



- (51) **International Patent Classification:**
H04N 19/65 (2014.01)
- (21) **International Application Number:**
PCT/US20 16/059862
- (22) **International Filing Date:**
1 November 2016 (01.11.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/249,474 2 November 2015 (02.11.2015) US
- (71) **Applicants:** DOLBY LABORATORIES LICENSING CORPORATION [US/US]; 1275 Market Street, San Francisco, California 94103 (US). DOLBY INTERNATIONAL AB [SE/NL]; Apollo Building, 3E Herikerbergweg 1-35, 1101 CN Amsterdam Zuidoost (NL).
- (72) **Inventors:** SCHUEUER, Klaas Heinrich; c/o Dolby Germany GmbH, Rheinstrasse 45 Aufgang, 112161 Berlin (DE). BROOKS, David; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US). TESHIGAWARA, Satoshi; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US). CHEN, Tao; c/o Dolby Laboratories, Inc., 1275 Market Street, San Francisco, California 94103 (US).
- (74) **Agents:** KONSTANTINDJES, Konstantinos et al; Dolby Laboratories, Inc., Intellectual Property Group, 1275 Market Street, San Francisco, California 94103 (US).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(H))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(in))

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

(54) **Title:** LAYERED REPRESENTATION CONTAINING CRC CODES AND DELIVERY OF HIGH DYNAMIC RANGE VIDEO

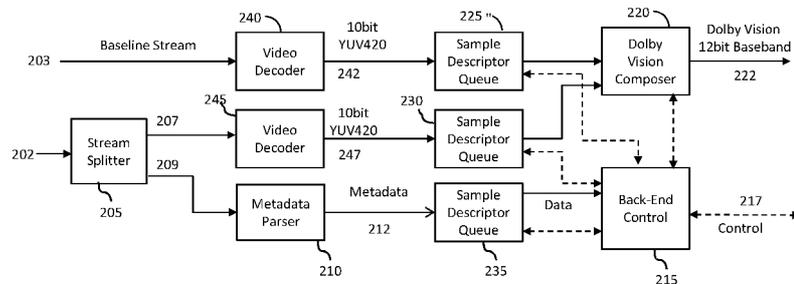


FIG. 2

(57) **Abstract:** In a method to reconstruct a high dynamic range video signal, a decoder receives a base layer standard dynamic range video signal, an enhancement layer video signal, a metadata bitstream for a reference processing unit and a CRC code related to the metadata. A decoder reconstructs a high-dynamic range video output signal based on the base layer video signal, the enhancement layer video signal, and the data syntax and metadata specified by the metadata bitstream.

WO2017/079132 A1

**LAYERED REPRESENTATION CONTAINING CRC CODES
AND DELIVERY OF HIGH DYNAMIC RANGE VIDEO**

CROSS-REFERENCE TO RELATED APPLICATIONS

5 [0001] This application claims the benefit of United States Provisional Application No. 62/249,474, filed November 2, 2015, which is incorporated by reference in its entirety.

TECHNOLOGY

[0002] The present invention relates generally to images. More particularly, an
10 embodiment of the present invention relates to the layered representation and delivery of high dynamic range video.

BACKGROUND

[0003] The development and adoption of international audio and video coding standards,
15 such as MPEG-2, MPEG-4, H.264, and more recently H.265 (also known as HEVC), has been instrumental for the rapid development, adoption, and proliferation of standard-based devices and communication standards, such as DVD players, Blu-Ray players, and the ATSC and DVB standards for digital television broadcasting.

[0004] Display technologies being developed by Dolby Laboratories, Inc., and others are
20 able to reproduce images having high dynamic range (HDR) and wide color gamut. Such displays can reproduce images that more faithfully represent real-world scenes than conventional displays of standard dynamic range (SDR).

[0005] As used herein, the term 'dynamic range' (DR) may relate to a capability of the
human visual system (HVS) to perceive a range of intensity (e.g., luminance, luma) in an
25 image, e.g., from darkest darks (blacks) to brightest whites (i.e., highlights). In this sense, DR relates to a 'scene-referred' intensity. DR may also relate to the ability of a display device to adequately or approximately render an intensity range of a particular breadth. In this sense, DR relates to a 'display-referred' intensity. Unless a particular sense is explicitly specified to have particular significance at any point in the description herein, it should be
30 inferred that the term may be used in either sense, e.g. interchangeably.

[0006] As used herein, the term high dynamic range (HDR) relates to a DR breadth that spans the some 14-15 orders of magnitude of the human visual system (HVS). In practice, the DR over which a human may simultaneously perceive an extensive breadth in intensity range may be somewhat truncated, in relation to HDR. As used herein, the terms enhanced

dynamic range (EDR) or visual dynamic range (VDR) may individually or interchangeably relate to the DR that is perceivable within a scene or image by a human visual system (HVS) that includes eye movements, allowing for some light adaptation changes across the scene or image. As used herein, EDR may relate to a DR that spans 5 to 6 orders of magnitude. Thus
5 while perhaps somewhat narrower in relation to true scene referred HDR, EDR nonetheless represents a wide DR breadth and may also be referred to as HDR.

[0007] In practice, images comprise one or more color components (e.g., luma Y and chroma Cb and Cr) wherein each color component is represented by a precision of n -bits per pixel (e.g., $n=8$). Using linear luminance coding, images where $n \leq 8$ (e.g., color 24-bit JPEG
10 images) are considered images of standard dynamic range, while images where $n > 8$ may be considered images of enhanced dynamic range. EDR and HDR images may also be stored and distributed using high-precision (e.g., 16-bit) floating-point formats, such as the OpenEXR file format developed by Industrial Light and Magic.

[0008] A reference electro-optical transfer function (EOTF) for a given display
15 characterizes the relationship between color values (e.g., luminance) of an input video signal to output screen color values (e.g., screen luminance) produced by the display. For example, ITU Rec. ITU-R BT. 1886, "Reference electro-optical transfer function for flat panel displays used in HDTV studio production," (03/2011), which is included herein by reference in its entity, defines the reference EOTF for flat panel displays based on measured characteristics
20 of the Cathode Ray Tube (CRT). Given a video stream, information about its EOTF is typically embedded in the bit stream as metadata. As used herein, the term "metadata" relates to any auxiliary information that is transmitted as part of the coded bitstream and assists a decoder to render a decoded image. Such metadata may include, but are not limited to, color space or gamut information, reference display parameters, and auxiliary signal parameters, as
25 those described herein.

[0009] Most consumer desktop displays currently support luminance of 200 to 300 cd/m^2 or nits. Most consumer HDTVs range from 300 to 500 nits with new models reaching 1000 nits (cd/m^2). Such conventional displays thus typify a lower dynamic range (LDR), also referred to as a standard dynamic range (SDR), in relation to HDR or EDR. As the
30 availability of HDR content grows due to advances in both capture equipment (e.g., cameras) and HDR displays (e.g., the PRM-4200 professional reference monitor from Dolby Laboratories), HDR content may be color graded and displayed on HDR displays that support higher dynamic ranges (e.g., from 1,000 nits to 5,000 nits or more). Such displays may be defined using alternative EOTFs that support high luminance capability (e.g., 0 to 10,000

nits). An example of such an EOTF is defined in SMPTE ST 2084:2014 "High Dynamic Range EOTF of Mastering Reference Displays," which is incorporated herein by reference in its entirety. In general, without limitation, the methods of the present disclosure relate to any dynamic range higher than SDR. As appreciated by the inventors here, improved techniques for the coding, delivery, and decoding of high-dynamic range images are desired.

5 [00010] The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section. Similarly,
10 issues identified with respect to one or more approaches should not assume to have been recognized in any prior art on the basis of this section, unless otherwise indicated.

BRIEF DESCRIPTION OF THE DRAWINGS

[00011] An embodiment of the present invention is illustrated by way of example, and not
15 in way by limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[00012] FIG. 1 depicts an example system for the coding, delivery, and display of HDR video according to an embodiment of this invention;

[00013] FIG. 2 depicts an example process for dual-layer video decoding of HDR video
20 according to an embodiment of this invention;

[00014] FIG. 3 depicts an example process for base layer processing in HDR video decoding according to an embodiment of this invention;

[00015] FIG. 4 depicts an example process for enhancement layer processing in HDR video decoding according to an embodiment of this invention; and

25 [00016] FIG. 5A and FIG. 5B depict example processes for graphics overlay on HDR video according to an embodiment of this invention.

DESCRIPTION OF EXAMPLE EMBODIMENTS

[00017] Layered representation and delivery of high dynamic range video (HDR) is
30 described herein. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are not described

in exhaustive detail, in order to avoid unnecessarily occluding, obscuring, or obfuscating the present invention.

OVERVIEW

5 [0001 8] Example embodiments described herein relate to the layered representation and delivery of high-dynamic range images. In a method for generating high-dynamic range video data with a decoder, the decoder:

receives reference processing data;

10 parses a cyclic redundancy check (CRC) parameter from the reference processing data;

extracts from the reference processing data at least an RPU data header, RPU data payload data, and an RPU-alignment-zero-bit flag;

computes a CRC value on the entire reference processing data excluding the CRC parameter;

15 compares the CRC value with the CRC parameter to detect an error;

receives a base-layer image;

receives an enhancement layer image; and

combines the base-layer image and the enhancement layer image based on the reference processing data to generate the output image.

20 TERMS AND ABBREVIATIONS

Abbreviation	Description
API	Application programming interface
BL	Base layer
EL	Enhancement layer
HDMI	High-Definition Multimedia Interface
LSB	Least-significant bit
LUT	Lookup table
MMR	Multivariate multiple regression
MSB	Most-significant bit
NAL	Network Abstraction Layer
NLQ	Nonlinear quantization
CRC	Cyclic redundancy check

TECHNOLOGY OVERVIEW

[00019] Dolby Vision™ for consumer applications is an end-to-end technology suite that enables the creation and distribution of content mastered with a high dynamic range and wide color gamut. Dolby Vision display management matches the capability of a given television
 5 by using a series of algorithms to map the signal to any Dolby Vision consumer television. This creates an optimal and seamless video experience for the viewer. As used herein, the term "Dolby Vision video" denotes EDR or HDR video.

[00020] FIG. 1 depicts an example system for the coding, delivery, and display of HDR video according to an embodiment of this invention. As depicted in FIG. 1, in an

10 embodiment of a system for Dolby Vision consumer applications, a Dolby Vision video signal (107) is sent from a Dolby Vision source (105) (e.g., a Dolby Vision capable Ultra-HD Blu-ray player and the like) to a Dolby Vision sink (110) (e.g., a television, a monitor, and the like) via a digital interface (e.g., HDMI).

[00021] FIG. 2 depicts an example processing pipeline for the decoding and playback of
 15 Dolby Vision content on an HDR-capable media player according to an embodiment. The video elementary stream (EL) splitter block (205) strips apart the enhancement video data (207) and the metadata information (209) contained in the Dolby Vision EL video stream (202). Given a Dolby Vision input stream (202) (see [9]), the output of (205) consists of an EL video elementary stream (207), and a Dolby Vision metadata bitstream (209).

20 [00022] The metadata parser (210) decodes Dolby Vision metadata from the bitstream. The input metadata bitstream contains:

- Dolby Vision specific metadata (i.e. composer (220) prediction coefficients)
- Static metadata as defined in SMPTE 2086 in Ref. [7].
- Dynamic scene-based metadata (e.g., as may be defined in WD SMPTE ST 2094).

25 [00023] The output (212) contains a composing metadata structure and a display management metadata structure which are described in more detail later on.

[00024] The back-end control block (215) is responsible for configuration and control of all composer and display management processing blocks.

The inputs are:

- 30
- BL video decoder output sample descriptor queue (225)
 - EL video decoder output sample descriptor queue (230)
 - Dolby Vision metadata parser output sample descriptor queue (235)
 - Dolby Vision application control interface parameters

The outputs are:

- The values of the composer hardware registers
- The values of the display management hardware registers and look-up tables
- The controlling signal (217) of the Dolby Vision HDMI hardware block.

5 [00025] The Dolby Vision composer (220) accepts synchronized decoded BL (BDMV HDR) and EL video frames, metadata information, and back-end control block data, and uses the metadata to reconstruct Dolby Vision video frames.

The inputs are:

- A BL 10-bit YUV420 signal (242) from a BL video decoder (240) buffer
- 10 • An EL 10-bit YUV420 signal (247) from an EL video decoder (245) buffer
- A metadata structure containing composing coefficients, parameters, and others

The output is a reconstructed Dolby Vision signal YUV420 buffer (222). More detailed information about the Dolby Vision composer is provided later on.

15 **Dolby Vision Enhancement Layer Definitions**

[00026] A Dolby Vision EL video bitstream (202) consists of one MPEG HEVC video stream with embedded Dolby Vision metadata. The stream is a byte stream of NAL units as defined in Annex B of Ref. [3].

[00027] The input of a video elementary stream splitter is a Dolby Vision EL video 20 bitstream (202). The video elementary stream splitter application (205) splits the combined EL video and metadata elementary stream input into two output streams:

- The EL video elementary stream (207)
- A metadata bitstream (209)

[00028] To identify each individual stream in a video elementary stream, the splitter 25 searches the Dolby Vision video elementary byte stream for the 3-byte NAL start code 0x000001 or the 4-byte NAL start code 0x00000001. The first two bytes immediately following the start code identify the layer to which the current NAL unit belongs, as listed in Table 1.

Table 1: Dolby Vision Enhancement Layer NAL unit type

Layer Description	First Two Bytes After Start Code
EL video NAL unit	!0x7C01
Dolby Vision metadata NAL unit	0x7C01

30 Note: The BL video elementary stream NAL unit bypasses the video elementary stream splitter.

[00029] An EL video elementary stream NAL unit can be passed to the EL video decoder without any change.

[00030] For the Dolby Vision metadata bitstream NAL unit, the first two bytes (0x7C01) immediately following the start code must be removed before they are sent to the Dolby

5 Vision metadata parser application. A complete definition of the Dolby Vision metadata bitstream syntax and semantics can be found in Annex A.

[00031] Dolby Vision metadata, as defined in Annex A and conforming to Ref. [9], are subject to the requirements listed in Table 2.

Table 2: Metadata Setup

Metadata	Syntax
vdr_rpu_profile	1
vdr_rpu_level	0
num_y_partitions_minus1	0
num_x_partitions_minus1	0
num_pivots_minus2 [0]	For polynomial prediction: Must be less or equal to 7 For MMR prediction: Must be less or equal to 3 For power prediction: Must be equal to 0
num_pivots_minus2 [1] num_pivots_minus2 [2]	For polynomial prediction: Must be less or equal to 3 For MMR or power prediction: Must be equal to 0
nlq_num_pivots_minus2	0
prev_vdr_rpu_id	Must be no greater than 0 if this syntax element is present
BL_bit_depth_minus8	2
EL_bit_depth_minus8	2
Vdr_bit_depth_minus8	An integer no greater than 6
spatial_resampling_filter_flag	0
mapping_color_space	0 (YCbCr)
mapping_idc [0]	Set to 0 when coefficient_data_type is equal to 0
mapping_idc [1] mapping_idc [2]	Must be 0 or 1 when coefficient_data_type is equal to 0

coefficient_log2_denorm	An integer no greater than 23
nlq_method_idc	Set to 0 when coefficient_data_type is equal to 0
BL_chroma_format_idc	0
EL_chroma_format_idc	0
mapping_chroma_format_idc	0

Dolby Vision Composer Definition

[00032] The Dolby Vision composer (220) accepts decoded BL and EL video frames, metadata information, and back-end control block data, and uses the metadata to reconstruct

5 Dolby Vision video frames.

Composer IO Definitions

Composer Inputs

[00033] Decoded Base Layer Video Frame

- 10 · The decoded BL video input conforms to the following values:
 - The BL video signal is called S_{cmp} , where $0 \leq cmp \leq 2$ represents the three color channels.
 - The video format is YCbCr 4:2:0.
 - The video resolution is 3840x2160 or 1920x1080.
- 15 · The bit depth is equal to 10bit.
- The EOTF is according to SMPTE ST 2084 in Ref. [6].

[00034] Decoded Enhancement Layer Video Frame

The decoded EL video input conforms to the following values:

- 20 · The enhancement-layer video signal is called R_{cmp} , where $0 \leq cmp \leq 2$ represents the three color channels.
- The video format is YCbCr 4:2:0.
- The video resolution is 1920x1080
- The bit depth is equal to 10bit.

[00035] Composer Metadata

The coefficients and parameters used in the metadata for the Dolby Vision composer are as follows.

5 **Coefficient_log_2_denom**

[00036] This parameter specifies the number of fractional bits for Dolby Vision decoding coefficients if `coefficient_data_type = 0`.

BL_bit_depth

10 [00037] This parameter specifies the bit depth of the BL video signal. It is derived by $BL_bit_depth = EL_bit_depth_minus_8 + 8$, where `BL_bit_depth_minus_8` is as defined in Ref. [3]. For Ultra-HD Blu-ray `BL_bit_depth` must be equal to 10.

EL_bit_depth

15 [00038] This parameter specifies the bit depth of the EL video signal. It is derived by $EL_bit_depth = EL_bit_depth_minus_8 + 8$, where `EL_bit_depth_minus_8` is as defined in Ref. [3]. For Ultra-HD Blu-ray `EL_bit_depth` must be equal to 10.

vdr_bit_depth

20 [00039] This parameter specifies the bit depth of the reconstructed Dolby Vision video signal. It is derived by $vdr_bit_depth = vdr_bit_depth_minus_8 + 8$, where `vdr_bit_depth_minus_8` is as defined in Ref. [3].

mapping_idc

25 [00040] This parameter specifies the prediction method for each color channel of the BL. For the luma channel, only 0 is allowed. For the chroma channels, `mapping_idc` must be in the range of 0 to 1.

spatial_resampling_filter_lag

30 [00041] For Ultra-HD Blu-ray this flag must be equal to 0.

spatial_resampling_explicit_filter_flag

[00042] For Ultra-HD Blu-ray this flag must be equal to 0.

Base layer Prediction Coefficients

[00043] For the Dolby Vision for Ultra-HD Blu-ray decoder, the prediction method is a piece-wise polynomial mapping up to the second order. The maximum number of channel segments allowed is eight.

5 [00044] The polynomial function coefficients are $\{jpoly_coef_{c,j,i}\}$, where $0 \leq c \leq 2$ (c is the color channel index), $0 \leq j \leq 7$ (j is the segment index) and $0 \leq i \leq 2$ (i is the coefficient order):

- $jpoly_coef_{c,j,i}$ are the coefficients for each color channel c , each segment j and each order i .

10
$$jpoly_coef_{c,j,i} = (poly_coef_int_{[0][0][c][j][i]} \ll coefficient_log2_denom) \mid poly_coef_{[0][0][c][j][i]}$$

[00045] Each polynomial coefficient is a real number with a bit depth of $coefficient_log2_denom+7$, as specified in the following list:

- 15
- $5/[coefficient_log2_denom+6]$: Sign bit
 - $B/[coefficient_log2_denom+5 : coefficient_log2_denom]$: Integer part
 - $Bz/[coefficient_log2_denom-1 : 0]$: Fractional part

[00046] Alternatively, the chroma channels can be predicted using a multiple multivariate regression function up to third order. The multi-variate multiple regression coefficients are $\{m_{c,i}\}$, where $1 \leq c \leq 2$ (c is the color channel index) and $0 \leq i \leq 21$ (us the coefficient number):

- m_f is the multivariate regression constant.

25
$$m_f = (mmr_constant_int_{[0][0][j][0]} \ll coefficient_log2_denom) \mid mmr_constant_{[0][0][j][0]}$$

- $m_{c,i}$ are the multivariate regression coefficients.

"
$$m_{c,i} = (mmr_coef_int_{[0][0][j][0][m][n]} \ll coefficient_log2_denom) \mid mmr_coef_{[0][0][j][0][M][n]}$$

30
$$i = 7*(m-1) + n \text{ with } 1 \leq m \leq 3 \text{ and } 1 \leq n \leq 7.$$

[00047] Each polynomial coefficient is a real number with a bit depth of $\text{coefficient_log2_denom}+17$, as specified in the following list:

- $\text{Bit}[\text{coefficient_log2_denom}+16]$: Sign bit
- $\text{5z}[\text{coefficient_log2_denom}+15 : \text{coefficient_log2_denom}]$: Integer part
- 5 • $\text{Bz}[\text{coefficient_log2_denom}-1 : 0]$: Fractional part

Note: `pivot_value`, `poly_coef_int`, `poly_coef`, `mmr_constant_int`, `mmr_constant`, `mmr_coef_int`, and `mmr_coef` are decoding metadata syntax elements, as defined in Ref. [11].

10

Enhancement Layer Inverse Quantization Coefficients

[00048] For the Dolby Vision for Ultra-HD Blu-ray decoder, the inverse quantization method is linear with dead zone. The maximum number of segments allowed in is one.

15 [00049] The linear-with-dead-zone function coefficients of the luma and chroma channels are

- $\{S[j], M[j], T[j], Rmax[j] \mid J, \text{ where } 0 \leq j \leq 2 \text{ (} j \text{ is the color channel index) and } i = 0 \text{ (} i \text{ is the segment index):}$
- $S[j]$ is the slope coefficient of the linear-with-dead-zone function.
- 20 • $S[j] = ((\text{linear_deadzone_slope_int}[\text{0}][\text{0}][i][j] \ll \text{coefficient_log2_denom}) \mid \text{linear_deadzone_slope}[\text{0}][\text{0}][i][j]).$
- $M[j]$ is the offset coefficient of the linear-with-dead-zone function.
- $M[j] = \text{nlq_offset}[\text{0}][\text{0}][i][j].$
- $T[j]$ is the threshold coefficient of the linear-with-dead-zone function.
- 25 • $T[j] = ((\text{linear_deadzone_threshold_int}[\text{0}][\text{0}][i][j] \ll \text{coefficient_log2_denom}) \mid \text{linear_deadzone_threshold}[\text{0}][\text{0}][i][j]).$
- $Rmax[j]$ is the maximum enhancement layer value coefficient of the linear-with-dead-zone function.
- 30 • $Rmax[j] = ((\text{vdr_in_max_int}[\text{0}][\text{0}][i][j] \ll \text{coefficient_log2_denom}) \mid \text{vdr_in_max}[\text{0}][\text{0}][i][j]).$

- M_{jj} is an integer with a value in the range of $[0, 2^{EL_bit_depth} - 1]$ and a bit depth of EL_bit_depth .

[00050] Other inverse quantization coefficients are real numbers with bit depths of
5 $coefficient_log2_denom+2$, as specified in the following list:

- $Z_{[coefficient_log2_denom+1]}$: Sign bit
- $Z_{[coefficient_log2_denom]}$: Integer part
- $Z_{[coefficient_log2_denom-1 : 0]}$: Fractional part

10 [00051] Note: $linear_deadzone_slope_int$, $linear_deadzone_slope$,
 nlq_offset , $linear_deadzone_threshold_int$,
 $linear_deadzone_threshold$, $vdr_in_max_int$, and vdr_in_max are decoding
metadata syntax elements, as defined in Ref. [11].

15 Composer Output

[00052] Composer output conforms to the following values:

- The output is a Dolby Vision signal called V_{cmp} , where $0 \leq cmp \leq 2$ represents the three color channels.
- The video format is YCbCr 4:2:0.
- The resolution is 3840x2160 or 1920x1080.
- Bit depth is equal to vdr_bit_depth .

Composer Decoding Process

25 Base-Layer Decoding Process

[00053] FIG. 3 depicts an example decoding process of the base layer (BL) according to an embodiment. The decoded BL samples (242) are given by the decoding process (240) of the baseline video stream (203). The decoded BL signal (242) is mapped to the dynamic range of the EL signal domain using pre-selected prediction coefficients signaled in the
30 Dolby Vision metadata (212). Predicted BL signal's decimal values are in the range of $[0,1)$. The predicted BL signal (312) will be added to the inverse quantized EL signal (304) to reconstruct the HDR Dolby Vision signal (222).

[00054] Different prediction methods can be applied. For the Dolby Vision for Ultra-HD Blu-ray decoder, the prediction method for the luma channel is a piece-wise polynomial up to the second order. For the chroma channels, the prediction method is a piece-wise polynomial up to the second order or multivariate multiple regression up to the third order.

5

Base Layer Inverse Mapping

[00055] To generate the prediction from BL to EL, the mapping from BL signal range to EL signal range is done with a prediction curve consisting of multiple segments. The pivots (or boundaries) of these segments are derived as follows.

10 [00056] The value of `num_pivots_minus2` indicates how many pivot points exist for the component `cmp` of the current BL video frame. The value of each pivot point is stored in a sorted array `pivot_value [cmp] []`, where `pivot_value [cmp] [0]` is a low end-point value and `pivot_value [cmp] [num_pivots_minus2 [cmp]+1]` is a high end-point value.

[00057] To find the pivot values for a given sample `Scmp`, the `pivot_value` array is
 15 traversed until the sample value lies between two pivot values; this determines the segment. The sample is capped at the low- and high-end pivot values.

MAPPING_POLYNOMIAL Method

[00058] This method is applied to the luma component or when `mapping_idc` is equal to
 20 zero. The resulting prediction from BL is constructed in a piecewise manner. Inputs are the BL signal of the current component and the pivot array introduced in the previous section. The variables `poly_order_minus1` and `linear_interp_f_lag` are held in the Dolby Vision metadata.

[00059] When `linear_interp_f_lag` is 1, linear interpolation is performed between the
 25 two pivot values of the current segment `j` where the BL pixel value `S` is located. Otherwise, full polynomial prediction for an `N`-order polynomial is calculated according to

$$R = \sum_{i=0}^N fpoly_coef_{j,i} * (S i 2^{BL_bit_depth}) \quad (1)$$

where `fpoly_coef` are prediction coefficients signaled in the Dolby Vision metadata. The calculation can be carried out in fixed point or 32-bit floating point.

30

MAPPING_MMR Method

[00060] This mapping applies to chroma only. In a first step, the BL luma component must be downsampled by using the downsampling filters specified in the following equations before performing the multivariate multiple regression prediction process.

- 5 • Horizontal downsampling filter:

$$\bullet \quad r[n] = \text{Clip3}(0, (y[2n-1] + 2 * y[2n] + y[2n+1] + 2) \gg 2, 1023); \quad (2)$$

- Vertical downsampling filter:

$$\bullet \quad r[n] = \text{Clip3}(0, (y[2n] + y[2n+1] + 1) \gg 1, 1023); \quad (3)$$

10 [00061] Horizontal downsampling is applied first, followed by vertical downsampling. If the pixel used in equations 2 and 3 is out of the picture boundary, it is set to the value of the pixel at the picture boundary. That is, image edges are handled by pixel replication. Values of s_{cmp} are clamped to highest and lowest pivot value respectively. Then, a multiple multivariate regression is performed:

15

$$R = m_0 + \sum_{i=0}^N \left(\begin{matrix} m_1[i] \cdot S_0^i + m_2[i] \cdot S_1^i + m_3[i] \cdot S_2^i + m_4[i] \cdot S_0^i \cdot S_1^i \\ + m_5[i] \cdot S_0^i \cdot S_2^i + m_6[i] \cdot S_1^i \cdot S_2^i + m_7[i] \cdot S_0^i \cdot S_1^i \cdot S_2^i \end{matrix} \right), \quad (4)$$

where the coefficients m are transmitted in the Dolby Vision metadata and depend on the current segment and color channel. The order N is specified in the Dolby Vision metadata as well. The calculation can be carried out in fixed point or 32bit floating point.

20

Enhancement Layer Decoding Process

[00062] For each BL video frame, there is a corresponding EL video frame that is reconstructed by the EL video decoder. The Dolby Vision decoder must guarantee that the decoded BL video frame and the decoded EL video frame are both in display order and frame aligned to each other. In an EL video frame, the sample values carry three quantized components of an EL signal in YCbCr color space. Once an EL video frame is aligned with its corresponding BL video frame, it must:

25

1. Upscale the EL signal to the same resolution of BL, if the EL resolution is lower than the BL resolution
- 30

2. Perform inverse quantization to translate the EL signal to the Dolby Vision signal domain.

The inverse quantized EL signal can now be added to the predicted BL signal. An example process for the EL decoding processing is shown FIG. 4.

5 Enhancement Layer Spatial Resampling

[00063] If the `el_spatial_resampling_filter_flag` is equal to 1, the Dolby Vision decoder must upsample the reconstructed signal from the EL video decoder in both the horizontal and vertical directions. If the `el_spatial_resampling_filter_flag` is equal to 0, the EL spatial resampling is bypassed.

10 [00064] The input is the reconstructed video frame from the EL video decoder with bit depth equal to `EL_bit_depth` and without any changes to the chroma format. The output of this process is a spatially upsampled, reconstructed EL signal with the same bit depth and chroma format as the input signal.

[00065] Vertical upsampling is applied first, followed by horizontal upsampling.

15

Enhancement Layer Nonlinear Dequantization

[00066] The input is a reconstructed EL signal with a bit depth equal to 10 (`EL_bit_depth = 10`). Based on the value of `coefficient_data_type`, the final output is either a fixed-point integer or a 32-bit single precision floating point value. The

20 NLdQ method to be used is determined as follows:

- If `nlq_method_idc` is equal to 0, `NLQ_LINEAR_DZ` applies.
- If `nlq_method_idc` is equal to 1, `NLQ_MU_LAW` applies.

`NLQ_LINEAR_DZ`

25 [00067] The linear with dead zone dequantization employs piece-wise linear functions to calculate the dequantized pel values. Each of the linear functions is specified by slope S , threshold T and offset M , with a deadzone defined at the offset. This process is operated individually on each component of each pel R and is given by:

$$dR = \begin{cases} 0, & R = M \\ S * ((R - M) - 0.5 * \text{sign}(R - M)) + T * \text{sign}(R - M) & R \neq M \end{cases}, \quad (5)$$

30 with

$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

The dequantized pixel value is given by dR .

[00068] The Dolby Vision signal reconstruction takes in the predicted (or mapped) BL signal and the inverse quantized EL signal, and the reconstructed pel value is a sum of the
 5 two inputs. Based on the value of **coefficient_data_type**, the final output is either a fixed-point integer or a 32-bit single precision floating point value. Appropriate offsetting, scaling and clipping are specified accordingly for each case to ensure a valid pel value.

Definition of Graphic Overlay Adaptation

10 [00069] Graphics processing for Dolby Vision is similar to the Graphics processing of the BDMV HDR case, but it is handled inside of the Dolby Vision Image Processing. Graphics processing for Dolby Vision adds an additional processing step of inverse display management to the graphics handling. This additional step maps the graphics signal to the Dolby Vision signal space.

15 [00070] FIG. 5A depicts an example of the video overlay process in a first embodiment (e.g., HDMV mode). It follows the conceptual model of presentation planes as described in Ref. [9]. In addition, inverse display management (inverse DM) is applied to the Primary and Secondary graphics planes (e.g., for presentation and/or interactive graphics) prior to the mixing process so that the dynamic ranges of the input primary and secondary graphics
 20 matches the dynamic range of the signal at the output of the Dolby Vision composer. As used herein, the term 'inverse display management' denotes a process to increase the dynamic range of a signal. An example of a dynamic range mapping function may be found in Ref. [12].

[00071] FIG. 5B depicts an example of the video overlay process in a second embodiment
 25 (e.g., BD-J mode). In addition to the current presentation plane model of BD-J mode, as described in [9], an inverse display management (Inverse DM) shall be applied to the existing graphic planes as shown in the figure.

Definition of the Dolby Vision HDMI Interface

30 [00072] Dolby Vision content is transmitted to the rendering device via HDMI. The HDMI device must support HD@60p, UHD@30p and UHD@60p. The Dolby Vision pixel format is 12-bit YCbCr 4:2:2.

HDMI Sink

[00073] A Dolby Vision display (i.e. the HDMI sink) indicates its support for Dolby Vision content in its HDMI E-EDID by setting a flag in the Dolby Vision Vendor Specific Video Data Block (VSVDB) conforming to CEA Extension version 3 as defined in section 5 7.5.7 of [8]. Further, it shall also indicate support for YCbCr 4:2:2 in its E-EDID.

[00074] The Dolby Vision VSVDB also contains the 12bit primary RGB chromaticities, white points and luminance characteristics of the display. The Dolby Vision display verifies that it is connected to a Dolby Vision capable source device (i.e. the Ultra HD Blu-ray player) by checking the validity of the metadata CRC (Cyclic Redundancy Check) as defined later.

10 HDMI Source

[00075] The Dolby Vision capable Ultra HD Blu-ray player (i.e. HDMI Source) transmits the Dolby Vision baseband video signal incorporating the embedded metadata stream. The embedded metadata and its CRC protection are used to indicate that a valid Dolby Vision baseband signal is present. HDMI source and sink negotiate the resulting Dolby Vision 15 format - if any - before transmission according to the HDMI protocol.

Dolby Vision Metadata

[00076] To accommodate the display properties, Dolby Vision Display Management (DM) metadata is sent to the Dolby Vision capable display on a per-frame basis. Nonetheless, the DM metadata can be shared among multiple frames, leading to a scene based DM metadata 20 package in the best case. Up to 16 DM metadata packages shall be buffered for reference.

[00077] A DM metadata structure holds information about its identifier, the previous frame's metadata to be replaced, scene cut information, and about color conversion matrices from YCbCr to display-RGB. The metadata is packetized into one or more DM metadata 25 transmission packets of 1024 bits each. These consist of packet type identifiers, the DM metadata, and a 32bit-CRC. There are four kinds of EDR metadata transmission packet types:

- 0b00: a single packet carries an entire Dolby Vision DM metadata structure
- 0b01: the first packet if multiple packets carry the Dolby Vision DM metadata structure
- 0b10: intermediate packet(s) if multiple packets carry the Dolby Vision DM 30 metadata structure

- Obi 1: the last packet if multiple packets carry the Dolby Vision DM metadata structure

Dolby Vision Metadata transmission

- [00078] For transmission, the Dolby Vision DM metadata package is put bit by bit into the least significant bit of each 12 bit chroma channel in the YCbCr 4:2:2 signal. When Dolby Vision DM metadata transmission packet is represented in a sequence of bits, the first byte goes first with the most significant bit of that byte first. For example, for a byte 0x01, the seven zeros go first followed by the one 1. The i -th bit of the Dolby Vision DM metadata transmission packet is at bit[m] ($0 \leq m \leq 7$) of byte[n] ($0 \leq n \leq 127$) where $i = (\llcorner * 8 + (7 - m))$.
- 10 [00079] Assuming that the Dolby Vision video frame has a resolution of $W \times H$ and a pixel sample can be represented using coordinates (y, x) , where $0 \leq y \leq H$ and $0 \leq x \leq W$. For each pixel, there are one luma sample and one chrominance sample, which is the Cb component for an even pixel or the Cr component for an odd pixel. In raster scan order the i -th pixel is at (y, x) and $i = (W \times y + x)$.
- 15 [00080] The z -th bit of a Dolby Vision DM metadata transmission packet shall be placed onto the least significant bit of chrominance sample of the i -th pixel (y, x) in raster scan order in the Dolby Vision video frame.
- [00081] To improve the imperceptibility of embedding the Dolby Vision DM metadata into the YCbCr signal, bit scrambling is applied to the metadata before embedding.
- 20 [00082] For the sake of robustness, each Dolby Vision DM metadata transmission packet shall be sent three times consecutively. The Dolby Vision sink shall check the CRC of the transmission packets and discard subsequent copies as soon as one CRC is valid. If the packet type is not 0b00, i.e. if the Dolby Vision DM metadata is split among multiple transmission packets, all these packets share the same metadata identifier and all shall be included within
- 25 one single video frame. The first transmission packet shall have packet type 0b01, the last one 0b11, the intermediate ones 0b10 in this case.
- [00083] An example of the metadata bits embedded into the samples for the first four luminance and associated chrominance samples is shown in Table 3 with 12 bits on the input chain.

30

Table 3: Layout of Dolby Vision DM metadata embedded in 12-bit EDR YCbCr 4:2:2

video data

D11	Cb[0][11]	Y[0][11]	Cr[0][11]	Y[1][11]	Cb[1][11]	Y[2][11]	Cr[1][11]	Y[3][11]
D10	Cb[0][10]	Y[0][10]	Cr[0][10]	Y[1][10]	Cb[1][10]	Y[2][10]	Cr[1][10]	Y[3][10]
D9	Cb[0][9]	Y[0][9]	Cr[0][9]	Y[1][9]	Cb[1][9]	Y[2][9]	Cr[1][9]	Y[3][9]
D8	Cb[0][8]	Y[0][8]	Cr[0][8]	Y[1][8]	Cb[1][8]	Y[2][8]	Cr[1][8]	Y[3][8]
D7	Cb[0][7]	Y[0][7]	Cr[0][7]	Y[1][7]	Cb[1][7]	Y[2][7]	Cr[1][7]	Y[3][7]
D6	Cb[0][6]	Y[0][6]	Cr[0][6]	Y[1][6]	Cb[1][6]	Y[2][6]	Cr[1][6]	Y[3][6]
D5	Cb[0][5]	Y[0][5]	Cr[0][5]	Y[1][5]	Cb[1][5]	Y[2][5]	Cr[1][5]	Y[3][5]
D4	Cb[0][4]	Y[0][4]	Cr[0][4]	Y[1][4]	Cb[1][4]	Y[2][4]	Cr[1][4]	Y[3][4]
D3	Cb[0][3]	Y[0][3]	Cr[0][3]	Y[1][3]	Cb[1][3]	Y[2][3]	Cr[1][3]	Y[3][3]
D2	Cb[0][2]	Y[0][2]	Cr[0][2]	Y[1][2]	Cb[1][2]	Y[2][2]	Cr[1][2]	Y[3][2]
D1	Cb[0][1]	Y[0][1]	Cr[0][1]	Y[1][1]	Cb[1][1]	Y[2][1]	Cr[1][1]	Y[3][1]
D0	Metadata	Y[0][0]	Metadata	Y[1][0]	Metadata	Y[2][0]	Metadata	Y[3][0]

Annex A

5 A 1. Dolby Vision Metadata Bitstream Syntax

[00084] This section provides information about the Dolby Vision metadata syntax elements for Ultra-HD Blu-ray using the H.265/ HEVC system in Ref. [3] as the reference system. The enhanced reference processing unit data syntax as specified in this section conveys the parameters necessary to predict the Dolby Vision signal efficiently from the reconstructed BL video signal, to de-quantize the reconstructed EL video signal, and to form the reconstructed Dolby Vision video signal.

A 1.1 Reference Processing Unit Data Raw Byte Sequence Payload Syntax

15 **Table A-1 - 1 Reference Processing Unit Data Raw Byte Sequence Payload Syntax**

	Descriptor
rpu_data_rbsp () {	
rpu_data ()	
rbsp_trailing_bits ()	
}	

A 1.2 Reference Processing Unit Data Syntax

Table A-1- 2 Reference Processing Unit Data Syntax

Code	Descriptor
<code>rpu_data() {</code>	
<code> rpu_data_header ()</code>	
<code> if (rpu_type == 2) {</code>	
<code> if (!use_prev_vdr_rpu_flag)</code>	
<code> vdr_rpu_data_payload ()</code>	
<code> if (vdr_dm_metadata_present_flag)</code>	
<code> vdr_dm_data_payload ()</code>	
<code> while (!byte_aligned ())</code>	
<code> rpu_alignment_zero_bit</code>	f(1)
<code> rpu_data_crc32</code>	u(32)
<code> } // EDR RPU data</code>	
<code> }</code>	

5

A 1.3 Reference Processing Unit Data Header Syntax

Table A-1- 3 Reference Processing Unit Data Header Syntax

Code	Descriptor
<code>rpu_data_header () {</code>	
<code> rpu_type</code>	u(6)
<code> rpu_format</code>	u(11)
<code> if (rpu_type == 2) {</code>	
<code> vdr_rpu_profile</code>	u(4)
<code> vdr_rpu_level</code>	u(4)
<code> vdr_seg_info_present_flag</code>	u(1)
<code> if (vdr_seg_info_present_flag) {</code>	
<code> chroma_resampling_explcit_filter_flag</code>	u(1)
<code> coefficient_data_type</code>	u(2)
<code> if (coefficient_data_type == 0) {</code>	
<code> coefficient_log2_denom</code>	ue(v)
<code> }</code>	
<code> }</code>	

<code>vdr_rpu_normalized_idc</code>	<code>u(2)</code>
<code>BL_video_full_range_flag</code>	<code>u(1)</code>
<code>if (rpu_fcrmat & 0x700 == 0) {</code>	
<code>BL_bit_depth_minus8</code>	<code>ue(v)</code>
<code>EL_bit_depth_minus8</code>	<code>ue(v)</code>
<code>vdr_bit_depth_minus8</code>	<code>ue(v)</code>
<code>spatial_resampling_filter_flag</code>	<code>u(1)</code>
<code>reserved_zero_3bits</code>	<code>u(3)</code>
<code>el_spatial_resampling_filter_flag</code>	<code>u(1)</code>
<code>disable_residual_flag</code>	<code>u(1)</code>
<code>} // end of sequence header</code>	
<code>} // end of EDR RPU sequence header</code>	
<code>vdr_dm_metadata_present_flag</code>	<code>u(1)</code>
<code>use_prev_vdr_rpu_flag</code>	<code>u(1)</code>
<code>if (use_prev_vdr_rpu_flag) {</code>	
<code>prev_vdr_rpu_id</code>	<code>ue(v)</code>
<code>}</code>	
<code>else {</code>	
<code>vdr_rpu_id</code>	<code>ue(v)</code>
<code>mapping_color_space</code>	<code>ue(v)</code>
<code>mapping_chroma_format_idc</code>	<code>ue(v)</code>
<code>for(cmp = 0; cmp < 3; cmp++) {</code>	
<code>num_pivots_minus2 [cmp]</code>	<code>ue(v)</code>
<code>for (pivot_idx = 0; pivot_idx < num_pivots_minus2 [cmp] + 2; pivot_idx ++)</code>	
<code>pred_pivot_value [cmp][pivot_idx]</code>	<code>u(v)</code>
<code>} // end of pivot points for BL three components</code>	
<code>if (rpu_format & 0x700 == 0 && !disable_residual_flag) {</code>	
<code>nlq_method_idc</code>	<code>u(3)</code>
<code>nlq_num_pivots_minus2 = 0</code>	
<code>} // end of vl.x architecture EL specific</code>	

<code>num_x_partitions_minus1</code>	ue(v)
<code>num_y_partitions_minus1</code>	ue(v)
<code>} // end of EDR RPU frame header</code>	
<code>} // end of EDR RPU header</code>	
<code>}</code>	

A 1.4 Reference Processing Unit Data Payload Syntax

Table A-1- 4 Reference Processing Unit Data Payload Syntax

	Descriptor
<code>vdr_rpu_data_payload () {</code>	
<code> rpu_data_mapping (0, 0)</code>	
<code> rpu_data_nlg (0, 0)</code>	
<code>}</code>	

5

A 1.5 Reference Processing Unit Data Mapping Parameters Syntax

Table A-1- 5 Reference Processing Unit Data Mapping Syntax

	Descriptor
<code>rpu_data_mapping (y, x) {</code>	
<code> num_cmps = 3</code>	
<code> for (cmp = 0; cmp < num_cmps; cmp++) {</code>	
<code> for (pivot_idx = 0; pivot_idx < num_pivots_minus2 [</code> <code> cmp] + 1; pivot_idx+-) {</code>	
<code> mapping_idc [y][x][cmp][pivot_idx]</code>	ue(v)
<code> if (num_mapping_param_predictors [y][x][cmp][</code> <code> pivot_idx] > 0)</code>	
<code> mapping_param_pred_f lag [y][x][cmp][</code> <code> pivot_idx]</code>	U(1)
<code> else</code>	
<code> mapping_param_pred_f lag [y][x][cmp][</code> <code> pivot_idx] = 0</code>	
<code> if (mapping_param_pred_f lag [y][x][cmp][</code> <code> pivot_idx] == 0)</code>	
<code> rpu_data_mapp_ng_param (y, x, cmp, pivot_idx)</code>	

else {	
if (num_mapping_param_predictors [y][x][cmp][pivot_idx] > 1)	
diff_pred_part_idx_mapping_minus1 [y][x][cmp][pivot_idx]	ue(v)
}	
} // pivot_idx	
} // cmp	
}	

Table A-1- 6 Reference Processing Unit Data Mapping Parameters Syntax

rpu_data_mapping_param (y, x, cmp, pivot_idx) {	Descriptor
if (mapping_idc [y][x][cmp][pivot_idx] == MAPPING_POLYNOMIAL) {	
poly_order_minus1 [y][x][cmp][pivot_idx]	ue(v)
if (poly_order_minus1 [y][x][cmp][pivot_idx] == 0)	
linear_interp_f_lag [y][x][cmp][pivot_idx]	u(l)
if (poly_order_minus1 [y][x][cmp][pivot_idx] == 0 && linear_interp_f_lag [y][x][cmp][pivot_idx]) {	
if (coefficient_data_type == 0)	
pred_linear_interp_value_int [y][x][cmp][pivot_idx]	ue(v)
pred_linear_interp_value [y][x][cmp][pivot_idx]	u(v)
if (pivot_idx == num_pivots_minus2 [cmp;])	
if (coefficient_data_type == 0)	
pred_linear_interp_value_int [y][x][cmp][pivot_idx + 1]	ue(v)
pred_linear_interp_value [y][x][cmp][pivot_idx + 1]	u(v)
} // Linear interpolation	
else {	

for(i = 0 ; i <= poly_order_minus1 [y] [x] [cmp] [pivot_idx] + 1; i ++) {	
if (coefficient_data_type == 0)	
poly_coef_int [y] [x] [cmp] [pivot_idx] [i]	se(v)
poly_coef [y] [x] [cmp] [pivot_idx] [i]	u(v)
} // the i-th order	
} // Non-linear	
} // Polynomial coefficients	
else if (mapping_idx [y] [x] [cmp] [pivot_idx] == MAPPING_MMR) {	
mmr_order_minus1 [y] [x] [cmp] [pivot_idx]	u(2)
if (coefficient_data_type == 0)	
mmr_constant_int [y] [x] [cmp] [pivot_idx]	se(v)
mmr_constant [y] [x] [cmp] [pivot_idx]	u(v)
for(i = 1; i <= mmr_order_minus1 + 1; i +-) {	
for (j = 0; j < 7; j++) {	
if (coefficient_data_type == 0)	
mmr_coef_int [y] [x] [cmp] [pivot_idx] [i] [j]	se(v)
mmr_coef [y] [x] [cmp] [pivot_idx] [i] [j]	u(v)
} // the j-th coefficients	
} // the i-th order	
} // MMR coefficients	
}	

A 1.6 Reference Processing Unit Data Nonlinear Quantization and Quantization Parameters Syntax

5

Table A-1- 7 Reference Processing Unit Data Nonlinear Quantization Syntax

	Descriptor
rpu_data_nlq (y, x) {	
num_cmps = 3	
for (pivot_idx = 0; pivot_idx < nlq_num_pivots_minus2 + 1; pivot_idx ++) {	

for (cmp = 0; cmp < num_cmps; cmp ++) {	
if (num_nlq_param_predictors [y][x][pivot_idx][cmp] > 0)	
nlq_param_pred_f lag [y][x][pivot_idx][cmp]	u(l)
else	
nlq_param_pred_f lag [y][x][pivot_idx][cmp] = 0	
if (nlq_param_pred_f lag [y][x][pivot_idx][cmp] == 0)	
rpu_data_nlq_param (y, x, pivot_idx, cmp)	
else {	
if (num_nlq_param_predictors [y][x][pivot_idx][cmp] > 1)	
diff_pred_part_idx_nlq_minus1 [y][x][pivot_idx][cmp]	ue(v)
}	
} // cmp	
} // pivot_idx	
}	

Table A-1- 8 Reference Processing Unit Data Nonlinear Quantization Parameters Syntax

	Descriptor
rpu_data_nlq_param (y, x, pivot_idx, cmp) {	
nlq_offset [y][x][pivot_idx][cmp]	u(v)
if (coefficient_data_type == 0)	
vdr_in_max_int [y][x][pivot_idx][cmp]	ue(v)
vdr_in_max [y][x][pivot_idx][cmp]	u(v)
if (nlq_method_idc == NLQ_LINEAR_DZ) {	
if (coefficient_data_type == 0)	
linear_deadzone_slope_int [y][x][pivot_idx][cmp]	ue(v)
linear_deadzone_slope [y][x][pivot_idx][cmp]	u(v)
if (coefficient_data_type == 0)	
linear_deadzone_threshld_int [y][x][pivot_idx][cmp]	ue(v)

<pre> linear_deadzone_threshold [y][x][pivot_idx][cmp] </pre>	u(v)
<pre> } // Linear dead zone coefficients </pre>	
<pre> } </pre>	

A 2. Dolby Vision Metadata Semantics

[00085] This section specifies semantics associated with the syntax elements in the Dolby Vision consumer coding system. The section does not include specifications of semantics associated with the existing syntax elements in the H.265/ HEVC system. Refer to Ref. [3] for information about these elements. When the semantics of a syntax element are specified using a table or a set of tables, the bitstream must not include any values not specified therein.

A 2.1 Reference Processing Unit Data Raw Byte Sequence Payload Semantics

[00086] The reference processing unit raw byte sequence payload semantics are detailed in Reference Processing Unit Data Semantics.

A 2.2 Reference Processing Unit Data Semantics

`rpu_alignment_zero_bit` shall be equal to 0.

`rpu_data_crc32` is a 32-bit CRC to allow error detection for `rpu_data()`. A model for generating this 32-bit CRC is defined in Annex A of [10]. The CRC value is calculated on the entire `rpu_data()` excluding `rpu_data_crc32`, using the following generator polynomial:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

A 2.3 Reference Processing Unit Data Header Semantics

`rpu_type` specifies the type of this RPU and must be 2 for the BDA Ultra HD Blu-ray Option-A HDR coding system.

`rpu_format` specifies further information related to the prediction and coding process given `rpu_type`. The value of `rpu_format` must remain the same for the entire sequence. For BDA Ultra-HD Blu-ray, `rpu_format` must be set to 0.

Table A-2- 1 Definition of rpu_format field

Major version rpu_format bit[10:8]	Minor version rpu_format bit[7:4]	Codec Version
0b000	0b0000	Dolby Consumer EDR codec v1.0 architecture
0b000. .0b001	0b0001.-.0b11 11	Reserved for future Use
0b010. .0b1 11	0b0000..0b11 11	Unspecified

5 **vdr_rpu_prof ile** and **vdr_rpu_level** specify the profile and the level constraints that the Dolby Vision reference processing conforms to, and must be 1 and 0, respectively, for BDA Option-A HDR.

10 **vdr_seq_inf o_present_f lag** indicates if Dolby Vision sequence information is present in the current reference processing unit. When Dolby Vision sequence level information is present, all values of included syntax elements must remain the same for the entire video sequence.

For BDA Ultra- HD Blu-ray, **vdr_seq_info_present_flag** must be set to 0.

chroma_resampling_explicit_f ilter_f lag must be equal to 0 for Ultra-HD Blu-ray.

15 **coef ficient_data_type** specifies the data types of the mapping parameters and the NLQ parameters used during Dolby Vision codec processing. The value of **coefficient_data_type** must be in the range of 0 to 3.

The allowed parameter data types are as shown in the following table.

20

Table A-2- 2 Dolby Vision Codec Coefficient Data Types

coefficient_data_type	Dolby Vision Decoding Coefficient Data Type
0	fixed point
1	32-bit single precision defined in IEEE-754-2008
2...3	Reserved

coefficient_log2_denom specifies the number of fractional bits for Dolby Vision decoding related coefficients if **coefficient_data_type** is equal to 0. **coefficient_log2_denom** must be in the range of 0 to 32.

- 5 **coefficient_log2_denom** is not present when **coefficient_data_type** is not equal to 0. When **coefficient_log2_denom** is not present, it is assumed to be 0.

- 10 **vdr_rpu_normalized_idc** specifies what type of normalization is applied to the input and output data of the Dolby Vision reference processing unit interlayer prediction process. For Ultra-HD Blu-ray **vdr_rpu_normalized_idc** must be 1.

BL_video_full_range_flag equal to 1 specifies the black level and range of the luma and chroma BL signals is in full range. If **BL_full_range_flag** is equal to 0, the black level and range of the luma and chroma BL signals is in standard range.

- 15 When the **BL_video_full_range_flag** syntax element is not present, the value of **BL_video_full_range_flag** is assumed to be equal to 0.

BL_bit_depth_minus8 is used to derive the bit depth of BL signal, **BL_bit_depth**.

- 20 **BL_bit_depth** is equal to **BL_bit_depth_minus8** + 8. For Ultra-HD Blu-ray **BL_bit_depth_minus8** must be equal to 2.

EL_bit_depth_minus8 is used to derive the bit depth of the EL signal, **EL_bit_depth**.

- 25 **EL_bit_depth** is equal to **EL_bit_depth_minus8** + 8. For Ultra-HD Blu-ray **EL_bit_depth_minus8** must be equal to 2.

vdr_bit_depth_minus8 is used to derive the bit depth of the reconstructed Dolby Vision signal, **vdr_bit_depth**. **vdr_bit_depth** is equal to **vdr_bit_depth_minus8** + 8. The value of **vdr_bit_depth_minus8** must be in the range of 0 to 6.

- 30 **spatial_resampling_filter_flag** must be equal to 0 for Ultra-HD Blu-ray.

reserved_zero_3bits must be equal to 0.

el_spatial_resampling_filter_flag equal to **1** specifies that the EL spatial resampling filter is used for EL decoding. If **0**, this means spatial resampling is bypassed during EL decoding.

The value of **el_spatial_resampling_filter_flag** must be in the range of **0** and **1**. The sum of **el_spatial_resampling_filter_flag** and **spatial_resampling_filter_flag** must be less than or equal to **1**.

disable_residual_flag must be equal to **0** for Ultra-HD Blu-ray.

vdr_dm_metadata_present_flag specifies whether the **vdr_dm_data_payload()** syntax structure is present. The following values are supported:

- **1**: **vdr_dm_data_payload()** syntax structure as defined in Annex A.2 is present
- **0**: **vdr_dm_data_payload()** syntax structure is not present.

If **vdr_dm_metadata_present_flag** is not present, it is assumed to be **0**. For Ultra-HD Blu-ray

vdr_dm_metadata_present_flag must be equal to **1**.

use_prev_vdr_rpu_flag specifies whether the current Dolby Vision reference processing unit is predicted from a previously sent reference processing unit. The following values are supported.

- **1**: A previously sent Dolby Vision reference processing unit is used to generate the interlayer prediction for the current picture
- **0**: The current Dolby Vision reference processing unit is transmitted explicitly, and a **vdr_rpu_id** is assigned to the current Dolby Vision reference processing unit.

When the BL picture is an instantaneous refresh picture, the value of **use_prev_vdr_rpu_flag** must be **0**.

prev_vdr_rpu_id specifies the **rpu_id** of the previously sent Dolby Vision reference processing unit that is used to process the current picture.

The value of **prev_vdr_rpu_id** must be in the range of **0** to **15**, further constrained by the number of Dolby Vision reference processing units transmitted since the last instantaneous

decoding refresh picture. If **prev_vdr_rpu_id** is not present, it must be set to **-1** (invalid **rpu_id**).

vdr_rpu_id **vdr_rpu_id** specifies the **rpu_id** assigned to the current Dolby Vision reference processing unit that is explicitly transmitted in the **rpu_data_payload** ().

The value of **vdr_rpu_id** must be in the range of 0 to 15. If the BL picture is an instantaneous decoding refresh picture, **vdr_rpu_id** must be equal to 0.

5 An instantaneous decoding refresh picture causes the decoding process to mark all reference processing units as unused for reference immediately after the reference processing unit bitstream decoding of the instantaneous decoding refresh picture. All reference processing units that follow a reference processing unit of the instantaneous decoding refresh picture can be decoded without
10 referencing from any reference processing unit that precedes the reference processing unit of the instantaneous decoding refresh picture.

mapping_color_space must be set to 0 for BDA Option-A.

mapping_chroma_format_idc must be set to 0 for BDA Option-A.

15 **num_pivots_minus2** [**cmp**] specifies the number of pivot points minus two used to perform the piecewise mapping process for the color component **cmp**. For example, if one mapping model is applied for the entire BL signal dynamic range, then the value of **num_pivots_minus2** is equal to 0.

20 The value of **num_pivots_minus2** must be in the range of 0 to 7, inclusive.

pred_pivot_value [**cmp**] [**i**] is used to derive the value **pivot_value** [**cmp**] [**i**] of the **i**-th pivot point for the component **cmp**. These pivot points collectively define the boundaries of the piecewise mapping process. The value of **pivot_value** [**cmp**] [**i**] is derived using the
25 following pseudocode. The array **pred_pivot_value** [] [] is used as input and the array **pivot_value** [] [] is used as output. The value of **i** must be in the range of 0 to (**num_pivots_minus2** + 1), inclusive.

The number of bits used for the representation of the **pred_pivot_value** [**cmp**] [**i**] syntax element is **BL_bit_depth** bits.

30 **assign_pivot_values**(**pred_pivot_value**[] [], **num_pivots_minus2**)
{
 int **pivot_value** [3] [**num_pivots_minus2**+2];

```

    int pivot_idx, cmp;

    for( cmp = 0; cmp < 3; cmp ++ ) {
        pivot_value [ cmp ][ 0 ] = pred_pivot_value [ cmp ][ 0 ];
5       for( pivot_idx = 1; pivot_idx < num_pivots_minus2[ cmp ] + 2; pivot_idx ++ )
            pivot_value [ cmp ][ pivot_idx ] =
                pivot_value [ cmp ][ pivot_idx - 1 ]+pred_pivot_value [ cmp ][ pivot_idx
];
    }
10    return pivot_value;
}

```

The component order is Y, Cb, Cr for `cmp=0, 1, 2`.

`nlq_method_idc` specifies the NLQ method used for current EL video frame. The value of
15 `nlq_method_idc` must be set to 0 for BDA Option-A.

`num_x_partitions_minus1` must be set to 0 for BDA Option-A.

`num_y_partitions_minus1` must be set to 0 for BDA Option-A.

20

A 2.4 Reference Processing Unit Data Payload Semantics

This section provides details on the semantics of the reference processing unit data payload for
`rpu_type` equal to 2.

A 2.4.1 Reference Processing Unit Data Mapping Syntax

25

`mapping_idc [y][x][cmp][pivot_idx]` specifies the mapping method used for
the the color component `cmp`, and the dynamic range identified by `pivot_idx`.

The value of `mapping_idc [y][x][cmp][pivot_idx]` must be in the range of 0 to
1, inclusive. The values of `mapping_idc [y][x][cmp][pivot_idx]` and the

30 corresponding mapping method currently supported in the Dolby Vision system for BDA Option-A
are specified in the following table.

Table A-2- 3 Interlayer Mapping Methods

mapping_idc	Name	Method
0	MAPPING_POLYNOMIAL	N* order polynomial mapping (N>=1)
1	MAPPINGJVIMR	Multivariate multiple regression mapping

5 mapping_param_pred_flag [y] [x] [cmp] [pivot_idx] specifies whether the mapping parameters associated with mapping_idc [y] [x] [cmp] [pivot_idx] are explicitly transmitted or predicted from a previous partition in the current picture reference processing unit. Supported values are:

- 0: The mapping parameters associated with mapping_idc [y] [x] [cmp] [pivot_idx] are explicitly transmitted.
- 10 • 1: The mapping parameters of a previous partition from the current picture are assigned to the mapping parameters associated with mapping_idc [y] [x] [cmp] [pivot_idx].

When the syntax element mapping_param_pred_flag [y] [x] [cmp] [pivot_idx] is not present, it shall be inferred to be 0.

15 If no predictor is available from previous partitions in the current picture (for example, if no previous partition from the current picture reference processing unit uses the same mapping method as indicated by mapping_idc [y] [x] [cmp] [pivot_idx]), then there will be no predictor available to predict mapping parameters for mapping_idc [y] [x] [cmp] [pivot_idx]. In this case, num_mapping_param_predictors [y] [x] [cmp] [pivot_idx] and mapping_param_pred_flag [y] [x] [cmp] [pivot_idx] are both set to 0.

Note - num_mapping_param_predictors [y] [x] [cmp] [pivot_idx] is the number of available mapping parameter predictors in the current picture associated with mapping_idc [y] [x] [cmp] [pivot_idx]. This number increments by one when a new set of mapping parameters associated with mapping_idc [y] [x] [cmp] [pivot_idx] is explicitly transmitted, i.e. the value of mapping_param_pred_flag [y] [x] [cmp] [pivot_idx] is 0.

`diff_pred_part_idx_mapping_minus1 [y][x][cmp][pivot_idx]` specifies a differential value used to derive the partition index, `pred_part_idx_mapping [y][x][cmp][pivot_idx]`, of the mapping parameter predictor in the current reference processing unit. The presence of `diff_pred_part_idx_mapping_minus1 [y][x][cmp][pivot_idx]` depends on `num_mapping_param_predictors [y][x][cmp][pivot_idx]`.

The value of `pred_part_idx_mapping [y][x][cmp][pivot_idx]` is derived and constrained as follows:

- If `num_mapping_param_predictors [y][x][cmp][pivot_idx]` is greater than 1, the value of `pred_part_idx_mapping [y][x][cmp][pivot_idx]` is set to $(curr_part_idx - diff_pred_part_idx_mapping_minus1 [y][x][cmp][pivot_idx] - 1)$.
Note - `curr_part_idx` is the index of current partition in raster scan order.
- Otherwise, if `num_mapping_param_predictors [y][x][cmp][pivot_idx]` is equal to 1, the value of `pred_part_idx_mapping [y][x][cmp][pivot_idx]` is set to be the partition index of the only previous partition in the current reference processing unit that can serve as the predictor of mapping parameters associated with `mapping_idc [y][x][cmp][pivot_idx]`.
- The value of `pred_part_idx_mapping [y][x][cmp][pivot_idx]` must be in the range of 0 to $(curr_part_idx - 1)$.

A 2.4.2 Reference Processing Unit Data Mapping Parameters Semantics

`poly_order_minus1 [y][x][cmp][pivot_idx]` specifies the order of the polynomial mapping method identified by `mapping_idc [y][x][cmp][pivot_idx]`. The value of `poly_order_minus1 [y][x][cmp][pivot_idx]` must be 0 or 1.

`linear_interp_flag [y][x][cmp][pivot_idx]` specifies whether the first order polynomial mapping parameters are indicated using linear interpolation method.

- 1: The first-order polynomial mapping parameters are indicated using linear interpolation method.

- 0: The first-order polynomial mapping parameters are indicated using polynomial coefficients.

Note - linear interpolation is another way to specify the first order polynomial. The inverse mapped value for the end points for each piece is specified and the values in between the end points are interpolated. Linear interpolation can be used to implement look-up tables.

pred_linear_interp_value_int[y][x][cmp][pivot_idx] specifies the integer portion of fp_pred_linear_interp_value [y][x][cmp][pivot_idx] when coefficient_data_type is equal to 0. If coefficient_data_type equal to 1, pred_linear_interp_value_int [y][x][cmp][pivot_idx] is not present.

Note - fp_pred_linear_interp_value [y][x][cmp][pivot_idx] is used to derive the value of linear_interp_value [y][x][cmp][pivot_idx] associated with mapping_idc [y][x][cmp][pivot_idx] when linear_interp_flag [y][x][cmp][pivot_idx] is equal to 1 and coefficient_data_type is equal to 0.

pred_linear_interp_value[y][x][cmp][pivot_idx] specifies the fractional portion of fp_pred_linear_interp_value[y][x][cmp][pivot_idx] when coefficient_data_type is equal to 0. If coefficient_data_type equal to 1, pred_linear_interp_value [y][x][cmp][pivot_idx] is used to derive the value of linear_interp_value [y][x][cmp][pivot_idx] associated with mapping_idc [y][x][cmp][pivot_idx] when linear_interp_flag [y][x][cmp][pivot_idx] is equal to 1. If coefficient_data_type is equal to 0, the length of the pred_linear_interp_value [y][x][cmp][pivot_idx] syntax element is coefficient_log2_denom bits. If coefficient_data_type is equal to 1, the length of the pred_linear_interp_value [y][x][cmp][pivot_idx] syntax element is 32 bits.

The values of linear_interp_value [y][x][cmp][pivot_idx] is derived and constrained as follows:

- If pivot_idx is equal to 0, the value of linear_interp_value [y][x][cmp][pivot_idx] is equal to pred_linear_interp_value [y][x][cmp][pivot_idx] if coefficient_data_type is equal to 1. If coefficient_data_type is equal to 0, fp_pred_linear_interp_value [y][x][cmp][pivot_idx] = (pred_linear_interp_value_int [y][x][cmp][pivot_idx] << coefficient_log2_denom) + pred_linear_interp_value [y][x][cmp][

$\text{pivot_idx}]$ and $\text{linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx}] =$
 $\text{fp_pred_linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx}]$.

- Otherwise, the value of $\text{linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx}]$ is equal to $\text{pred_linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx}] +$
 5 $\text{linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx} - 1]$ if $\text{coefficient_data_type}$ is equal to 1. If $\text{coefficient_data_type}$ is equal to 0,
 $\text{fp_pred_linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx}] =$
 $(\text{pred_linear_interp_value_int}[y][x][\text{cmp}][\text{pivot_idx}] \ll$
 $\text{coefficient_log2_denom}) + \text{pred_linear_interp_value}[y][x][\text{cmp}][$
 10 $\text{pivot_idx}]$ and $\text{linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx}] =$
 $\text{fp_pred_linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx}] +$
 $\text{linear_interp_value}[y][x][\text{cmp}][\text{pivot_idx} - 1]$.
- If $\text{vdr_rpu_normalized_idc}$ is equal to 0, $\text{linear_interp_value}[y][x][\text{cmp}][$
 $i]$ shall be in the range of $[0, (2^{\text{vdr_bit_depth}} - i)]$. If $\text{vdr_rpu_normalized_idc}$ is equal
 15 to 1, $\text{linear_interp_value}[y][x][\text{cmp}][i]$ shall be in the decimal valued
 range of $[0, 1)$.

$\text{poly_coef_int}[y][x][\text{cmp}][\text{pivot_idx}][i]$ specifies the integer portion of
 $\text{fp_poly_coef}[y][x][\text{cmp}][\text{pivot_idx}][i]$ when $\text{coefficient_data_type}$ is
 20 equal to 0. If $\text{coefficient_data_type}$ is equal to 1, $\text{poly_coef_int}[y][x][\text{cmp}][$
 $\text{pivot_idx}][i]$ is not present.

Note - $\text{fp_poly_coef}[y][x][\text{cmp}][\text{pivot_idx}][i]$ is used to derive the value of
 the i-th order polynomial coefficient associated with $\text{mapping_idc}[y][x][\text{cmp}][$
 $\text{pivot_idx}]$ when $\text{coefficient_data_type}$ is equal to 0.

$\text{poly_coef}[y][x][\text{cmp}][\text{pivot_idx}][i]$ specifies the fractional portion of
 $\text{fp_poly_coef}[y][x][\text{cmp}][\text{pivot_idx}][i]$ when $\text{coefficient_data_type}$ is
 equal to 0. If $\text{coefficient_data_type}$ is equal to 1, $\text{poly_coef}[y][x][\text{cmp}][$
 $\text{pivot_idx}][i]$ is used to derive the value of the i-th order polynomial coefficient associated
 30 with $\text{mapping_idc}[y][x][\text{cmp}][\text{pivot_idx}]$. If $\text{coefficient_data_type}$ is equal to
 0, the length of the $\text{poly_coef}[y][x][\text{cmp}][\text{pivot_idx}][i]$ syntax element is
 $\text{coefficient_log2_denom}$ bits. If $\text{coefficient_data_type}$ is equal to 1, the length of the
 $\text{poly_coef}[y][x][\text{cmp}][\text{pivot_idx}][i]$ syntax element is 32 bits.

The value of the i-th order polynomial coefficient associated with mapping_idc [y][x][cmp][pivot_idx] is derived as follows:

- If coefficient_data_type is equal to 0, the value of the i-th order polynomial coefficient is equal to $fp_poly_coef [y][x][cmp][pivot_idx][i] = (poly_coef_int [y][x][cmp][pivot_idx][i] \ll coefficient_log2_denom) + poly_coef [y][x][cmp][pivot_idx][i]$.
- If coefficient_data_type is equal to 1, the value of the i-th order polynomial coefficient is equal to $poly_coef [y][x][cmp][pivot_idx][i]$.

10 **mmr_order_minus1 [y][x][cmp][pivot_idx]** specifies the order of the multivariate multiple regression mapping method identified by mapping_idc [y][x][cmp][pivot_idx]. The value of **mmr_order_minus1 [y][x][cmp][pivot_idx]** must be in the range of 0 to 2.

15 **mmr_constant_int [y][x][cmp][pivot_idx]** specifies the integer portion of $fp_mmr_constant [y][x][cmp][pivot_idx]$ when coefficient_data_type is equal to 0. If coefficient_data_type is equal to 1, **mmr_constant_int [y][x][cmp][pivot_idx]** is not present.

Note - $fp_mmr_constant [y][x][cmp][pivot_idx]$ is used to derive the value of the multivariate multiple regression constant coefficient associated with mapping_idc [y][x][cmp][pivot_idx] when coefficient_data_type is equal to 0.

25 **mmr_constant [y][x][cmp][pivot_idx]** specifies the fractional portion of $fp_mmr_constant [y][x][cmp][pivot_idx]$ when coefficient_data_type is equal to 0. If coefficient_data_type is equal to 1, **mmr_constant [y][x][cmp][pivot_idx]** is used to derive the value of multivariate multiple regression constant coefficient associated with mapping_idc [y][x][cmp][pivot_idx]. If coefficient_data_type is equal to 0, the length of the **mmr_constant [y][x][cmp][pivot_idx]** syntax element is coefficient_log2_denom bits. If coefficient_data_type is equal to 1, the length of the **mmr_constant [y][x][cmp][pivot_idx]** syntax element is 32 bits.

The value of multivariate multiple regression constant coefficient associated with mapping_idc [y][x][cmp][pivot_idx] is derived as follows:

- If `coefficient_data_type` is equal to 0, the value of multivariate multiple regression constant coefficient is equal to `fp_mmr_constant [y][x][cmp][pivot_idx] = (mmr_constant_int [y][x][cmp][pivot_idx] << coefficient_log2_denom) + mmr_constant [y][x][cmp][pivot_idx]`.
- If `coefficient_data_type` is equal to 1, the value of multivariate multiple regression constant coefficient is equal to `mmr_constant [y][x][cmp][pivot_idx]`.

`mmr_coef_int [y][x][cmp][pivot_idx][i][j]` specifies the integer portion of `fp_mmr_coef [y][x][cmp][pivot_idx][i][j]` when `coefficient_data_type` is equal to 0. If `coefficient_data_type` is equal to 1, `mmr_coef_int [y][x][cmp][pivot_idx][i][j]` is not present.

Note - `fp_mmr_coef [y][x][cmp][pivot_idx][i][j]` is used to derive the value of the j-th multivariate multiple regression coefficient at the i-th order associated with `mapping_idc [y][x][cmp][pivot_idx]` when `coefficient_data_type` is equal to 0.

`mmr_coef [y][x][cmp][pivot_idx][i][j]` specifies the fractional portion of `fp_mmr_coef [y][x][cmp][pivot_idx][i][j]` when `coefficient_data_type` is equal to 0. If `coefficient_data_type` is equal to 1, `mmr_coef [y][x][cmp][pivot_idx][i][j]` is used to derive the value of the j-th multivariate multiple regression coefficient at the i-th order associated with `mapping_idc [y][x][cmp][pivot_idx]`. If `coefficient_data_type` is equal to 0, the length of the `mmr_coef [y][x][cmp][pivot_idx][i][j]` syntax element is `coefficient_log2_denom bits`. If `coefficient_data_type` is equal to 1, the length of the `mmr_coef [y][x][cmp][pivot_idx][i][j]` syntax element is 32 bits.

The value of the j-th multivariate multiple regression coefficient at the i-th order associated with `mapping_idc [y][x][cmp][pivot_idx]` is derived as follows:

- If `coefficient_data_type` is equal to 0, the value of the j-th multivariate multiple regression coefficient at the i-th order is equal to `fp_mmr_coef [y][x][cmp][pivot_idx][i][j] = (mmr_coef_int [y][x][cmp][pivot_idx][i][j] << coefficient_log2_denom) + mmr_coef [y][x][cmp][pivot_idx][i][j]`.
- If `coefficient_data_type` is equal to 1, the value of the j-th multivariate multiple regression coefficient at the i-th order is equal to `mmr_coef [y][x][cmp][pivot_idx][i][j]`.

A 2.4.3 Reference Processing Unit Data Nonlinear Quantization Semantics

nlq_param_pred_f_lag[*y*][*x*][**pivot_idx**][**cmp**] specifies whether NLQ parameters associated with the partition, vertical coordinate *y*, horizontal coordinate *x*, **pivot_idx**, and **cmp** are explicitly transmitted or predicted from a previous partition in the current picture reference processing unit.

The two supported values are:

- \emptyset : The NLQ parameters associated with vertical coordinate *y*, horizontal coordinate *x*, **pivot_idx**, and **cmp** are explicitly transmitted.
- 1: The NLQ parameters of a previous partition in the reference processing unit from the current picture are assigned to the NLQ parameters associated with the partition with vertical coordinate *y*, horizontal coordinate *x*, **pivot_idx**, and **cmp**.

When the syntax element **nlq_param_pred_f_lag** [*y*][*x*][**pivot_idx**][**cmp**] is not present, it is assumed to be \emptyset .

Occasionally, no predictor from previous partitions in the current reference processing unit is available. For example, if no previous partition reference processing unit from the current picture uses the same NLQ method as indicated by the partition with vertical coordinate *y*, horizontal coordinate *x*, **pivot_idx**, and **cmp**, there will be no predictor available to predict NLQ parameters for the partition with vertical coordinate *y* and horizontal coordinate *x*, **pivot_idx**, and **cmp**. In this case, **num_nlq_param_predictors** [*y*][*x*][**pivot_idx**][**cmp**] and **nlq_param_pred_f_lag** [*y*][*x*][**pivot_idx**][**cmp**] are both set to 0.

Note - The **num_nlq_param_predictors** [*y*][*x*][**pivot_idx**][**cmp**] is the number of available NLQ parameter predictors in the current reference processing unit for the current partition with vertical coordinate *y*, horizontal coordinate *x*, **pivot_idx**, and **cmp**. This number is incremented by one when a new set of NLQ parameters is explicitly transmitted. (That is, when the value of **nlq_param_pred_f_lag** [*y*][*x*][**pivot_idx**][**cmp**] is 0).

diff_pred_part_idx_nlq_minus1[*y*][*x*][**pivot_idx**][**cmp**] specifies a differential value used to derive the partition index, **pred_part_idx_nlq** [*y*][*x*][**pivot_idx**][**cmp**], of the NLQ parameter predictor from the reference processing unit of the current picture. The

presence of the syntax element `diff_pred_part_idx_nlq_minus1[y][x][pivot_idx][cmp]` depends on the value of `num_nlq_param_predictors[y][x][pivot_idx][cmp]`.

5 The value of `pred_part_idx_nlq[y][x][pivot_idx][cmp]` is derived and constrained as follows:

- If `num_nlq_param_predictors[y][x][pivot_idx][cmp]` is greater than 1, the value of `pred_part_idx_nlq[y][x][pivot_idx][cmp]` is set to $(\text{curT_part_idx} - \text{diff_pred_part_idx_nlq_minus1}[y][x][pivot_idx][cmp] - 1)$.
- 10 • If `num_nlq_param_predictors[y][x][pivot_idx][cmp]` is equal to 1, the value of `pred_part_idx_nlq[y][x][pivot_idx][cmp]` is set to be the partition index of the only previous partition in the reference processing unit of the current picture that can serve as the predictor of NLQ parameters associated with the partition in vertical coordinate `y`, horizontal coordinate `x`, `pivot_idx`, and `cmp`.
- 15 • The value of `pred_part_idx_nlq[y][x][pivot_idx][cmp]` must be in the range of \emptyset and $(\text{cur r_part_idx} - 1)$.

A 2.4.4 Reference Processing Unit Data Nonlinear Quantization Parameter Semantics

20 **`nlq_offset[y][x][pivot_idx][cmp]`** specifies the NLQ offset coefficient associated with the partition in vertical coordinate `y`, horizontal coordinate `x`, `pivot_idx`, and `cmp`.

The number of bits used for the representation of `nlq_offset[y][x][pivot_idx][cmp]` syntax element is `EL_bit_depth`. The value of `nlq_offset[y][x][pivot_idx][cmp]` is in the range of $[0, 2^{\text{EL_bit_depth}} - 1]$.

25 **`vdr_in_max_int[y][x][pivot_idx][cmp]`** specifies the integer portion of `fp_vdr_in_max[y][x][pivot_idx][cmp]` when `coefficient_data_type` is equal to \emptyset . If `coefficient_data_type` is equal to 1, `vdr_in_max_int[y][x][pivot_idx][cmp]` is not present.

30 Note `-fp_vdr_in_max[y][x][pivot_idx][cmp]` is used to derive the value of non-linear quantization maximum EL signal value coefficient associated the partition with vertical

coordinate y and horizontal coordinate x , pivot index $pivot_idx$ and component cmp when $coefficient_data_type$ is equal to 0.

5 **vdr_in_max**[y][x][**pivot_idx**][cmp] specifies the fractional portion of $fp_vdr_in_max$ [y][x][$pivot_idx$][cmp] when $coefficient_data_type$ is equal to \emptyset .

If $coefficient_data_type$ is equal to 1, **vdr_in_max** [y][x][$pivot_idx$][cmp] is used to derive the value of the NLQ maximum EL signal value coefficient associated with the partition in vertical coordinate y , horizontal coordinate x , $pivot_idx$, and cmp .

10 If $coefficient_data_type$ is equal to 0, the length of the **vdr_in_max** [y][x][$pivot_idx$][cmp] syntax element is $coefficient_log2_denom$ bits.

If $coefficient_data_type$ is equal to 1, the length of the **vdr_in_max**[y][x][$pivot_idx$][cmp] syntax element is 32 bits.

15 The value of the NLQ maximum EL signal value coefficient associated with the partition in vertical coordinate y , horizontal coordinate x , $pivot_idx$, and cmp is derived as follows:

- If $coefficient_data_type$ is equal to \emptyset the value of the NLQ maximum EL signal value coefficient is equal to $fp_vdr_in_max$ [y][x][$pivot_idx$][cmp], where,

20 $fp_vdr_in_max$ [y][x][$pivot_idx$][cmp] = $(vdr_in_max_int$ [y][x][$pivot_idx$][cmp] \ll $coefficient_log2_denom$) + vdr_in_max [y][x][$pivot_idx$][cmp].

- If $coefficient_data_type$ is equal to 1, the value of NLQ maximum EL signal value coefficient is equal to vdr_in_max [y][x][$pivot_idx$][cmp].

25

linear_deadzone_slope_int [y][x][**pivot_idx**][cmp] specifies the integer portion of $fp_linear_deadzone_slope$ [y][x][$pivot_idx$][cmp] when $coefficient_data_type$ is equal to 0. If $coefficient_data_type$ is equal to 1, $linear_deadzone_slope_int$ [y][x][$pivot_idx$][cmp] is not present.

30 Note - $fp_linear_deadzone_slope$ [y][x][$pivot_idx$][cmp] is used to derive the value of linear dead zone slope coefficient associated the partition with vertical coordinate y and horizontal coordinate x , pivot index $pivot_idx$ and component cmp when $coefficient_data_type$ is equal to \emptyset .

linear_deadzone_slope[**y**][**x**][**pivot_idx**][**cmp**] specifies the fractional portion of **fp_linear_deadzone_slope** [**y**][**x**][**pivot_idx**][**cmp**] when **coefficient_data_type** is equal to 0.

5 If **coefficient_data_type** is equal to 1, **linear_deadzone_slope** [**y**][**x**][**pivot_idx**][**cmp**] is used to derive the value of linear dead zone slope coefficient associated with the partition with vertical coordinate **y** and horizontal coordinate **x**, pivot index **pivot_idx** and component **cmp**.

10 If **coefficient_data_type** is equal to 0, the length of the **linear_deadzone_slope** [**y**][**x**][**pivot_idx**][**cmp**] syntax element is **coefficient_log2_denom bits**.

If **coefficient_data_type** is equal to 1, the length of the **linear_deadzone_slope** [**y**][**x**][**pivot_idx**][**cmp**] syntax element is 32 bits.

15 The value of linear dead zone slope coefficient associated with the partition with vertical coordinate **y** and horizontal coordinate **x**, pivot index **pivot_idx** and component **cmp** is derived as follows:

- If **coefficient_data_type** is equal to 0, the value of linear dead zone slope coefficient is equal to **fp_linear_deadzone_slope** [**y**][**x**][**pivot_idx**][**cmp**] = **(linear_deadzone_slope_int** [**y**][**x**][**pivot_idx**][**cmp**] << **coefficient_log2_denom**) + **linear_deadzone_slope** [**y**][**x**][**pivot_idx**][**cmp**]
- If **coefficient_data_type** is equal to 1, the value of linear dead zone slope coefficient is equal to **linear_deadzone_slope** [**y**][**x**][**pivot_idx**][**cmp**]

25 **linear_deadzone_threshold_int**[**y**][**x**][**pivot_idx**][**cmp**] specifies the integer portion of **fp_linear_deadzone_threshold** [**y**][**x**][**pivot_idx**][**cmp**] when **coefficient_data_type** is equal to 0. If **coefficient_data_type** is equal to 1,

linear_deadzone_threshold_int [**y**][**x**][**pivot_idx**][**cmp**] is not present.

30 **Note** - **fp_linear_deadzone_threshold** [**y**][**x**][**pivot_idx**][**cmp**] is used to derive the value of linear dead zone threshold coefficient associated the partition with vertical coordinate **y** and horizontal coordinate **x**, pivot index **pivot_idx** and component **cmp** when **coefficient_data_type** is equal to 0.

linear_deadzone_threshold[y][x][pivot_idx][cmp] specifies the fractional portion of `fp_linear_deadzone_threshold[y][x][pivot_idx][cmp]` when `coefficient_data_type` is equal to 0.

5 If `coefficient_data_type` is equal to 1, `linear_deadzone_threshold[y][x][pivot_idx][cmp]` is used to derive the value of linear dead zone threshold coefficient associated with the partition with vertical coordinate `y` and horizontal coordinate `x`, pivot index `pivot_idx` and component `cmp`.

If `coefficient_data_type` is equal to 0, the length of the `linear_deadzone_threshold[y][x][pivot_idx][cmp]` syntax element is `coefficient_log2_denom` bits.

10 If `coefficient_data_type` is equal to 1, the length of `linear_deadzone_threshold[y][x][pivot_idx][cmp]` is 32 bits.

The value of linear dead zone threshold coefficient associated with the partition with vertical coordinate `y` and horizontal coordinate `x`, pivot index `pivot_idx` and component `cmp` is derived as follows:

15 · If `coefficient_data_type` is equal to 0, the value of linear dead zone threshold coefficient is equal to $fp_linear_deadzone_threshold[y][x][pivot_idx][cmp] = (linear_deadzone_threshold_int[y][x][pivot_idx][cmp] \ll coefficient_log2_denom) + linear_deadzone_threshold[y][x][pivot_idx][cmp]$.

20 · If `coefficient_data_type` is equal to 1, the value of linear dead zone threshold coefficient is equal to `linear_deadzone_threshold[y][x][pivot_idx][cmp]`.

Note - The value of linear dead zone threshold coefficient shall be greater than or equal to zero. The value of linear dead zone maximum Dolby Vision EL signal value coefficient shall be greater than or equal to the value of linear dead zone threshold coefficient.

25

A 3. Dolby Vision Display Management Metadata Bitstream Syntax

[00087] This section specifies the syntax and semantics of the Dolby Vision display management metadata bitstream. Dolby Vision display management metadata is not required for reconstructing Dolby Vision signals by the decoding process. Conforming decoders are not required to process this information. When present in the bitstream, Dolby Vision display management metadata must follow the syntax and semantics specified in A 3.1 and A 4.

30

When the Dolby Vision display management metadata is conveyed for the application by some means other than presence within the bitstream, the representation of the Dolby Vision display management metadata is not required to use the same syntax specified in this section.

A 3.1 Dolby Vision Display Management Metadata Bistream Syntax

5

Table A-3- 1 Dolby Vision Display Management Metadata Bistream Syntax

Vdr_dm_data_payload () {	Descriptor
affected_dm_metadata_id	ue(v)
current_dm_metadata_id	ue(v)
scene_refresh_flag	ue(v)
YCtoRGB_coef0	i(16)
YCtoRGB_coef1	i(16)
YCtoRGB_coef2	i(16)
YCtoRGB_coef3	i(16)
YCtoRGB_coef4	i(16)
YCtoRGB_coef5	i(16)
YCtoRGB_coef6	i(16)
YCtoRGB_coef7	i(16)
YCtoRGB_coef8	i(16)
YCtoRGB_offset0	u(32)
YCtoRGB_offset1	u(32)
YCtoRGB_offset2	u(32)
RGBtoLMS_coef 0	i(16)
RGBtoLMS_coef 1	i(16)
RGBtoLMS_coef 2	i(16)
RGBtoLMS_coef 3	i(16)
RGBtoLMS_coef 4	i(16)
RGBtoLMS_coef 5	i(16)
RGBtoLMS_coef 6	i(16)
RGBtoLMS_coef 7	i(16)
RGBtoLMS_coef 8	i(16)
signal_eof	u(16)

<code>signal_eotf_param0</code>	<code>u(16)</code>
<code>signal_eotf_param1</code>	<code>u(16)</code>
<code>signal_eotf_param2</code>	<code>u(32)</code>
<code>signal_bit_depth</code>	<code>u(5)</code>
<code>signal_color_space</code>	<code>u(2)</code>
<code>signal_chroma_format</code>	<code>u(2)</code>
<code>signal_full_range_flag</code>	<code>u(2)</code>
<code>source_min_PQ</code>	<code>u(12)</code>
<code>source_max_PQ</code>	<code>u(12)</code>
<code>source_diagonal</code>	<code>u(10)</code>
<code>num_ext_blocks</code>	<code>ue(v)</code>
<code>if (num_ext_blocks) {</code>	
<code> while (!byte_aligned ())</code>	
<code> dm_alignment_zero_bit</code>	<code>f(1)</code>
<code> for(i = 0; i < num_ext_blocks; i ++) {</code>	
<code> ext_metadata_block ()</code>	
<code> }</code>	
<code>}</code>	

A 3.2 Dolby Vision Display Management Extended Metadata Bistream Syntax

5

Table A-3- 2 Dolby Vision Display Management Extended Metadata Bistream Syntax

<code>Ext_metadata_block () {</code>	Descriptor
<code> ext_block_length</code>	<code>ue(v)</code>
<code> ext_block_level</code>	<code>u(8)</code>
<code> ext_block_payload (ext_block_length, ext_block_level)</code>	
<code>}</code>	

A 3.3 Dolby Vision Display Management Extended Metadata Payload Bistream Syntax

Table A-3-3 Dolby Vision Display Management Extended Metadata Payload Bistream Syntax

ext_block_payload (ext_block_length, ext_block_level) {	Descriptor
ext_block_len_bits = 8 * ext_block_length	
ext_block_use_bits = 0	
if (ext_block_level == 1) {	
min_PQ	u(12)
max_PQ	u(12)
avg_PQ	u(12)
ext_block_use_bits += 36	
}	
if (ext_block_level == 2) {	
target_max_PQ	u(12)
trim_slope	u(12)
trim_of_fset	u(12)
trim_power	u(12)
trim_chroma_weight	u(12)
trim_saturation_gain	u(12)
ms_weight	i(13)
ext_block_use_bits += 85	
}	
while (ext_block_use_bits++ < ext_block_len_bits)	
ext_dm_alignment_zero_bit	f(1)
}	

5 **A 4. Dolby Vision Display Management Metadata Semantics**

A 4.1 Dolby Vision Display Management Metadata Bistream Semantics

affected_dm_metadata_id indicates the id number of the Dolby Vision video frame that is relevant for this dm metadata. The value shall be in the range of 0 and 15, inclusive. If **affected_dm_metadata_id** is not present, it shall be inferred to be 0.

5

current_dm_metadata_id indicates the id number of the current Dolby Vision video frame. The value shall be in the range of 0 and 15, inclusive. If **current_dm_metadata_id** is not present, it shall be inferred to be 0.

Note: If DM metadata is relevant for the current Dolby Vision video frame,

10

affected_dm_metadata_id shall be set to **current_dm_metadata_id**. If DM metadata regards upcoming Dolby Vision video frame, **affected_dm_metadata_id** shall be set to $(\text{current_dm_metadata_id} + 1) \pmod{16}$. **affected_dm_metadata_id** shall be set to either **current_dm_metadata_id** OR $(\text{current_dm_metadata_id} + 1) \pmod{16}$, therefore, only current and upcoming Dolby Vision video frame's metadata may be described.

15

scene_refresh_flag equal to 1 specifies the video frame, which DM metadata affects, is a scene refresh point. Please refer to the DM design document for the definition of scene refresh point. The value shall be in the range of 0 and 1, inclusive. If **scene_refresh_flag** is not present, it shall be inferred to be 0.

20

YCCtoRGB_coef0 specifies the first coefficient of YCCtoRGB transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **YCCtoRGB_coef0** is not present, it shall be inferred to be 8192.

25

YCCtoRGB_coef1 specifies the second coefficient of YCCtoRGB transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **YCCtoRGB_coef1** is not present, it shall be inferred to be 0.

30

YCCtoRGB_coef2 specifies the third coefficient of YCCtoRGB transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **YCCtoRGB_coef2** is not present, it shall be inferred to be 12900.

YCCtoRGB_coef3 specifies the fourth coefficient of YCCtoRGB transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **YCCtoRGB_coef3** is not present, it shall be inferred to be **8192**.

5 **YCCtoRGB_coef4** specifies the fifth coefficient of YCCtoRGB transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **YCCtoRGB_coef4** is not present, it shall be inferred to be **-1534**.

10 **YCCtoRGB_coef5** specifies the sixth coefficient of YCCtoRGB transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **YCCtoRGB_coef5** is not present, it shall be inferred to be **-3836**.

15 **YCCtoRGB_coef6** specifies the seventh coefficient of YCCtoRGB transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **YCCtoRGB_coef6** is not present, it shall be inferred to be **8192**.

20 **YCCtoRGB_coef7** specifies the eighth coefficient of YCCtoRGB transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **YCCtoRGB_coef7** is not present, it shall be inferred to be **15201**.

YCCtoRGB_coef8 specifies the ninth coefficient of YCCtoRGB transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **YCCtoRGB_coef8** is not present, it shall be inferred to be 0.

25 **YCCtoRGB_off set0** specifies the first offset of YCCtoRGB transform matrix. The value shall be in the range of **0** to **4294967295**, inclusive. If **YCCtoRGB_off set0** is not present, it shall be inferred to be **0**.

30 **YCCtoRGB_off set1** specifies the second offset of YCCtoRGB transform matrix. The value shall be in the range of **0** to **4294967295**, inclusive. If **YCCtoRGB_off set1** is not present, it shall be inferred to be **536870912**.

YCCtoRGB_offset2 specifies the third offset of YCCtoRGB transform matrix. The value shall be in the range of 0 to 4294967295, inclusive. If **YCCtoRGB_offset2** is not present, it shall be inferred to be 536870912.

5 **RGBtoLMS_coef0** specifies the first coefficient of RGBtoLMS transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **RGBtoLMS_coef 0** is not present, it shall be inferred to be 5845.

10 **RGBtoLMS_coef1** specifies the second coefficient of RGBtoLMS transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **RGBtoLMS_coef 1** is not present, it shall be inferred to be 9702.

15 **RGBtoLMS_coef2** specifies the third coefficient of RGBtoLMS transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **RGBtoLMS_coef 2** is not present, it shall be inferred to be 837.

20 **RGBtoLMS_coef3** specifies the fourth coefficient of RGBtoLMS transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **RGBtoLMS_coef 3** is not present, it shall be inferred to be 2568.

RGBtoLMS_coef4 specifies the fifth coefficient of RGBtoLMS transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **RGBtoLMS_coef 4** is not present, it shall be inferred to be 12256.

25 **RGBtoLMS_coef5** specifies the sixth coefficient of RGBtoLMS transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **RGBtoLMS_coef 5** is not present, it shall be inferred to be 1561.

30 **RGBtoLMS_coef6** specifies the seventh coefficient of RGBtoLMS transform matrix. The value shall be in the range of -32768 to 32767, inclusive. If **RGBtoLMS_coef 6** is not present, it shall be inferred to be 0.

RGBtoLMS_coef7 specifies the eighth coefficient of RGBtoLMS transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **RGBtoLMS_coef7** is not present, it shall be inferred to be **679**.

5 **RGBtoLMS_coef8** specifies the ninth coefficient of RGBtoLMS transform matrix. The value shall be in the range of **-32768** to **32767**, inclusive. If **RGBtoLMS_coef8** is not present, it shall be inferred to be **15705**.

signal_eof shall be set to **65535** for **Ultra -HD Blu- ray**.

10

signal_eof_param0 shall be set to **0** for **Ultra -HD Blu- ray**.

signal_eof_param1 shall be set to **0** for **Ultra -HD Blu- ray**.

15

signal_eof_param2 shall be set to **0** for **Ultra -HD Blu- ray**.

signal_bit_depth indicates the bit depth of Dolby Vision signal. The value shall be in the range of **8** to **16**, inclusive. If **signal_bit_depth** is not present, it shall be inferred to be **14**.

20

signal_color_space specifies the Dolby Vision signal color space. The value shall be in the range of **0** to **3**, inclusive. If **signal_color_space** is not present, it shall be inferred to be **0**. The corresponding color spaces are defined in the following table:

Table A-4- 1 Definition of Dolby Vision signal color space Dolby

signal_color_space	Color Space Representation
0	YCbCr
1	RGB
2	IPT
3	Reserved

25

signal_chroma_format specifies the Dolby Vision signal chroma format if **signal_color_space** is equal to **0**. The value shall be in the range of **0** to **2**, inclusive. If **signal_color_space** is larger

than 0, signal_chroma_format shall be set to 2. If signal_chroma_format is not present, it shall be inferred to be 0. The corresponding chroma formats are specified in the following table:

Table A-4- 2 Definition of Dolby Vision signal chroma format

signal_chroma_format	Chroma Format Representation
0	4:2:0
1	4:2:2
2	4:4:4

5

signal_full_range_flag specifies the Dolby Vision signal range. The value shall be in the range of 0 to 3, inclusive. If signal_eotf is equal to 65535, signal_full_range_flag shall be set to 1. If signal_full_range_flag is not present, it shall be inferred to be 1. Refer to [6] for the signal range naming conventions. The corresponding signal ranges are specified in the following table:

10

Table A-4- 3 Definition of Dolby Vision signal full range flag

signal_full_range_flag	signal range
0	Narrow range
1	Full range
2	SDI range
3	Reserved

source_min_PQ specifies the minimum luminance value of source display in 12-bit PQ encoding. The value shall be in the range of 0 to 4095, inclusive. If source_min_PQ is not present, it shall be inferred to be 62.

15

Note: the 12-bit PQ encoded luminance value is calculated as follows:

$$L = C/10,000$$

$$N = \left(\frac{c_1 + c_2 L^{m_1}}{1 + c_3 L^{m_1}} \right)^{m_2}$$

$$PQ = \text{Floor}((2^b - 1) \times N + 0.5),$$

where

C denotes an optical value in cd/m²

20

L denotes the corresponding normalized linear value

N denotes the corresponding nonlinear value

m_i is the number $2610/4096 \times \frac{1}{4} = 0.1593017578125$

mi is the number $2523/4096 \times 128 = 78.84375$

c_i is the number $3424/4096 = 0.8359375 = c_3 - c_2 + 1$

c_2 is the number $2413/4096 \times 32 = 18.8515625$

5 c_3 is the number $2392/4096 \times 32 = 18.6875$

b is the number 12

The unary function *Floor* yields the largest integer not greater than its argument.

PQ is the resulting digital representation

10 **source_max_PQ** specifies the maximum luminance value of source display in 12-bit PQ encoding. The value shall be in the range of 0 to 4095, inclusive. If **source_max_PQ** is not present, it shall be inferred to be 3696.

15 **source_diagonal** indicates the diagonal size of source display in inch. The value shall be in the range of 0 to 1023, inclusive. If **source_diagonal** is not present, it shall be inferred to be 42.

num_ext_blocks specifies the number of extended metadata blocks. The value shall be in the range of 0 to 254, inclusive. If **num_ext_blocks** is not present, it shall be inferred to be 0.

20 **dm_alignment_zero_bit** shall be equal to 0. **dm_alignment_zero_bit** is not present if **num_ext_blocks** is equal to 0.

A 4.2 Dolby Vision Display Management Extended Metadata Bistream Semantics

25

ext_block_length is used to derive the size of current extended metadata block payload in bytes. The value shall be in the range of 0 to 4294967295, inclusive. **ext_block_length** is not present if **num_ext_blocks** is equal to 0. If **ext_block_length** is not present, it shall be inferred to be 0.

30

ext_block_level specifies the level of payload contained in the current extended metadata block. The value shall be in the range of 0 to 255, inclusive. The corresponding block levels are defined in the following table. If ext_block_level is not present, it shall be inferred to be 0.

5

Table A-4- 4 Definition of extended metadata block type

ext_block_level	extended metadata block type
0	Reserved
1	Level 1 Metadata - Content Range
2	Level 2 Metadata - Trim Pass
3...255	Reserved

10

Note: If there is more than one extension block with ext_block_level equal to 1, the decoder shall only use the latest level 1 extension block transmitted in the current frame. If there are more than 16 extension blocks with ext_block_level equal to 2, the decoder shall only use the first 16 level 2 extension blocks transmitted in current frame. If there is an extension block with ext_block_level equal to reserved values, the decoder shall ignore that extension block. If there is none of extension block transmitted in current frame, the decoder shall fall back to the values of level 1 and level 2 extended metadata as specified in A 4.3.

15

A 4.3 Dolby Vision Display Management Extended Metadata Payload Semantics

20

min_PQ specifies the minimum luminance value of current scene in 12-bit PQ encoding. The value shall be in the range of 0 to 4095, inclusive. If min_PQ is not present, it shall be inferred to be equal to the value of source_min_PQ.

25

max_PQ specifies the maximum luminance value of current scene in 12-bit PQ encoding. The value shall be in the range of 0 to 4095, inclusive. If max_PQ is not present, it shall be inferred to be equal to the value of source_max_PQ.

avg_PQ specifies the midpoint luminance value of current scene in 12-bit PQ encoding. The value shall be in the range of 0 to 4095, inclusive. If **avg_PQ** is not present, it shall be inferred to be equal to the value of $(\text{source_min_PQ} + \text{source_max_PQ})/2$.

5 **target_max_PQ** specifies the maximum luminance value of a target display in 12-bit PQ encoding. The value shall be in the range of 0 to 4095, inclusive. If **target_max_PQ** is not present, it shall be inferred to be equal to the value of **source_max_PQ**.

Note: There are more than one extension block with **ext_block_level** equal to 2, those blocks shall have no duplicated **target_max_PQ**.

10

trim_slope specifies the slope metadata. The value shall be in the range of 0 to 4095, inclusive. If **trim_slope** is not present, it shall be inferred to be 2048. Note that the 12-bit slope value is calculated as follows:

$$\text{trim_slope} = (S+0.5) \times 4096$$

15

trim_of f set specifies the offset metadata. The value shall be in the range of 0 to 4095, inclusive. If **trim_of f set** is not present, it shall be inferred to be 2048. Note that the 12-bit offset value is calculated as follows:

$$\text{trim_offset} = (O+0.5) \times 4096$$

20

trim_power specifies the power metadata. The value shall be in the range of 0 to 4095, inclusive. If **trim_power** is not present, it shall be inferred to be 2048. Note that the 12-bit power value is calculated as follows:

$$\text{trim_power} = (P+0.5) \times 4096$$

25

trim_chroma_weight specifies the chroma weight metadata. The value shall be in the range of 0 to 4095, inclusive. If **trim_chroma_weight** is not present, it shall be inferred to be 2048. Note that the 12-bit chroma weight value is calculated as follows:

$$\text{trim_chroma_weight} = (CW+0.5) \times 4096$$

30

`trim_saturation_gain` specifies the saturation gain metadata. The value shall be in the range of 0 to 4095, inclusive. If `trim_saturation_gain` is not present, it shall be inferred to be 2048.

Note that the 12-bit saturation gain value is calculated as follows:

$$\text{trim_saturation_gain} = (SG+0.5) \times 4096$$

5

`ms_weight` specifies the multiscale weight metadata. The value shall be in the range of -1 to 4095, inclusive. If `ms_weight` is not present, it shall be inferred to be 2048. Note that the 12-bit multiscale weight value is calculated as follows:

$$\text{ms_weight} = (MS+1.0) \times 2048$$

10

Note: If `ms_weight` is equal to -1, it means the multiscale weight value shall be overridden by the local settings.

`ext_dm_alignment_zero_bit` shall be equal to 0.

15

REFERENCES

The following literature supplements the information in this document and each one is incorporated by reference in its entirety for all purposes:

[1] ITU-R Recommendation BT.709, Parameter Values for the HDTV Standards for *Production and International Program Exchange*, available from

20

<http://www.itu.int>

[2] ITU-R Recommendation BT.2020, Parameter Values for Ultra-High-Definition Television Systems for Production and International Program Exchange, available from <http://www.itu.int>

25

[3] ITU-T H.265 (10/2014), *High-Efficiency Video Coding*, available at <http://www.itu.int>

[4] High-Definition Multimedia Interface™ Specification, version 1.4b, available from <http://www.hdmi.org>

[5] High-Definition Multimedia Interface™ Specification, version 2.0, available from <http://www.hdmi.org>

30

[6] SMPTE ST 2084:2014, High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays, August 16, 2014

[7] SMPTE ST 2086:2014, Mastering Display Color Volume Metadata Supporting High Luminance and Wide Color Gamut Images, October 13, 2014

[8] CEA-861-F, A DTV Profile for Uncompressed High Speed Digital Interfaces.

5 [9] Audio Visual Application Format Specifications for BD-ROM Version 3.0, White paper Blu-Ray Disc Read-only Format (Ultra HD Blu-ray), Blu-Ray Disc Association, July 2015

[10] ISO/IEC 13818-1:2013, *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Systems*, available from **www.iso.org**

10 [11] U.S. Patent Application Ser. No. 14/364,979, filed on June 12, 2014, "Specifying Visual Dynamic Range Coding Operations and Parameters," by S. Qu et al., published as U.S. Patent Application Pub. No. 2014/0341305.

[12] A. Ballestad and A. Kostin, "Method and Apparatus for Image Data Transformation," US. Patent 8,593,480.

15 EXAMPLE COMPUTER SYSTEM IMPLEMENTATION

[00088] Embodiments of the present invention may be implemented with a computer system, systems configured in electronic circuitry and components, an integrated circuit (IC) device such as a microcontroller, a field programmable gate array (FPGA), or another
20 configurable or programmable logic device (PLD), a discrete time or digital signal processor (DSP), an application specific IC (ASIC), and/or apparatus that includes one or more of such systems, devices or components. The computer and/or IC may perform, control, or execute instructions relating to layered representation and decoding of images with enhanced dynamic range, such as those described herein. The computer and/or IC may compute any of
25 a variety of parameters or values that relate to layered representation and decoding of images with enhanced dynamic range described herein. The image and video embodiments may be implemented in hardware, software, firmware and various combinations thereof.

[00089] Certain implementations of the invention comprise computer processors which execute software instructions which cause the processors to perform a method of the
30 invention. For example, one or more processors in a display, an encoder, a set top box, a transcoder or the like may implement methods related to layered representation and decoding of images with enhanced dynamic range as described above by executing software instructions in a program memory accessible to the processors. The invention may also be

provided in the form of a program product. The program product may comprise any non-transitory medium which carries a set of computer-readable signals comprising instructions which, when executed by a data processor, cause the data processor to execute a method of the invention. Program products according to the invention may be in any of a wide variety of forms. The program product may comprise, for example, physical media such as magnetic data storage media including floppy diskettes, hard disk drives, optical data storage media including CD ROMs, DVDs, electronic data storage media including ROMs, flash RAM, or the like. The computer-readable signals on the program product may optionally be compressed or encrypted.

10 [00090] Where a component (e.g. a software module, processor, assembly, device, circuit, etc.) is referred to above, unless otherwise indicated, reference to that component (including a reference to a "means") should be interpreted as including as equivalents of that component any component which performs the function of the described component (e.g., that is functionally equivalent), including components which are not structurally equivalent to the disclosed structure which performs the function in the illustrated example embodiments of the invention.

EQUIVALENTS, EXTENSIONS, ALTERNATIVES AND MISCELLANEOUS

[00091] Example embodiments that relate to the efficient layered representation and decoding of images with enhanced dynamic range are thus described. In the foregoing specification, embodiments of the present invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

- 5 1. A decoder for generating high-dynamic range video data, wherein the decoder comprises one or more processors and non-transitory memory, wherein generating an output image with the decoder comprises:
- receiving reference processing data and storing at least a portion of the reference processing data in the non-transitory memory;
- 10 parsing a cyclic redundancy check (CRC) parameter from the reference processing data;
- extracting from the reference processing data at least an RPU data header, RPU data payload data, and an RPU-alignment-zero-bit flag;
- computing by the decoder a CRC value on the entire reference processing data
- 15 excluding the CRC parameter; and
- comparing the CRC value with the CRC parameter to detect an error.
2. The decoder of claim 1, wherein generating an output image with the decoder further comprises extracting from the reference processing data display management payload data.
- 20 3. The decoder of claim 1, wherein generating an output image with the decoder further comprises:
- receiving a base-layer image;
- receiving an enhancement layer image;
- 25 and combining the base-layer image and the enhancement layer image based on the reference processing data to generate the output image.
4. The decoder of claim 1, wherein the RPU payload data comprise RPU prediction mapping data and RPU non-linear-quantizer (NLQ) data.
- 30 5. The decoder of claim 1, wherein generating an output image with the decoder further comprises:
- extracting an RPU type flag from the RPU data header; and

parsing additional data from the reference processing data only if the value of the RPU type flag matches a predetermined value.

6. The decoder of claim 5, wherein the RPU type flag comprises the `rpujype` flag which specifies a type of RPU processing, and the predetermined value is equal to 2.

7. The decoder of claim 1, wherein generating an output image with the decoder further comprises:

extracting a `use-previous-rpu-data` flag from the RPU data header and extracting the RPU data payload data only if the `use-previous-rpu-data` flag is set to zero.

8. The decoder of claim 7, wherein the `use-previous-rpu-data` flag comprises an `use_prev_vdr_rpu_flag` flag which, when set to 1 specifies that the current RPU data is predicted from previously received RPU data and when set to 0 specifies that the current RPU data are explicitly received and instructs the decoder to generate a new RPU data ID.

9. The decoder of claim 2, wherein display management payload data are parsed only if a `display-management-data-present-flag` flag extracted from the RPU data header is set to one.

10. The decoder of claim 9, wherein the `display-management-data-present-flag` comprises an `vdr_dma_metadata_present_flag` flag, which when set to 0 specifies that no display management payload data is present, and when set to 1 specifies that display management payload data is present according to a `vdr_dm_data_payload()` syntax structure.

11. The decoder of claim 1, wherein the RPU-alignment-zero-bit flag comprises an `rpu_alignment_zero_bit flag = 0`.

12. The decoder of claim 4, wherein the RPU prediction mapping data are extracted according to an `rpu_data_mapping()` syntax structure and the RPU non-linear-quantizer (NLQ) data are extracted according to an `rpu_data_nlq()` syntax structure.

13. A method for generating high-dynamic range video data with a decoder, wherein the method comprises:

receiving reference processing data using the decoder;

- parsing a cyclic redundancy check (CRC) parameter from the reference processing data;
- extracting from the reference processing data at least an RPU data header, RPU data payload data, and an RPU-alignment-zero-bit flag;
- 5 computing by the decoder a CRC value on the entire reference processing data excluding the CRC parameter; and
- comparing the CRC value with the CRC parameter to detect an error.
14. The method of claim 13, wherein generating an output image with the decoder further
- 10 comprises parsing from the reference processing data display management payload data.
15. The method of claim 13, wherein generating an output image with the decoder further comprises:
- receiving a base-layer image;
- 15 receiving an enhancement layer image;
- and combining the base-layer image and the enhancement layer image based on the reference processing data to generate the output image.
16. The method of claim 13, wherein the RPU payload data comprise RPU prediction
- 20 mapping data and RPU non-linear-quantizer (NLQ) data.
17. The method of claim 13, wherein generating an output image with the decoder further comprises:
- extracting an RPU type flag from the RPU data header; and
- 25 parsing additional data from the reference processing data only if the value of the RPU type flag matches a predetermined value.
18. The method of claim 17, wherein the RPU type flag comprises the `rpujtype` flag which specifies a type of RPU processing, and the predetermined value is equal to 2.
- 30
19. The method of claim 13, wherein generating an output image with the decoder further comprises:
- extracting a `use-previous-rpu-data` flag from the RPU data header and extracting the RPU data payload data only if the `use-previous-rpu-data` flag is set to zero.

20. The method of claim 19, wherein the use-previous-rpu-data flag comprises a **use_prev_vdr_rpu_flag** flag, which when set to 1 specifies that the current RPU data is predicted from previously received RPU data and when set to 0 specifies that the current RPU data are explicitly received and instructs the decoder to generate a new RPU data ID.
21. The method of claim 14, wherein display management payload data are parsed only if a display-management-data -present-flag flag extracted from the RPU data header is set to one.
22. The method of claim 21, wherein the display-management-data -present-flag comprises a **vdr_dma_metadata_present_flag** flag, which when set to 0 specifies that no display management payload data is present, and when set to 1 specifies that display management payload data is present according to a **vdr_dm_data_payload()** syntax structure.
23. The method of claim 13, wherein the RPU-alignment-zero-bit flag comprises an **rpu_alignment_zero_bit flag = 0**.
24. The method of claim 16, wherein the RPU prediction mapping data are extracted according to an **rpu_data_mapping()** syntax structure and the RPU non-linear-quantizer (NLQ) data are extracted according to an **rpu_data_nlq()** syntax structure.
25. The method of claim 15, further comprising:
- receiving a primary graphics plane image and a secondary graphics plane image in a dynamic range lower than the dynamic range of the output image;
 - applying inverse display management step to the primary graphics plane image and the secondary graphics plane image to generate corresponding third and fourth graphic images in the dynamic range of the output image; and
 - blending the output image and the third and fourth graphic images to generate a blended image.
26. The method of claim 15, further comprising:
- receiving a primary graphics plane image, a background plane image, and a secondary graphics plane image in a dynamic range lower than the dynamic range of the output image;

applying inverse display management step to the primary graphics plane image, the background plane image, and the secondary graphics plane image to generate corresponding third, fourth, and fifth graphic images in the dynamic range of the output image; and

5 blending the output image and the third, fourth, and fifth graphic images to generate a blended image.

27. A non-transitory computer-readable storage medium having stored thereon computer-executable instruction for executing a method with one or more processors in accordance with any one of the claims 13-26.

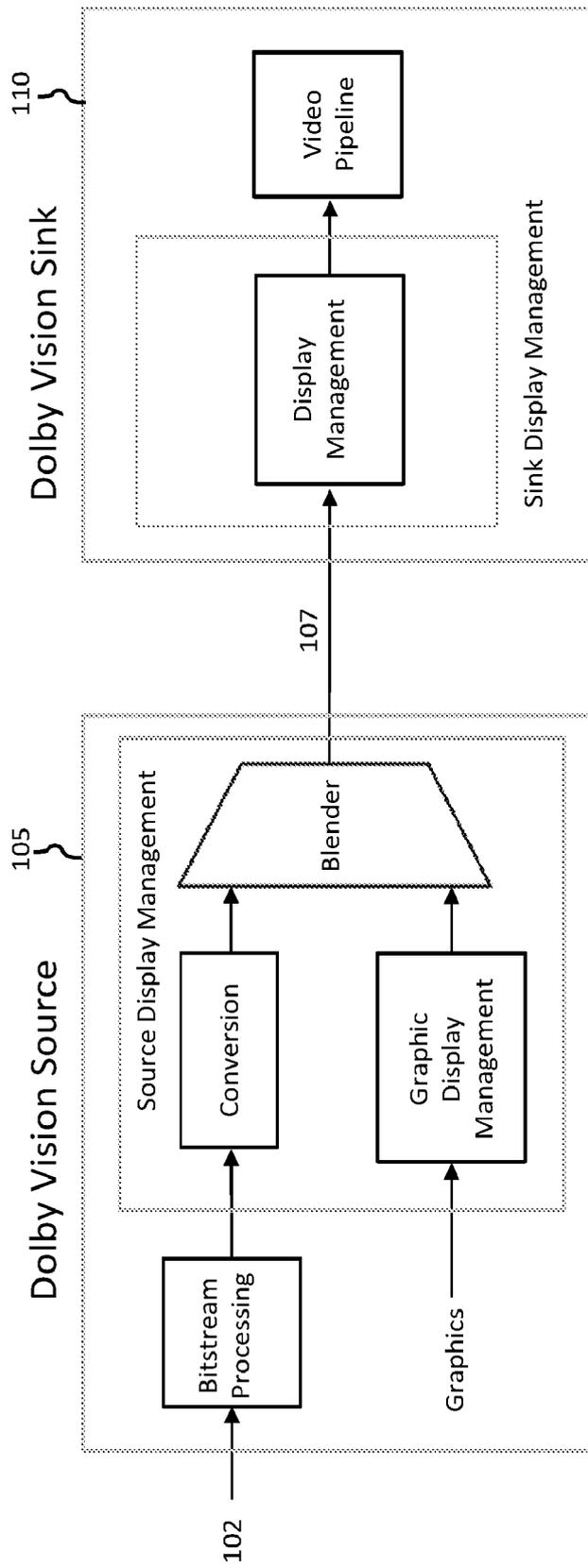


FIG. 1

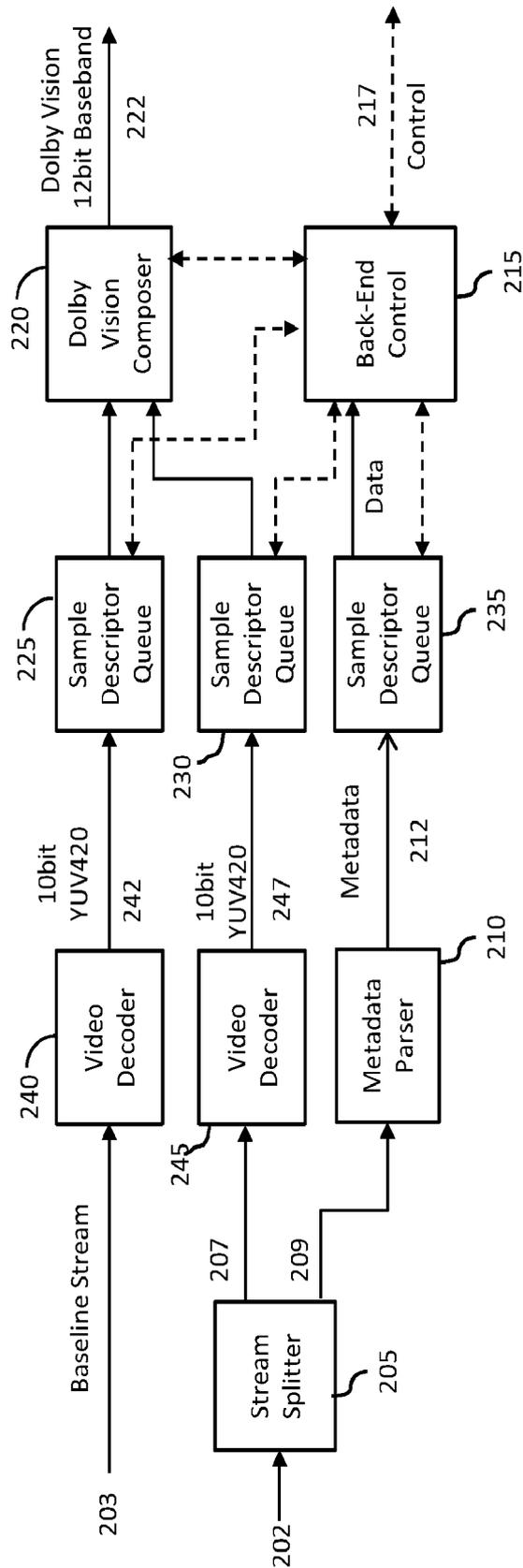


FIG. 2

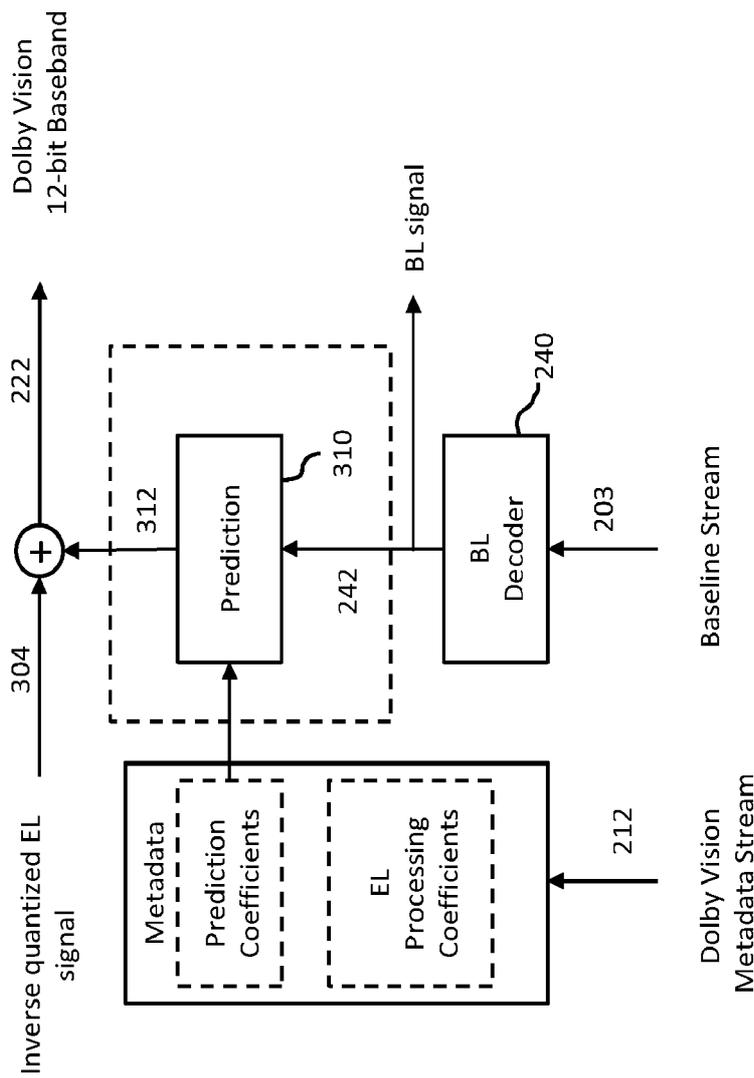


FIG. 3

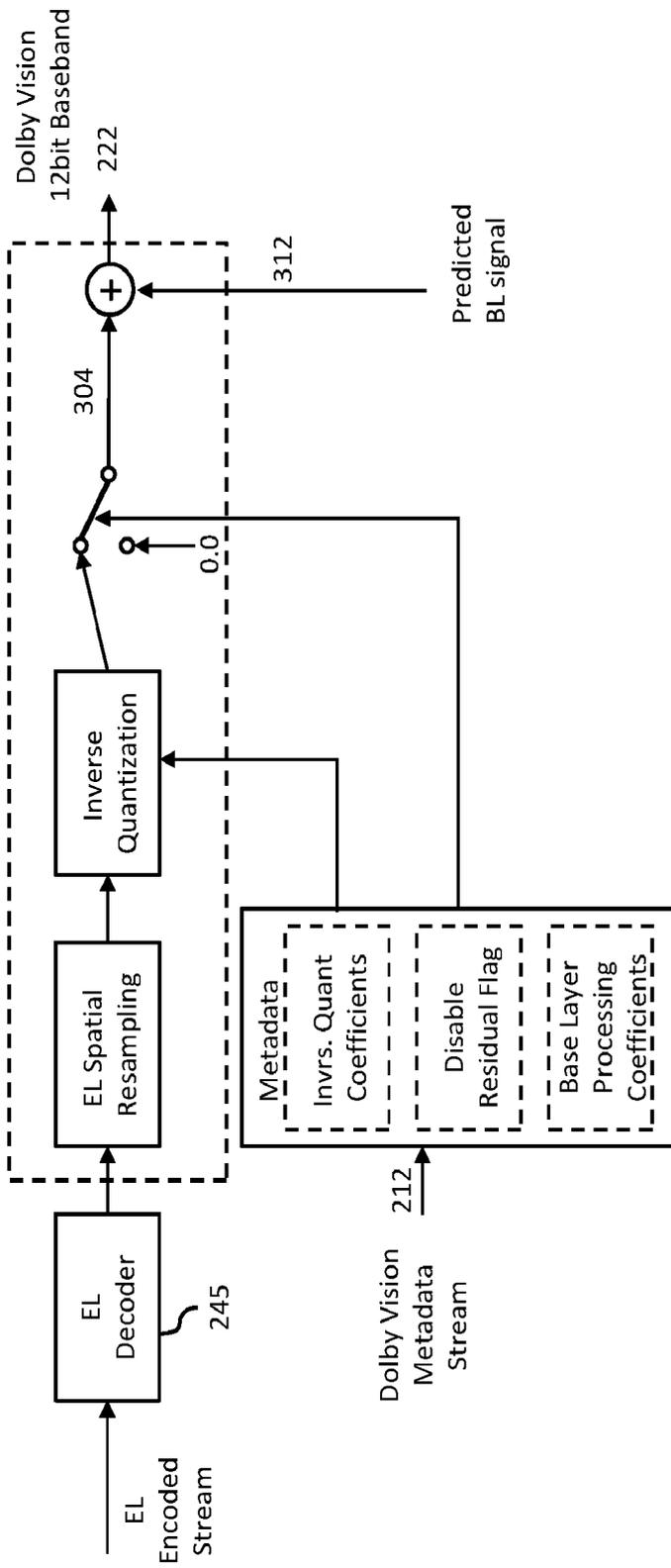


FIG. 4

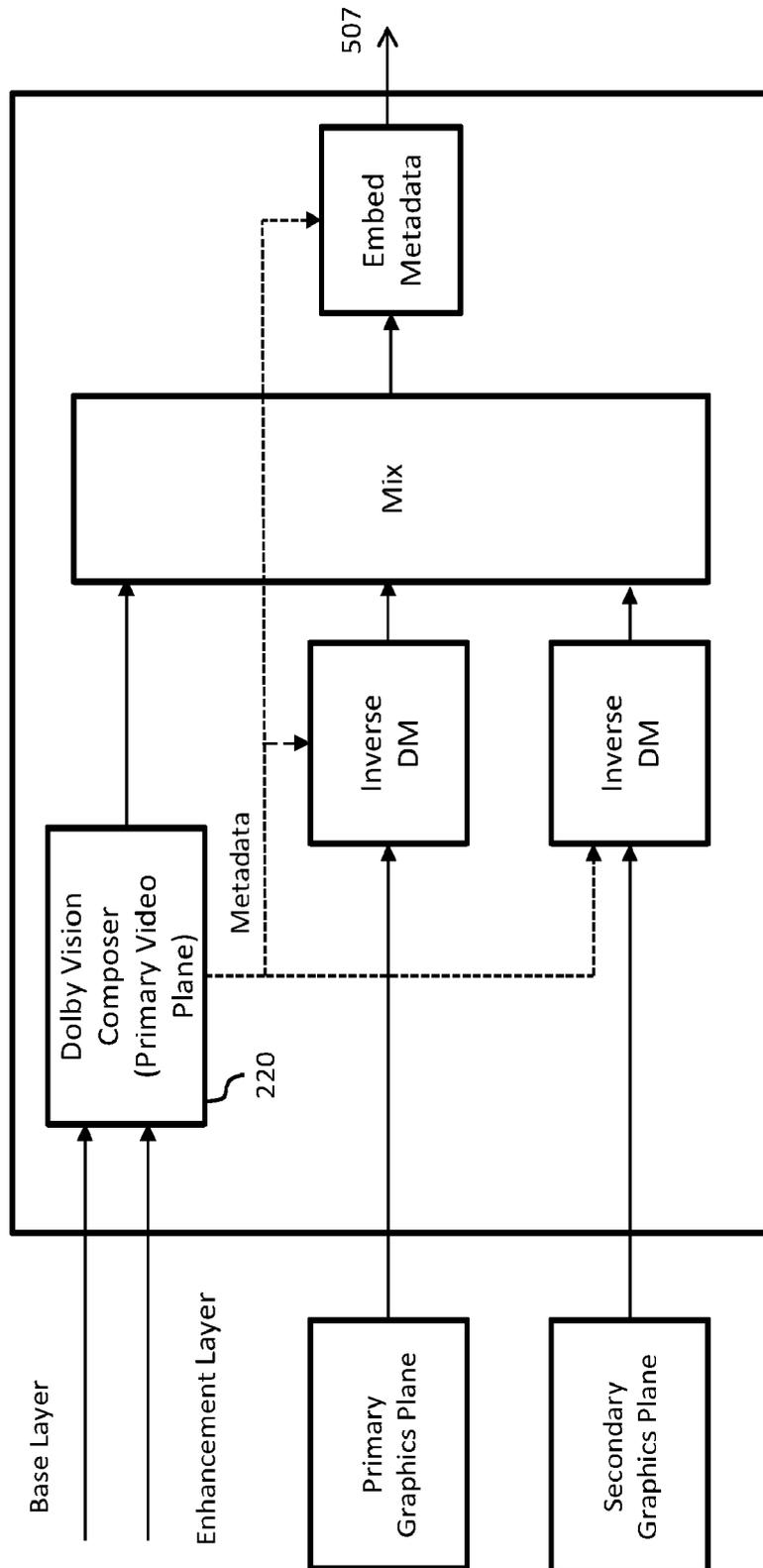


FIG. 5A

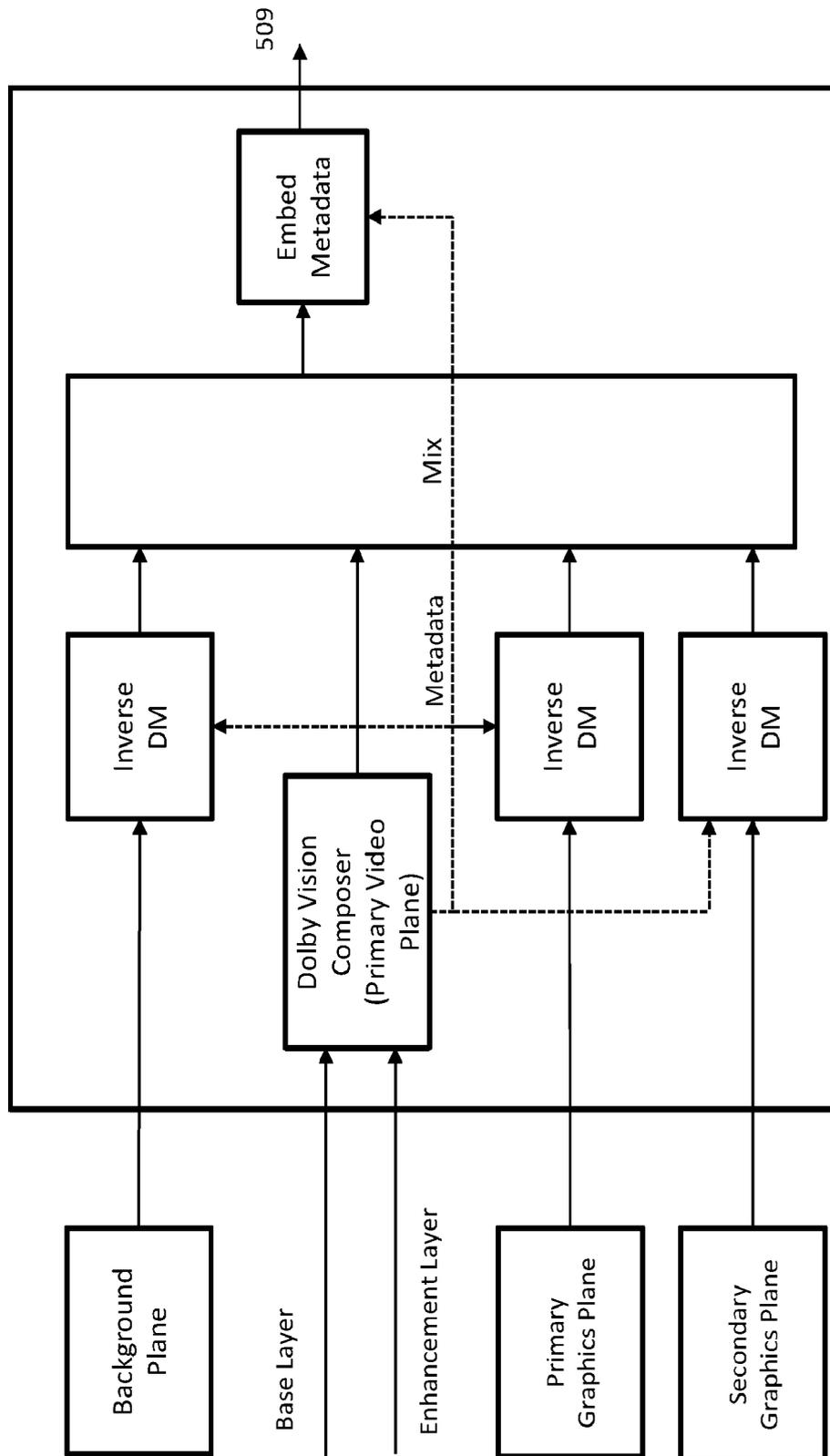


FIG. 5B

INTERNATIONAL SEARCH REPORT

International application No PCT/US2016/059862
--

A. CLASSIFICATION OF SUBJECT MATTER
 INV. H04N19/65
 ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
 EPO-Internal , WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 2 866 450 A1 (DOLBY LAB LICENSING CORP [US]) 29 April 2015 (2015-04-29)	I, 3, 4, II, 13, 15, 16, 23, 27
Y	paragraph [0024]	2, 5-10, 14, 17-22
A	----- W0 2014/176019 A1 (DOLBY LAB LICENSING CORP [US]) 30 October 2014 (2014-10-30) paragraph [0050]	1-11, 13-23, 27
Y	----- US 2014/341305 A1 (QU SHENG [US] ET AL) 20 November 2014 (2014-11-20) cited in the application paragraph [0053] paragraph [0079] - paragraph [0081] paragraph [0090] - paragraph [0091] -----	2, 5-10, 14, 17-22

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents :

<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>
---	---

Date of the actual completion of the international search 7 February 2017	Date of mailing of the international search report 04/04/2017
--	--

Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Raeymaekers, Peter
--	--

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/059862

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos. :

4. No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos. :

1-11 , 13-23 , 27

Remark on Protest

The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-11, 13-23, 27

addition of display management data to RPU data;

1.1. claims: 5, 6, 17, 18

using a RPU type flag in the RPU header

1.2. claims: 7, 8, 19, 20

use of a flag (use_prev_vdr_rpu_flag) for indicating use of previous RPU data

2. claims: 12, 24

use of syntax structures for parsing RPU prediction mapping data and RPU non-linear quantizer data

3. claims: 25, 26

applying dynamic range adaptations to graphic planes

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No PCT/US2016/059862
--

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2866450	AI	29-04-2015	EP 2866450 AI
			US 2015117551 AI

WO 2014176019	AI	30-10-2014	CN 105144729 A
			EP 2989798 AI
			HK 1214051 AI
			US 2016105695 AI
			WO 2014176019 AI

US 2014341305	AI	20-11-2014	CN 104041036 A
			EP 2801192 AI
			HK 1203013 AI
			JP 5989800 B2
			JP 2015503873 A
			JP 2016208542 A
			KR 20140120320 A
			KR 20160029134 A
			TW 201347542 A
			TW 201642656 A
			US 2014341305 AI
			WO 2013103522 AI
