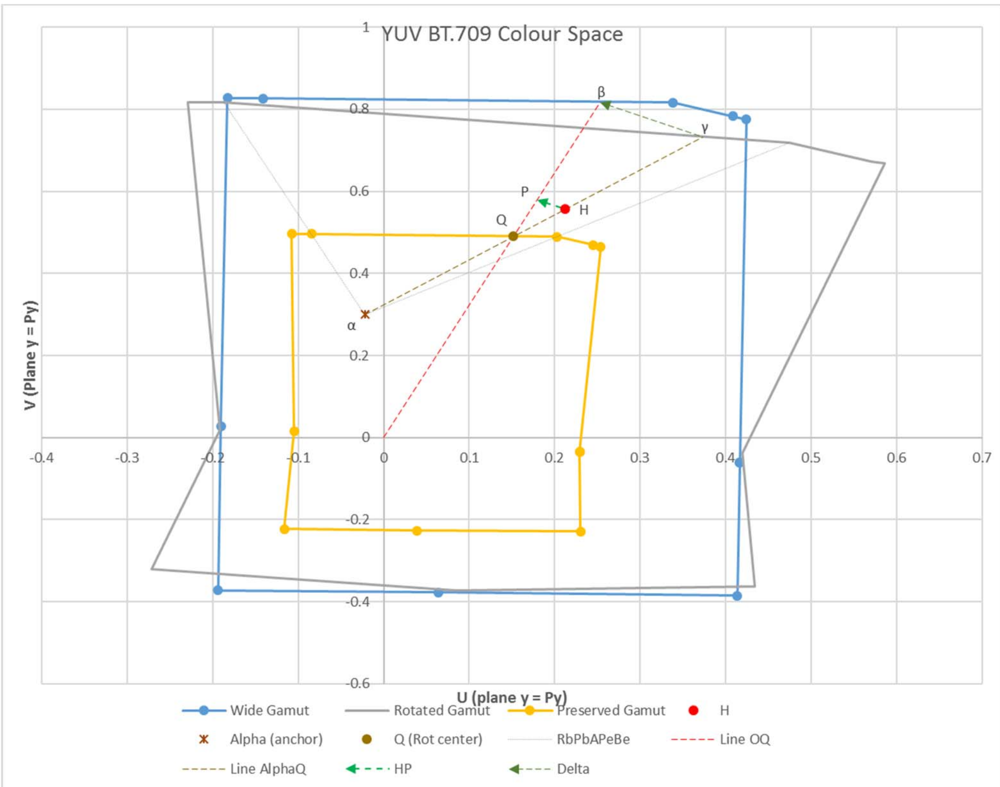

Figure D.18: Hue remapping with preserved area

D.4.6 Adaptation to output format

This process takes as inputs:

- the lightness value Y_{map} generated in clause D.4.4 and the chrominance components values U_{map} and V_{map} generated in clause D.4.5 and that are located in the gamut $\mathcal{W}$;
- the variables related to SDR colour space and the mastering display colour volume: **sdrPicColourSpace**, and **hdrDisplayColourSpace**.

The process generates as output:

- linear-light values R_{wide}, G_{wide} and B_{wide} in the wide gamut $\mathcal{K}$.

The colour space conversion is performed in linear-light domain using the canonical 3x3 matrix conversions.

The process converts a linear-light YUV sample into a linear-light RGB sample as follows:

$$\begin{bmatrix} R_{wide} \\ G_{wide} \\ B_{wide} \end{bmatrix} = M^{\mathcal{K}}_{XYZ-to-RGB} \times M^{\mathcal{S}}_{RGB-to-XYZ} \times M^{\mathcal{S}}_{YUV-to-RGB} \times \begin{bmatrix} Y_{map} \\ U_{map} \\ V_{map} \end{bmatrix} \quad \text{(D.92)}$$

where:

- $M^{\mathcal{S}}_{YUV-to-RGB}$ is a canonical 3x3 Y'CbCr to R'G'B' (computed with **sdrPicColourSpace**. In this version of the present document, this matrix is specified in Recommendation ITU-R BT.709-6 [6]);
- $M^{\mathcal{S}}_{RGB-to-XYZ}$ is a canonical 3x3 RGB to XYZ matrix (computed with **sdrPicColourSpace** and SMPTE RP 177 [i.9]);
- $M^{\mathcal{K}}_{XYZ-to-RGB}$ is a canonical 3x3 XYZ to RGB matrix (computed with **hdrDisplayColourSpace** and SMPTE RP 177 [i.9]).

According to the downstream system connected to the SL-HDR system and the associated negotiation relating to the signal to be carried on the interface, a possible next step is the application of a transfer function, which is out-of-scope of the present clause.

Annex E (informative): HDR-to-HDR 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 (**hdrDisplayMaxLuminance** and **hdrDisplayMinLuminance**, see clause 6.3.3).

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

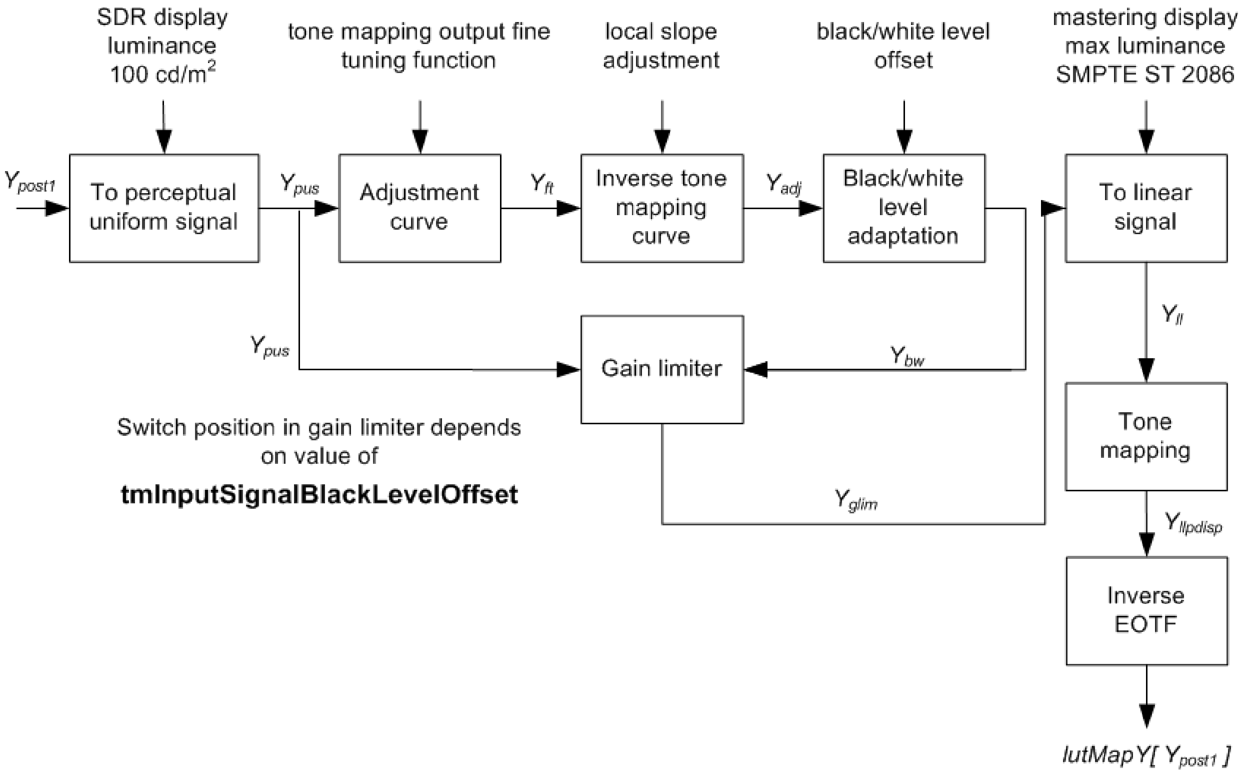

Figure E.1: Display adaptation process

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.

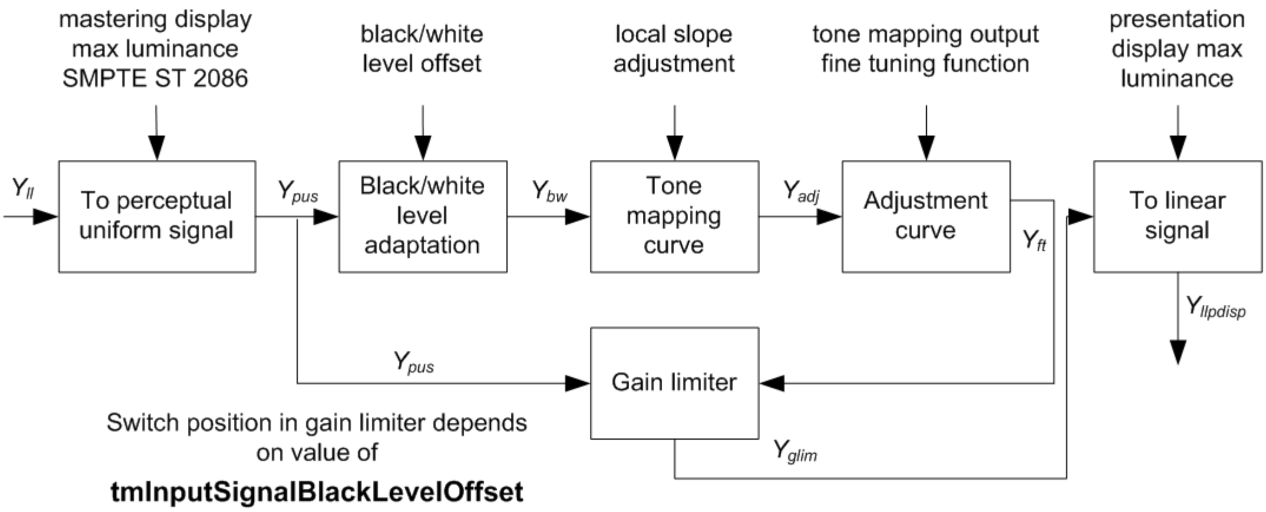

Figure E.2: Tone mapping process for display adaptation

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).

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

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

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

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

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

where:

- L_{HDR} is the maximum display mastering luminance from the variable **hdrDisplayMaxLuminance** in the structure hdr_characteristics( ) of the reconstruction metadata (clause 6.3.3.4);
- 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 2: 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).

$$TMWLO_{DA} = TMWLO\times Max(scaleHor;0) \quad (E.6)$$

where:

- $TMWLO$ is the **tmInputSignalWhiteLevelOffset** as stored in the structure luminance_mapping_variables( ) of the 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.

$$TMBLO_{DA} = TMBLO\times Max(scaleHor;0) \quad (E.7)$$

where:

- $TMBLO$ is the **tmInputSignalBlackLevelOffset** as stored in the structure luminance_mapping_variables( ) of the 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).

$$MIDX = \frac{1-HGC}{SGC-HGC} \quad (E.8)$$

$$MIDX_{DA} = \frac{MIDX\times(SGC-1)}{2}\times(1-scale)+MIDX \quad (E.9)$$

$$MIDY_{DA} = -1\times MIDX_{DA} + MIDX\times(SGC+1) \quad (E.10)$$

$$SGC_{DA} = \frac{MIDY_{DA}}{MIDX_{DA}} \quad (E.11)$$

where:

- SGC and HGC are computed according to equations (C.27) and (C.28).

$$para_{DA} = v(Abs(scale), L_{HDR})\times para \quad (E.12)$$

where:

- $para$ is computed according to equation (C.29);
- and $v(x,y)$ is taken from equations (2) and (3).

$$HGC_{DA} = \begin{cases} 0, & \text{if } MIDX_{DA} - 1 = 0 \\ Max\left(\frac{MIDY_{DA}-1}{MIDX_{DA}-1}\,;0\right), & otherwise \end{cases} \quad (E.13)$$

$$\mathbf{shadowGain_{DA}} = \left(\frac{SGC_{DA}}{\lambda} - 0.5\right)\times 4 \quad (E.14)$$

where:

- **shadowGain$_{DA}$** is the recomputed **shadowGain** to be used in the block "Tone mapping curve" in Figure E.2.

$$\mathbf{highlightGain_{DA}} = HGC_{DA}\times 4 \quad (E.15)$$

where:

- **highlightGain$_{DA}$** is the recomputed **highlightGain** to be used in the block "Tone mapping curve" in Figure E.2.

$$\mathbf{midToneWidthAdjFactor_{DA}} = para_{DA}\times 2 \quad (E.16)$$

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 should be applied and then the inverse black/white adaptation, see equation (E.17).

$$x_{i_{HDR}} = BWAD_{inv}\left(TMO_{inv}(x_i)\right) \quad (E.17)$$

where:

- x_i is the **tmOutputFineTuningX**[i] as stored in the structure luminance_mapping_variables( ) of the 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 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).

$$x_{i_{DA}} = TMO_{DA}\left(BWAD_{DA}\left(x_{i_{HDR}}\right)\right) \quad (E.18)$$

where:

- $x_{i_{DA}}$ is the recomputed **tmOutputFineTuningX**[i] to be used in the block "Adjustment curve" in Figure E.2;
- $BWAD_{DA}(Y_{pus}) = Y_{bw}$, as computed by equations (C.16), (C.17) and (C.18) and using the recomputed **tmInputSignalBlackLevelOffset** and **tmInputSignalWhiteLevelOffset** from equations (E.6) to (E.7);
- and $TMO_{DA}(X)$ is $TMO(X)$ from (C.20) up to and including equation (C.29), 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).

$$y_{i_{DA}} = Min\left((y_i - x_i) \times scaleVer + x_{i_{DA}}\ ;\ 1\right) \qquad \text{(E.19)}$$

where:

- y_i is the **tmOutputFineTuningY**[i] as stored in the structure luminance_mapping_variables( ) of the 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 "Inverse EOTF" in the decoder, see clause 7.2.3.1.9.

In the processing specified in clause 7.2.3.1.9 , clause 7.2.3.2 and clause 7.2.4, the value for $modFactor$ is set according to equation (E.20):

$$modFactor = \left(L_{pdisp} - 100\right) \div \left(L_{HDR} - 100\right) \qquad \text{(E.20)}$$

NOTE 3: $modFactor = 1$ in case no display adaptation is performed.

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): Error-concealment: recovery in post-processor from metadata loss or corruption

F.1 Introduction

SL-HDR1 streams are designed to be supported by legacy and future video distribution workflows. In the present document, SL-HDR1 parameters are conveyed in SEI messages that are seamlessly embedded in the coded video bitstream. In the unlikely event that a portion or all the SEI messages related to SL-HDR1 are pruned by a distribution equipment (e.g. when an SL-HDR1 stream is decoded, mixed, re-encoded, redistributed etc. by certain affiliate networks), this annex provides means to recover parameters default values for use in the SL-HDR1 post-processor. Dynamic metadata loss affects the reconstructed picture fidelity to the source picture. However, unlike blind inverse tone mapping operators, the recovered values of SL-HDR1 parameters allow containing a portion of the artistic intent of the source picture as the source picture was decomposed with an SL-HDR1 pre-processor which processing characteristics are very close to the SL-HDR1 post-processor (invertibility property). The methods proposed in this annex are also applicable in case that a corruption of metadata is detected.

It is expected that a distribution network leveraging an SL-HDR1 stream indicates an SL-HDR-enabled service at the system layer level. Thus, a loss of metadata related to the SL-HDR1 stream could be detected.

Recovery values helpful for reconstructing the HDR picture are provided in clause F.2.

It is expected that the static metadata carried in a Mastering Display Colour Volume SEI message (or equivalent message carrying ST 2086 information), helpful during the SDR-to-HDR reconstruction process, are prone to resist to all sorts of distribution workflows as these static metadata are specified both by SMPTE (production side for contribution networks) and MPEG/ITU-T (distribution side for distribution networks using AVC or HEVC). Besides, static metadata are generally defined and fixed for an entire stream or content. Eventually, MDCV SEI message/SMPTE ST 2086 metadata are being documented in all major applicative standards (ATSC, DVB, CTA...) and it is likely that interfaces to provide this information may be supported by most of the industry stakeholders. Thus, a recovery strategy may consist of recovering adjusted (but suboptimal) SL-HDR1 parameter values thanks to information carried in MDCV SEI/ST 2086 messages. A recovery procedure for the variable **shadow_gain_control** based on this assumption is provided in clause F.3.

In case all SEI messages related to SL-HDR1 are lost, a recovery procedure for the variable **shadow_gain_control** is documented in clause F.4.

In case an inverse gamut mapping is detected as being necessary after the HDR reconstruction process and relating metadata are lost or corrupted, it is recommended to apply a reverse colour space conversion as described in SMPTE RP 177 [i.9].

F.2 Metadata values for recovery mode

The metadata used for obtaining the HDR reconstructed picture may have their values recovered in case of loss or corruption. The following Table F.1 proposes recovery values for the syntax elements of the SL-HDR Information SEI message that are involved in the SDR-to-HDR reconstruction process.

It is noted that **matrix_coefficient_value**[i] default values correspond to the canonical coefficients of the Y'CbCr-to-R'G'B' conversion matrix for either BT.2020 or BT.709 colour space.

Typically, the values of **matrix_coefficient_value[i]** provided in Table F.1 are computed as follows:

$$\mathbf{matrix\_coefficient\_value}[\,\mathbf{i}\,] = Floor(\,c(i) \times 256 + 512 + 0{,}5) \qquad \text{(F.1)}$$

with $c(i) = \{1{,}4746;\ -0{,}1646;\ -0{,}5714; 1{,}8814\}$, if BT.2020 primaries (coefficients computed from Recommendation ITU-R BT.2020-2 [7])

or $c(i) = \{1{,}5748;\ -0{,}1874;\ -0{,}4681; 1{,}8556\}$, if BT.709 primaries (coefficients computed from Recommendation ITU-R BT.709-6 [6]).

By default, the BT.2020 matrix coefficients may be selected for the recovery procedure. However, it is possible that the service layer provides information on the colour space in which the SDR pre-processed picture is represented.

Table F.1: Default metadata values for recovery mode

Syntax element	Recovery value
sl_hdr_payload_mode	0
matrix_coefficient_value[i]	{889; 470; 366; 994}, if BT.2020 {915; 464; 392; 987}, if BT.709
chroma_to_luma_injection[i]	{0;1638}
k_coefficient_value[i]	{0; 0; 0}
tone_mapping_input_signal_black_level_offset	0
tone_mapping_input_signal_white_level_offset	0
shadow_gain_control	if MDCV SEI message is present, see clause G.3 otherwise, see clause G.4
highlight_gain_control	255
mid_tone_width_adjustment_factor	64
tone_mapping_output_fine_tuning_num_val	0
saturation_gain_num_val	1
saturation_gain_x[0]	0
saturation_gain_y[0]	118, if BT.2020 115, if BT.709

F.3 Recovery of shadow_gain_control with MDCV SEI message

This clause proposes a recovery procedure, for the value of the parameter **shadow_gain_control**, that is applicable when MDCV SEI/ST 2086 messages are available.

Indeed, the Mastering Display Colour Volume SEI message contains an information on the source picture mastering display nominal maximum luminance (**max_display_mastering_luminance** that is mapped to **hdrDisplayMaxLuminance** as specified by clause A.3.2) that may be used to adjust the value of **shadow_gain_control** as follows:

$$\mathbf{shadow\_gain\_control} = Clip3(0\,;255\,;\,Floor(r_s(\mathbf{hdrDisplayMaxLuminance})\times 127{,}5+0{,}5)) \quad \text{(F.2)}$$

with $r_s(x) = \frac{7{,}5}{\ln\left(1+4{,}7\times\left(\frac{x}{100}\right)^{\frac{1}{2{,}4}}\right)} - 2$

F.4 Recovery of shadow_gain_control without MDCV SEI message

It is likely that at the service level information or for a specific workflow the value of **hdrDisplayMaxLuminance** is known. This value can be input in the recovery procedure described in clause F.3. **hdrDisplayMaxLuminance** is set to the maximum luminance of the presentation display when available, otherwise it is arbitrarily set to a value of 1 000 cd/m². This value corresponds to the currently observed reference maximum display mastering luminance in most of the HDR markets.

Annex G (informative): ETSI TS 103 433 signalling in CTA-861-G

G.1 Introduction

CTA-861-G [i.8] specifies how ETSI TS 103 433 multi-part deliverable metadata can be carried on CE digital interfaces (e.g. HDMI). SL-HDR metadata may be delivered over a CTA-861-G interface to an SL-HDR capable sink in case an upstream source is not able to process SL-HDR metadata. This annex documents additional signalling and mapping for the HDR Dynamic Metadata Block (clause G.2) and the HDR Dynamic Metadata Extended InfoFrame (clause G.3) specified in CTA-861-G [i.8].

G.2 HDR Dynamic Metadata Data Block

The HDR Dynamic Metadata Data Block is used for signalling a sink device's specific HDR dynamic metadata support capabilities to a source device. ETSI TS 103 433 metadata support is signalled by a sink when the Supported HDR Dynamic Metadata Type has a value of 0x0002.

In the byte specified as Support Flags for Supported HDR Dynamic Metadata Type 0x0002, the bits 0-3 define the **ts_103_433_spec_version** syntax element and the bits 4-6 are used to define a **sl_hdr_mode_support** syntax element**.**

The present ETSI TS 103 433 multi-part deliverable specifies several modes that use the same SEI messages (SL-HDR Information and Mastering Display Colour Volume SEI messages) to carry SL-HDR metadata. Somehow, the processing associated to the different parts may be different and is specified in each part of ETSI TS 103 433. In order to signal sink capability with regards to the support of each part of this version of the multi-part deliverable ETSI TS 103 433, the new **sl_hdr_mode_support** syntax element is used.

The **ts_103_433_spec_version** syntax element value should be mapped to the value of **sl_hdr_spec_major_version_idc** (specified in clause A.2.2). **ts_103_433_spec_version** value should be equal or greater than 1.

For version V1.1.1 of the ETSI TS 103 433 [i.12], the **ts_103_433_spec_version** syntax element value should be mapped to the value of **ts_103_433_spec_version**. **ts_103_433_spec_version** value should be equal to 0.

- When the value of **ts_103_433_spec_version** is equal to or greater than 1, the **sl_hdr_mode_support** syntax element value should be mapped bitwise as specified in Table G.1.

Table G.1: Mapping of sl_hdr_mode_support bits for Supported HDR Dynamic Metadata Type 0x0002

sl_hdr_mode_support bitfield	Value
b_4	sl_hdr_mode_value_minus1 == 0 ? 1: 0
b_5	sl_hdr_mode_value_minus1 == 1 ? 1: 0
b_6	sl_hdr_mode_value_minus1 == 2 ? 1: 0

- b_4 should be equal to 1 when the sink is ETSI TS 103 433-1 capable. b_4 should be equal to 0 when the sink is not ETSI TS 103 433-1 capable.
- b_5 should be equal to 1 when the sink is ETSI TS 103 433-2 capable. b_5 should be equal to 0 when the sink is not ETSI TS 103 433-2 capable.
- b_6 should be equal to 1 when the sink is ETSI TS 103 433-3 capable. b_6 should be equal to 0 when the sink is not ETSI TS 103 433-3 capable.
- When **ts_103_433_spec_version** value is equal to 0, **sl_hdr_mode_support** bits 4-6 should be equal to 0 and ignored by the source.

G.3 HDR Dynamic Metadata Extended InfoFrame

In CTA-861-G [i.8], the HDR Dynamic Metadata Extended InfoFrame is used by a source device for identifying and delivering HDR dynamic metadata to a sink device.

HDR Dynamic Metadata for ETSI TS 103 433 are delivered by the source device using the Extended InfoFrame Type value equal to 0x0002 indicating "HDR Dynamic Metadata carried in Supplemental Enhancement Information (SEI) messages according to ETSI TS 103 433".

The sink device should interpret and use ETSI TS 103 433 multi-part deliverable metadata with its SL-HDR processing mode based on the value of **sl_hdr_mode_value_minus1** syntax element present in the SL-HDR Information SEI message as specified in clauses A.2.2 and A.2.3.

Annex H (informative): Change History

Date	Version	Information about changes
February 2017	1.1.2	Early draft: Plan of the specification Revise foreword (enabling multi-part deliverables) New annexes F, G, H, I Foreword
March 2017	1.1.3	Stable draft
May 2017	1.1.4	Second stable draft
June 2017	1.1.5	Draft for approval
July 2017	1.1.6	Draft for approval: addressing Edithelp! comments

History

Document history		
V1.1.1	August 2016	Publication as ETSI TS 103 433
V1.2.1	August 2017	Publication