TECHNICAL REPORT

ISO/IEC TR 23008-14

First edition
2018-08

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 14: Conversion and coding practices for HDR/WCG Y′CbCr 4:2:0 video with PQ transfer characteristics

Technologies de l'information — Codage à haut rendement et fourniture de supports dans les environnements hétérogènes —

Partie 14: Conversion et pratiques de codage pour la vidéo HDR/WCG Y′CbCr 4:2:0 avec caractéristiques de transfert PQ

Reference number
ISO/IEC TR 23008-14:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information* in collaboration with ITU-T. A technically aligned twin text is published as ITU-T H.Sup15.

A list of all parts in the ISO/IEC 23008 series can be found on the ISO website.

Introduction

High dynamic range (HDR) video is a type of video content in which the sample values span a larger luminance range than conventional standard dynamic range (SDR) video. HDR video can provide an enhanced viewer experience and can more accurately reproduce scenes that include, within the same image, dark areas and bright highlights, such as emissive light sources and reflections. On the other hand, wide colour gamut (WCG) video is video characterized by a wider spectrum of colours compared to what has been commonly available in conventional video. Recent advances in capture and display technology have enabled consumer distribution of HDR and WCG content. However, given the characteristics of such content, special considerations may need to be made, in terms of both processing and compression, compared to conventional content.

This document provides a set of recommended guidelines on processing of consumer distribution HDR/WCG video. This includes recommendations for converting a video signal, in a linear light RGB representation with Rec. ITU-R BT.2020 colour primaries, to a 10-bit, narrow range, PQ encoded (as defined in SMPTE ST 2084 and Rec. ITU-R BT.2100), 4:2:0, non-constant luminance Y′CbCr representation. These guidelines may also apply to other representations with higher bit depth or other colour formats, such as 4:4:4 Y′CbCr 12 bit video. The scope of this document is illustrated in Figure 1.

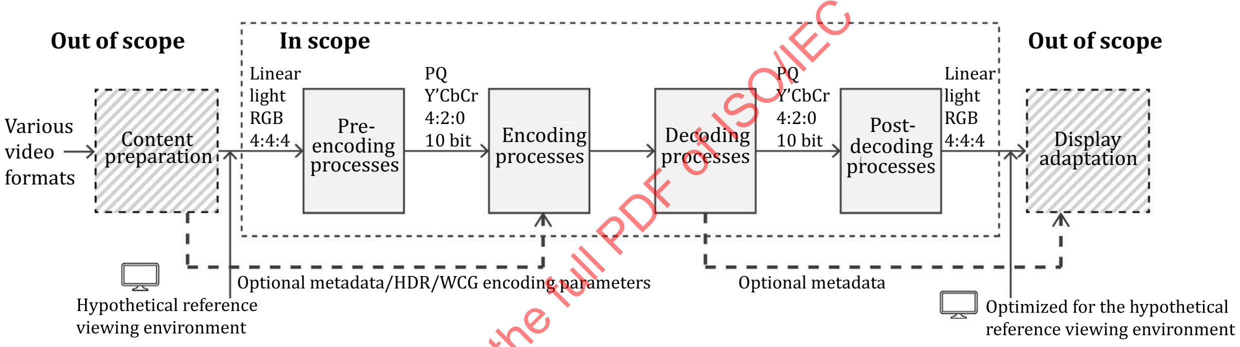


Figure 1 — Illustration of the scope of this document

The content preparation step, as well as the display adaptation step, are considered to be out of the scope of this document. However, metadata generated during the content preparation step may be passed through the encoder-decoder chain and can significantly affect the display adaptation step. The content preparation step may include filtering and image enhancement processing such as de-noising, colour correction, and sharpening, as well as other processes. Such methods are deliberately not described in this document. The processing steps described in this document are made available for reference only and the document does not contain any elements of normative nature. It is possible to replace one or more of the processing steps described in this document; for example, in order to reduce computational complexity or to improve fidelity. This document's intention is to provide some guidelines for operating an HDR/WCG video system that is constrained to code a 10-bit, PQ (as defined in SMPTE ST 2084 and Rec. ITU-R BT.2100), 4:2:0, non-constant luminance Y′CbCr signal representation. This configuration is also aligned with the HDR10 media profile defined in DECE v2.1, the interface defined in CTA 861G and the restrictions in the BD-ROM specifications. The processing steps in this document are designed for the case when the same hypothetical reference viewing environment (HRVE) is used before and after the HDR/WCG system. This document does not account for when the viewing environment used after the HDR/WCG system is different from the viewing environment used as the HRVE. In particular, display adaptation, such as the techniques described in the SMPTE ST 2094 standards, are not considered in this document. Report ITU-R BT.2390 contains additional information on viewing environments and examples of parameters that may be appropriate to apply for practical HDR/WCG systems. This document does not provide a description of any preferred HRVE, but acknowledges the fact that in many applications of HDR/WCG video, it may be desirable to have a well-defined HRVE description in order to ensure alignment between content preparation and content consumption.

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 14: Conversion and coding practices for HDR/WCG Y′CbCr 4:2:0 video with PQ transfer characteristics

1 Scope

This document provides guidance on the processing of high dynamic range (HDR) and wide colour gamut (WCG) video content. The purpose of this document is to provide a set of publicly referenceable recommended guidelines for the operation of AVC or HEVC video coding systems adapted for compressing HDR/WCG video for consumer distribution applications. This document includes a description of processing steps for converting from 4:4:4 RGB linear light representation video signals into non-constant luminance (NCL) Y′CbCr video signals that use the perceptual quantizer (PQ) transfer function defined in SMPTE ST 2084 and Rec. ITU-R BT.2100. Although the focus of this document is primarily on 4:2:0 Y′CbCr 10 bit representations, these guidelines are also applicable to other representations with higher bit depth or other colour formats, such as 4:4:4 Y′CbCr 12 bit video. In addition, this document provides some high-level recommendations for compressing these signals using either the AVC or HEVC video coding standards. A description of post-decoding processing steps is also included for converting these NCL Y′CbCr signals back to a linear light, 4:4:4 RGB representation.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Rec. ITU-T H.264 | ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*

Rec. ITU-T H.265 | ISO/IEC 23008-2, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in Rec. ITU-T H.264 | ISO/IEC 14496-10, Rec. ITU-T H.265 | ISO/IEC 23008-2 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
electro-optical transfer function
EOTF
function used in the post-decoding process to convert from a non-linear representation to a linear representation

3.2
full range
range in a fixed-point (integer) representation that spans the full range of values that could be expressed with that bit depth, such that, for 10-bit signals, black corresponds to code value 0 and peak white corresponds to code value 1023 for Y′

Note 1 to entry: As per the full range definition from Rec. ITU-R BT.2100.

3.3
inverse electro-optical transfer function
inverse EOTF
function used in the pre-encoding process to convert from a linear representation to a non-linear representation, computed as the inverse of the *EOTF* (3.1)

Note 1 to entry: In this document, the pre-encoding process is assumed to operate on HDR/WCG video content that has been prepared for a hypothetical reference viewing environment as shown in Figure 1. The content preparation step may contain processing such as applying an *opto-optical transfer function (OOTF)* (3.6), in which the HDR/WCG video is converted from one linear representation (corresponding to the scene) to another linear representation (corresponding to the display). The OOTF has the role of applying a "rendering intent". In systems where no such OOTF is applied in the content preparation step, the process of converting from a linear representation (corresponding to the scene) to a non-linear representation is typically called the *opto-electrical transfer function (OETF)* (3.5).

3.4
narrow range
range in a fixed-point (integer) representation that does not span the *full range* (3.2) of values that could be expressed with that bit depth such that, for 10 bit representations, the range from 64 (black) to 940 (peak white) is used for Y′ and the range from 64 to 960 is used for Cb and Cr

Note 1 to entry: As per the narrow range definition from Rec. ITU-R BT.2100.

Note 2 to entry: Narrow range is, in some applications, called by synonyms such as: "limited range", "video range", "legal range", "SMPTE range" or "standard range".

3.5
opto-electrical transfer function
OETF
function that converts linear scene light into the video signal, typically within a camera

3.6
opto-optical transfer function
OOTF
function that maps relative scene linear light (typically the camera output signal) to display linear light (typically, the signal driving a mastering monitor)

3.7
random access point access unit
RAPAU
access unit in the bitstream containing an intra coded picture with the property that all pictures following the intra coded picture in output order can be correctly decoded without using any information preceding the random access point access unit in the bitstream

3.8
transfer function
function that can be *EOTF* (3.1), *inverse EOTF* (3.3), *OETF* (3.5), inverse OETF, *OOTF* (3.6), or inverse OOTF

4 Abbreviated terms

AVC	advanced video coding (see Rec. ITU-T H.264 | ISO/IEC 14496-10)
CL	constant luminance
EOTF	electro-optical transfer function
FIR	finite impulse response
HD	high definition
HDR	high dynamic range
HEVC	high efficiency video coding (see Rec. ITU-T H.265 | ISO/IEC 23008-2)
HRVE	hypothetical reference viewing environment
HVS	human visual system
LUT	look-up table
MAD	mean absolute difference
NCL	non-constant luminance
PQ	perceptual quantizer (as defined in SMPTE ST 2084 and Rec. ITU-R BT.2100)
QP	quantization parameter
RAPAU	random access point access unit
RGB	colour system using red, green, and blue components
SSE	sum of squared errors
SDR	standard dynamic range
SEI	supplemental enhancement information
OETF	opto-electrical transfer function
OOTF	opto-optical transfer function
VUI	video usability information
WCG	wide colour gamut
XYZ	The CIE 1931 colour space. Y corresponds to the luminance signal.
Y′CbCr	colour space representation commonly used for video/image distribution as a way of encoding RGB information, also commonly expressed as YCbCr, $Y'C_BC_R$, or $Y'C'_BC'_R$. The relationship between Y′CbCr and RGB is dictated by certain signal parameters, such as colour primaries, transfer characteristics, and matrix coefficients. Unlike the (constant luminance) Y component in the XYZ representation, Y′ in this representation might not be representing the same quantity. Y′ is commonly referred to as “luma”. Cb and Cr are commonly referred to as “chroma”.

5 Conventions

5.1 General

The mathematical operators used in this document are similar to those used in the C programming language. However, the results of integer division and arithmetic shift operations are defined more precisely and additional operations are defined, such as exponentiation and real-valued division. Numbering and counting conventions generally begin from 0, e.g. “the first” is equivalent to the 0-th, “the second” is equivalent to the 1-th, etc.

5.2 Arithmetic operators

+ Addition

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

* Multiplication, including matrix multiplication

x^y Exponentiation. Denotes 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.

÷ Used to denote division in mathematical formulae where no truncation or rounding is intended.

$\frac{x}{y}$ Used to denote division in mathematical formulae where no truncation or rounding is intended.

$\sum_{i=x}^{y} f(i)$ The summation of $f(i)$ with i taking all integer values from x up to and including y.

x % y Modulus. Remainder of x divided by y, defined only for integers x and y with x ≥ 0 and y > 0.

5.3 Bit-wise operators

& Bit-wise “and”. When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

| Bit-wise “or”. When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

^ Bit-wise “exclusive or”. When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

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

x << y Arithmetic left shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values of y. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

5.4 Assignment operators

= Assignment operator

++ Increment, i.e. x+ + is equivalent to x = x + 1; when used in an array index, evaluates to the value of the variable prior to the increment operation.

−− Decrement, i.e. x− − is equivalent to x = x − 1; when used in an array index, evaluates to the value of the variable prior to the decrement operation.

+= Increment by amount given, i.e. x += 3 is equivalent to x = x + 3, and x += (−3) is equivalent to x = x + (−3).

−= Decrement by amount given, i.e. x −= 3 is equivalent to x = x − 3, and x −= (−3) is equivalent to x = x − (−3).

5.5 Relational, logical, and other operators

== Equality operator

!= Not equal to operator

!x Logical negation “not”

> Larger than operator

< Smaller than operator

≥ Larger than or equal to operator

≤ Smaller than or equal to operator

&& Conditional/logical “and” operator. Performs a logical “and” of its Boolean operators, but only evaluates the second operand if necessary.

| | Conditional/logical “or” operator. Performs a logical “or” of its Boolean operators, but only evaluates the second operand if necessary.

a ? b : c Ternary conditional. If condition a is true, then the result is equal to b; otherwise the result is equal to c.

5.6 Mathematical functions

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

Ceil(x) the smallest integer greater than or equal to x

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

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

$\mathrm{EOTF}^{-1}(x)$ the inverse EOTF used to convert a linear light representation to a non-linear light representation.

$$\mathrm{Max}(x,y)=\begin{cases} x & ; \quad x<y \\ y & ; \quad \text{otherwise} \end{cases}$$

$$\mathrm{Max}(x,y,z)=\begin{cases} x & ; \quad x>\mathrm{Max}(y,z) \\ y & ; \quad y>\mathrm{Max}(x,z) \\ z & ; \quad \text{otherwise} \end{cases}$$

$$\mathrm{Min}(x,y)=\begin{cases} x & ; \quad x<y \\ y & ; \quad \text{otherwise} \end{cases}$$

$$\mathrm{Min}(x,y,z)=\begin{cases} x & ; \quad x<\mathrm{Min}(y,z) \\ y & ; \quad y<\mathrm{Min}(x,z) \\ z & ; \quad \text{otherwise} \end{cases}$$

$$\mathrm{Round}(x)=\mathrm{Sign}(x)*\mathrm{Floor}(\mathrm{Abs}(x)+0.5)$$

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

EOTF(x) the EOTF used to convert a non-linear light representation x to a linear light representation.

5.7 Order of operations

When order of precedence in an expression is not indicated explicitly by use of parentheses, the following rules apply:

- operations of a higher precedence are evaluated before any operation of a lower precedence;
- operations of the same precedence are evaluated sequentially from left to right.

Table 1 specifies the precedence of operations from highest to lowest; a higher position in the table indicates a higher precedence.

NOTE For those operators that are also used in the C programming language, the order of precedence used in this document is the same as used in the C programming language.

Table 1 — Operation precedence from highest (at top of table) to lowest (at bottom of table)

Operations (with operands x, y, and z)
“x++”, “x−−”
“!x”, “−x” (as a unary prefix operator)
“x^y”
“x * y”, “x / y”, “x ÷ y”, “$\frac{x}{y}$”, “x % y”
“x + y”, “x − y” (as a two-argument operator), “$\sum_{i=x}^{y} f(i)$”
“x << y”, “x >> y”
“x < y”, “x ≤ y”, “x > y”, “x ≥ y”
“x = = y”, “x != y”
“x & y”
“x | y”
“x && y”
“x || y”
“x ? y : z”
“x..y”
“x = y”, “x += y”, “x −= y”

6 Overview

The HDR/WCG System described in this document consists of four major stages:

- a pre-encoding stage consisting of several pre-processing processes (Clause 7);
- an encoding stage (Clause 8);
- a decoding stage (Clause 9);
- a post-decoding stage, also consisting of several post-processing processes (Clause 10).

These four stages are applied sequentially, with the output of one stage being used as input to the next stage according to the above-mentioned order (see Figure 1).

It is assumed that both the input to and the output of the HDR/WCG System are 4:4:4, linear light, floating-point signals, in an RGB colour representation using the same colour primaries. The output signal is targeted to resemble the input video signal as closely as possible. Other video formats can be used as input to the HDR/WCG System by first converting them to the above defined input signal representation. The HDR/WCG System described in this document is, in practice, a system for both HDR and WCG video since it is assumed that the input video is represented with colour primaries in accordance with Rec. ITU-R BT.2020 and Rec. ITU-R BT.2100.

Two different models, *the simple reference model* and *the enhanced reference model*, are described in this document for the pre-encoding and encoding processes. The *simple reference model* corresponds to the reference configuration used in the MPEG call for evidence (CfE) on HDR and WCG[16], while the *enhanced reference model* corresponds to a new reference configuration that was developed in MPEG following the CfE. Both of these models were tested in the JCT-VC verification test on HDR/WCG Video Coding using HEVC Main 10 Profile[15]. For the decoding process and post-decoding processes, a single model is described.

The primary purpose of the pre-encoding process is to convert the video input from its 4:4:4 RGB linear light, floating-point signal representation to a signal that is suitable for a video encoder. The conversion to a non-linear representation is performed in an attempt to exploit the characteristics of the human visual system (HVS) that could allow the re-quantization of the signal at a limited precision.

NOTE 1 For a fixed-point linear HDR/WCG video representation, approximately a 28-bit integer representation would be required to avoid introducing visible quantization/banding errors due to the 28 f-stop linear light dynamic range (0.000 05 cd/m^2 to 10 000 cd/m^2) that is spanned by the PQ EOTF. In practice, the input to the HDR/WCG System will typically be in a non-linear representation that would either need to be first converted to linear light data or be directly converted to a non-linear representation using the PQ EOTF.

It is assumed that encoding and decoding is performed in a 4:2:0, 10-bit representation. An encoder is expected to make the best use of the encoding tools available according to a particular specification, profile, and level, given also the characteristics of the content and the limitations of the intended application and implementation. In particular, different encoding algorithms, such as algorithms for motion estimation, mode decision, rate allocation, rate control, and post-filtering control among other aspects, may have to be considered when encoding HDR/WCG material, in a given representation, compared to SDR material. The decoding process, on the other hand, is fully specified in the respective HEVC and AVC video coding standards, and a decoder should fully comply to the intended profile and level to properly decode and output the reconstructed video samples from a given input bitstream.

NOTE 2 The focus of this document is on consumer and direct-to-home applications, which are expected to use, at least in the near term future, a 4:2:0 10-bit format. Processes similar to the ones described in this document can be used for conversion and compression of other formats, such as 4:2:2 and 4:4:4 chroma formats or video with a bit depth higher than 10 bits.

The steps in the post-decoding process are aligned with what is commonly referred to as the non-constant luminance representation (NCL) in which colour conversion, to R′G′B′, is performed prior to applying the EOTF to produce linear RGB sample values.

There is no specific or minimum bit depth required for performing the operations described in the pre-encoding process and the post-encoding process. Using the precision associated with 64 bit floating-point operations will give high accuracy, but it is also possible to use fixed-point arithmetic or floating-point operations with precision lower than 64 bits. It is recommended to avoid using too low precision since it could potentially lead to loss of precision in the output video. The input to the encoding step and the output of the decoding step are, however, 10-bit integer representations.

7 Pre-encoding process

7.1 General

The pre-encoding process described in this document includes the following components:

a) a conversion component from a linear data representation to a non-linear data representation using the appropriate EOTF;

b) a colour format conversion component that converts data to the non-constant luminance Y′CbCr representation;

c) a conversion component that converts a floating-point to a fixed-point representation (e.g. 10 bits);

d) a chroma down-conversion component that converts data from 4:4:4 to 4:2:0.

NOTE Picture resolution scaling may also be a vital component of the pre-encoding process; for example, if the target system requires a particular image resolution be delivered to the decoder. It may be desirable, for example, to rescale a source from a 1 920 × 1 080 to a 3 840 × 2 160 resolution, or vice versa. Such scaling is not included in the scope of this document.

Figure 2 presents a diagram of how these components are combined in the *simple reference* model to generate the desirable outcome, in a conventional manner. In this model, all blocks work independently, whereas chroma subsampling is performed using fixed-point arithmetic and at the same precision as the target outcome.

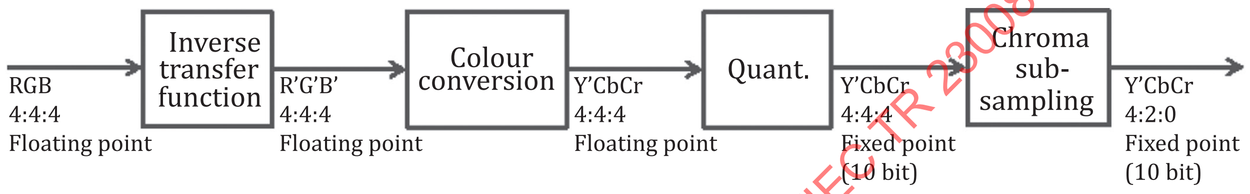


Figure 2 — Conventional pre-encoding process system diagram

Although this combination could be the most appropriate for some implementations, it has several limitations that can affect both its performance and implementation complexity. In this clause, the pre-encoding process components are first introduced in more detail and then the alternative configuration corresponding to the *enhanced reference model* is presented in 7.3. Recommendations on how to best utilize some of the conversion components are also presented.

7.2 Pre-encoding process stages

7.2.1 Conversion from a linear to a non-linear light representation: RGB to R′G′B′

Conversion from a linear to a non-linear light representation is performed using an inverse EOTF or as is commonly referred to in other specifications, an opto-electrical transfer function (OETF). In this document, the PQ EOTF defined in SMPTE ST 2084 and Rec. ITU-R BT.2100 is used.

More specifically, the non-linear light representation V of a linear light intensity signal L_o, which takes values normalized to the range [0, 1], can be computed as Formula (1):

$$V = \mathrm{EOTF}^{-1}(L_o) = \left(\frac{c_1 + c_2 * L_o^n}{1 + c_3 * L_o^n} \right)^m \tag{1}$$

where c_1, c_2, c_3, m, and n are constants, which are defined as given in Formulae (2) to (6):

$$c_1 = c_3 - c_2 + 1 = 3\,424 \div 4\,096 = 0.835\,937\,5 \tag{2}$$

$$c_2 = 2\,413 \div 128 = 18.851\,562\,5 \tag{3}$$

$$c_3 = 299 \div 16 = 18.687\,5 \tag{4}$$

$$m = 2\,523 \div 32 = 78.843\,75 \tag{5}$$

$$n = 1\,305 \div 8\,192 = 0.159\,301\,757\,812\,5 \tag{6}$$

The peak value of 1 for L_o is ordinarily intended to correspond to an intensity level of 10 000 candelas per square metre (cd/m^2), while the value of 0 for L_o is ordinarily intended to correspond to an intensity level of 0 cd/m^2. The behaviour of the inverse PQ EOTF in relationship to the Rec. ITU-R BT.709 OETF and the inverse of the Rec. ITU-R BT.1886 EOTF is shown in Figure 3.

NOTE A direct comparison of the inverse of the PQ EOTF with the Rec. ITU-R BT.709 OETF might not be appropriate since Rec. ITU-R BT.709 may assume the use of an OOTF during decoding.

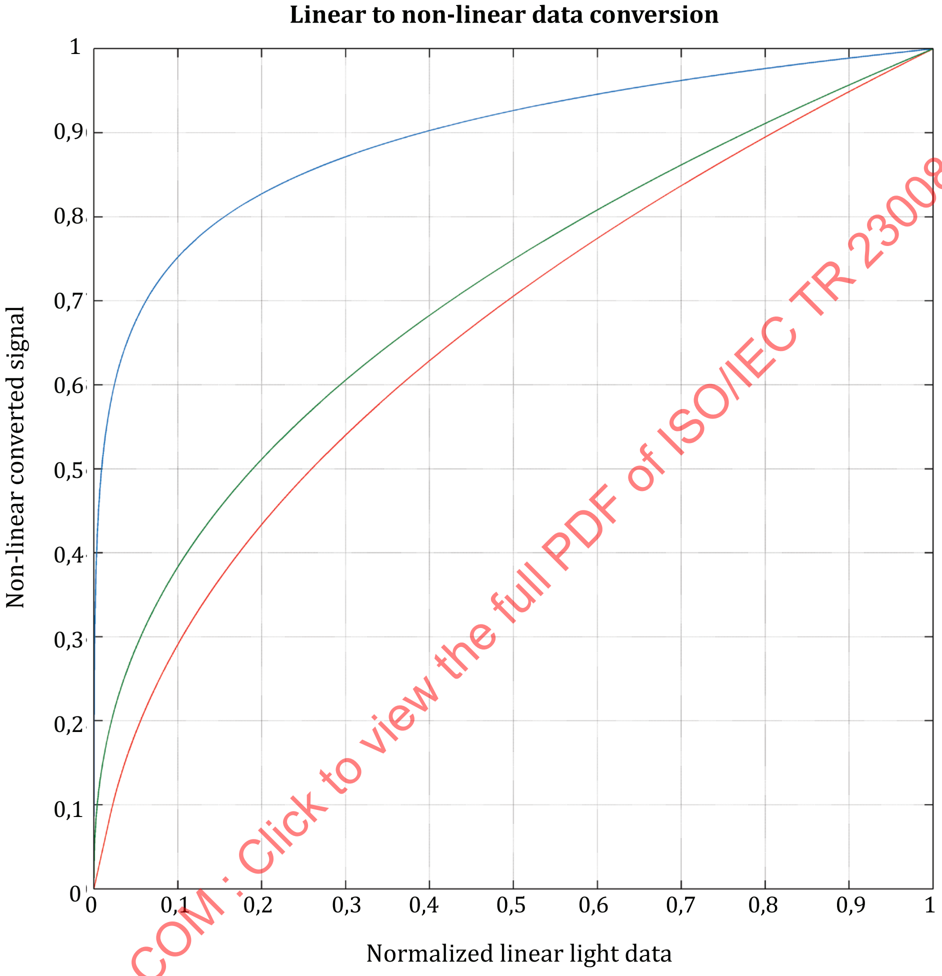


Figure 3 — Inverse of the PQ EOTF (top curve), inverse of the Rec. ITU-R BT.1886 EOTF (middle curve), and the Rec. ITU-R BT.709 OETF (bottom curve)

This process is applied to all R, G, and B linear light samples, where each component is a number between 0.0 (representing no light) and 1.0 (representing 10 000 cd/m^2). This results in their non-linear counterparts R′, G′, and B′ as given in Formulae (7) to (9):

$$R' = \mathrm{EOTF}^{-1}(R) \tag{7}$$

$$G' = \mathrm{EOTF}^{-1}(G) \tag{8}$$

$$B' = \mathrm{EOTF}^{-1}(B) \tag{9}$$

The resulting values for R′, G′, and B′ are numbers between 0.0 and 1.0. Although it is, in general, recommended to perform this conversion process using Formula (1) directly, this, however, may not be possible in some implementations given the complexity of the computation. Instead, look-up tables (LUT) may be preferred. Due to the characteristics of the conversion and the desire to achieve high precision for both low/dark and high values, it is highly recommended that, in such scenario, a non-uniformly indexed LUT interpolator is used as described in Reference [17]. Such schemes can achieve relatively high accuracy/minimum approximation error for the conversion, while achieving considerable memory savings.

7.2.2 Colour representation conversion: R′G′B′ to non-constant luminance Y′CbCr

Conversion from the R′G′B′ to the non-constant luminance Y′CbCr representation is commonly performed using a 3 × 3 matrix conversion process of the form given in Formula (10):

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} w_{YR} & w_{YG} & w_{YB} \\ w_{CbR} & w_{CbG} & w_{CbB} \\ w_{CrR} & w_{CrG} & w_{CrB} \end{bmatrix} * \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \mathbf{W} * \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{10}$$

where w_{YR}, w_{YG}, w_{YB}, w_{CbR}, w_{CbG}, w_{CbB}, w_{CbR}, w_{CbG}, and w_{CbB} are constants. The values for w_{YR}, w_{YG}, and w_{YB}, are set to exactly the same values used to convert R, G, and B data to the CIE 1931 Y (luminance) signal. For Rec. ITU-R BT.2100 colour primaries, these are defined as given in Formulae (11) to (13):

$$w_{YR} = 0.262\,7 \tag{11}$$

$$w_{YG} = 0.678\,0 \tag{12}$$

$$w_{YB} = 0.059\,3 \tag{13}$$

The resulting value for Y′ will be between 0.0 and 1.0. The values of the constants w_{CbR}, w_{CbG}, w_{CbB}, w_{CrR}, w_{CrG}, and w_{CrB} are computed in a manner that the resulting Cb and Cr components are always within the [−0.5, 0.5] range. This results in the following values given in Formulae (14) to (19):

$$w_{CbR} = -\frac{w_{YR}}{2*(1-w_{YB})} = -0.139\,630\,063 \tag{14}$$

$$w_{CbG} = -\frac{w_{YG}}{2*(1-w_{YB})} = -0.360\,639\,937 \tag{15}$$

$$w_{CbB} = 0.5 \tag{16}$$

$$w_{CrR} = 0.5 \tag{17}$$

$$w_{CrG} = -\frac{w_{YG}}{2*(1-w_{YR})} = -0.459\,785\,705 \tag{18}$$

$$w_{\mathrm{CrB}} = -\frac{w_{\mathrm{YB}}}{2*(1-w_{\mathrm{YR}})} = -0.040\ 214\ 295 \tag{19}$$

An alternative method to perform the same conversion process is presented in Rec. ITU-R BT.2020 and Rec. ITU-R BT.2100, where the chroma components are computed after the conversion of the luma component according to Formula (10) as given in Formulae (20) and (21):

$$\mathrm{Cb} = \frac{\mathrm{B}' - \mathrm{Y}'}{\mathrm{alpha}} \tag{20}$$

$$\mathrm{Cr} = \frac{\mathrm{R}' - \mathrm{Y}'}{\mathrm{beta}} \tag{21}$$

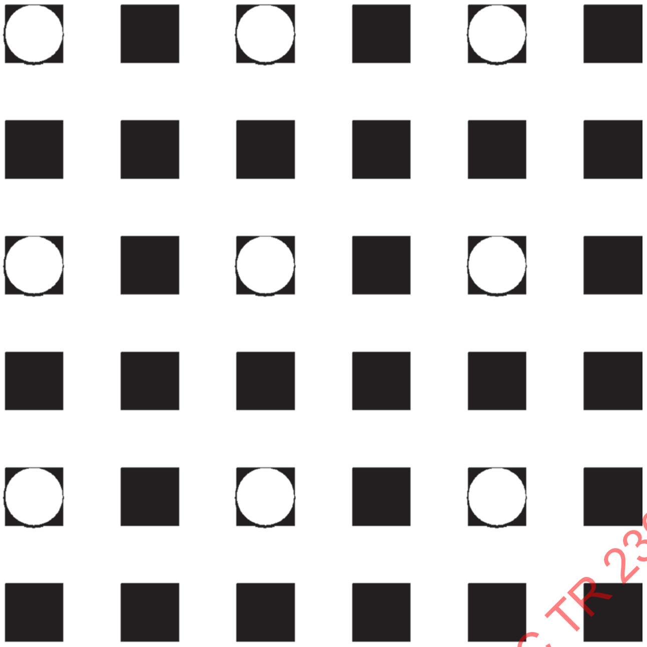
with alpha = 2* (1 − w_{YB}) and beta = 2* (1 − w_{YR}).

This can be seen as equivalent to the matrix presented in Formula (10).

The inverse process, i.e. converting Y′, Cb, and Cr data back to R′, G′, and B′ data, is described in 10.4.

7.2.3 Chroma down-conversion

Converting the HDR/WCG video data from a 4:4:4 representation to a 4:2:0 representation nominally involves filtering and down-converting/subsampling the two chroma planes in both the horizontal and vertical directions. It is, though, possible to apply more complex chroma down-conversion methods that preserve edges and thus reduce the impact of interpolated colour values that did not exist in the local neighborhood of a pixel in the original 4:4:4 representation. It is also a requirement, according to both Rec. ITU-R BT.2020 and Rec. ITU-R BT.2100, that the resulting chroma samples are co-sited with those of luma at even horizontal and vertical positions (see Figure 4) where the first sample and line are counted starting from zero.



Key

■ luma sample positions

○ chroma sample positions (chroma sample type 2)

Figure 4 — Chroma and luma sample location relationship

It is anticipated that a considerable amount of consumer electronics conversion systems would use 2-D separable finite impulse response (FIR) linear filters for low-pass filtering the chroma data before subsampling (2:1 decimation step). Such filters would basically be of the form given in Formula (22):

$$y\left[n\right]=\sum_{i=-N}^{N} b_i * x\left[n+i\right] \tag{22}$$

where

x[n] is the input chroma signal;

y[n] is the filtered output chroma signal;

(2 * N) is the filter order;

(2 * N + 1) is the number of taps of the filter;

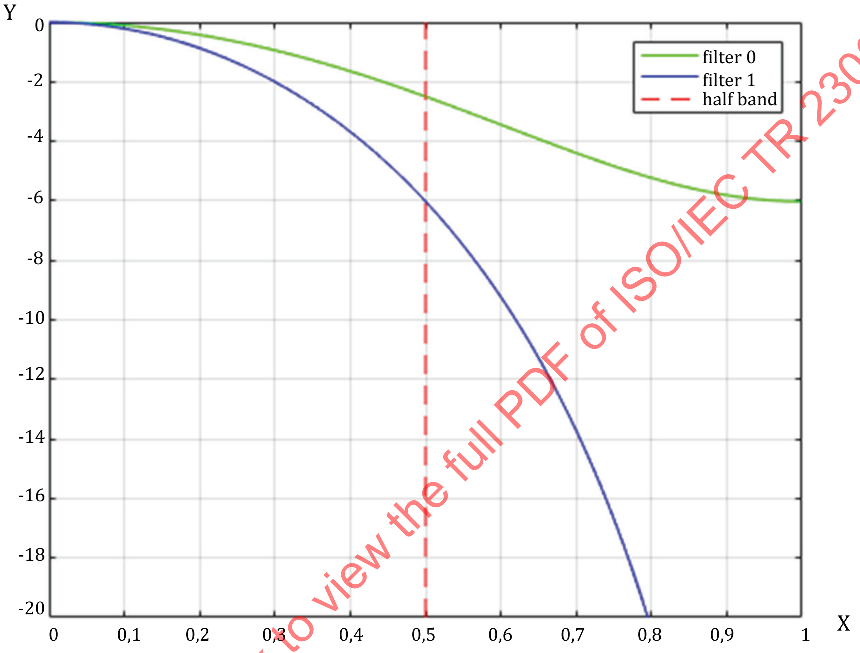
b_i is the coefficient of the filter at position *i*.

It has been observed that, especially due to the nonlinear characteristics of the PQ EOTF and its effect on quantization, special caution needs to be exercised when selecting the filter coefficients of such a resampling filter, in order to mitigate chroma "leakage" as defined in Reference [11]. Conventional filters, such as linear filters, that are commonly used for down-conversion of SDR chroma signals may potentially result in visual artefacts when applied to HDR/WCG signals. This document, however, only considers two short-tap-length linear FIR filters, which have been used in experiments for development of this report. Such filters can be utilized for both vertical and horizontal filtering of the chroma samples. Both *simple reference* and *enhanced reference* models use filter f_0 given in Table 2.

Table 2 — Suggested filters for chroma down-sampling

Filter	Filter coefficients		
	b_{-1}	b_0	b_1
f_0	1 ÷ 8	6 ÷ 8	1 ÷ 8
f_1	1 ÷ 4	2 ÷ 4	1 ÷ 4

The characteristics (magnitude and phase) of these filters are shown in Figure 5. Filter f_1 has a stronger attenuation that is equal to −6 dB at 0.5 π rad/s, whereas filter f_0 could potentially cause some aliasing artefacts due to having a significant amount of energy remaining in its stop-band.



Key

X normalized frequency (xπ radians/sample)

Y magnitude (dB)

Figure 5 — Frequency response of filters f_0 and f_1 for chroma down-sampling

The up-conversion process, i.e. from a 4:2:0 representation back to a 4:4:4 representation, is discussed in 10.3.

7.2.4 Floating-point to fixed-point (narrow range) 10 bit conversion

A key component of the pre-encoding process is the conversion from a floating-point representation to a fixed-point, narrow range, 10-bit representation. This process is essentially a quantization step that would introduce some distortion. In general, the conversion process can be expressed as Formulae (23) and (24):

$$\mathrm{D}' = \mathrm{Clip3}\left[0, 2^{b} - 1, \mathrm{Round}\left(\mathrm{E}' * \mathrm{scale} + \mathrm{offset}\right)\right] \tag{23}$$

or equivalently:

$$\mathrm{D}' = \mathrm{Clip3}\left[0, \left(1 << b\right) - 1, \mathrm{Round}\left(\mathrm{E}' * \mathrm{scale} + \mathrm{offset}\right)\right] \tag{24}$$

where

E′ is the floating-point representation of a particular component;

D′ is the resulting quantized value using b bits

In this document, b =10. The scale and offset constants depend on the target range (narrow versus full range video) and the component type (luma, chroma, or colour primary components). More specifically, for the narrow range NCL representation, the scale and offset for the luma component are set as Formulae (25) to (27):

$$\mathrm{scale} = 219 * 2^{b-8} = 219 * \left[1 << \left(b - 8\right)\right] \tag{25}$$

$$\mathrm{offset} = 2^{b-4} = 1 << \left(b - 4\right) \tag{26}$$

$$\mathrm{DY}' = \mathrm{Clip3}\left\langle 0, \left(1 << b\right) - 1, \mathrm{Round}\left\{\mathrm{Y}' * 219 * \left[1 << \left(b - 8\right)\right] + 1 << \left(b - 4\right)\right\}\right\rangle \tag{27}$$

On the other hand, the fixed-point narrow range representation for the two chroma components can be computed as Formulae (28) and (29):

$$\mathrm{scale} = 224 * 2^{b-8} = 224 * \left[1 << \left(b - 8\right)\right] \tag{28}$$

$$\mathrm{offset} = 2^{b-1} = 1 << \left(b - 1\right) \tag{29}$$

and Formulae (30) and (31):

$$\mathrm{DCb} = \mathrm{Clip3}\left\langle 0, \left(1 << b\right) - 1, \mathrm{Round}\left\{\mathrm{Cb} * 224 * \left[1 << \left(b - 8\right)\right] + 1 << \left(b - 1\right)\right\}\right\rangle \tag{30}$$

$$\mathrm{DCr} = \mathrm{Clip3}\left\langle 0, \left(1 << b\right) - 1, \mathrm{Round}\left\{\mathrm{Cr} * 224 * \left[1 << \left(b - 8\right)\right] + 1 << \left(b - 1\right)\right\}\right\rangle \tag{31}$$

NOTE For a 10 bit narrow range representation, DY′ results in a value within the range of [64, 940]. Similarly, DCb and DCr result in values within the range of [64, 960].

Figure 6 presents the mapping of non-normalized, gray (R=G=B) linear light values to a non-linear representation according to certain transfer functions; HDR (PQ) and SDR (gamma 2.4 for Rec. ITU-R BT.709/Rec. ITU-R BT.2020, assuming the use of the Rec. ITU-R BT.1886 EOTF during display) specifications.

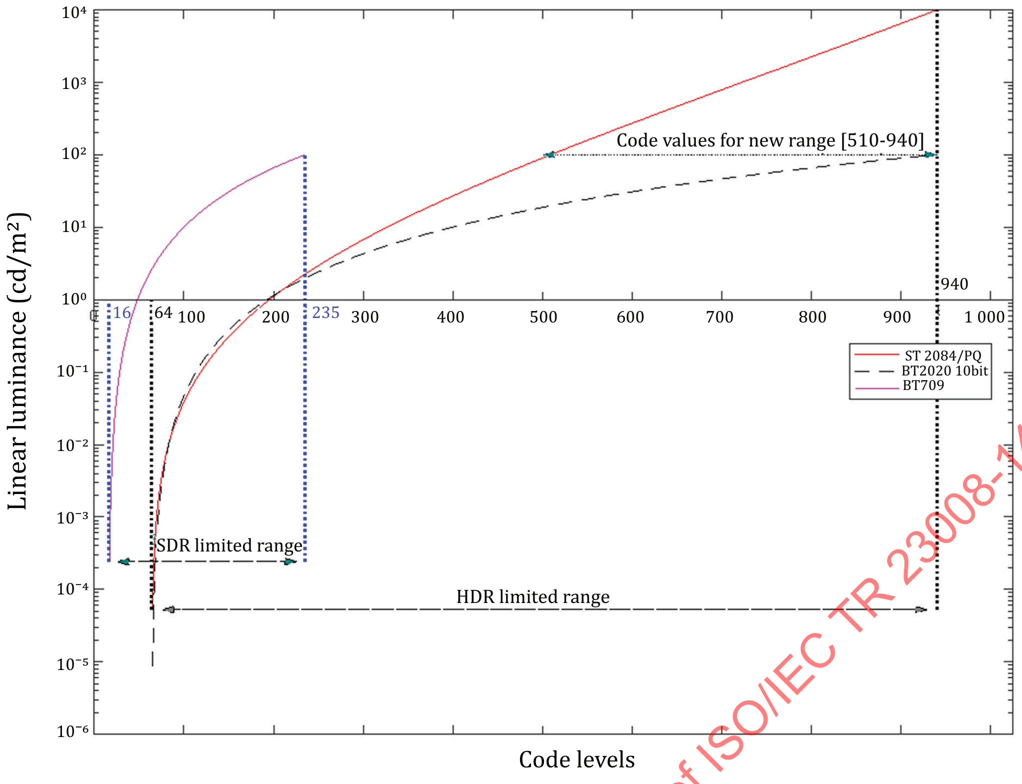


Figure 6 — Mapping of "gray" linear light values to quantized 8 (SDR only) and 10 bit (SDR and HDR) values

The inverse conversion process, i.e. converting from a fixed-point representation back to a floating-point representation, is discussed in detail in 10.2.

7.3 Closed loop pre-encoding conversion — Luma adjustment

7.3.1 General

As mentioned in 7.2.3, chroma leakage can occur in the NCL representation, primarily due to chroma down-sampling, potentially resulting in objectionable artefacts.

This subclause presents an alternative conversion method, which can considerably alleviate this problem. This method is called the luma adjustment method and it is basically a closed loop conversion process where the impact of chroma down-sampling, quantization, inverse quantization, and up-sampling, is accounted for during the luma conversion process. An example schematic diagram of such a system is presented in Figure 7. An iterative luma adjustment method, which is used in the enhanced reference model, is presented in 7.3.2. A closed form approach is presented in 7.3.3 that requires no iterations and is considerably faster than the iterative method.

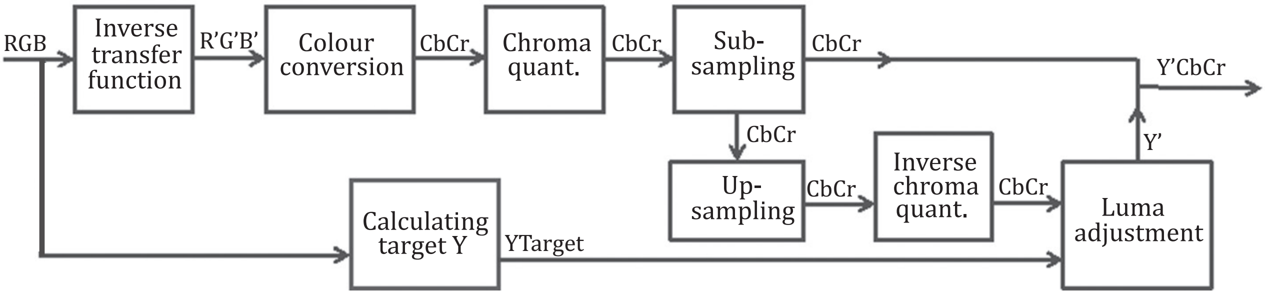


Figure 7 — Example schematic diagram of a closed loop pre-encoding conversion system

7.3.2 Luma adjustment — Iterative approach

7.3.2.1 General

7.2.2 and Formula (10) presented a conversion process from R′G′B′ to the Y′CbCr NCL representation. Given this process, the Cb and Cr components can be computed as Formulae (32) and (33):

$$\mathrm{Cb} = w_{\mathrm{CbR}} * \mathrm{R}' + w_{\mathrm{CbG}} * \mathrm{G}' + w_{\mathrm{CbB}} * \mathrm{B}' \qquad (32)$$

$$\mathrm{Cr} = w_{\mathrm{CrR}} * \mathrm{R}' + w_{\mathrm{CrG}} * \mathrm{G}' + w_{\mathrm{CrB}} * \mathrm{B}' \qquad (33)$$

Converting these two components to their target resolution, i.e. using the steps described in 7.2.3, followed by conversion back to the original representation resolution, as presented in 10.3, provides the opportunity to analyse the error introduced into the signal and potentially compensate for it. More specifically, performing quantization, down-scaling (QD), and subsequently up-scaling and inverse quantization (IQU) onto these components would result into the reconstructed $\widetilde{\mathrm{Cb}}$ and $\widetilde{\mathrm{Cr}}$ components, which are defined as given in Formulae (34) and (35):

$$\widetilde{\mathrm{Cb}} = \mathrm{IQU}\left[\mathrm{QD}\left(\mathrm{Cb}\right)\right] \qquad (34)$$

$$\widetilde{\mathrm{Cr}}=\mathrm{IQU}\left[\mathrm{QD}(\mathrm{Cr})\right] \tag{35}$$

Luminance (Y), unlike luma (Y′), is computed given the linear R, G, and B component values using a formulation of the form given in Formula (36):

$$\mathrm{Y}=w_{\mathrm{YR}}*\mathrm{R}+w_{\mathrm{YG}}*\mathrm{G}+w_{\mathrm{YB}}*\mathrm{B} \tag{36}$$

Since R = EOTF (R′), this can be rewritten as Formula (37):

$$\mathrm{Y}=w_{\mathrm{YR}}*\mathrm{EOTF}(\mathrm{R}')+w_{\mathrm{YG}}*\mathrm{EOTF}(\mathrm{G}')+w_{\mathrm{YB}}*\mathrm{EOTF}(\mathrm{B}') \tag{37}$$

However, since the reconstructed $\widetilde{\mathrm{Cb}}$ and $\widetilde{\mathrm{Cr}}$ values will likely differ from the original Cb and Cr values, the reconstructed $\tilde{\mathrm{R}}',\tilde{\mathrm{G}}'$, and $\tilde{\mathrm{B}}'$ values will also differ from the original R′, G′, and B′ values. Therefore, the reconstructed luminance $\mathrm{Y}_{\mathrm{rec}}$, which is equal to Formula (38):

$$\mathrm{Y}_{\mathrm{rec}}=w_{\mathrm{YR}}*\mathrm{EOTF}\left(\tilde{\mathrm{R}}'\right)+w_{\mathrm{YG}}*\mathrm{EOTF}\left(\tilde{\mathrm{G}}'\right)+w_{\mathrm{YB}}*\mathrm{EOTF}\left(\tilde{\mathrm{B}}'\right) \tag{38}$$

will also differ from the original luminance Y. Using Formula (73), it can be computed as Formula (39):

$$\begin{aligned}\mathrm{Y}_{\mathrm{rec}}=\;&w_{\mathrm{YR}}*\mathrm{EOTF}\left(\mathrm{Y}'+a_{\mathrm{RCr}}*\widetilde{\mathrm{Cr}}\right)+w_{\mathrm{YG}}*\mathrm{EOTF}\left(\mathrm{Y}'+a_{\mathrm{GCb}}*\widetilde{\mathrm{Cb}}+a_{\mathrm{GCr}}*\widetilde{\mathrm{Cr}}\right)+\\&w_{\mathrm{YB}}*\mathrm{EOTF}\left(\mathrm{Y}'+a_{\mathrm{BCb}}*\widetilde{\mathrm{Cb}}\right)\end{aligned} \tag{39}$$

Chroma component dependent factors can then be defined as Formulae (40) to (43):

$$\mathrm{Crfactor}=a_{\mathrm{RCr}}*\widetilde{\mathrm{Cr}} \tag{40}$$

$$\mathrm{Gfactor}=a_{\mathrm{GCb}}*\widetilde{\mathrm{Cb}}+a_{\mathrm{GCr}}*\widetilde{\mathrm{Cr}} \tag{41}$$

$$\mathrm{Cbfactor}=a_{\mathrm{BCb}}*\widetilde{\mathrm{Cb}} \tag{42}$$

resulting in Formula (43) for $\mathrm{Y}_{\mathrm{rec}}$:

$$\mathrm{Y}_{\mathrm{rec}}=w_{\mathrm{YR}}*\mathrm{EOTF}(\mathrm{Y}'+\mathrm{Crfactor})+w_{\mathrm{YG}}*\mathrm{EOTF}(\mathrm{Y}'+\mathrm{Gfactor})+w_{\mathrm{YB}}*\mathrm{EOTF}(\mathrm{Y}'+\mathrm{Cbfactor}) \tag{43}$$

The intent of the luma adjustment method is to try and locate the value of Y′ that would minimize distortion for $\mathrm{Y}_{\mathrm{rec}}$ compared to the original luminance value Y. Unfortunately, due to the non-linear characteristics of the EOTF, solving for Y′ given Y and the values of Crfactor, Gfactor, and Cbfactor is not a straightforward process. However, root-finding numerical methods, such as the bisection method, can be used instead. The performance, and more specifically the convergence speed and accuracy of these methods, is considerably impacted by the selection of the initial interval as well as the computations performed during the search.

NOTE Alternative methods that try to approximate the impact on Y or methods that evaluate the impact on other components have also been suggested and have remained under study.

The target of the luma adjustment process is to minimize luminance distortion, which can be realized through the following ordered steps.

a) Calculate the luminance value Y from the original R, G, and B, e.g. using Formula (36). This will be referred to as the $\mathrm{Y}_{\mathrm{target}}$ value.

b) Convert the R, G, and B data to their R′, G′, and B′ representation.

c) Given the R′, G′, and B′ planes generate the Cb and Cr chroma planes.

d) Down-scale and quantize the chroma planes to their target representation.

e) De-quantize and up-convert the chroma planes back to their original representation, i.e. $\widetilde{\mathrm{Cb}}$ and $\widetilde{\mathrm{Cr}}$

f) Calculate Crfactor, Gfactor, and Cbfactor from $\widetilde{\mathrm{Cb}}$ and $\widetilde{\mathrm{Cr}}$.

g) Given the reconstructed chroma planes, try to find for each luma position an appropriate Y′ value, i.e. $\mathrm{Y'}_{\mathrm{adjust}}$, that would potentially result in a minimum distortion for a particular aspect of the signal, i.e. in this case minimum luminance distortion. The value of $\mathrm{Y'}_{\mathrm{adjust}}$ at each luma position would be the value used for the encoding of the luma signal. In particular, $\mathrm{Y'}_{\mathrm{adjust}}$ would be the solution to Formula (44):

$$\begin{aligned}\mathrm{Y}_{\mathrm{target}} = {} & w_{\mathrm{YR}} * \mathrm{EOTF}\left(\mathrm{Y'}_{\mathrm{adjust}} + \mathrm{Crfactor}\right) + w_{\mathrm{YG}} * \mathrm{EOTF}\left(\mathrm{Y'}_{\mathrm{adjust}} + \mathrm{Gfactor}\right) + \\ & w_{\mathrm{YB}} * \mathrm{EOTF}\left(\mathrm{Y'}_{\mathrm{adjust}} + \mathrm{Cbfactor}\right)\end{aligned} \tag{44}$$

7.3.2.2 Bisection search

The bisection search method is an iterative technique that is commonly used to derive the roots of an equation of the form f(x) = 0. The function f(x) is assumed to be continuous, and defined over an interval [a, b], where f(a) and f(b) need to have opposite signs. In this application, it is desirable to have a unique solution. For this to be guaranteed, the behaviour of f(x) within this interval needs to also be strictly monotonic (i.e. consistently increasing or decreasing). At each iteration of the search, the interval is divided by two, i.e. it is divided at the midpoint $c = \frac{(a+b)}{2}$. Then, the value of the function f(c) is computed at this point. Depending on the value of f(c) and its relationship with f(a) and f(b), a new smaller interval is defined that satisfies the opposite sign condition. This search is repeated until a root is found, the interval is sufficiently small, or if a certain maximum number of iterations has been achieved.

This method can be used to find the $\mathrm{Y'}_{\mathrm{adjust}}$ value for luma, as discussed in the previous clause, in the following way:

Let x be the value that represents the quantized representation of the luma component, as defined in 7.2.1. Furthermore, let g(x) be the de-quantization function [Formula (72)], that maps the value of x back to its original representation and essentially a value between [0, 1]. Then, the narrow range, 10-bit representation can be computed as Formula (45):

$$\mathrm{g}(x) = (x - 64.0) \div 876.0 \tag{45}$$

Now, let f(x) be the function, as given in Formula (46):

$$\begin{aligned}\mathrm{f}(x) = {} & w_{\mathrm{YR}} * \mathrm{EOTF}\left[\mathrm{g}(x) + \mathrm{Crfactor}\right] + w_{\mathrm{YG}} * \mathrm{EOTF}\left[\mathrm{g}(x) + \mathrm{Gfactor}\right] + \\ & w_{\mathrm{YB}} * \mathrm{EOTF}\left[\mathrm{g}(x) + \mathrm{Cbfactor}\right] - \mathrm{Y}_{\mathrm{target}}\end{aligned} \tag{46}$$

The initial interval can be set as the entire range, i.e. [a, b] = [64, 940]. Given the characteristics of the PQ EOTF it is expected that f(a) ≤ 0 and f(b) ≥ 0, which satisfy the bisection conditions for a unique solution. The value of f(x) at position x can be computed as $x = \frac{(a+b)}{2} = 502$ and the interval can then be adjusted accordingly. If, for example, f(502) >0 then the interval will be adjusted to [64, 502]. The next evaluation point will then be the middle point of this new interval, i.e. the point at $x = \frac{(64+502)}{2} = 283$. At this point, if now f(283) < 0 then the interval will be adjusted to [283, 502]. The process can continue until either a value is found that satisfies f(x) = 0, or when the interval is of the form [k, k+1], e.g. [343, 344]. In this case, both values can be evaluated and the one resulting in the smallest distortion for Y or EOTF^{-1} (Y) can be used.

A critical component of this method is the selection of the initial interval. The brute force approach is to use the entire valid range as the initial range, for example, for the 10-bit narrow representation the range of [64, 940] as in the previous example. This, though, may require, in the worst-case, a number of iterations equal to the target bit depth of the content to reach an interval of size one. However, using information about the original colour and the chroma values Cb* and Cr*, upper and lower bounds can be found for the value of Y'_{adjust}, greatly reducing the size of the initial interval. This can considerably reduce the average number of iterations and thus have a direct impact on the number of computations performed and the overall complexity of the process as described in Reference [19]. Three such bounds are preferably used. The first is described in Reference [12] and uses the definitions in Formulae (47) to (49):

$$R'_{\mathrm{bound}} = \mathrm{EOTF}^{-1}\left(Y_{\mathrm{target}}\right) - \mathrm{Crfactor} \tag{47}$$

$$G'_{\mathrm{bound}} = \mathrm{EOTF}^{-1}\left(Y_{\mathrm{target}}\right) - \mathrm{Gfactor} \tag{48}$$

$$B'_{\mathrm{bound}} = \mathrm{EOTF}^{-1}\left(Y_{\mathrm{target}}\right) - \mathrm{Cbfactor} \tag{49}$$

Y′ will always be in the interval $\left[\mathrm{Min}\left(R'_{\mathrm{bound}}, G'_{\mathrm{bound}}, B'_{\mathrm{bound}}\right), \mathrm{Max}\left(R'_{\mathrm{bound}}, G'_{\mathrm{bound}}, B'_{\mathrm{bound}}\right)\right]$. The proof for this is out of the scope of this document, but is presented in Reference [12]. The second bound uses the fact that all three variables $R'_{\mathrm{bound}}, G'_{\mathrm{bound}}$, and B'_{bound} are smaller than 1, which leads to a tighter upper bound, i.e. $Y' \leq \mathrm{EOTF}^{-1}\left(Y_{\mathrm{target}}\right)$. This makes use of the fact that the EOTF is convex as described in Reference [12]. The third bound relies on the fact that all three of the reconstructed colour components $\tilde{R}, \tilde{G}$, and $\tilde{B}$ cannot simultaneously be smaller (or all simultaneously larger) than the original colour components R′, G′, and B′ for the computation of Y'_{adjust}. By using the definitions given in Formulae (50) to (52):

$$Y'_{\tilde{\mathrm{Cb}}} = R' + \mathrm{Crfactor} \tag{50}$$

$$Y'_{\mathrm{G}} = G' + \mathrm{Gfactor} \tag{51}$$

$$Y'_{\tilde{\mathrm{Cr}}} = B' + \mathrm{Cbfactor} \tag{52}$$

It can be shown that Y'_{adjust} should be in the interval Formula (53):

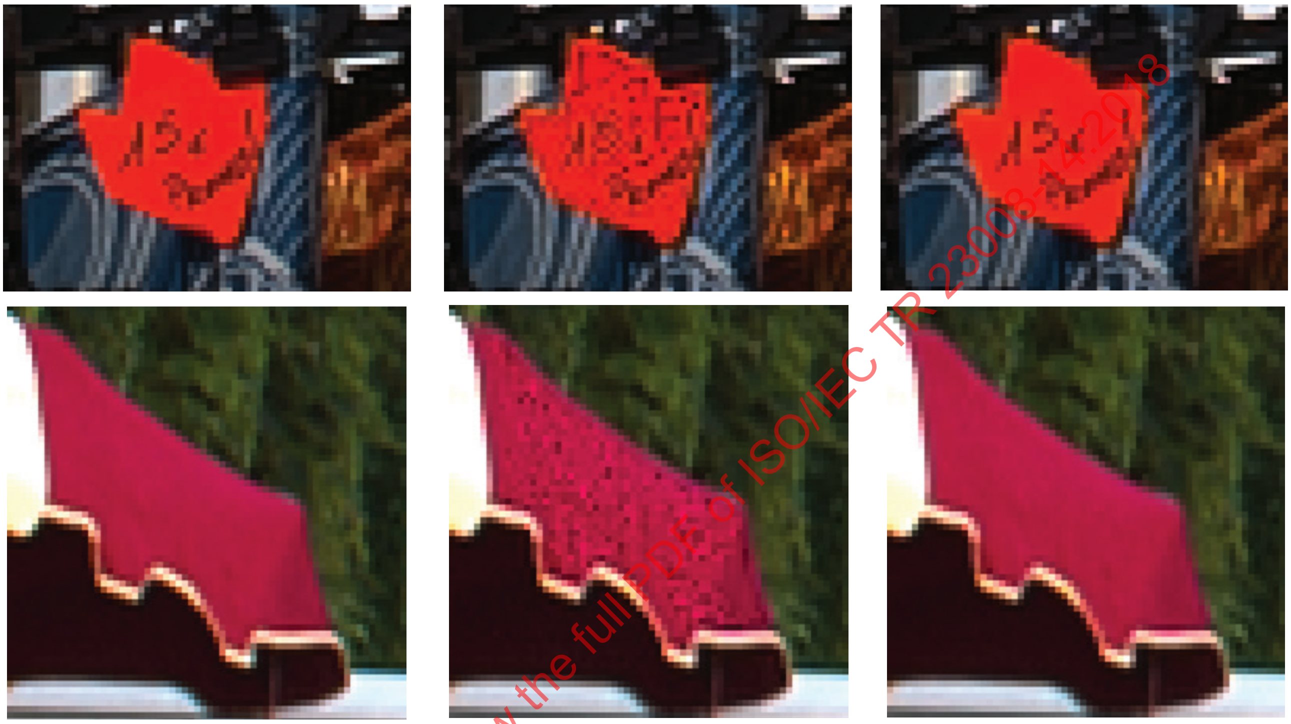
$$\left[Y'_{\min}, Y'_{\max}\right] = \left[\mathrm{Min}\left(Y'_{\tilde{\mathrm{Cb}}}, Y'_{\mathrm{G}}, Y'_{\tilde{\mathrm{Cr}}}\right), \mathrm{Max}\left(Y'_{\tilde{\mathrm{Cb}}}, Y'_{\mathrm{G}}, Y'_{\tilde{\mathrm{Cr}}}\right)\right] \tag{53}$$

By combining all three bounds together, the minimum and maximum bounds for Y′ can be computed as Formula (54):

$$Y'_{\mathrm{low\_bound}} = \mathrm{Max}\left(\mathrm{Min}\left(R'_{\mathrm{bound}}, G'_{\mathrm{bound}}, B'_{\mathrm{bound}}\right), Y'_{\min}\right) \tag{54}$$

$$Y'_{high\_bound} = \begin{cases} \mathrm{Min}\left[\mathrm{EOTF}^{-1}\left(Y_{target}\right), Y'_{max}\right] & ;\ \text{if}\ \mathrm{Max}\left(R'_{bound}, G'_{bound}, B'_{bound}\right) < 1 \\ \mathrm{Min}\left[\mathrm{Max}\left(R'_{bound}, G'_{bound}, B'_{bound}\right), Y'_{max}\right] & ;\ \text{otherwise} \end{cases} \tag{55}$$

Finally, the initial interval [a, b] for x is calculated as $a = \mathrm{Floor}\left[g^{-1}\left(Y'_{low\_bound}\right)\right]$ and $b = \mathrm{Ceil}\left[g^{-1}\left(Y'_{high\_bound}\right)\right]$ where $g^{-1}(x)$ is the inverse of Formula (45).



a) Originals in 4:4:4

b) Traditional subsampling

c) Luma adjustment based subsampling

Figure 8 — Tone-mapped examples showing improvements of the luma adjustment method

Figure 8 shows an example of what the difference can be when performing traditional subsampling, according to the simple reference model as described in 7.2.3 and using the luma component unchanged [Figure 8 b)] compared to the luma adjustment method that is described in this clause [Figure 8 c)] and is used in the enhanced reference model. These images were processed using the Rec. ITU-R BT.709 colour primaries to more easily demonstrate the differences. Since the printed medium cannot reproduce HDR images, tone-mapped versions are calculated using Formula (56):

$$R_{SDR} = \mathrm{Clip3}\left[0,255,255 * \left(R_{HDR} * 2^{c}\right)^{\frac{1}{\gamma}}\right] \tag{56}$$

where

γ = 2.2;

c is an exposure parameter set to make the SDR image look similar to the HDR image.

The artefacts are clearly more visible in the simple reference model subsampling case compared to when using the enhanced reference model method.

7.3.3 Luma adjustment — Closed form solution

The bisection search method described in the previous clause, has a worst-case complexity for a 10 bit data representation of ten iterations. This might be a problem for certain real-time applications, and in particular for hardware implementations. Quite often, hardware systems are designed by taking into account the worst-case scenario and by assuming that the same number of processing steps is required for each block or sample. Even though the complexity of the bisection method could be further bounded by limiting the maximum number of iterations, it may be beneficial for such applications to use a closed-form solution that is able to determine an appropriate luma value in a single step.

A closed-form solution can be found as follows. First, down-sampled chroma samples are obtained. Then, chroma is up-sampled to the original (luma) resolution by applying a chosen up-sampling filter. Then, for every pixel, the algorithm estimates a luma value $\mathrm{Y'}_{\mathrm{adjust}}$ that, in combination with the up-sampled $\widetilde{\mathrm{Cb}}$ and $\widetilde{\mathrm{Cb}}$ values, will result in a reconstructed $\widetilde{\mathrm{RGB}}$ pixel with colour component values of $\left\{\tilde{\mathrm{R}},\tilde{\mathrm{G}},\tilde{\mathrm{B}}\right\}$. It is highly desirable that $\widetilde{\mathrm{RGB}}$ is as close as possible to the original linear light RGB value according to a chosen distance metric. The difference between the two RGB values is denoted as Formula (57):

$$D=\mathrm{Diff}\left(\mathrm{RGB},\widetilde{\mathrm{RGB}}\right) \tag{57}$$

Depending on the chosen distance metric Diff(x), different closed-form solutions to the optimization problem can be obtained. In the following, a solution based on the weighted sum of the differences of linear R, G, and B data is described.

In particular, the square of the sum of the weighted differences between the individual R, G, and B components can be computed as Formula (58):

$$D=\left[\mathrm{w_R}*\left(\tilde{\mathrm{R}}-\mathrm{R}\right)+\mathrm{w_G}*\left(\tilde{\mathrm{G}}-\mathrm{G}\right)+\mathrm{w_B}*\left(\tilde{\mathrm{B}}-\mathrm{B}\right)\right]^2 \tag{58}$$

This is also equivalent to Formula (59):

$$D=\left\{\mathrm{w_R}*\left[\mathrm{EOTF}\left(\tilde{\mathrm{R}}'\right)-\mathrm{EOTF}\left(\mathrm{R}'\right)\right]+\mathrm{w_G}*\left[\mathrm{EOTF}\left(\tilde{\mathrm{G}}'\right)-\mathrm{EOTF}\left(\mathrm{G}'\right)\right]+\mathrm{w_B}*\left[\mathrm{EOTF}\left(\tilde{\mathrm{B}}'\right)-\mathrm{EOTF}\left(\mathrm{B}'\right)\right]\right\}^2 \tag{59}$$

where EOTF(x) is the PQ EOTF. If weights $\mathrm{w_R}$, $\mathrm{w_G}$, and $\mathrm{w_B}$ are set equal to the contribution of the linear R, G, and B components to the luminance component, this cost function would be minimizing the squared difference between the luminance values.

NOTE 1 It is also possible to use other error functions, such as the sum of the R, G, and B component squared errors, which can result in different solutions.

Finding a closed-form solution for Y′ may be difficult because of the non-trivial form of the PQ EOTF. In order to obtain a closed form solution, the EOTF(x) is approximated with a first-degree polynomial using the truncated Taylor series expansion given in Formula (60):

$$\mathrm{EOTF}\left(x_i+\Delta\right)\approx\mathrm{EOTF}\left(x_i\right)+\mathrm{EOTF}'\left(x_i\right)*\Delta_x \tag{60}$$

where EOTF′(x_i) is the value of the derivative of EOTF(x) with respect to x at point x_i and Δ_x is the change in the value of x. Substituting Formula (60) into Formula (59), the cost function is approximated as Formula (61):

$$D=\left[w_{\mathrm{R}}*\mathrm{EOTF}'\left(\mathrm{R}'\right)*\Delta_{\mathrm{R}}+w_{\mathrm{G}}*\mathrm{EOTF}'\left(\mathrm{G}'\right)*\Delta_{\mathrm{G}}+w_{\mathrm{B}}*\mathrm{EOTF}'\left(\mathrm{B}'\right)*\Delta_{\mathrm{B}}\right]^2 \tag{61}$$

Colour component values $\tilde{\mathrm{R}}',\tilde{\mathrm{G}}'$, and $\tilde{\mathrm{B}}'$ in the EOTF domain can be obtained from Y′, $\widetilde{\mathrm{Cb}}$, and $\widetilde{\mathrm{Cr}}$ data using the inverse colour transformation described in 10.4 using Formula (73).

Substituting Formula (73) into Formula (61), results in Formula (62):

$$D=\left\{\begin{array}{l} w_{\mathrm{R}} * \mathrm{EOTF}'(\mathrm{R}') * \left[a_{\mathrm{RY}} * \mathrm{Y}'_{\mathrm{adjust}} + a_{\mathrm{RCb}} * \widetilde{\mathrm{Cb}} + a_{\mathrm{RCr}} * \widetilde{\mathrm{Cr}} - \left(a_{\mathrm{RY}} * \mathrm{Y}' + a_{\mathrm{RCb}} * \mathrm{Cb} + \mathrm{a}_{\mathrm{RCr}} * \mathrm{Cr}\right)\right] + \\ w_{\mathrm{G}} * \mathrm{EOTF}'(\mathrm{G}') * \left[a_{\mathrm{GY}} * \mathrm{Y}'_{\mathrm{adjust}} + a_{\mathrm{GCb}} * \widetilde{\mathrm{Cb}} + a_{\mathrm{GCr}} * \widetilde{\mathrm{Cr}} - \left(a_{\mathrm{GY}} * \mathrm{Y}' + a_{\mathrm{GCb}} * \mathrm{Cb} + \mathrm{a}_{\mathrm{GCr}} * \mathrm{Cr}\right)\right] + \\ w_{\mathrm{B}} * \mathrm{EOTF}'(\mathrm{B}') * \left[a_{\mathrm{BY}} * \mathrm{Y}'_{\mathrm{adjust}} + a_{\mathrm{BCb}} * \widetilde{\mathrm{Cb}} + a_{\mathrm{BCr}} * \widetilde{\mathrm{Cr}} - \left(a_{\mathrm{BY}} * \mathrm{Y}' + a_{\mathrm{BCb}} * \mathrm{Cb} + \mathrm{a}_{\mathrm{BCr}} * \mathrm{Cr}\right)\right] \end{array}\right\}^{2} \tag{62}$$

Sorting the expressions inside the brackets and substituting with their numerical values those coefficients in matrix **A** from Formula (73) that are equal to 0 and 1 results in Formula (63):

$$D=\left[w_{\mathrm{R}} * \mathrm{EOTF}'(\mathrm{R}') * \left(\mathrm{Y}'_{\mathrm{adjust}} - e_{\mathrm{R}}\right) + \mathrm{w}_{\mathrm{G}} * \mathrm{EOTF}'(\mathrm{G}') * \left(\mathrm{Y}'_{\mathrm{adjust}} - e_{\mathrm{G}}\right) + w_{\mathrm{B}} * \mathrm{EOTF}'(\mathrm{B}') * \left(\mathrm{Y}'_{\mathrm{adjust}} - e_{\mathrm{B}}\right)\right]^{2} \tag{63}$$

where e_R, e_G, and e_B are defined as Formulae (64) to (66):

$$e_{\mathrm{R}} = \mathrm{Y}' - \left(\widetilde{\mathrm{Cr}} - \mathrm{Cr}\right) * a_{\mathrm{RCr}} \tag{64}$$

$$e_{\mathrm{G}} = \mathrm{Y}' - \left(\widetilde{\mathrm{Cb}} - \mathrm{Cb}\right) * a_{\mathrm{GCb}} - \left(\widetilde{\mathrm{Cr}} - \mathrm{Cr}\right) * a_{\mathrm{GCr}} \tag{65}$$

$$e_{\mathrm{B}} = \mathrm{Y}' - \left(\widetilde{\mathrm{Cb}} - \mathrm{Cb}\right) * a_{\mathrm{BCb}} \tag{66}$$

Then, in order to find the local minimum, D is differentiated with respect to $\mathrm{Y}'_{\mathrm{adjust}}$ (e_R, e_G, and e_B do not depend on $\mathrm{Y}'_{\mathrm{adjust}}$), the derivative is set equal to zero, and the resulting formula is solved with respect to Y′. The value of Y′ is then equal to Formula (67):

$$\mathrm{Y}'_{\mathrm{adjust}} = \frac{w_{\mathrm{R}} * \mathrm{EOTF}'(\mathrm{R}') * e_{\mathrm{R}} + w_{\mathrm{G}} * \mathrm{EOTF}'(\mathrm{G}') * e_{\mathrm{G}} + w_{\mathrm{B}} * \mathrm{EOTF}'(\mathrm{B}') * e_{\mathrm{B}}}{w_{\mathrm{R}} * \mathrm{EOTF}'(\mathrm{R}') + w_{\mathrm{G}} * \mathrm{EOTF}'(\mathrm{G}') + w_{\mathrm{B}} * \mathrm{EOTF}'(\mathrm{B}')} \tag{67}$$

Applying Formulae (64), (65), (66), and (67), one can obtain the adjusted value of Y′ in a single step. This approach can be adopted by applications that would benefit from or require lower complexity and a fixed number of operations per luma sample. Experimental results suggest that the algorithm described above can achieve a considerable complexity reduction, while at the same time having only a small difference in terms of objective metric performance compared to the bisection search method. In a particular reference implementation, a speedup factor of around 2.5 times for the total colour conversion runtime compared to the bisection search method was reported in Reference [18]. The differences in the objective measurements are mostly due to the approximation of the EOTF with its tangent. Subjective performance appears to be similar to that of the bisection search method.

The derivative EOTF′(x) in Formula (67) can be computed using a formula obtained by the differentiation of the EOTF or by the definition of a derivative, i.e. by dividing the change in the function value by the increment of the function argument. Alternatively, the derivative values can be pre-computed and stored in a LUT.

As mentioned earlier, the weights w_R, w_G, and w_B can be chosen based on the desired precision or importance of each component. For example, they can be set equal to 1 or based on the contribution of each colour component to the luminance CIE 1931 Y component.

The above algorithm can be summarized as follows.

a) Convert the original R, G, and B data to their R′, G′, and B′ representation, if needed.

b) Given the R′, G′, and B′ planes generate the Cb and Cr chroma planes.

c) Down-scale chroma planes to 4:2:0 or 4:2:2 representation and quantize the samples.

d) De-quantize and up-convert the chroma back to their original resolution to obtain $\widetilde{\mathrm{Cb}}$ and $\widetilde{\mathrm{Cr}}$.

e) For each luma sample, calculate e_R, e_G, and e_B based on Formulae (64), (65), and (66), respectively.

f) Calculate $\mathrm{Y'}_{\mathrm{adjust}}$ based on Formula (67).

NOTE 2 The formulae described above do not take into account the effects of clipping of the R, G, and B data, within the range of 0 cd/m^2 to 10 000 cd/m^2, when applying the colour transformation from Y′CbCr to RGB. This can decrease the precision of the $\mathrm{Y'}_{\mathrm{adjust}}$ estimation obtained from Formula (67) when R, G, and B values are close to their upper limit of 10 000 cd/m^2. This clipping effect can be mitigated by modifying Formula (67) when one or more of the R, G, and B samples are clipped at 10 000 cd/m^2. The details of such a modification are considered as being out of the scope of this document. It can be argued that this effect would not be significant for most of the currently available HDR/WCG content given that, due to limitations of existing displays, HDR/WCG content are rarely mastered with a peak luminance value close to 10 000 cd/m^2. Therefore, the results obtained with this solution are likely difficult to distinguish from the results generated using the bisection search.

NOTE 3 Several other methods for performing the luma adjustment process using a closed form process, such as methods involving look-up tables, have also been suggested.

8 Encoding process

8.1 General

After pre-processing, the data is ready for compression. The HDR/WCG data coming out of the pre-processing step will exhibit slightly different characteristics than typical, standard dynamic range (SDR) data. This means that it can be possible to increase perceptual/subjective quality if the encoder is configured in a slightly different manner compared to when compressing SDR data. This clause presents two such differences in data characteristics and gives guidance on how an encoder can be configured to better exploit these differences.

8.2 Perceptual luma quantization

8.2.1 General

When processing SDR data, a power law transfer function such as the one described in Rec. ITU-R BT.709 is typically used. As is described above, the HDR/WCG data has instead undergone processing using the PQ transfer function defined in SMPTE ST 2084 and Rec. ITU-R BT.2100. This will in itself give a different characteristic of the processed data. One way to see this is to pre-process the same SDR data using both the Rec. ITU-R BT.709 transfer function and the PQ transfer function. For a 10-bit representation, if the original data has a peak brightness of 100 cd/m^2, the luma component will occupy all code levels from 64 to 940 if the Rec. ITU-R BT.709 transfer function is used. However, only code levels from 64 to 509 will be used in the case of the PQ transfer function. Since the step sizes are different in the two cases, a perturbation of +/− 1 code level around code level 509 (100 cd/m^2) in the PQ case will be equivalent of roughly +/− 4 code levels around code level 940 (also 100 cd/m^2) in the Rec. ITU-R BT.709 case. At the same time, a perturbation of +/− 1 code level around code level 80 (0.01 cd/m^2) in the PQ case will be roughly equivalent to a perturbation of +/− 1 code level around code level 80 (0.01 cd/m^2) in the Rec. ITU-R BT.709 case. Thus, if an encoder is wired to treat an error of one code level the same way regardless if it is at level 80 or 509, such an encoder will allow errors that are four times larger in the bright areas (at around 100 cd/m^2) if it uses the PQ transfer function compared to the use of Rec. ITU-R BT.709. In other words, by switching from a Rec. ITU-R BT.709 transfer function to PQ, a lot of bits will be redistributed from the bright areas of the image to the dark areas.

Thus, if an encoder with a certain setting has achieved a good balance between bright and dark areas for the Rec. ITU-R BT.709 transfer function, using the encoder with the same settings for PQ can produce images in which bright and dark areas are allocated too few and too many bits, respectively. This may result in more objectionable compression artefacts in the bright areas, while no perceivable improvement may be observed in the dark areas. For HDR/WCG data, this effect can be even more pronounced; the luminance increase from a code level increase is even higher at, for example, 4 000 cd/

m^2 than it is at 100 cd/m^2. Furthermore, it might be the case that the HDR/WCG content contains considerable amounts of noise in the dark areas, which can have a further impact on performance.

One way to ameliorate this effect is for the encoder to calculate the average luma value in a block and, using this value, adaptively adjust the block's quantization (QP) parameter. In particular, an encoder may increase or decrease the QP for the block if it is classified as a *dark* or a *bright* block, respectively. In this way, it can be possible to shift bits back from dark regions to bright regions and potentially achieve a result that can be perceptually more pleasing.

Shifting bits from dark to bright areas works in the opposite direction of the inverse EOTF, which assigns more code levels to dark values. However, the inverse EOTF is based on the best-case sensitivity for the human visual system. For instance, if all colours in a picture are dark, it predicts well how a small perturbation can be detected. However, if some colours are dark and others bright, it is harder for the visual system to detect perturbations in the dark areas and hence it is reasonable to move bits from dark to bright areas.

NOTE In the development of this technique, only the local luma characteristics were analysed, without trying to adapt performance based on regional or global brightness characteristics, among other potential considerations. Other aspects, such as the noise present in some of the material, can have also impacted coding performance and affected the design of the scheme. Further study may result in a revision of the described methods if additional evidence on the behaviour of the technology is obtained.

Many existing coders already use some form of adaptive QP method. As an example, such methods can be used to increase the QP in areas of very high variance (where it is perceptually hard to see errors) and decrease the QP in areas of lower variance (where errors are typically more visible). In some other systems, brightness, edges, motion, as well as other features, may also be considered. However, these methods are likely to have been designed based on SDR content characteristics. Given the above observations regarding the transfer function relationships, it is advised that, when compressing PQ encoded data, a QP adaptation method is considered that also takes into account these relationships. Other characteristics, such as colour, could also be considered.

A simple example QP adaptation method, which is used in the enhanced reference model, is presented below. This method was found to result in better subjective, as well as objective performance compared to the fixed QP coding configuration that is used in the simple reference model.

8.2.2 Example of luma-dependent adaptive quantization

The purpose of this approach is to try and match a similar level of distortion to a particular, gray level, luminance value x when either the power law transfer function of Rec. ITU-R BT.709 $f_{709}(x)$ or the PQ transfer function $f_{PQ}(x)$ are used, in combination with 10 bit quantization, as well as a codec's quantization level. More specifically, it is highly desirable to determine the QP value QP_{PQ} to be used with a PQ encoded value x, that would result in the same or similar distortion, or equivalently the same or similar quantization behavior Quant( ), if that same value was encoded using the Rec. ITU-R BT.709 transfer function and a known QP value QP_{709}. That is Formula (68):

$$\mathrm{Quant}\left[f_{709}(x), QP_{709}\right] \cong \mathrm{Quant}\left[f_{PQ}(x), QP_{PQ}\right] \tag{68}$$

The linear characteristics of the transformations employed on residual data in codecs such as AVC and HEVC enable the consideration of these formulations even after such transformations are performed. However, these also limit the consideration of such an optimization at a block level. Based on the characteristics of the Rec. ITU-R BT.709 and PQ transfer function and Formula (68), an approximate relationship between QP_{PQ} and QP_{709} can be computed as Formula (69):

$$QP_{PQ} = QP_{709} + dQP(x) \tag{69}$$

This relationship is depicted in Table 3, as well as in Figure 9, with int_L replacing the value of x. More specifically, in a particular implementation, int_L is computed by obtaining the average luma value of a 64 × 64 CTU block, $L_{average}$, and then rounding this quantity, i.e. $\mathrm{int}_L = \mathrm{Round}(L_{average})$. Based on this relationship, for every CTU, the QP will be adjusted according to its brightness by this dQP value.

Table 3 — Look-up table of the dQP value from the average of the luma value

luma int_L range	dQP
$int_L < 301$	3
$301 \le int_L < 367$	2
$367 \le int_L < 434$	1
$434 \le int_L < 501$	0
$501 \le int_L < 567$	−1
$567 \le int_L < 634$	−2
$634 \le int_L < 701$	−3
$701 \le int_L < 767$	−4
$767 \le int_L < 834$	−5
$int_L \ge 834$	−6

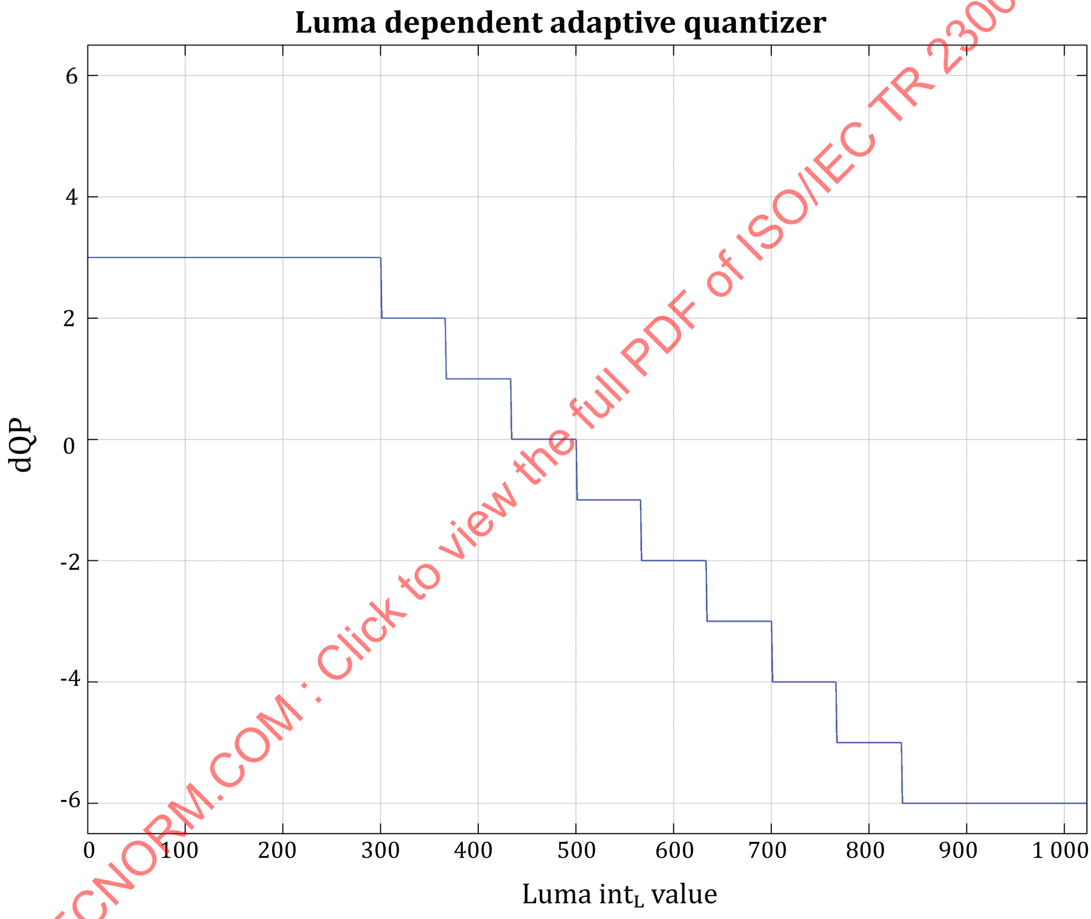


Figure 9 — Difference in QP value as a function of the average luma value in a 64 × 64 block

8.3 Chroma QP offset

8.3.1 General

Another major difference between HDR/WCG and SDR data has been observed in the characteristics of the chroma channels Cb and Cr. For 10 bit SDR content encoded using the Rec. ITU-R BT.709 transfer function and the Rec. ITU-R BT.709 colour space, typically all three components Y, Cb, and Cr use the entire allowed range, i.e. Y′ will use up most of the range [64, 940] and Cb and Cr will populate most of [64, 960]. However, for HDR/WCG data using the Rec. ITU-R BT.2020 colour space and the PQ transfer function, the Cb and Cr distributions will be clustered closer to the mid-point of 512, which represents a value of Cb and Cr equal to 0. On the other hand, the Y′ component may still populate most of its allowed range. Furthermore, if the content does not exercise the entire Rec. ITU-R BT.2020 colour space, the Cb and Cr distributions will be even more tightly clustered around 0. In particular, if SDR content is instead represented using the PQ transfer function and the Rec. ITU-R BT.2020 colour space, the distribution of the Cb and Cr will be reduced substantially compared to its original Rec. ITU-R BT.709 representation. However, the luminance distribution may not be as affected.

The above observations may have a considerable impact on the encoding process. An existing encoder setting may have been able to achieve a good balance between luma and chroma for SDR content using the Rec. ITU-R BT.709 representation. However, the same encoder with the same settings will likely not achieve the same performance for the same content if the content is represented using the PQ transfer function and the Rec. ITU-R BT.2020 colour space. Given the characteristics of the new representation, this will result in a bitrate allocation shift from chroma to luma. However, if chroma is not allocated enough bits, this can give rise to visible chroma artefacts. These artefacts may, for example, appear in white areas, where miscolourations in the direction of cyan and magenta can become visible, as seen in Figure 10 a).

One way to ameliorate this is for the encoder to apply a negative chroma QP offset value. This will lower the QP value used for quantizing the chroma coefficients and has an effect similar to stretching out the Cb and Cr distributions. This effectively shifts bits back from luma to chroma, thus allowing the encoder to achieve a better balance between chroma and luma quality.

Since chroma artefacts typically become more visible at low bit rates, applying a large negative chroma QP offset at such rates can potentially help reduce these artefacts significantly. However, after a certain rate point, chroma quality may be considered as being good enough. At this point, it may no longer be necessary to shift bits from luma to chroma. Thus, at higher rates, the chroma QP offset can be set to a smaller value or even be set to zero.

A special case occurs when it is known that the content is in a restricted subset of the colour gamut defined by the Rec. ITU-R BT.2020 and Rec. ITU-R BT.2100 colour primaries. As an example, if a mastering display limited to the P3D65 colour primaries, as defined in SMPTE RP 431-2, was used to grade the content, then it is likely that the content does not also venture outside of this colour gamut. In this case, it might be known in advance that the chroma values will never go outside a certain interval that is much smaller than the allowed [64, 960] range. Under such circumstances, it may be advantageous to use a larger negative chroma QP offset compared to the QP offset that may be used for content that makes use of the entire colour gamut defined by the Rec. ITU-R BT.2020 and Rec. ITU-R BT.2100 colour primaries.

8.3.2 Example of chroma QP offset settings

In the following example, it is assumed that the colour primaries of the mastering display/capture device are known.

Based on this knowledge, a model is used to assign QP offsets for Cb and Cr based on the luma QP and a factor based on the capture and representation colour primaries. The model is expressed as Formulae (70) and (71):

$$\mathrm{QPoffsetCb} = \mathrm{Clip3}\left\{-12, 0, \mathrm{Round}\left[c_{cb} * \left(k * \mathrm{QP} + l\right)\right]\right\} \tag{70}$$

$$\mathrm{QPoffsetCr} = \mathrm{Clip3}\left\{-12, 0, \mathrm{Round}\left[c_{cr} * \left(k * \mathrm{QP} + l\right)\right]\right\} \tag{71}$$

where c_{cb} = 1 if the capture colour primaries are the same as the representation colour primaries, c_{cb} = 1.04 if the capture colour primaries are equal to the P3D65 primaries and the representation colour primaries are equal to the Rec. ITU-R BT.2020 primaries, and c_{cb} = 1.14 if the capture colour primaries are equal to the Rec. ITU-R BT.709 primaries and the representation primaries are equal to the Rec. ITU-R BT.2020 primaries.

Similarly, c_{cr} = 1 if the capture colour primaries are the same as the representation colour primaries, c_{cr} = 1.39 if the capture colour primaries are equal to the P3D65 primaries and the representation colour primaries are equal to the Rec. ITU-R BT.2020 primaries, and c_{cr} = 1.78 if the capture colour primaries are equal to the Rec. ITU-R BT.709 primaries and the representation primaries are equal to the Rec. ITU-R BT.2020 primaries.

Finally, k = −0.46 and l = 9.26.

The constants c_{cr} and c_{cb} have been calculated as the ratio of the range in the different colour representations. As an example, a maximally red colour represented using Rec. ITU-R BT.709 primaries is the colour RGB_{709} = (1,0,0). This gives a fully saturated Cr component of 0.5, i.e. $YCbCr_{709}$ = (0.213, −0.115, 0.500). Conversion to Rec. ITU-R BT.2020 primaries results in RGB_{2020} = (0.627, 0.069, 0.016) and $YCbCr_{2020}$ = (0.213, −0.104, 0.281). Likewise, the colour with the smallest Cr component is cyan that has RGB and YCbCr values of RGB_{709} = (0, 1, 1) and $YCbCr_{709}$ = (0.787, 0.115, −0.500), respectively. Conversion to Rec. ITU-R BT.2020 will result in RGB_{2020} = (0.373, 0.931, 0.984) and $YCbCr_{2020}$ = (0.787, 0.104, −0.281). The Cr component range has therefore shrunk from [−0.5, 0.5] to [−0.281, 0.281] and in this case the constant c_{cr} is calculated as (0.5−(−0.5)) ÷ (0.281−(−0.281)) = 1.78.

For HEVC, if no other chroma QP offset is desired on a picture level by other means of the encoding process, the syntax elements pps_cb_qp_offset and pps_cr_qp_offset can be set equal to QPoffsetCb and QPoffsetCr, respectively. Finer control of the chroma QP offset can be achieved at the slice level.

Similarly, for AVC, if no other chroma QP offset is desired on a picture level by other means of the encoding process, the syntax elements chroma_qp_index_offset and second_chroma_qp_index_offset can be set equal to QPoffsetCb and QPoffsetCr, respectively.

An example of the effect of this method is shown in Figure 10. Figure 10 a) shows a segment of a tone-mapped result using Formula (56) for an HDR/WCG image that was compressed without the use of either the luma QP or chroma QP offset modifications described above. On the other hand, Figure 10 b) shows the same segment compressed at the same bit rate using both of these modifications. It can be seen that the large chroma artefacts, especially on the white window shutter and on the inside of the umbrella, have been ameliorated. Furthermore, the luma, especially in the wall areas, has also been improved.

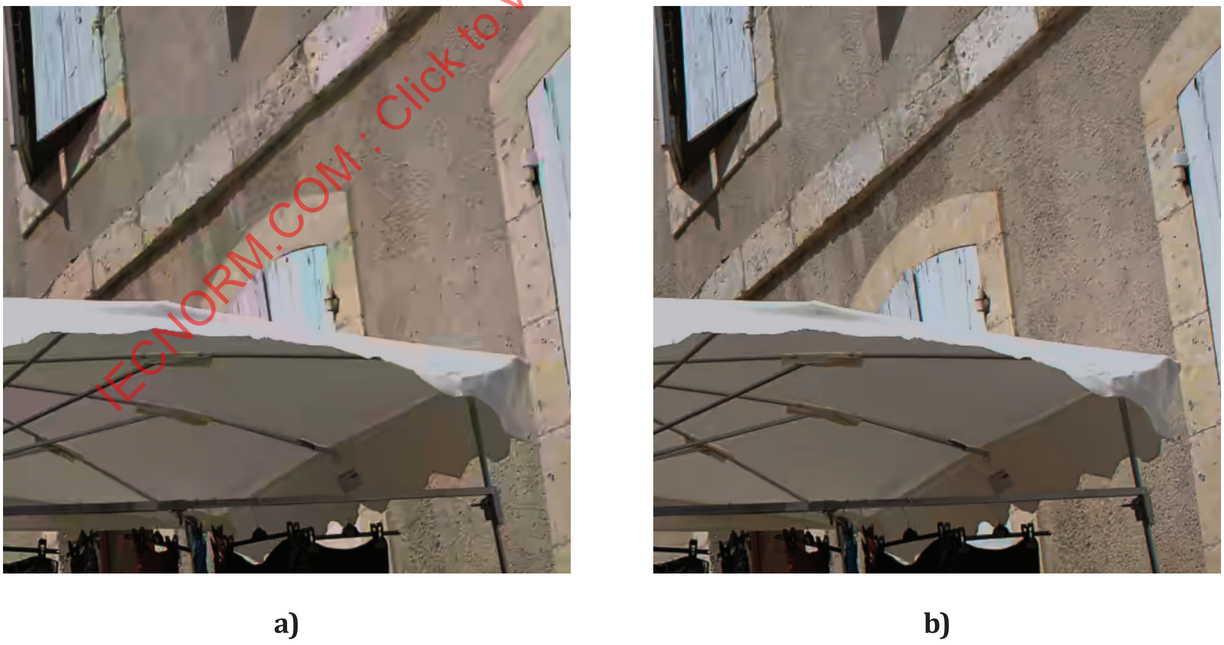


Figure 10 — Image quality without a) and with b) the presented QP modifications

8.4 Other encoding aspects

Apart from modifying the QP allocation in the encoder, it may also be desirable for an encoder manufacturer to adjust other non-normative encoding processes in their encoders, such as the motion estimation, intra and inter mode decision, trellis quantization, and rate control among others. These processes commonly consider simple distortion metrics such as mean absolute difference (MAD), or sum of squared errors (SSE), for making a variety of decisions for the decision process, and may have been tuned based on SDR content characteristics. Given, however, the earlier observations about the differences in the characteristics between SDR and HDR/WCG content, these processes may also need to be appropriately adjusted. Furthermore, other metrics may also be more appropriate for these encoding decisions. These aspects are not explored in the context of this document.

8.5 HEVC encoding

When creating the HEVC bitstream, it is recommended to set the syntax elements listed in Table 4 to the values listed in Table 4 in the sequence parameter set (SPS) of the bitstream. The syntax elements in Table 4 are conveyed in the video usability information syntax branch of the SPS defined in Rec. ITU-T H.265 | ISO/IEC 23008-2. They may also be duplicated and carried in various application-layer headers.

Table 4 — Recommended settings for HEVC encoding

Syntax element	Location	Recommended value
general_profile_space	profile_tier_level()	0
general_profile_idc	profile_tier_level()	2 (Main 10)
vui_parameters_present_flag	seq_parameter_set_rbsp( )	1
video_signal_type_present_flag	vui_parameters( )	1
video_full_range_flag	vui_parameters( )	0
colour_description_present_flag	vui_parameters( )	1
colour_primaries	vui_parameters( )	9
transfer_characteristics	vui_parameters( )	16
matrix_coeffs	vui_parameters( )	9
chroma_loc_info_present_flag	vui_parameters( )	1
chroma_sample_loc_type_top_field	vui_parameters( )	2
chroma_sample_loc_type_bottom_field	vui_parameters( )	2

For HDR/WCG content represented with the colour primaries of Rec. ITU-R BT.2020 and Rec. ITU-R BT.2100 and the PQ transfer function, the video characteristics is typically different compared to the video characteristics of SDR content represented with Rec. ITU-R BT.709 colour primaries and Rec. ITU-R BT.709 OETF (Rec. ITU-R BT.1886 EOTF) transfer function. Chroma QP adjustment, as described in 8.3 can be performed by adjusting and controlling the HEVC syntax elements pps_cb_qp_offset, slice_cb_qp_offset, pps_cr_qp_offset and slice_cr_qp_offset. Similarly, perceptual luma quantization as discussed in 8.2 could be achieved by adjusting the syntax elements cu_qp_delta_abs and cu_qp_delta_sign_flag.

NOTE See Annex A for SEI messages specified in AVC or HEVC that can be particularly relevant for HDR/WCG video.

8.6 AVC encoding

When creating the AVC bitstream, it is recommended to set the syntax elements listed in Table 5 to the values listed in Table 5 in the SPS of the bitstream. The syntax elements in Table 5 are conveyed in the video usability information syntax branch of the SPS defined in Rec. ITU-T H.264 | ISO/IEC 14496-10. They may also be duplicated and carried in various application-layer headers.